Task Performance with List-Mode Data

by

Luca Caucci

A Dissertation Submitted to the Faculty of the

College of Optical Sciences

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy

In the Graduate College

The University of Arizona

2012

The University of Arizona
Graduate College

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Luca Caucci entitled *Task Performance with List-Mode Data* and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Harrison H. Barrett Date: July 18, 2012

Lars R. Furenlid Date: July 18, 2012

Matthew A. Kupinski Date: July 18, 2012

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Harrison H. Barrett Date: July 18, 2012
Dissertation Director: Dr. Harrison H. Barrett

Statement by Author

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Signed: Luca Caucci

Acknowledgements

I feel enormously grateful to my advisor Dr. Harrison H. Barrett for his mentorship and encouragement throughout my graduate studies. I would also like to thank Dr. Lars R. Furenlid and Dr. Matthew A. Kupinski for serving on the defense committee. Their thorough revisions have greatly improved the dissertation. This work has been supported by NIH grants R01 EB000803 and P41 EB002035.

Table of Contents

Table of Contents—*Continued*

Table of Contents—*Continued*

List of Figures

List of Figures—*Continued*

List of Figures—*Continued*

List of Figures—*Continued*

List of Figures—*Continued*

List of Tables

List of Tables—*Continued*

Abstract

This dissertation investigates the application of list-mode data to detection, estimation, and image reconstruction problems, with an emphasis on emission tomography in medical imaging.

We begin by introducing a theoretical framework for list-mode data and we use it to define two observers that operate on list-mode data. These observers are applied to the problem of detecting a signal (known in shape and location) buried in a random lumpy background. We then consider maximum-likelihood methods for the estimation of numerical parameters from list-mode data, and we characterize the performance of these estimators via the so-called Fisher information matrix.

Reconstruction from PET list-mode data is then considered. In a process we called "double maximum-likelihood" reconstruction, we consider a simple PET imaging system and we use maximum-likelihood methods to first estimate a parameter vector for each pair of gamma-ray photons that is detected by the hardware. The collection of these parameter vectors forms a list, which is then fed to another maximum-likelihood algorithm for volumetric reconstruction over a grid of voxels.

Efficient parallel implementation of the algorithms discussed above is then presented. In this work, we take advantage of two low-cost, mass-produced computing platforms that have recently appeared on the market, and we provide some details on implementing our algorithms on these devices.

We conclude this dissertation work by elaborating on a possible application of list-mode data to X-ray digital mammography. We argue that today's CMOS detectors and computing platforms have become fast enough to make X-ray digital mammography list-mode data acquisition and processing feasible.

CHAPTER 1
Introduction

1.1 Motivation

The field of medicine relies heavily on images for the diagnosis of pathologies, their treatment and monitoring, and for the discovery and development of new drugs [11]. To answer the need for images that allow physicians to do diagnosis and carry out research, many different imaging modalities have been developed [12]. Among them, we want to list here X-ray radiography, ultrasound, fluorescence and bioluminescence optical imaging, computed tomography (CT), positron-emission tomography (PET), single-photon emission computed tomography (SPECT), magnetic-resonance imaging (MRI) and spectroscopy (MRS), endoscopy, medical thermography, medical photography and microscopy.

Preclinical (animal) studies and drug development make heavy use of medical imaging. Animal models are used in early stages of drug development to study, among other things, feasibility, and safety of a potential drug or treatment [13]. Animal studies have been crucial in deepening our knowledge of carcinogenesis via tumor modeling, monitoring of their growth, and detection of metastasis [8, 14]. Small animals, such as mice and rats, have many advantages over larger animals: low housing and maintenance costs, short reproductive cycle, ease of transport, and well-developed protocols and methodologies for genetic manipulation [15, 16]. With non-invasive *in-vivo* imaging, it is not necessary to sacrifice the animal at each time point of interest; this allows working with fewer animals and it also increases the statistical validity of the data by reducing the level of experimental variation [9, 17].

Building an imaging device for a small animal is usually less expensive than building a human-size imaging system. However, the small size of the animal used in preclinical testing also translates into the requirement for higher sensitivity and spatial resolution to maintain the equivalent accuracy as human studies [18–21]. A comparison between clinical and preclinical PET and SPECT imaging systems is reported in Table 1.1.

Modality		**Sensitivity**	**Resolution**	**Field of view (FOV)**
Clinical	PET	1–3%	≈ 5 mm	≈ 50 cm
	SPECT	0.01–0.03%	≈ 10 mm	≈ 50 cm
Preclinical	PET	2–4%	1–2 mm	≈ 7 cm
	SPECT	0.3%	0.5–2 mm	≈ 8 cm

Table 1.1. Comparison between clinical and preclinical PET/SPECT imaging systems (adapted from [8])

Many imaging devices have been proposed in the past. Along with hardware, software—in the form of reconstruction algorithms—has been developed. The goal of this dissertation is to attack the problem of image reconstruction (or, more specifically, signal detection and parameter estimation) with the formal mathematical methods of image quality assessment. This, in turn, will show how the information collected by the hardware can be optimally processed to increase detection/estimation performance.

The underlying requirement to accomplish this goal is *list-mode* data storage and processing. In a sentence, list-mode data representation consists of storing (hence, preserving) in a list all the information learned about each detected photon. Using list-mode data, common pre-processing steps—such as binning of photons into detector pixels or voxels—are avoided, along with the inherent information loss that these steps might introduce.

List-mode data representation has traditionally been overshadowed by other data representations, such as the already-mentioned binned data. Perhaps, this is due to the fact that processing list-mode data necessitates high-performance computational hardware and—to a lesser extent—to the lack of a detailed theoretical framework. This dissertation strives to provide a theoretical framework with which to interpret and manipulate list-mode data and, at the same time, illustrate how today's mass-produced parallel commodity computing hardware can be used to process list-mode data.

1.2 Nuclear Medicine Imaging

Generally speaking, we can define nuclear medicine imaging as any imaging technique that relies on radioactive isotopes (radioisotopes) to generate images of a patient's organs in a non-invasive way. Different radiotracers have different biochemical properties and tend to concentrate in particular organs. For example, ^{131}I tends to accumulate in the thyroid gland while ^{11}C is often used to study pathologies that affect metabolism, such as: gout, anemia, diabetes, and acromegaly (hormonal disorder characterized by excessive production of growth hormone in the anterior pituitary).

The beginning of nuclear medicine can be traced back to 1895 when Wilhelm Conrad Röntgen (March 27, 1845–February 10, 1923) discovered X-rays and Marie Curie (November 7, 1867–July 4, 1934) laid down a theory for radioactivity and discovered two radioactive elements: polonium and radium. Another important contribution came from George Charles de Hevesy (August 1, 1885–July 5, 1966) who played a key role in the development of radioactive tracers to study chemical processes such as the metabolic processes in animals.

Among all the possible modalities we listed at the beginning of this chapter, we will focus our attention on those falling under the category of *emission tomography* [10]. In emission tomography imaging, a radiopharmaceutical is administered to a patient. The radiopharmaceutical is a radioactive drug in which radioactive elements (in the form of radionuclides) are bound to a pharmaceutically-active molecule. This process is carried out by synthesizing a molecule in which one of the elements is replaced by a radioactive isotope (radionuclide) [1, 2], as shown in Figure 1.1. Some parameters of interest of commonly used radionuclides are reported in Table 1.2.

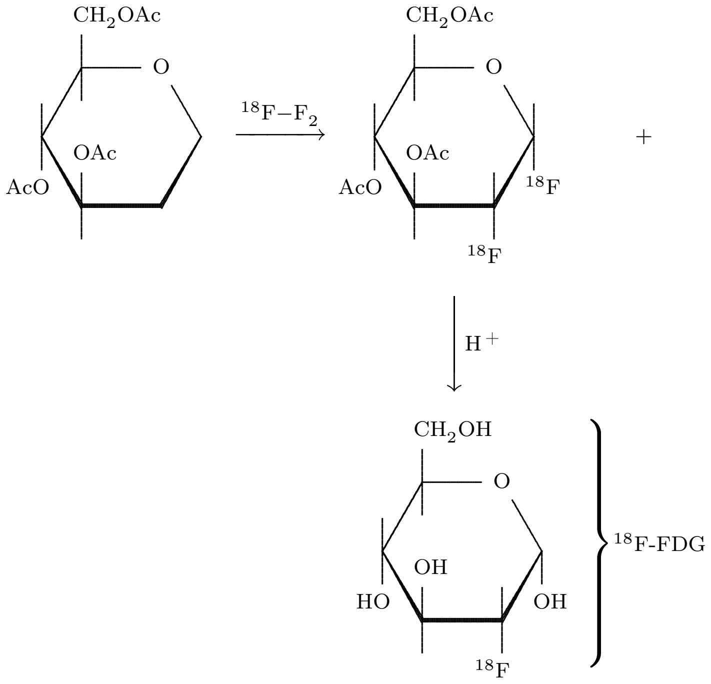


FIGURE 1.1. Synthesis of ^{18}F-FDG by electrophilic fluorination (adapted from [1, 2])

Depending on their biochemical nature, radiopharmaceuticals tend to accumulate and get metabolized in specific organs or tissues. When the radioactive part of a radiopharmaceutical (i.e., the radionuclide) undergoes radioactive decay, gamma-ray photons are emitted isotropically, thus forming a gamma-ray emission density corresponding to the body's uptake. These gamma-ray photons can be collected over time by specialized detecting devices (known as gamma-ray cameras), and this information is recorded for processing. This allows physicians to detect and localize tumors or lesions, identify regions of the brain influenced by drugs or external stimuli, and so on. Leading medical imaging techniques encompassed by emission tomography include single-photon emission computed tomography (SPECT) and positron emission tomography (PET). We will devote the next two sections to the description of these

Radionuclide	**Decay Mode**	**Half-Life**	**Principal γ-Ray(s)**	**Modality**
^{11}C	β^+ (100%)	20.39 m	511 keV	PET
^{13}N	β^+ (100%)	9.97 m	511 keV	PET
^{15}O	β^+ (100%)	2.04 m	511 keV	PET
^{18}F	β^+ (96.7%)	109.77 m	511 keV	PET
^{62}Cu	β^+ (100%)	9.74 m	511 keV	PET
^{64}Cu	β^+ (17.6%)	12.70 h	511, 1346 keV	PET
	EC (43.4%)		1346 keV	
	β^- (39.0%)		no γ-rays	
^{68}Ga	β^+ (100%)	68.1 m	511 keV	PET
^{76}Br	β^+ (100%)	16.1 h	511 keV	PET
$^{99\mathrm{m}}$Tc	IT (100%)	6.02 h	142 keV	SPECT
^{111}In	EC (100%)	2.80 d	171, 245 keV	SPECT
^{123}I	EC (100%)	13.22 h	159 keV	SPECT
^{124}I	β^+ (22.8%)	4.18 d	511, 603, 723, 1691 keV	PET
^{131}I	β^- (100%)	8.03 d	364 keV	SPECT
^{201}Tl	EC (100%)	3.04 d	68–80 keV x-rays	SPECT

β^+ = beta decay with positron emission; β^- = beta decay with electron emission; EC = electron capture, IT = isomeric transition.

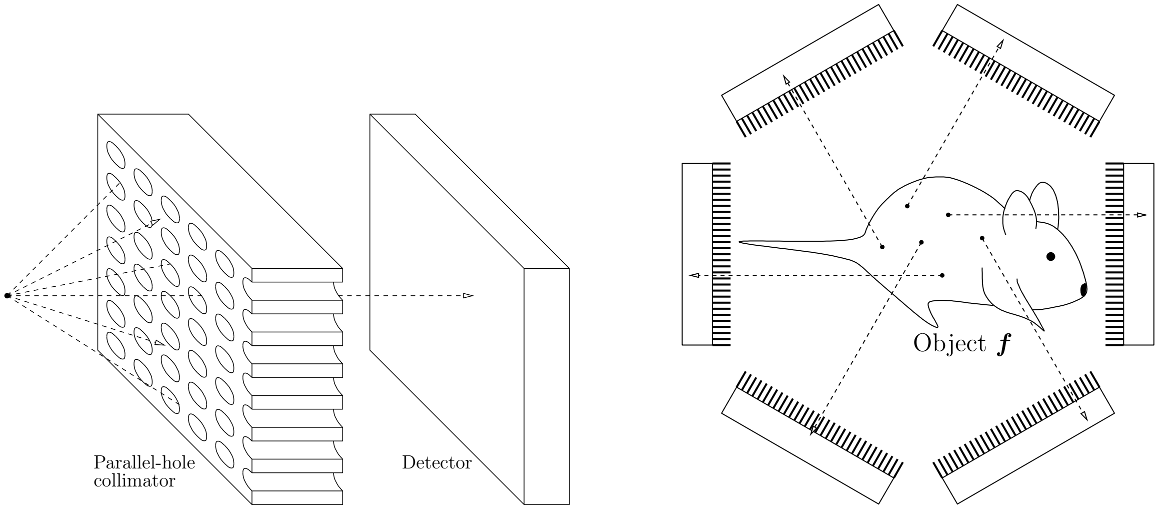
Table 1.2. Common PET and SPECT radionuclides used for imaging (adapted from [8,9])

two imaging modalities.

Even though some of the mathematical tools that we will develop in the remainder of this dissertation are very general and applicable to other imaging modalities as well (such as X-ray imaging) we will usually give preference to SPECT and PET imaging.

1.3 Single-Photon Emission Computed Tomography (SPECT)

As remarked above, SPECT imaging [10] takes advantage of the fact that some radionuclides are unstable and, when they undergo decay to form another isotope, a high-energy, externally detectable gamma-ray photon is emitted. Radionuclides commonly used in SPECT imaging include $^{99\mathrm{m}}$Tc, ^{111}In, ^{123}I, and ^{201}Tl [10]. Table 1.2 lists some of the properties of these and other radionuclides.



(a) Imaging with a parallel-hole collimator

(b) Schematic diagram of a SPECT detector arrangement

FIGURE 1.2. SPECT imaging

Photons are emitted isotropically from each point of the object reached by the radiotracer, and the mean number of photons emitted during a sufficiently long time interval is proportional to the concentration of the radiotracer distribution within the object. The location of emission and direction of propagation of these photons are ill-defined [22] quantities. The direction of propagation can be restricted by placing a collimator between the object and the detector. The collimator is usually made of tungsten (or any other high-density radiation-absorbing material such as lead) and its job is to absorb and block gamma-ray photons that do not propagate along specific paths. As an example, Figure 1.2a, shows the case of a parallel-hole collimator: photons that propagate in a direction oblique to the axes of the collimator's apertures strike on the walls of the collimator and get absorbed. By selecting—via the collimator(s)—some of the gamma-ray photons based on their direction of propagation, a correspondence between points in object space and image space is established, and this correspondence is used to create an image on the detector.

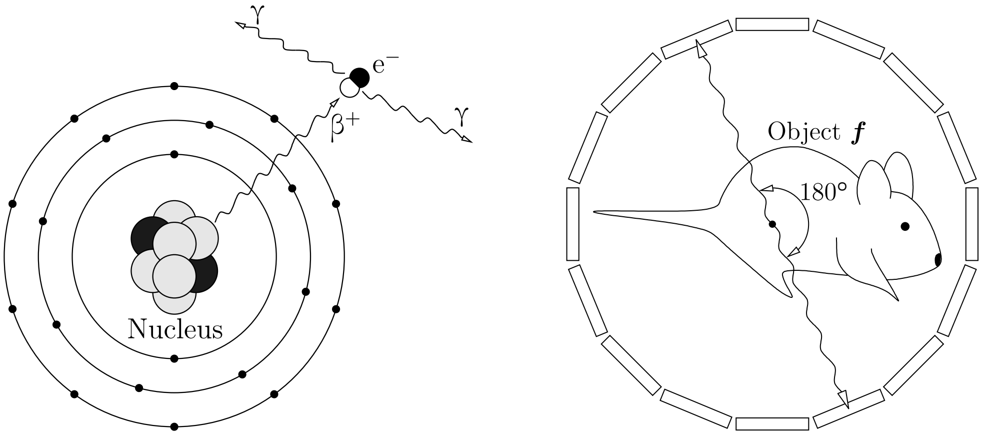
SPECT imaging is carried out by collecting data along different projections, as shown in Figure 1.2b. This setup can be accomplished by either rotating the object or by moving the collimator-detector assembly around the object, as shown in Figure 1.2b. Systems with multiple collimator-detector assemblies have been developed as well [20, 23–26]. An alternative approach to SPECT imaging is by means of coded apertures [27–31].

As a final note on SPECT imaging, we remark that the SPECT radionuclides listed in Table 1.2 allow for a wide variety of gamma-ray photon energies being emitted. If high-energy resolution detectors are used [32], dual-isotope SPECT imaging is possible [33–35]. Dual-isotope imaging enables the investigation of multiple biological processes at the same time, thus opening new areas in research and drug development.

1.4 Positron Emission Tomography (PET)

Similarly to SPECT, PET imaging [10, 36] uses gamma-ray photons emitted when unstable radionuclides (such as ^{11}C, ^{13}N, ^{15}O, ^{18}F [10]) decay. The decay mechanism used in PET imaging is, however, different. We can trace back [37,38] the development of positron emission tomography to the middle of the 1950s, when it was recognized that some radionuclides exhibit chemical properties that render them suitable for the tracing of important physiological pathways [38], such as oxygen tension in malignant neoplasms. Some examples of preclinical PET scanners are discussed in [39–41].

Positron emission (also called β^+ decay) occurs when a proton is converted into a neutron via the weak nuclear force. This process is immediately followed by the release of a positron (the antimatter counterpart of the electron). The positron travels for a short distance (no more than a few millimeters [42]) before it annihilates with a nearby electron. Positron-electron annihilation results in the emission of two gamma-ray photons—traveling at nearly 180° apart—each one with energy equal to the rest mass of the electron (511 keV) [43, 44]. Figure 1.3a schematically display this whole process.



(a) Electron-positron annihilation

(b) Schematic diagram of a PET detector arrangement

Figure 1.3. PET imaging

The gamma-ray photons are detected by a set of gamma-ray cameras surrounding the object. Figure 1.3b shows a possible setup. When two photons are detected within a short period of time (say, 10 ns [36]) by opposite detectors, a coincidence event occurs. Specialized hardware and software are used to pair two detected photons and form a coincidence event [45, 46], as shown in Figure 1.4.

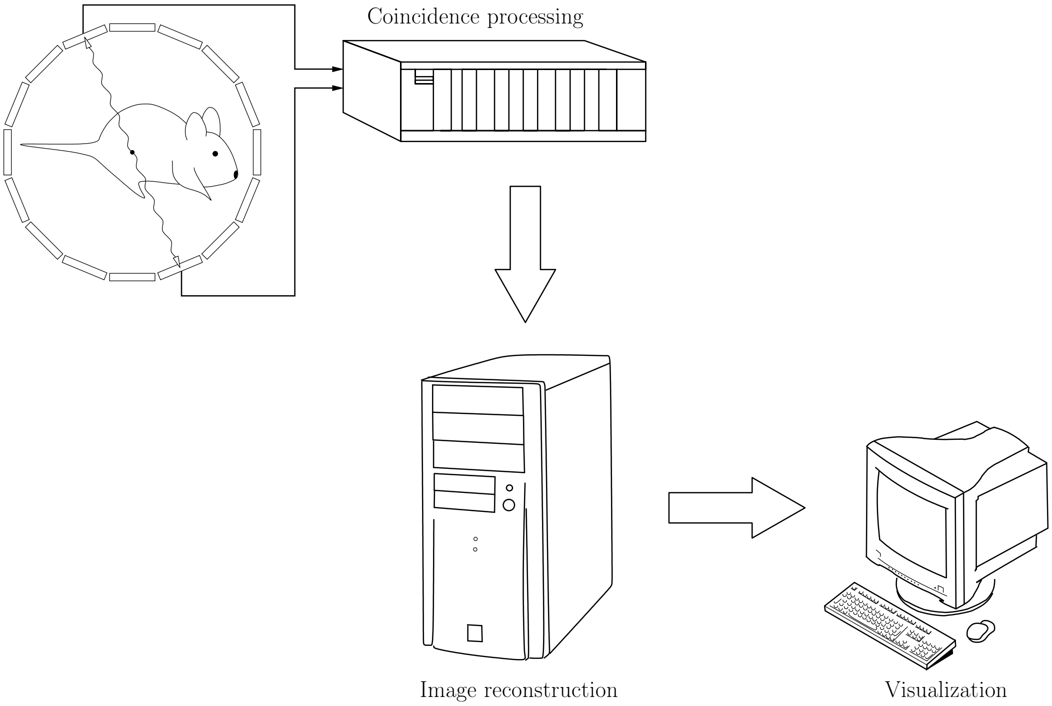


FIGURE 1.4. Schematic diagram of a PET imager

The two locations in the gamma-ray cameras at which photons have been detected define a line, called line of response (LOR). All we can learn by a coincidence event is that a positron-electron annihilation occurred somewhere along the LOR or in its close proximity. It is important to note that the knowledge of the LOR defines the direction of propagation of the two gamma-ray photons: contrary to the case of SPECT, no physical collimation is needed. Coincidentally, the process of pairing photon interaction to form a LOR is usually referred to as *electronic collimation*.

The short decay time of some commercially available gamma-ray camera crystals allows for better coincidence timing resolution so that the location of positron-electron annihilation along the LOR can be estimated by measuring the time-of-flight of the two coincident photon-crystal interactions [47]. Currently, time resolutions of the order of 500 ps are achievable, and this figure translates into a $\Delta x = 7.50$ cm constraint [48] on the position of the positron-electron annihilation along the LOR. As such, reconstructions based on time-of-flight measurements will not be of benefit for small-animal PET imaging. For objects of the size of the transverse human body, reductions of noise variance of a factor of about 5 has been predicted and obtained [48] when time-of-flight measurement is included in the reconstruction algorithm [49].

The absence of physical collimators is one of the major advantages of PET over SPECT, as a larger number of photons reach the detectors and might get detected, thus resulting in higher sensitivity than SPECT. To increase the fraction of photons that are absorbed and therefore detected by a gamma-ray camera, thick (say, 20 mm or thicker) crystals are needed. Thick crystals give high sensitivity at the expense of requiring estimation of the depth of interaction (DOI) [50–52]. Knowledge of the DOI increases spatial resolution as the location of the LOR defined from each coincidental photon pair is determined with less uncertainty (see Figure 1.5) and degradations in the reconstructed data due to parallax errors are reduced [53].

A wide variety of variations on the basic design of a gamma-ray camera have been proposed to allow or improve depth of interaction resolution. This would include placing different crystals (with different decay times) one on top of the other [54, 55], or in more complicated arrangements [56]. Information on the depth of interaction has also been obtained by calculating the ratio of light collected by two adjacent cameras in a detector ring arrangement [57]. Alternatively, an array of photodiodes has been used, in which each photodiode was coupled to one end of a scintillation crystal. The opposite ends of the crystals were coupled to a photomultiplier tube. Each crystal was coated with a lossy reflector, so the ratio of light detected in the photodiode and photomultiplier tube depends on the interaction depth in the crystal, and it is used to determine the depth of interaction on an event by event basis [58]. Algorithms for the estimation the depth of interaction have been proposed as well [53, 59–63].

Compared to SPECT, PET imaging has some disadvantages as well: PET radiopharmaceuticals are more expensive to produce than SPECT radiopharmaceuticals, and the short half-life of PET radionuclides requires an on-site or nearby cyclotron

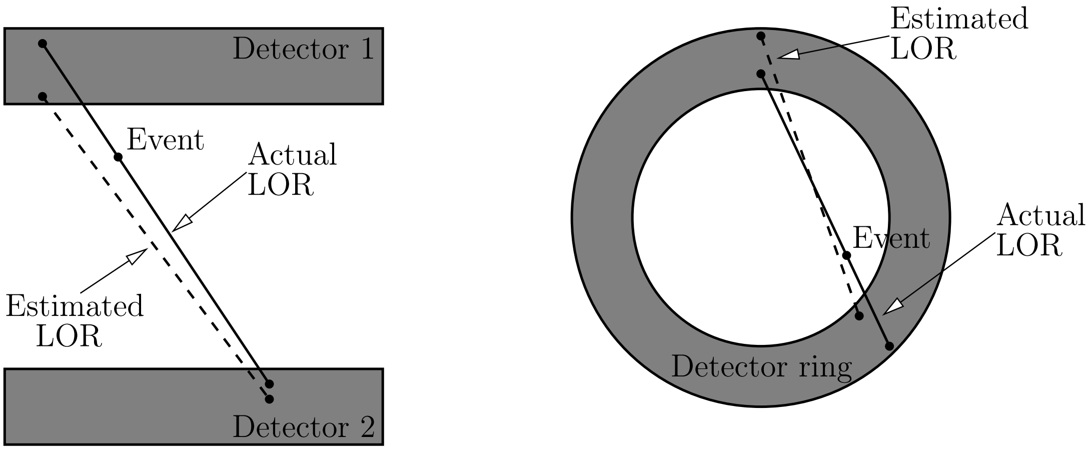


Figure 1.5. Examples of parallax errors on the estimated line of response for two different detector geometries

for their production. PET imagers are more expensive than SPECT scanners and, because of the non-zero positron range, there are inherent physical limitations to their spatial resolution [64]. This translates into a resolution of the order of 1–2 mm for preclinical PET images (see Table 1.1).

1.5 Intent of the Dissertation

The significant progress achieved in recent years on gamma-ray cameras for small-animal imaging along with the development of fast general-purpose computing architectures opened the possibility to process raw gamma-ray camera data in a different and theoretically sound way. Indeed, massively parallel commodity computing hardware, such as Sony's Cell Broadband Engine (Cell BE) and NVIDIA's graphics processing units (GPUs), has made it possible to apply maximum-likelihood estimation methods directly to gamma-ray camera outputs. Accurate list-mode data processing has now become computationally feasible, naturally leading to "optimal" reconstruction codes based on maximum-likelihood.

Even though the concept of fast maximum-likelihood list-mode processing permeates the dissertation in its entirety, the main research goals can be partitioned into four categories, which we now describe in some more detail.

Maximum-likelihood estimation in medical imaging

In maximum-likelihood estimation (MLE) we estimate a parameter $\boldsymbol{\theta}$ by maximizing for $\boldsymbol{\theta}$ the probability (or "likelihood") $\mathrm{pr}(\boldsymbol{g} \mid \boldsymbol{\theta})$ of the observed data $\boldsymbol{g}$. ML estimates $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ have many favorable mathematical properties. For example, as the size of the data $\boldsymbol{g}$ increases, ML estimates are—in a sense that we will clarify later—asymptotically optimal. Another property of ML estimates states that they are unbiased: $\langle \hat{\boldsymbol{\theta}}_{\mathrm{ML}} \rangle_{\boldsymbol{g}} = \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the true, unknown value of the parameter being estimated and $\langle \ldots \rangle_{\boldsymbol{g}}$ denotes average over $\boldsymbol{g}$. One goal of this dissertation is to show how ML techniques—in conjunction with list-mode data—can be effectively used in medical imaging. That would include fast estimation of parameters from photomultiplier tube (PMT) data and reconstruction algorithms (such as the list-mode maximum-likelihood expectation-maximization algorithm, or LMMLEM algorithm) as well.

Development and evaluation of reconstruction codes

As mentioned above, Cell BE and GPU technologies are revolutionizing the world of scientific computing with fast, cheap, and massively parallel-computing devices. In this dissertation, we will elaborate on some of the capabilities and benefits that these technologies offer. That includes 2D and 3D ML position-of-interaction estimation from PMT data and list-mode reconstruction for a simple geometry. Other pieces of code that will be developed are a method for the generation of random numbers on a multicore device (such as a GPU device) and parallel code for evaluation of expectations via Markov chain Monte Carlo (MCMC). These tools will be developed as "utility" tools: they are not the main topic of the dissertation but, nonetheless, they are crucial in the calculation of results for which a closed form cannot be found.

Signal detection with list-mode data

The performance of any detection task is rigorously studied using the receiver operating characteristic (ROC) curve and by measuring the area under it. This area, which is usually called the area under the curve (AUC), is a meaningful figure of merit to assess task performance. Two different observers can be compared by calculating the corresponding AUCs, and the observer with the largest AUC outperforms the other in terms of detection performance. From the theory of signal detection, it is well known that the observer that maximizes the AUC uses the likelihood ratio $\Lambda(\boldsymbol{g})$ as a test statistic

$$\Lambda(\boldsymbol{g}) = \frac{\mathrm{pr}(\boldsymbol{g} \mid H_1)}{\mathrm{pr}(\boldsymbol{g} \mid H_0)},$$

where $\mathrm{pr}(\boldsymbol{g} \mid H_1)$ is the probability density function of the data $\boldsymbol{g}$ under the signal-present hypothesis (H_1) and $\mathrm{pr}(\boldsymbol{g} \mid H_0)$ is the same probability density function but under the signal-absent hypothesis (H_0).

In many practical cases, knowledge of these probability density functions represents an obstacle that is avoided by forcing the observer to be of a particular form, for which calculations are easily carried out without the need of knowing the complete statistics of $\boldsymbol{g}$. Traditionally, the optimal linear observer—known as the Hotelling observer—has represented the only viable alternative to the ideal observer.

List-mode data provide an ideal framework to study signal detection. Indeed, an expression for the probability density function $\mathrm{pr}(\hat{\mathscr{A}} \mid H_k)$ of the list-mode data $\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\}$ given the hypothesis H_k can be derived and used to write a manageable expression for the likelihood ratio on the data $\hat{\mathscr{A}}$

$$\Lambda(\hat{\mathscr{A}}) = \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)}.$$

From realizations of $\Lambda(\hat{\mathscr{A}})$, many quantities (such as the AUC) linked to task performance can be calculated and used to compare the performance of the list-mode ideal observer with other observers, such as the Hotelling observer.

Information content of a photon

In this research goal, we want to quantify how much each photon (or, in general, each attribute vector $\hat{\boldsymbol{A}}^{(j)}$ of the list $\hat{\mathscr{A}}$) contributes to task performance. The figure of merit we will use is the ideal observer's signal-to-noise ratio (SNR). The theory can be worked out quite easily for the signal-known-exactly/background-known-exactly (SKE/BKE) case [5]. In such a case, the likelihood of the data $\hat{\mathscr{A}}$ under the hypothesis H_k is

$$\begin{aligned}\mathrm{pr}(\hat{\mathscr{A}} \mid H_k) = \left[\frac{1}{\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_k(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}\right]^J \prod_{j=1}^{J} \bigg\{ \int_\infty \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{A}^{(j)}) \times \\ \times \left[\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_k(\boldsymbol{r}) \mathrm{pr}(\boldsymbol{A}^{(j)} \mid \boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right] \mathrm{d}\boldsymbol{A}^{(j)} \bigg\},\end{aligned}$$

where the function $f_k(\boldsymbol{r})$ denotes the real object under the hypothesis H_k, the integer number J is the number of elements in the list $\hat{\mathscr{A}}$, and, finally, the function $s(\boldsymbol{r})$ denotes the sensitivity of the imaging system for point $\boldsymbol{r}$ in the field of view. The subscript "FOV" in $\int_{\mathrm{FOV}} \ldots \mathrm{d}\boldsymbol{r}$ denotes integration over the field of view, while "∞" in $\int_\infty \ldots \mathrm{d}\boldsymbol{A}^{(j)}$ denotes integration over the space of attribute vector $\boldsymbol{A}^{(j)}$.

If we take the logarithm of $\mathrm{pr}(\hat{\mathscr{A}} \mid H_k)$ and do some manipulations

$$\ln \mathrm{pr}(\hat{\mathscr{A}} \mid H_k) = \sum_{j=1}^{J} \left\{ \ln \int_{\mathrm{FOV}} s(\boldsymbol{r}) f_k(\boldsymbol{r}) \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\,\mathrm{d}\boldsymbol{r} - \ln \int_{\mathrm{FOV}} s(\boldsymbol{r}) f_k(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \right\}.$$

The ideal observer uses the likelihood ratio $\Lambda(\hat{\mathscr{A}}) = \mathrm{pr}(\hat{\mathscr{A}} \mid H_1)/\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)$ or its logarithm $\lambda(\hat{\mathscr{A}}) = \ln \Lambda(\hat{\mathscr{A}})$ as test statistic. It is easy to see that

$$\begin{aligned}\lambda(\hat{\mathscr{A}}) &= \sum_{j=1}^{J} \left\{ \ln \frac{\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_1(\boldsymbol{r}) \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_0(\boldsymbol{r}) \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} - \ln \frac{\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_1(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} s(\boldsymbol{r}) f_0(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\} = \\ &= \sum_{j=1}^{J} \lambda(\{\hat{\boldsymbol{A}}^{(j)}\}),\end{aligned}$$

in which the last part of the expression above shows that $\lambda(\hat{\mathscr{A}})$ is the sum of a large number of independent quantities $\lambda(\{\hat{\boldsymbol{A}}^{(j)}\})$. Thus, it is reasonable to assume that the log-likelihood ratio $\lambda(\hat{\mathscr{A}})$ is normally distributed. Normally distributed log-likelihood ratios enjoy many properties [65], one of them can be summarized as follows

$$\mathrm{SNR}_{\lambda}^{2} = -2\big\langle \lambda(\hat{\mathscr{A}}) \big\rangle_{\hat{\mathscr{A}}|H_0} = -2 \sum_{j=1}^{J} \big\langle \lambda(\{\hat{\boldsymbol{A}}^{(j)}\}) \big\rangle_{\hat{\boldsymbol{A}}^{(j)}|H_0},$$

where $\langle \ldots \rangle_{\hat{\mathscr{A}}|H_0}$ denotes expectation over the list-mode data $\hat{\mathscr{A}}$, statistically conditioned on H_0, and

$$\big\langle \lambda(\{\hat{\boldsymbol{A}}\}) \big\rangle_{\hat{\boldsymbol{A}}|H_0} = \int_{\infty} \lambda(\{\hat{\boldsymbol{A}}\}) \mathrm{pr}(\hat{\boldsymbol{A}} \mid H_0)\, \mathrm{d}\hat{\boldsymbol{A}}.$$

The quantity above numerically quantifies the contribution of each event to the $\mathrm{SNR}_{\lambda}^{2}$.

1.5.1 Structure of the Dissertation

The remainder of the dissertation is structured in five chapters. In Chapter 2 we will review the main concepts of image quality and we will discuss the two observers that we will be concerned with in this dissertation. In Chapter 3 we will describe maximum-likelihood estimation and we will review two algorithms for maximum-likelihood estimation. The next chapter—Chapter 4—begins by presenting a treatment of gamma-ray cameras with an emphasis on the statistical characterization of the data they produce. This is followed by the description of two maximum-likelihood algorithms: the first to perform estimation of position of interaction from gamma-ray PMT data and the second to carry out image reconstruction with list-mode data. Task performance with list-mode data and its processing for the case of emission tomography conclude Chapter 4. Parallel computational methods is the main topic of Chapter 5. Finally, Chapter 6 summarizes our findings and discusses in some detail a possible application of list-mode data to CADe/CADx X-ray mammography.

CHAPTER 2
Assessment of Image Quality in Medical Imaging

Images are routinely produced in the clinic to diagnose or monitor one or more diseases. The accuracy with which medical experts can correctly diagnose a pathology depends on their degree of expertise and on the "quality" of the images as well. In medical imaging we are interested in objectively assessing image quality to devise a well-defined parameter with which characterize the performance of trained experts (or "observers") and imaging systems for a particular clinical task, say, tumor detection.

This chapter begins by presenting the basic notions of assessment of image quality. The treatment is then made more formal by discussing statistical decision theory and signal detection. Objective assessment of image quality is applied to the problem of signal detection by introducing the so-called receiver operating characteristic curve. The chapter continues with the description of the two decision-making tools used in this dissertation: the ideal observer and the Hotelling observer. While the theory developed in this chapter is mainly presented for the case of images represented as finite sets of numbers, the same concepts hold true if we define images as functions of continuous variables, as discussed towards the end of the chapter.

NOTE: Parts of this chapter have been adapted from L. Caucci, *"Point Detection and Hotelling Discriminant: An Application in Adaptive Optics,"* Master's Thesis, The University of Arizona, Tucson, AZ, 2007 [66].

2.1 General Considerations

The field of image science and data processing is concerned with the task of generating and manipulating data to be presented to an observer who is then asked to make a decision about the presence of a signal, or to estimate one or more numerical parameters from the data. For example, in the case of radiology, the goal of an image scientist is to devise an imaging system and/or an image reconstruction algorithm that will provide images adequate for a correct diagnosis by a radiologist. In the image processing literature, the concept of image quality is not always stated and defined precisely. Indeed, some papers simply limit themselves to the description of a new image enhancement/reconstruction algorithm and then show a few images obtained with it along with the images that one would obtain with a previous method; no well-defined objective definition of image quality is provided, and the interpretation of the results is left to the reader who is supposed to visually compare these images and conclude that the new algorithm "performs better" because it provides images that "look better." When the interpretation of image quality is not left to the reader, metrics—such as resolution, noise power spectrum, and modulation transfer function—are provided [67]. Even though these physical measurements are easy to calculate and understand, it is not clear, in general, to what extent they correlate with task performance. As it turns out, any naive way of defining image quality will almost surely fail [68].

Images are usually generated to fulfill a particular purpose. For this reason, any meaningful way to assess the performance of a system producing images should take into account the purpose—or "task"—for which these images are produced. Assessing image quality means establishing how much the images produced suit the task of interest and how well an observer can perform that task with these images [69–73]. The observer performance can be assessed with a cost function or by evaluating the probability that an error is made. Generally, we can consider two types of tasks: classification or estimation. In the case of classification, we want to "label" the object that produced the image by associating it to a particular category or class. For example, in the case of a satellite image of a field, we may want to classify the field according to what is grown on it, or we may want to classify an astronomical image—

or a sequence of related astronomical images—depending on whether or not a planet orbiting a star is present at a particular location. In the case of medical imaging, we might be interested in knowing if a tumor is present or not. It is not necessary to assume the classification task be binary: any m-ary classification can be implemented by performing a sequence of binary classifications. In the case of estimation, we are interested in estimating numerical information from the image(s) we are observing. For example, from the aerial view of a field, we may want to estimate the extent of such a field. In astronomy, we may want to estimate the motion of a newly-discovered planet orbiting a star, while in medical imaging we might be interested in the location and size of a tumor mass.

In some cases, assessing image quality can be a laborious and time-consuming process. Indeed, if we consider again the case of a radiologist examining images to detect, for example, a tumor, objective assessment of image quality should take into consideration the eye-brain system and the complicated processes that take place within it. Such processes are still so poorly understood that psychophysical studies are usually necessary. In a psychophysical study, trained human observers are presented images and they are asked to make a decision about the presence of a signal. The results (for example, fraction of incorrect classifications for each observer) are then averaged over all observers. A more practical alternative to psychophysical studies is to use mathematical observers. These mathematical observers can replace a human observer if their performance is found to be in good agreement with human performance evaluated through psychophysical studies.

For the case of a binary classification, and if we denote with $\boldsymbol{g}$ the data we have measured, we can assume that a binary decision is based on the value that a test statistic function $t(\boldsymbol{g})$ takes on the observed data $\boldsymbol{g}$. A yes/no decision can then be made by comparing $t(\boldsymbol{g})$ with a decision threshold or cutoff t_{cut} [69, 74]. As it will be discussed later, changing the value of t_{cut} allows the plotting of a curve from which meaningful figures of merit for the task of interest can be calculated. Two particular test statistics $t(\boldsymbol{g})$ will be discussed in this chapter: the ideal observer and the Hotelling observer. We will show that, as its name suggests, the ideal observer is optimal in the sense that it sets the upper limit on the performance of any observer on the task of interest. On the other hand, the Hotelling observer has the advantage of being linear and, in general, easy to compute and handle mathematically.

2.2 Statistical Decision Theory

The objective of statistical decision theory [75–77] is to help a decision maker select one alternative among many. The selection of one alternative out of all the possible ones is based on some sort of partial, inaccurate, or noisy data about an event, and it is carried out by taking into account the consequences and risks associated with each available alternative. These kinds of studies began in the middle of the 18th century when Thomas Bayes (c. 1701–1761) analyzed the problem of deducing which one of two possible causes most likely resulted in the observed data [78]. Usually, there is a cost associated to each alternative and, because the cost depends on the alternative and on the actual causes that produced the observed data, the objective is to make it possible for the decision maker to select the alternative that best explains the observed data and minimizes the average cost.

Signal-detection theory [74] results from the combination of statistical decision theory and the study of the statistical properties and distribution of noise and signals. When a signal is transmitted through a communication system and received by a receiver, it usually undergoes some distortion due to noise. In many cases, the noise is random, so the receiver is faced with the problem of deciding which signal was most likely transmitted. The task of interest in signal-detection theory is to determine whether or not a weak non-random auditory or visual signal is present in a noisy random background. In other words, signal detection is the task that asks the observer: *"Do you see a signal?"* In psychology, detection theory is used in medical and non-medical diagnosis and for the study of memory, cognition, and sensory psychology [79]. Detection theory is extensively used in radar applications [80, 81] and medicine [69, 82, 83] as well.

Detection theory embodies a way to assess observer performance for the task of interest. If such a task is the detection of a signal in a noisy background, any error the observer commits can be of two types: we can have "misses" or "false negatives" (the signal is actually present but the observer concludes that it is not) and "false alarms" or "false positives" (the signal is not present but the observer believes it is). For example, a radiologist may be looking at a radiograph in order to determine whether a patient has a cancer or not. The radiologist will issue a false negative if he/she detects no tumor but the patient has one. On the other hand, the radiologist will issue a false alarm if, from inspection of the radiograph, he/she concludes that the patient has a cancer, but the patient is in fact healthy. Generally, the cost of a miss is higher than the cost of a false alarm, so it is desirable to keep the number of misses very small with respect the number of false alarms.

According to [74], there are three psychophysical procedures for decision making. They are: "yes/no detection task," "rating procedure," and "m-alternative forced choice." In a yes/no detection task, the observer is presented one stimulus at a time, and he/she is asked to say whether or not a particular signal is present in the stimulus he/she perceives. So the output the observer produces is a yes/no answer. The reiteration of this experiment many times with different stimuli gives a "stimulus-response matrix" [74] for that particular observer. This matrix, which is of size 2×2, reports the number of misses (false negatives), the number of hits (true positives), the number of false alarms (false positives) and the number of correct rejections (true negatives). Table 2.1 summarizes these possible outcomes.

In the case of a detection task, the probability of a false positive and the probability of a true positive can be estimated from the stimulus-response matrix. This information can be represented as a point on a curve [74, 84], which is called "re-

Signal **Response**	**Present**	**Absent**
“Yes”	“hit” or “true positive”	“false alarm” or “false positive”
“No”	“miss” or “false negative”	“correct rejection” or “true negative”

Table 2.1. Possible scenarios for a yes/no detection task

ceiver operating characteristic” (ROC) curve. More specifically, an ROC curve is a plot of the probability of false positive, which is also called “false-positive fraction” (FPF), versus the probability of true positive, which is also called “true-positive fraction” (TPF) [69, 85].

In a rating procedure, the observer is presented one stimulus at a time, and the output the observer produces is a number (in a rating scale) which expresses the likelihood that the observation was caused by signal plus noise as opposed to just noise [74]. For example, in a tumor-detection task, a 1–5 scale could be used: “1” may mean “tumor definitely absent,” and “5” may mean “tumor definitely present” [69]. Intermediate values are to be interpreted as intermediate levels of certainty/uncertainty that the tumor is present. A rating procedure can easily be converted into a yes/no detection task by thresholding. Varying the threshold allows one to get more points on the FPF-TPF plane, which can be fit to get an ROC curve [74].

In an m-alternative forced choice (mAFC), the observer is presented m stimuli at a time. Only one stimulus out of the $m > 1$ contains the signal of interest. The observer—who knows that only one stimulus contains the signal—is asked to select the stimulus that he/she believes most likely contains the signal in addition to noise [74]. A particular case of the m-alternative forced choice is the 2-alternative forced choice (2AFC), in which the observer is presented two stimuli at a time, one where the signal is present and the other where it is not, and the observer is asked to decide in which stimulus he/she believes the signal is present [74].

2.3 The ROC Curve

As briefly alluded to earlier, the ROC curve is a plot of the true-positive fraction (also called "sensitivity" [69,86]) versus the false-positive fraction. The true-negative fraction is also called "specificity" [69,86]. In order to plot an ROC curve, different values for the true-positive fraction (TPF) and the false-positive fraction (FPF) are needed. As noted above, this can be carried out by changing some of the observer's internal parameters such as the threshold t_{cut} used to convert a rating procedure into a yes/no detection task. While ROC analysis plays a key role in radiology [73,87–96], it has found applications in numerous other fields, such as information retrieval, weather forecasting, psychology, aptitude testing, survey research, medical imaging, materials testing, and polygraph lie detection [97–101].

For a 2-alternative forced choice test [92,102], we can introduce a "figure of merit" for the decision maker as the probability of a correct answer. It can be proved [74] that the area between the ROC curve for $0 \leqslant \mathrm{FPF} \leqslant 1$ and the horizontal axis (called "area under the ROC curve" or AUC) is the probability of a correct answer in a 2-alternative forced choice test [92]. The AUC provides a figure of merit for the decision maker in the sense that it provides a way to assess its performance. This figure of merit ranges from $1/2$ for a worthless decision maker, to 1 for a perfect one.

Besides the area under the ROC curve, other figures of merit for the assessment of the observer's ability to detect a signal have been introduced. The general idea is to use a single number as a measure of system performance for the task of interest. For example, if both data distributions under the signal-present hypothesis H_1 and the signal-absent hypothesis H_0 are univariate Gaussian with means μ_{H_0} and μ_{H_1}, and variances $\sigma_{H_0}^2$ and $\sigma_{H_1}^2$, respectively, the quantity [74,103]

$$d_A^2 = \frac{\left[\mu_{H_0} - \mu_{H_1}\right]^2}{\Pr(H_0)\,\sigma_{H_0}^2 + \Pr(H_1)\,\sigma_{H_1}^2} \tag{2.1}$$

is a meaningful figure of merit for the assessment of the observer's ability to detect the signal of interest. In the expression above, $\Pr(H_0)$ and $\Pr(H_1)$ denote the probability of the hypotheses H_0 and H_1, respectively. The quantity d_A just introduced is known in the literature as "index of detectability" [103]. For the special case of $\sigma_{H_0}^2 = \sigma_{H_1}^2 = \sigma^2$ the index of detectability d_A is denoted as d' and it is given by [103–106]

$$d'^2 = \frac{\left[\mu_{H_0} - \mu_{H_1}\right]^2}{\sigma^2}. \tag{2.2}$$

We have seen that if we want to get an ROC curve, we need to vary the observer's false-positive fraction and true-positive fraction. To do this, we may think of the observer as composed of two independent mechanisms: the "sensory" mechanism and the "decision" mechanism [74]. For example, the sensory mechanism can output a number, say $t(\boldsymbol{g})$, where $\boldsymbol{g}$ is the observed data. The function $t(\boldsymbol{g})$ is called a "test statistic." We will make the assumption that the vector $\boldsymbol{g}$ is a vector in a Hilbert space. In many practical cases, $\boldsymbol{g}$ is a vector in an Euclidean space. However, if we are carrying out a theoretical analysis, we may interpret $\boldsymbol{g}$ as a function $g(\boldsymbol{r})$ of, say, a 2D or 3D spatial variable $\boldsymbol{r}$.

Given a fixed threshold t_{cut}, a decision is made by the decision mechanism using the following rule: say "signal present" if $t(\boldsymbol{g}) > t_{\text{cut}}$, otherwise say "signal absent." Two restrictions are implicit in the process just outlined. First, no randomness in the decision rule is allowed and, second, for every $\boldsymbol{g}$, one and only one alternative between "signal present" and "signal absent" is selected [69]. This decision rule partitions the set of vectors $\boldsymbol{g}$ into two distinct and non-overlapping regions. A value for the threshold t_{cut} corresponds to a point on the ROC curve, and so t_{cut} parametrizes the ROC curve. More formally, if $\boldsymbol{s}$ is the signal vector we want to detect and the random vector $\boldsymbol{n}$ is the noise, the observed data vector $\boldsymbol{g}$ can be written under hypotheses H_0 and H_1 as:

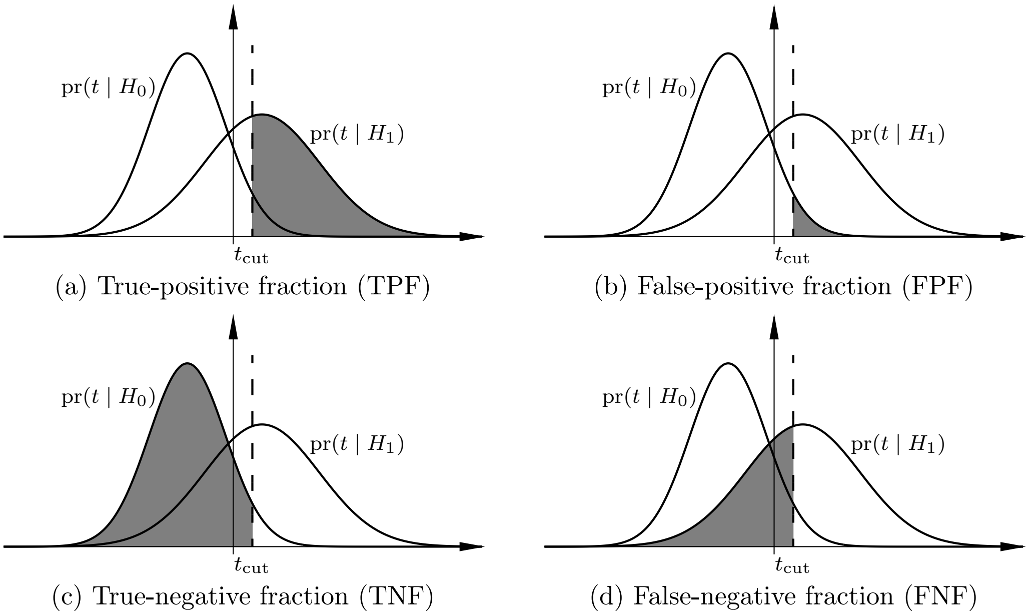
$$\begin{aligned} H_0 :& \quad \boldsymbol{g} = \boldsymbol{n}, \\ H_1 :& \quad \boldsymbol{g} = \boldsymbol{s} + \boldsymbol{n}, \end{aligned}$$

respectively. Vectors $\boldsymbol{g}$, $\boldsymbol{n}$, and $\boldsymbol{s}$ all belong to the same Hilbert space. The test statistic function $t(\boldsymbol{g})$ is a scalar function of the random vector $\boldsymbol{g}$ and so it is a random variable itself. It makes sense to consider the probability density functions $\mathrm{pr}(t \mid H_0)$ and $\mathrm{pr}(t \mid H_1)$ of $t(\boldsymbol{g})$ under the hypotheses H_0 and H_1, respectively. From these probability density functions, the true-positive fraction, false-positive fraction, true-negative fraction, and false-negative fraction (all of them as functions of the threshold t_{cut}) can be formally defined as [65, 69]:

$$\mathrm{TPF}(t_{\mathrm{cut}}) = \int_{t_{\mathrm{cut}}}^{\infty} \mathrm{pr}(t \mid H_1)\,\mathrm{d}t, \qquad \mathrm{FPF}(t_{\mathrm{cut}}) = \int_{t_{\mathrm{cut}}}^{\infty} \mathrm{pr}(t \mid H_0)\,\mathrm{d}t,$$

$$\mathrm{TNF}(t_{\mathrm{cut}}) = \int_{-\infty}^{t_{\mathrm{cut}}} \mathrm{pr}(t \mid H_0)\,\mathrm{d}t, \qquad \mathrm{FNF}(t_{\mathrm{cut}}) = \int_{-\infty}^{t_{\mathrm{cut}}} \mathrm{pr}(t \mid H_1)\,\mathrm{d}t.$$

These probabilities are shown as shaded areas in Figure 2.1. Figure 2.2 shows the corresponding ROC curve.



(a) True-positive fraction (TPF)

(b) False-positive fraction (FPF)

(c) True-negative fraction (TNF)

(d) False-negative fraction (FNF)

FIGURE 2.1. Probabilities for the four possible outcomes for a yes/no detection task

If the plots of the densities $\mathrm{pr}(t \mid H_0)$ and $\mathrm{pr}(t \mid H_1)$ are well separated and do not overlap, the system performance will be very high because a wrong decision will

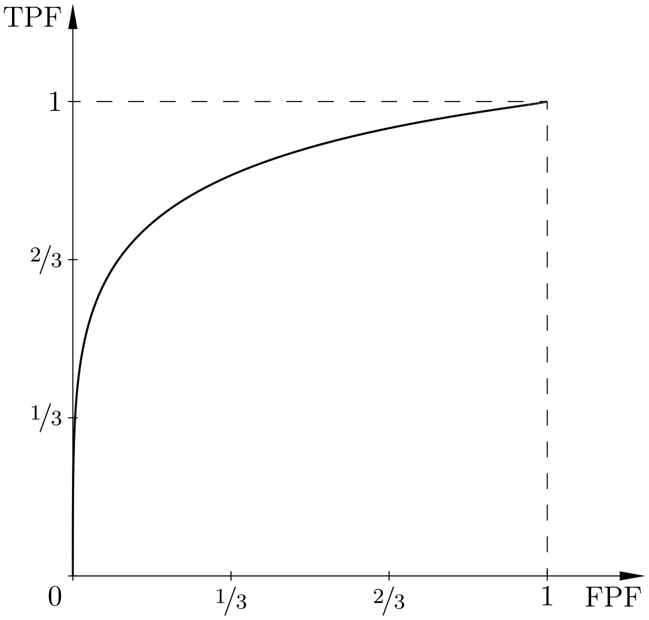


Figure 2.2. ROC curve for a yes/no detection task

have a very small probability. The AUC [69, 98]

$$\mathrm{AUC} = -\int_{-\infty}^{\infty} \mathrm{TPF}(t_{\mathrm{cut}}) \frac{\mathrm{dFPF}(t_{\mathrm{cut}})}{\mathrm{d}t_{\mathrm{cut}}} \, \mathrm{d}t_{\mathrm{cut}} \tag{2.3}$$

will then be close to 1. The minus sign in (2.3) arises since $\mathrm{FPF}(t_{\mathrm{cut}}) \to 1$ as $t_{\mathrm{cut}} \to -\infty$ [69]. On the other hand, if the densities $\mathrm{pr}(t \mid H_0)$ and $\mathrm{pr}(t \mid H_1)$ almost completely overlap, the probability of a detection error will be close to the probability of a correct decision: the observer is doing almost nothing but guessing and, consequently, $\mathrm{TPF}(t_{\mathrm{cut}}) \approx \mathrm{FPF}(t_{\mathrm{cut}})$ for all t_{cut}. The AUC will therefore be close to $1/2$.

Another figure of merit we can consider is the signal-to-noise ratio (SNR) associated to the test statistic $t(\boldsymbol{g})$ [69]

$$\mathrm{SNR}^2_{t(\boldsymbol{g})} = \frac{\left[\left\langle t(\boldsymbol{g})\right\rangle_{H_1} - \left\langle t(\boldsymbol{g})\right\rangle_{H_0}\right]^2}{\frac{1}{2}\left(\sigma^2_{t(\boldsymbol{g})|H_1} + \sigma^2_{t(\boldsymbol{g})|H_0}\right)}, \tag{2.4}$$

where

$$\langle t(\boldsymbol{g})\rangle_{H_k} = \int_{-\infty}^{\infty} t \, \mathrm{pr}(t \mid H_k) \, \mathrm{d}t,$$

and

$$\sigma^2_{t(\boldsymbol{g})|H_k} = \int_{-\infty}^{\infty} \left[t - \langle t(\boldsymbol{g})\rangle_{H_k}\right]^2 \mathrm{pr}(t \mid H_k) \, \mathrm{d}t,$$

for $k \in \{0, 1\}$ are the mean and variance of $t(\boldsymbol{g})$ under the hypothesis H_k. When the test statistic $t(\boldsymbol{g})$ is normally distributed under both hypotheses, the AUC and $\mathrm{SNR}_{t(\boldsymbol{g})}$ are related via [65, 69]:

$$\mathrm{AUC} = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\mathrm{SNR}_{t(\boldsymbol{g})}}{2}\right), \tag{2.5}$$

in which

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} \, \mathrm{d}u.$$

The expression in (2.5) allows us to calculate $\mathrm{SNR}_{t(\boldsymbol{g})}$ if we know the AUC [65, 69]:

$$\mathrm{SNR}_{t(\boldsymbol{g})} = 2 \operatorname{erf}^{-1}(2\,\mathrm{AUC} - 1).$$

It is important to note that the relation above is valid only when $t(\boldsymbol{g})$ follows normal statistics under the H_0 and H_1 hypotheses. If the hypothesis of normality of $t(\boldsymbol{g})$ is not satisfied, we can use the expression above to define another figure of merit for detection performance [65, 69]:

$$d_A = 2 \operatorname{erf}^{-1}(2\,\mathrm{AUC} - 1). \tag{2.6}$$

To conclude this discussion on ROC curves, we note that, in many applications, we are interested in detecting a signal and, if the signal is present, in estimating its location (or some other parameter). An example from the field of medicine is the detection of a tumor, paired with finding its location (within some tolerance). A non-medical example would be the detection and localization of a dim planet orbiting a star [66, 107]. It therefore makes sense to extend the concept of ROC curves to such hybrid problems and define "localization ROC curves" (LROCs) [108, 109] or, in the more general case, "estimation ROC curves" (EROCs) [110, 111]. Areas under these curves (ALROC and AEROC, respectively) can be considered as well.

2.4 The Ideal Observer

We noted before that whenever a decision is made, there is a cost associated to it. For example, if a radiologist believes that a patient has cancer, a biopsy is usually carried out in order to verify if the radiologist's diagnosis is correct and the patient needs treatment.

Recall that the hypotheses H_0 and H_1 were defined as "signal absent" and "signal present," respectively. Let $\Pr(H_0)$ be the probability of the event "hypothesis H_0 is true" and let $\Pr(H_1) = 1 - \Pr(H_0)$ be the probability of the event "hypothesis H_1 is true." Hypotheses H_0 and H_1 are mutually exclusive and exactly one of them is true. We can consider a cost matrix [69, 112] of the following form

$$\mathbf{C} = \begin{pmatrix} c_{0,0} & c_{0,1} \\ c_{1,0} & c_{1,1} \end{pmatrix},$$

where each $c_{i,j}$ is defined as the cost (or, if negative, the profit) incurred by choosing hypothesis H_i when hypothesis H_j is actually true. With this notation, the cost of a false alarm is given by $c_{1,0}$, while the cost of a miss is $c_{0,1}$. Suppose that our observer uses the decision function $\Lambda(\boldsymbol{g})$ and, when presented with the data vector $\boldsymbol{g}$, selects decision D_1 "signal present" if the value $\Lambda(\boldsymbol{g})$ is greater than or equal to a fixed decision level t_{cut}; the observer selects decision D_0 "signal absent" otherwise. Define the following sets [113]:

$$\Gamma_0 = \{\boldsymbol{g} \mid \Lambda(\boldsymbol{g}) < t_{\text{cut}}\}, \qquad \text{and} \qquad \Gamma_1 = \{\boldsymbol{g} \mid \Lambda(\boldsymbol{g}) \geqslant t_{\text{cut}}\}.$$

The goal is to select the function $\Lambda(\boldsymbol{g})$ in such a way that the expected average cost (or "risk") [112]

$$\begin{aligned}\bar{c} = \Pr(H_0) &\left[c_{0,0} \int_{\Gamma_0} \text{pr}(\boldsymbol{g} \mid H_0)\, \text{d}\boldsymbol{g} + c_{1,0} \int_{\Gamma_1} \text{pr}(\boldsymbol{g} \mid H_0)\, \text{d}\boldsymbol{g}\right] + \\ &+ \Pr(H_1) \left[c_{0,1} \int_{\Gamma_0} \text{pr}(\boldsymbol{g} \mid H_1)\, \text{d}\boldsymbol{g} + c_{1,1} \int_{\Gamma_1} \text{pr}(\boldsymbol{g} \mid H_1)\, \text{d}\boldsymbol{g}\right]\end{aligned}$$

is minimized. This strategy takes the name of "Bayes criterion" [78] and the value of $\bar{c}$ is called "Bayes risk." A detector that employs the Bayes criterion is called "Bayes detector." The Bayes criterion is said to be optimal in the sense that it minimizes the expect average cost $\overline{c}$. With some algebraic manipulation, the average cost $\overline{c}$ can be rewritten in the following way [69, 114, 115]:

$$\begin{aligned}\overline{c} &= c_{0,1}\mathrm{Pr}(H_1) + c_{0,0}\mathrm{Pr}(H_0) + \\ &+ \int_{\Gamma_1} \big[(c_{1,0} - c_{0,0})\mathrm{Pr}(H_0)\mathrm{pr}(\boldsymbol{g} \mid H_0) - (c_{0,1} - c_{1,1})\mathrm{Pr}(H_1)\mathrm{pr}(\boldsymbol{g} \mid H_1)\big]\, \mathrm{d}\boldsymbol{g}.\end{aligned} \tag{2.7}$$

Because the vector $\boldsymbol{g}$ is assumed to belong to Γ_1 if $\Lambda(\boldsymbol{g}) \geqslant t_{\mathrm{cut}}$ and to Γ_0 if $\Lambda(\boldsymbol{g}) < t_{\mathrm{cut}}$, finding the optimal Γ_1 is equivalent to finding a test statistic $\Lambda(\boldsymbol{g})$ that is optimal, and vice versa. The quantity $c_{0,1}\mathrm{Pr}(H_1) + c_{0,0}\mathrm{Pr}(H_0)$ that appears in (2.7) does not depend on Γ_1, so $c_{0,1}\mathrm{Pr}(H_1) + c_{0,0}\mathrm{Pr}(H_0)$ can be ignored when finding the choice of Γ_1 that minimizes the cost $\overline{c}$. Assuming that $c_{1,0} > c_{0,0}$ and $c_{0,1} > c_{1,1}$ (the cost incurred in making a wrong decision is greater than the cost incurred in making a correct decision), a way to minimize the quantity in (2.7) is to have the integrand negative whenever $\boldsymbol{g} \in \Gamma_1$ [115]. In other words, for all $\boldsymbol{g} \in \Gamma_1$ we want to have [113, 115]

$$(c_{0,1} - c_{1,1})\mathrm{Pr}(H_1)\mathrm{pr}(\boldsymbol{g} \mid H_1) > (c_{1,0} - c_{0,0})\mathrm{Pr}(H_0)\mathrm{pr}(\boldsymbol{g} \mid H_0).$$

Equivalently [116]

$$\frac{\mathrm{pr}(\boldsymbol{g} \mid H_1)}{\mathrm{pr}(\boldsymbol{g} \mid H_0)} > \frac{c_{1,0} - c_{0,0}}{c_{0,1} - c_{1,1}} \frac{\mathrm{Pr}(H_0)}{\mathrm{Pr}(H_1)}. \tag{2.8}$$

This tells us how the function $\Lambda(\boldsymbol{g})$ and the threshold t_{cut} should be selected. Indeed we have

$$\Lambda(\boldsymbol{g}) = \frac{\mathrm{pr}(\boldsymbol{g} \mid H_1)}{\mathrm{pr}(\boldsymbol{g} \mid H_0)},$$

and

$$t_{\mathrm{cut}} = \frac{c_{1,0} - c_{0,0}}{c_{0,1} - c_{1,1}} \frac{\mathrm{Pr}(H_0)}{\mathrm{Pr}(H_1)}.$$

This particular choice of $\Lambda(\boldsymbol{g})$ is called "likelihood ratio" [112, 115]. It can be shown that the likelihood ratio test statistic maximizes the true-positive fraction for every value of the false-positive fraction, so it also maximizes the area under the ROC curve (see Figure 2.2) [65, 69]. For this reason, the "ideal observer" is defined as the observer that uses $\Lambda(\boldsymbol{g})$ (or any monotonic function of $\Lambda(\boldsymbol{g})$) as test statistic.

As a special case, consider data vectors $\boldsymbol{g}$ (with M components) normally distributed under both hypotheses H_0 and H_1, with known means $\overline{\boldsymbol{g}_0}$ and $\overline{\boldsymbol{g}_1}$, and known covariance matrices $\mathbf{K}_{\boldsymbol{g}|H_0} = \mathbf{K}_{\boldsymbol{g}|H_1} = \mathbf{K}_{\boldsymbol{g}}$ (whenever $\mathbf{K}_{\boldsymbol{g}|H_0}$ and $\mathbf{K}_{\boldsymbol{g}|H_1}$ are the same, we say that the set of probability distributions of $\boldsymbol{g}$ under hypotheses H_0 and H_1 is "homoscedastic" [69]). In other words, the densities $\mathrm{pr}(\boldsymbol{g} \mid H_0)$ and $\mathrm{pr}(\boldsymbol{g} \mid H_1)$ are

$$\mathrm{pr}(\boldsymbol{g} \mid H_k) = \frac{1}{\sqrt{(2\pi)^M \det(\mathbf{K}_{\boldsymbol{g}})}} \exp\left[-\frac{1}{2}(\boldsymbol{g} - \overline{\boldsymbol{g}_k})^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} (\boldsymbol{g} - \overline{\boldsymbol{g}_k})\right],$$

for $k \in \{0, 1\}$. In this case, we have

$$\begin{aligned} \Lambda(\boldsymbol{g}) = \frac{\mathrm{pr}(\boldsymbol{g} \mid H_1)}{\mathrm{pr}(\boldsymbol{g} \mid H_0)} &= \exp\left[-\frac{1}{2}\left(\overline{\boldsymbol{g}_1}^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} \overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} \overline{\boldsymbol{g}_0}\right)\right] \times \\ &\times \exp\left[\frac{1}{2}\left(\overline{\boldsymbol{g}_1}^{\mathsf{T}} - \overline{\boldsymbol{g}_0}^{\mathsf{T}}\right) \mathbf{K}_{\boldsymbol{g}}^{-1} \boldsymbol{g} + \frac{1}{2} \boldsymbol{g}^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\right] = \\ &= C \exp\left[(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} \boldsymbol{g}\right]. \end{aligned}$$

Because the logarithm is a strictly increasing function, we can take the logarithm on both sides of (2.8) and leave the inequality still satisfied. For the case of the normally distributed data vector above, we can then define

$$\lambda(\boldsymbol{g}) = \ln\left[\Lambda(\boldsymbol{g})\right] = (\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})^{\mathsf{T}} \mathbf{K}_{\boldsymbol{g}}^{-1} \boldsymbol{g} + \ln C = \boldsymbol{w}^{\mathsf{T}} \boldsymbol{g} + \ln C.$$

This shows that, apart from an additive constant $\ln C$—that can be incorporated in the logarithm of the threshold t_{cut}—the test statistic for the case of normally distributed data is linear and given by the dot product between the vector $\boldsymbol{w} = (\mathbf{K}_{\boldsymbol{g}}^{-1})^{\mathsf{T}} (\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}) = \mathbf{K}_{\boldsymbol{g}}^{-1} (\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})$ and the data vector $\boldsymbol{g}$. The function $\lambda(\boldsymbol{g}) = \ln \Lambda(\boldsymbol{g})$ is usually called "log-likelihood ratio" [69].

2.5 The Hotelling Observer

Linear discriminant analysis [117,118] is concerned with the study of classifiers of the form $t(\boldsymbol{g}) = \boldsymbol{w}^{\mathsf{T}} \boldsymbol{g}$, where $\boldsymbol{g}$ is the data vector. One reason for using linear test statistics is that they are easy to calculate and analyze mathematically. Furthermore, there is some evidence that linear test statistics might deliver performances comparable to those of the human observer [119] for a detection task. Computing the dot product $\boldsymbol{w}^{\mathsf{T}}\boldsymbol{g}$ can be interpreted as projecting the vector $\boldsymbol{g}$ along a line in the direction of $\boldsymbol{w}$. A binary classification is then carried out by comparing $t(\boldsymbol{g})$ to a threshold t_{cut}: if the projection $t(\boldsymbol{g}) = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{g}$ is greater than or equal to t_{cut}, then it is assumed that the vector $\boldsymbol{g}$ belongs to the class Γ_1; if $\boldsymbol{w}^{\mathsf{T}}\boldsymbol{g} < t_{\text{cut}}$, then $\boldsymbol{g}$ is assumed to belong to the class Γ_0. Equivalently, we can say that decision D_k is taken whenever the data vector $\boldsymbol{g}$ is thought to belong to the class Γ_k [69]. Clearly, the goal is to find a vector $\boldsymbol{w}$ that, in a given sense, maximizes the "separation" between the projection along $\boldsymbol{w}$ of the vectors in Γ_0 and the projection along $\boldsymbol{w}$ of the vectors in Γ_1. Many measures of separations between two classes have been proposed in the past; among them is the "coefficient of racial likeness T^2" [120] introduced in 1931 by Harold Hotelling (September 29, 1895–December 26, 1973) to assess the likelihood that two sets of samples were drawn from distributions with the same mean.

The study of linear discriminant functions continued with the analysis carried out in 1936 by Ronald A. Fisher [121]. In order to classify an object into one of two classes and measure class separability, Fisher employed a linear function of the data and proposed to use the ratio of the among-classes sum-of-squares of this linear function to its within-classes sum-of-squares as measure of discrimination between the two classes. This ratio, which turned out to be strongly related [122] to Hotelling's T^2 coefficient of racial likeness, can be calculated by means of the "between classes" and "within classes" scatter matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, which are defined as:

$$\mathbf{S}_1 = \frac{|\Gamma_0|}{|\Gamma_0| + |\Gamma_1|} \left(\overline{\boldsymbol{g}} - \overline{\boldsymbol{g}_0}\right)\left(\overline{\boldsymbol{g}} - \overline{\boldsymbol{g}_0}\right)^{\mathsf{T}} + \frac{|\Gamma_1|}{|\Gamma_0| + |\Gamma_1|} \left(\overline{\boldsymbol{g}} - \overline{\boldsymbol{g}_1}\right)\left(\overline{\boldsymbol{g}} - \overline{\boldsymbol{g}_1}\right)^{\mathsf{T}}$$

and

$$\mathbf{S}_2 = \sum_{\boldsymbol{g}\in\Gamma_0} \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0}\right)\left(\boldsymbol{g} - \overline{\boldsymbol{g}_0}\right)^{\mathsf{T}} + \sum_{\boldsymbol{g}\in\Gamma_1} \left(\boldsymbol{g} - \overline{\boldsymbol{g}_1}\right)\left(\boldsymbol{g} - \overline{\boldsymbol{g}_1}\right)^{\mathsf{T}},$$

where

$$\overline{\boldsymbol{g}_k} = \frac{1}{|\Gamma_k|} \sum_{\boldsymbol{g}\in\Gamma_k} \boldsymbol{g},$$

$$\overline{\boldsymbol{g}} = \frac{1}{|\Gamma_0| + |\Gamma_1|} \sum_{\boldsymbol{g}\in\Gamma_0\cup\Gamma_1} \boldsymbol{g},$$

and $|\Gamma_k|$ denotes the number of samples in the set Γ_k. The discriminant measure $J(\boldsymbol{w})$ between the classes Γ_0 and Γ_1 is then [121]

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\mathsf{T}\mathbf{S}_1\boldsymbol{w}}{\boldsymbol{w}^\mathsf{T}\mathbf{S}_2\boldsymbol{w}}.$$

With some algebra, the matrix $\mathbf{S}_1$ can be rewritten as follows

$$\begin{aligned}\mathbf{S}_1 = & \frac{|\Gamma_0|}{|\Gamma_0| + |\Gamma_1|}\left(\frac{|\Gamma_1|}{|\Gamma_0| + |\Gamma_1|}\right)^2 \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)^\mathsf{T} + \\ & + \left(\frac{|\Gamma_0|}{|\Gamma_0| + |\Gamma_1|}\right)^2 \frac{|\Gamma_1|}{|\Gamma_0| + |\Gamma_1|} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)^\mathsf{T} = \\ = & \frac{|\Gamma_0|}{|\Gamma_0| + |\Gamma_1|}\frac{|\Gamma_1|}{|\Gamma_0| + |\Gamma_1|} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)^\mathsf{T}.\end{aligned}$$

Substituting we get

$$J(\boldsymbol{w}) = \frac{|\Gamma_0|\,|\Gamma_1|}{\left(|\Gamma_0| + |\Gamma_1|\right)^2} \frac{\boldsymbol{w}^\mathsf{T}\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)^\mathsf{T}\boldsymbol{w}}{\boldsymbol{w}^\mathsf{T}\mathbf{S}_2\boldsymbol{w}} = \frac{|\Gamma_0|\,|\Gamma_1|}{\left(|\Gamma_0| + |\Gamma_1|\right)^2} \frac{\left(\overline{t_1} - \overline{t_0}\right)^2}{\boldsymbol{w}^\mathsf{T}\mathbf{S}_2\boldsymbol{w}},$$

where

$$\overline{t_k} = \frac{1}{|\Gamma_k|} \sum_{\boldsymbol{g}\in\Gamma_k} t(\boldsymbol{g}) = \boldsymbol{w}^\mathsf{T}\overline{\boldsymbol{g}_k}.$$

The expression found above shows that maximizing $J(\boldsymbol{w})$ is equivalent to finding the vector $\boldsymbol{w}$ that maximizes the distance between the means of the two classes while minimizing the variance within each class. In order to find the vector $\boldsymbol{w}$ that maximizes $J(\boldsymbol{w})$, we can consider the gradient of $J(\boldsymbol{w})$ with respect to $\boldsymbol{w}$, set it to zero, and solve for $\boldsymbol{w}$ the resulting system. It is easy to see [123] that any vector $\boldsymbol{w}$ that maximizes $J(\boldsymbol{w})$ must satisfy

$$\mathbf{S}_1\boldsymbol{w} = \alpha\mathbf{S}_2\boldsymbol{w},$$

for some constant α. Noting that $\mathbf{S}_1\boldsymbol{w} \propto (\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})\left[(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})^\mathsf{T}\boldsymbol{w}\right]$ and observing that multiplying by any positive constant the optimal vector $\boldsymbol{w}_{\text{opt}}$ (i.e., the one that maximizes $J(\boldsymbol{w})$) does not change its optimality, we have that the vector $\boldsymbol{w}_{\text{opt}}$ sought can be written as

$$\boldsymbol{w}_{\text{opt}} = \mathbf{S}_2^{-1}\left[\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right], \tag{2.9}$$

provided that the inverse of $\mathbf{S}_2$ exists. The vector $\boldsymbol{w}_{\text{opt}}$ is the "template" vector for the Fisher's linear discriminant.

In practice, this method is seldom used because $\mathbf{S}_2$ is almost always singular [69, 124]. Indeed, it can be proved that a necessary condition for having $\mathbf{S}_2$ invertible is to require that the number of vectors $\boldsymbol{g}$ in $\Gamma_0 \cup \Gamma_1$ is at least the number of components of $\boldsymbol{g}$ [69]. In some cases—for example when the vectors in $\Gamma_0 \cup \Gamma_1$ are the raster-scan representations of images of size 512 × 512 pixels—it may be impossible to collect or simulate such a huge number of vectors.

A more viable option is the Hotelling observer [120, 125] where, instead of using a sample covariance matrix, an ensemble covariance matrix is used [69]. This covariance matrix can sometimes be derived analytically from the statistics of the original objects, the properties of the imaging system, and the noise. If $\mathbf{K}_{\boldsymbol{g}|H_k}$ for $k \in \{0, 1\}$ are the covariance matrices under the two hypotheses H_0 and H_1, we introduce the mean covariance matrix as

$$\mathbf{K}_{\text{av}} = \frac{1}{2}\Big[\mathbf{K}_{\boldsymbol{g}|H_0} + \mathbf{K}_{\boldsymbol{g}|H_1}\Big]$$

and define the Hotelling test statistic as

$$t_{\text{Hot}}(\boldsymbol{g}) = \left[\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right]^\mathsf{T}\mathbf{K}_{\text{av}}^{-1}\boldsymbol{g} = \left[\boldsymbol{w}_{\text{Hot}}\right]^\mathsf{T}\boldsymbol{g}, \tag{2.10}$$

where

$$\boldsymbol{w}_{\text{Hot}} = \mathbf{K}_{\text{av}}^{-1}\left[\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right]$$

is the template vector for the Hotelling test statistic. It can be shown that if the classes Γ_0 and Γ_1 are equiprobable, then the template vector for the Hotelling test statistic and the one for the Fisher's linear discriminant are the same (up to a constant) [122]. Another remarkable advantage of the Hotelling observer is that it does not require knowledge of the densities under both hypotheses, but only knowledge of the mean covariance matrix $\mathbf{K}_{\mathrm{av}}$ and the ability to invert it [69].

Linear test statistics are given by the sum of a large number of random variables and so they are usually normally distributed by dint of the central-limit theorem [126–128]. Furthermore, as it was shown in § 2.4, in the case of homoscedastic normal distributions under the hypotheses H_0 and H_1, the log-likelihood ratio $\lambda(\boldsymbol{g})$ turns out to be a linear function. Thus, for normally distributed data, the Hotelling observer and the log-likelihood ratio are the same (in the sense that the two test statistics differ only by a constant) and optimal (in the sense that both maximize the area under the ROC curve). Furthermore, as shown in [129], the Fisher's discriminant function $t(\boldsymbol{g}) = [\boldsymbol{w}_{\mathrm{opt}}]^{\mathsf{T}}\boldsymbol{g}$, where $\boldsymbol{w}_{\mathrm{opt}}$ is the vector defined in (2.9), minimizes the probability of misclassification in the case of normally distributed homoscedastic test statistics. This relates the problem of assessing class separation to the problem of optimal classification of an object into one of two classes.

The covariance matrix $\mathbf{K}_{\mathrm{av}}$ that appears in (2.10) is symmetric and positive definite and so, taking advantage of its eigenvalue decomposition, we can find a matrix $\mathbf{K}_{\mathrm{av}}^{1/2}$ such that $\mathbf{K}_{\mathrm{av}}^{1/2}\mathbf{K}_{\mathrm{av}}^{1/2} = \mathbf{K}_{\mathrm{av}}$ [69]. With some algebra, we can rewrite (2.10) as

$$t_{\mathrm{Hot}}(\boldsymbol{g}) = \left[\mathbf{K}_{\mathrm{av}}^{-1/2}\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}\right)\right]^{\mathsf{T}}\left[\mathbf{K}_{\mathrm{av}}^{-1/2}\boldsymbol{g}\right],$$

which shows that the Hotelling observer can be interpreted as the simple matched filter between "prewhitened" versions of the vectors $\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}$ and $\boldsymbol{g}$ [69]. For the Hotelling observer, means of $t_{\mathrm{Hot}}(\boldsymbol{g})$ under the signal-absent and signal-present hypotheses are given by

$$\begin{aligned}
\langle t_{\mathrm{Hot}}(\boldsymbol{g})\rangle_{H_0} &= \left\langle[\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}]^{\mathsf{T}}\mathbf{K}_{\mathrm{av}}^{-1}\boldsymbol{g}\right\rangle_{H_0} = [\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}]^{\mathsf{T}}\mathbf{K}_{\mathrm{av}}^{-1}\overline{\boldsymbol{g}_0},\\
\langle t_{\mathrm{Hot}}(\boldsymbol{g})\rangle_{H_1} &= \left\langle[\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}]^{\mathsf{T}}\mathbf{K}_{\mathrm{av}}^{-1}\boldsymbol{g}\right\rangle_{H_1} = [\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0}]^{\mathsf{T}}\mathbf{K}_{\mathrm{av}}^{-1}\overline{\boldsymbol{g}_1},
\end{aligned}$$

in which we used the $\langle \ldots \rangle_{H_k}$ notation to denote ensemble average conditioned on the hypothesis H_k. It follows that

$$\langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_1} - \langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_0} = \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right).$$

For the variance of $t(\boldsymbol{g})$ conditioned on the hypothesis H_0, we have

$$\begin{aligned}
\sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})|H_0} &= \left\langle \left[t_{\mathrm{Hot}}(\boldsymbol{g}) - \langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_0} \right]^2 \right\rangle_{H_0} = \\
&= \left\langle \left[\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \boldsymbol{g} - \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \overline{\boldsymbol{g}_0} \right]^2 \right\rangle_{H_0} = \\
&= \left\langle \left[\left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right) \right]^2 \right\rangle_{H_0} = \\
&= \left\langle \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right) \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right) \right\rangle_{H_0} = \\
&= \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \underbrace{\left\langle \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right) \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \right\rangle_{H_0}}_{\mathbf{K}_{\mathrm{av}}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right) = \\
&= \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right),
\end{aligned}$$

in which we have recognized $\left\langle \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right) \left(\boldsymbol{g} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \right\rangle_{H_0} = \mathbf{K}_{\mathrm{av}}$. Similarly

$$\sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})|H_1} = \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right) = \sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})|H_0} = \sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})}.$$

Because $t_{\mathrm{Hot}}(\boldsymbol{g})$ is normally distributed under hypotheses H_0 and H_1, we can use (2.2) to calculate the detectability d'_{Hot} for the Hotelling observer

$$d'^2_{\mathrm{Hot}} = \frac{\left[\langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_1} - \langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_0} \right]^2}{\sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})}} = \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right).$$

With similar calculations, we can derive an expression for the SNR for the Hotelling observer

$$\mathrm{SNR}^2_{\mathrm{Hot}} = \frac{\left[\langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_1} - \langle t_{\mathrm{Hot}}(\boldsymbol{g}) \rangle_{H_0} \right]^2}{\frac{1}{2} \left(\sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})|H_1} + \sigma^2_{t_{\mathrm{Hot}}(\boldsymbol{g})|H_0} \right)} = \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right)^{\mathsf{T}} \mathbf{K}_{\mathrm{av}}^{-1} \left(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0} \right).$$

Our treatment assumed that vectors $\boldsymbol{g}$, $\boldsymbol{s}$, and $\boldsymbol{n}$ above were vectors of size M in the same Euclidean space. Consequently, covariances were matrices of size $M \times M$. In a practical application, such as the analysis of radiological images for the detection of lesions, the number M is the number of pixels in the image being evaluated. Hence, M can be quite large and easily exceed 10^6, thus making the matrix $\mathbf{K}_{\text{av}}$ challenging to store and calculate with today's computer technology. This problem has already been spotted in [66, 69, 124, 130], and clever ways to calculate $\mathbf{K}_{\text{av}}^{-1}$ or $\boldsymbol{w}_{\text{Hot}}$ have been proposed [131]. Alternatively, we can reduce the size of our data structures by first passing the data $\boldsymbol{g}$ through a bank of channels [73, 132–136]. More specifically, we introduce $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and we implement channel $\mathcal{C}_k$ as the dot product between $\boldsymbol{g}$ and the channel template vector $\boldsymbol{c}_k$: $\mathcal{C}_k(\boldsymbol{g}) = \boldsymbol{c}_k^{\mathsf{T}}\boldsymbol{g}$, for $k = 1, \ldots, K$. Because now each $\mathcal{C}_k(\boldsymbol{g})$ is simply a scalar value—as opposed to a vector of size M—the size of the covariance matrix for the channelized data is just $K \times K$, much smaller than $M \times M$.

As a final note, we remark that all the notions developed in this chapter can be extended to the generic case in which the data vector $\boldsymbol{g}$ is a vector in a Hilbert space. For example, $\boldsymbol{g}$ could be a continuous function of a 2D or 3D spatial variable $\boldsymbol{r}$. We can still consider the Hotelling observer and associated figures of merit. This time, however, instead of the covariance matrix $\mathbf{K}_{\text{av}}$ we will be dealing with a covariance operator $\mathcal{K}_{\text{av}}$ and we will denote its integral kernel [137] as $[\mathcal{K}_{\text{av}}](\boldsymbol{r}, \boldsymbol{r}')$. The definition of the Hotelling that was given in (2.10) now assumes the form

$$t_{\text{Hot}}(\boldsymbol{g}) = \int \boldsymbol{w}_{\text{Hot}}(\boldsymbol{r})\boldsymbol{g}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \boldsymbol{w}_{\text{Hot}}^{\dagger}\boldsymbol{g},$$

in which the symbol "†" denotes the adjoint of a vector or operator in a Hilbert space. The template vector for the Hotelling observer is now given by

$$\boldsymbol{w}_{\text{Hot}}(\boldsymbol{r}) = \int \big[\mathcal{K}_{\text{av}}^{-1}\big](\boldsymbol{r}, \boldsymbol{r}')\big(\overline{\boldsymbol{g}_1}(\boldsymbol{r}') - \overline{\boldsymbol{g}_0}(\boldsymbol{r}')\big)\,\mathrm{d}\boldsymbol{r}',$$

where $\mathcal{K}_{\text{av}}^{-1}$ is the inverse of the operator $\mathcal{K}_{\text{av}}$. Similarly, the SNR for the Hotelling observer is rewritten as

$$\begin{aligned}\text{SNR}_{\text{Hot}}^2 &= \iint \big(\overline{\boldsymbol{g}_1}(\boldsymbol{r}) - \overline{\boldsymbol{g}_0}(\boldsymbol{r})\big)\big[\mathcal{K}_{\text{av}}^{-1}\big](\boldsymbol{r}, \boldsymbol{r}')\big(\overline{\boldsymbol{g}_1}(\boldsymbol{r}') - \overline{\boldsymbol{g}_0}(\boldsymbol{r}')\big)\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}' = \\ &= (\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})^{\dagger}\mathcal{K}_{\text{av}}^{-1}\,(\overline{\boldsymbol{g}_1} - \overline{\boldsymbol{g}_0})\,. \end{aligned} \tag{2.11}$$

CHAPTER 3
Maximum-Likelihood Estimation

This chapter provides an introduction to maximum-likelihood estimation and its main properties. Once the basic ideas of maximum-likelihood estimation are presented, a simple and efficient algorithm to perform maximum-likelihood estimation is described. This algorithm works very well when we just need to estimate a few parameters. For larger problems, a specialized iterative maximum-likelihood estimation algorithm—called “maximum-likelihood expectation-maximization” or MLEM—is derived. The chapter concludes with a treatment of the basic properties of the MLEM algorithm.

3.1 General Idea

Maximum-likelihood estimation (MLE) is a statistical method used to estimate model parameters from observed noisy data [138, 139]. For a good historical development of the concept of maximum-likelihood estimation, the interested reader can consult [140].

In broad terms, given a set of observed data and an underlying model (which depends on some unknown parameters), MLE tries to determine the values of the model parameters that better explain the observed data [69]. One important fact to be aware of and that will be reminded throughout our treatment, is that the observed data that we use to perform maximum-likelihood estimation are to be interpreted as realizations of random variables. Thus, parameters we estimate from these data are to be considered as realizations of random variables as well.

Mathematically, we can introduce the set of parameters we want to estimate as the vector $\boldsymbol{\theta}$. The model itself is characterized by a probability density function (PDF) of the form $\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})$. The vector $\boldsymbol{x}$, which belongs to the set $\mathbb{X}$, represents the *complete data* [141]. The observed data vector $\boldsymbol{y}$, which we will also call *incomplete* *data*, belongs to the set $\mathbb{Y}$. We stress the fact that we do not observe $\boldsymbol{x}$ directly, but only indirectly and through the vector $\boldsymbol{y}$ [141]. We will assume that vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are statistically related via the PDF $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x})$. Probability density functions $\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})$ and $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x})$ allow us to write

$$\begin{aligned}\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta}) &= \int_{\mathbb{X}} \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{x} = \\ &= \int_{\mathbb{X}} \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{x},\end{aligned} \tag{3.1}$$

in which $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the PDF of the observed data $\boldsymbol{y}$ given the parameter $\boldsymbol{\theta}$ and we made use of the fact that the statistics of $\boldsymbol{y}$ conditioned on $\boldsymbol{x}$ and $\boldsymbol{\theta}$ actually, depend on $\boldsymbol{x}$ only. This assumption is valid if we imagine two separate "mechanisms" that, when concatenated, produce a sample $\boldsymbol{y}$ from the value of $\boldsymbol{\theta}$. The first mechanism samples $\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})$ to produce the complete data $\boldsymbol{x}$; the second mechanism produces the incomplete data $\boldsymbol{y}$ from the value of $\boldsymbol{x}$. Thus, conditioning on $\boldsymbol{\theta}$ in $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$ is unnecessary and we can simply write $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x})$.

As an example, we can consider a PET system and assume that the vector $\boldsymbol{\theta}$ represents a 2D or 3D continuous radiotracer distribution. In other words, $\boldsymbol{\theta}$ is a function of a 2D or 3D spatial variable. As a side note, $\boldsymbol{\theta}$ is usually a vector in a Euclidean space, but that is actually not a requirement for our treatment. Continuing with our PET example, $\boldsymbol{x}$ could represent the information about J pairs of photons emitted by the object. The vector $\boldsymbol{x}$ would thus be made of J components and each of them would be a pair $x_j = (\boldsymbol{r}^{(j)}, \vec{\boldsymbol{s}}^{(j)})$, in which $\boldsymbol{r}^{(j)}$ denotes the location at which the j^{th} pair of gamma-ray photons is created and $\vec{\boldsymbol{s}}^{(j)}$ is a vector with norm (length) 1 that denotes the direction along which one of the photons of the j^{th} pair propagates (we are making the simplifying assumption in which the two photons propagate along the same straight line but in opposite directions). Photons propagate to gamma-ray cameras where they might get detected and produce measurable signals at the PMT outputs. If they get detected, attribute vectors $\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}$ will be available and will make up the vector $\boldsymbol{y} = (y_1, \ldots, y_J) = (\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)})$. It does not hurt to recall that we do not actually observe $\boldsymbol{x} = (x_1, \ldots, x_J)$ (which contains the locations of the photon pairs creation and the directions of propagation); we can only observe (measure) $\boldsymbol{y}$. Thus, for the example we are considering here, the probability density function $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{J} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}^{(j)}, \vec{\boldsymbol{s}}^{(j)})$ models the propagation and absorption of the photons and it also accounts for all the random processes occurring inside the gamma-ray cameras. On the other hand, density $\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \prod_{j=1}^{J} \mathrm{pr}(\boldsymbol{r}^{(j)}, \vec{\boldsymbol{s}}^{(j)} \mid \boldsymbol{\theta})$ models the photon pair creation spatial distribution and the distribution of the direction of propagation.

We said above that MLE solves the estimation problem by finding the parameter vector $\boldsymbol{\theta}$ that best explains the observed data $\boldsymbol{y}$. Mathematically, this concept is formalized as [69]

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{\theta}} \big[\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})\big] = \arg\max_{\boldsymbol{\theta}} \big[L(\boldsymbol{\theta}; \boldsymbol{y})\big], \tag{3.2}$$

in which $\boldsymbol{\theta}$ varies over the whole parameter vector space, the "hat" symbol denotes an estimated quantity, and we have introduced the *likelihood*

$$L(\boldsymbol{\theta}; \boldsymbol{y}) = \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta}).$$

We observer that $L(\boldsymbol{\theta}; \boldsymbol{y})$ has to be interpreted as a function of $\boldsymbol{\theta}$ for fixed (measured) $\boldsymbol{y}$ and with the "$\arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y})$" notation in (3.2) we denote the value of $\boldsymbol{\theta}$ that maximizes the likelihood. Recall that $\boldsymbol{y}$ is the result of a noisy measurement, so the actual value of $\boldsymbol{y}$ in (3.2) might change if the measurement is repeated. In other words, $\boldsymbol{y}$ is a random quantity, and this implies that the ML estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ is random as well. Therefore, we can—and in some cases we will—investigate the statistical properties of $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$.

An alternative way to calculate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ consists on rewriting (3.2) as [69]

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{\theta}} \big[\ln \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})\big] = \arg\max_{\boldsymbol{\theta}} \big[\ell(\boldsymbol{\theta}; \boldsymbol{y})\big], \tag{3.3}$$

in which we have introduced the *log-likelihood*

$$\ell(\boldsymbol{\theta}; \boldsymbol{y}) = \ln L(\boldsymbol{\theta}; \boldsymbol{y}) = \ln \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta}). \tag{3.4}$$

Because the logarithm is a strictly monotonic function, the expression in (3.3) is equivalent to (3.2). Often times, the log-likelihood $\ell(\boldsymbol{\theta};\boldsymbol{y})$ is numerically easier to treat than the likelihood $L(\boldsymbol{\theta};\boldsymbol{y})$.

From the definition (3.2) of the maximum-likelihood estimate, it follows that

$$\left.\frac{\partial \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}} = 0,$$

provided that $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ is differentiable at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$. It is important to note that the equation above might have more than one solution. In other words, the maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ might not be unique.

3.2 Important Properties

Maximum-likelihood estimates have many favorable properties. Some of these properties, which often have theoretical or practical importance, are summarized below without providing much mathematical detail or formal proof. Whenever possible, the interested reader can find a more formal treatment and proofs in the literature we reference.

Asymptotic efficiency. Before discussing asymptotic efficiency of MLE, we need to recall that the observed data $\boldsymbol{y}$ is a random noisy quantity. Thus, the maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ calculated from $\boldsymbol{y}$ is a random quantity as well. For the case in which $\boldsymbol{y}$ represents a set of repeated independent and identically distributed samples, asymptotic efficiency of MLE refers to the fact that, asymptotically and as the size of the set $\boldsymbol{y}$ increases, the variance of each component of the vector $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ attains the smallest possible value [142, 143], which is given by the *Cramér-Rao lower bound* [144, 145].

Functional invariance. Assume the MLE of a parameter $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ and consider a function $u(\boldsymbol{\theta})$ of the parameter $\boldsymbol{\theta}$. We can identify $u(\boldsymbol{\theta})$ with the parameter

$\boldsymbol{\mu}$, for which we can consider a maximum-likelihood estimate $\hat{\boldsymbol{\mu}}_{\mathrm{MLE}}$. Then [146, 147],

$$\hat{\boldsymbol{\mu}}_{\mathrm{MLE}} = u(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}).$$

The equation above shows that the property of being a maximum-likelihood estimate is conserved if we consider a function of the maximum-likelihood estimate itself.

Sufficiency. In statistics, a quantity calculated from some samples and used to estimate an unknown parameter $\boldsymbol{\theta}$ is said to be a *sufficient statistic* if no other quantity that can be calculated from the same samples would provide additional information regarding the value of the parameter $\boldsymbol{\theta}$ [138]. Intuitively, a sufficient statistic is a function of the samples that "compresses" the data $\boldsymbol{y}$ without loosing any information about $\boldsymbol{\theta}$. Sufficiency for a maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ can be stated by saying that $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ must be a function of a sufficient statistic for $\boldsymbol{\theta}$ [148, 149].

Consistency. Consistency of an estimator regards the behavior of the estimator as the sample size increases. Let us consider again the case in which $\boldsymbol{y}$ is a set of repeated independent and identically distributed samples. It is possible to show that, when the range of the elements of the set $\boldsymbol{y}$ is independent on the parameter $\boldsymbol{\theta}$, there exists a maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ that, as the size of the set $\boldsymbol{y}$ goes to infinity, converges in probability [127] to the true value $\boldsymbol{\theta}$ of the parameter [144, 150]. The property of consistency can be restated by saying that there exists a maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ that is *unbiased* [69] as more and more data are collected. It can be shown that a consistent maximum-likelihood estimate is unique [150].

Asymptotic normality. The property of asymptotic normality states that, as the sample size increases, the probability density function of maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ asymptotically approaches the probability density function of a normally-distributed random variable. More specifically, we have convergence [46,69,139,144,151] to a normally distributed random variable with mean equal the true parameter we want to estimate and covariance matrix equal to the inverse of the Fisher information matrix [114,152,153].

3.3 MLE for Low-Dimensional Problems

We saw in (3.2) that performing a maximum-likelihood estimation entails performing a search for a maximum of the likelihood $L(\boldsymbol{\theta}; \boldsymbol{y})$ or, equivalently, the log-likelihood $\ell(\boldsymbol{\theta}; \boldsymbol{y})$. In some cases, this search problem can be solved analytically. The example that follows shows one such case. At the same time, it provides a more concrete introduction to maximum-likelihood estimation.

Consider J independent identically distributed Poisson random variables $Y_1, \ldots, Y_J$ with parameter (mean) θ. In other words [127]:

$$\Pr(Y_j = y_j \mid \theta) = \frac{\theta^{y_j}}{y_j!} e^{-\theta},$$

in which $\Pr(Y_j = y_j \mid \theta)$ denotes the probability that the random variable Y_j assumes the value y_j, given that the parameter (mean) of the Poisson random variable Y_j is θ. The goal is to estimate the parameter θ from the realizations $y_1, \ldots, y_J$. Each number y_j is a non-negative integer number. We can collect the realizations of $Y_1, \ldots, Y_J$ in the vector $\boldsymbol{y} = (y_1, \ldots, y_J)$. By the hypothesis of independence, we can write the likelihood as

$$L(\theta; \boldsymbol{y}) = \Pr(Y_1 = y_1, \ldots, Y_J = y_J \mid \theta) = e^{-J\theta} \prod_{j=1}^{J} \frac{\theta^{y_j}}{y_j!},$$

and the log-likelihood is

$$\ell(\theta; \boldsymbol{y}) = \ln L(\theta; \boldsymbol{y}) = \sum_{j=1}^{J} \left[y_j \ln \theta - \ln(y_j!) \right] - J\theta.$$

In this simple case, we can find an estimate of the parameter θ by calculating the derivative of $\ell(\theta; \boldsymbol{y})$, setting it zero, and solving for θ the resulting equation. Specifically

$$\frac{\mathrm{d}\ell(\theta; \boldsymbol{y})}{\mathrm{d}\theta} = \left[\frac{1}{\theta} \sum_{j=1}^{J} y_j\right] - J,$$

which gives

$$\hat{\theta}_{\mathrm{MLE}} = \frac{1}{J} \sum_{j=1}^{J} y_j. \tag{3.5}$$

The maximum-likelihood estimate $\hat{\theta}_{\mathrm{MLE}}$ of θ turns out to be the sample mean of the observed data $y_1, \ldots, y_J$. This should not be a surprising result because the mean of a Poisson random variable corresponds to its parameter and the arithmetic mean of realizations $y_1, \ldots, y_J$ is an unbiased estimator of the mean of any of the Y_j:

$$\left\langle \frac{1}{J} \sum_{j=1}^{J} Y_j \right\rangle = \frac{1}{J} \sum_{j=1}^{J} \langle Y_j \rangle = \frac{1}{J} \sum_{j=1}^{J} \theta = \theta.$$

To show that the quantity found in (3.5) maximizes the log-likelihood, we can consider the second derivative of $\ell(\theta; \boldsymbol{y})$ and evaluate it for $\theta = \hat{\theta}_{\mathrm{MLE}}$:

$$\left. \frac{\mathrm{d}^2\ell(\theta; \boldsymbol{y})}{\mathrm{d}\theta^2} \right|_{\theta = \hat{\theta}_{\mathrm{MLE}}} = -\frac{1}{\hat{\theta}^2_{\mathrm{MLE}}} \sum_{j=1}^{J} y_j.$$

Because the numbers $y_1, \ldots, y_J$ are non-negative, the quantity at the right-hand side is always negative or zero. Hence, $\hat{\theta}_{\mathrm{MLE}}$ is a maximum of $\ell(\theta; \boldsymbol{y})$, as desired.

In many practical cases, however, an analytic solution of the maximum-likelihood estimation problem is not possible. This would be, for example, the case in which the gradient of the likelihood $L(\boldsymbol{\theta}; \boldsymbol{y})$ has a very complicated expression and the resulting equations cannot be solved explicitly for $\boldsymbol{\theta}$. If the number of components of the vector $\boldsymbol{\theta}$ is large (for example, greater than 5), an exhaustive search as the one suggested in (3.2) is not computationally feasible. In these cases, we are usually forced to use an optimization algorithm to find a point that solves (3.2). We will not try to face the problem of providing a survey of all the optimization methods that have appeared in the literature. For a good treatment of the topic, the interested reader can consult [154–172].

In the remainder of this section, we want to present an alternative technique that can be used to search for the maximum of the likelihood $L(\boldsymbol{\theta}; \boldsymbol{y})$. For reasons that will be clear later, this method is feasible when the number of components of the vector $\boldsymbol{\theta}$ is relatively small and $L(\boldsymbol{\theta}; \boldsymbol{y})$ is a "smooth" function of $\boldsymbol{\theta}$. To fix the ideas, let assume that $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ with, say, $N \leqslant 5$ and let assume that we know *a priori* that each θ_n can only take values between a_n and b_n. In other words, $a_n \leqslant \theta_n \leqslant b_n$. Cases like this frequently appear in practical problems. For example, if the parameter vector $\boldsymbol{\theta}$ to estimate is a 3D location of interaction of a photon within a scintillation crystal, then $N = 3$ and the point $\boldsymbol{\theta}$ cannot lie outside the detector's crystal. In other words, $a_n \leqslant \theta_n \leqslant b_n$ for $n = 1, 2, 3$ where quantities a_n and b_n define the crystal boundaries.

The search algorithm [45,173,174] starts by choosing Q points $\theta_n^{(1)}, \ldots, \theta_n^{(Q)}$ equally spaced in each of the $[a_n, b_n]$ intervals. The separation between points will be $\Delta_n = (b_n - a_n)/Q$. An N-dimensional grid of parameter vectors $\boldsymbol{\theta}^{(q_1,\ldots,q_N)} = \big(\theta_1^{(q_1)}, \ldots, \theta_N^{(q_N)}\big)$ is created by considering all the possible choices for the indices $q_1, \ldots, q_N$. Such a grid will thus be made up of Q^N parameter vectors $\boldsymbol{\theta}^{(q_1,\ldots,q_N)}$ and we can think of it as an N-dimensional grid with Q points in each dimension. Points are evenly spaced by Δ_n in the n^{th} dimension. This grid is centered at $\boldsymbol{\theta}_{\text{mid}} = \big(\frac{a_1+b_1}{2}, \ldots, \frac{a_N+b_N}{2}\big)$. Notice that $\boldsymbol{\theta}_{\text{mid}}$ might not necessarily belong to the grid. The likelihood $L(\boldsymbol{\theta}^{(q_1,\ldots,q_N)}; \boldsymbol{y})$ is then calculated for all the Q^N parameter vectors $\boldsymbol{\theta}^{(q_1,\ldots,q_N)}$ of the grid and the parameter vector $\boldsymbol{\theta}^{\star}$ of the grid that maximizes $L(\boldsymbol{\theta}^{(q_1,\ldots,q_N)}; \boldsymbol{y})$ is retained. A new N-dimensional grid is considered; this second grid will be centered at $\boldsymbol{\theta}^{\star}$ and will be obtained from the previous one by reducing the separation between points. In particular, we can decide that points in the n^{th} dimension of the new grid will now be separated by Δ_n/α, for $\alpha > 1$. For example, we can set $\alpha = 2$, meaning that the separation between adjacent points in the second grid is half the separation between adjacent points in the first grid. The process outlined above is then repeated: $L(\boldsymbol{\theta}^{(q_1,\ldots,q_N)};\boldsymbol{y})$ is calculated for all the parameter vectors of the second grid, a new parameter vector $\boldsymbol{\theta}^{\star}$ is selected, a third grid—smaller than the second—is considered, and so on for a prefixed number of iterations. Because the separation between points in the grid is divided by $\alpha > 1$ at each iteration, the algorithm is able to find a local maximum of $L(\boldsymbol{\theta}^{(q_1,\ldots,q_N)};\boldsymbol{y})$ with an accuracy that grows exponentially with the number of iterations. We will usually refer to the algorithm just described as the *contracting-grid* algorithm [45, 59, 173].

3.4 The MLEM Algorithm

As we alluded to above, many algorithms to find the point that maximizes the likelihood (or its logarithm) are available. These algorithms range from classical minimization algorithms (such as the steepest-descent and the conjugate-gradient methods) to derivative-free optimization routines, and from simulating annealing to and genetic algorithms. We also presented in some details a very simple algorithm [45] that runs fast when just a few parameters $\theta_1, \ldots, \theta_N$ need be estimated.

In medical applications, we might want to use MLE to reconstruct an image from observed data. In such a case, the vector $\boldsymbol{\theta}$ will stand for the reconstructed image. Typically, such an image is of size 128×128 pixels or larger, leading to $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ with $N \geqslant 128^2 = 16384$. Even the fastest algorithm described above might take an incredible amount of time to solve this maximum-likelihood estimation problem. Fortunately, we can take advantage of the mathematical structure—such as the relationship in (3.1)—of the problem at hand and devise a fast iterative algorithm to solve (3.2). The goal of this section is to briefly present a derivation of the expectation-maximization (EM) algorithm [141, 175–178] and also mention its main properties. The EM algorithm will be used to solve a maximum-likelihood (ML) problem for the particular case in which the noise in the data follows Poisson statistics, thus leading to the derivation of the maximum-likelihood expectation-maximization (MLEM) algorithm [179–184] in the form commonly used in medical applications.

The MLEM algorithm constructs a sequence $\hat{\boldsymbol{\theta}}^{(1)}, \hat{\boldsymbol{\theta}}^{(2)}, \ldots, \hat{\boldsymbol{\theta}}^{(k)}, \ldots$ of estimates such that

$$\lim_{k \to \infty} \hat{\boldsymbol{\theta}}^{(k)} = \hat{\boldsymbol{\theta}}^{(\infty)}, \qquad \big|\hat{\boldsymbol{\theta}}^{(\infty)}\big| < \infty,$$

and

$$\ell(\hat{\boldsymbol{\theta}}^{(1)}; \boldsymbol{y}) < \ell(\hat{\boldsymbol{\theta}}^{(2)}; \boldsymbol{y}) < \cdots < \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) < \cdots < \ell(\hat{\boldsymbol{\theta}}^{(\infty)}; \boldsymbol{y}) = \max_{\boldsymbol{\theta}} \big[\ell(\boldsymbol{\theta}; \boldsymbol{y})\big],$$

where $\ell(\boldsymbol{\theta}; \boldsymbol{y})$ is the log-likelihood defined in (3.4). Given one of the terms in the sequence $\hat{\boldsymbol{\theta}}^{(1)}, \hat{\boldsymbol{\theta}}^{(2)}, \ldots, \hat{\boldsymbol{\theta}}^{(k)}, \ldots$, we can consider the quantity $\ell(\boldsymbol{\theta}; \boldsymbol{y}) - \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y})$. Notice that this quantity is a function of $\boldsymbol{\theta}$ for fixed observed data $\boldsymbol{y}$. With some manipulations and using (3.1), we can write

$$\begin{aligned}
\ell(\boldsymbol{\theta}; \boldsymbol{y}) - \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) &= \ln \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta}) - \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \\
&= \ln \int_{\mathbb{X}} \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{x} - \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \\
&= \ln \int_{\mathbb{X}} \left[\frac{\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})} \right] \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})\, \mathrm{d}\boldsymbol{x} - \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \\
&= \ln \left\langle \frac{\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})} \right\rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}} - \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}),
\end{aligned}$$

in which we have recognized that the integral is nothing but the expectation over $\boldsymbol{x}$ given $\boldsymbol{y}$ and $\hat{\boldsymbol{\theta}}^{(k)}$ of the quantity in $[\ldots]$ above. Notice that the logarithm is a concave function. Thus, by Jensen's inequality [185] applied to $\langle \ldots \rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}}$:

$$\begin{aligned}
\ell(\boldsymbol{\theta}; \boldsymbol{y}) - \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) &\geqslant \left\langle \ln \left[\frac{\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}} - \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \\
&= \left\langle \ln \left[\frac{\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}} - \langle \ln \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)}) \rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}} = \\
&= \left\langle \ln \left[\frac{\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)})\, \mathrm{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}}.
\end{aligned}$$

If we define

$$\tilde{\ell}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) = \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) + \left\langle \ln \left[\frac{\operatorname{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})}{\operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}) \operatorname{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}},$$

then, by the result derived above:

$$\tilde{\ell}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) \leqslant \ell(\boldsymbol{\theta}; \boldsymbol{y}).$$

We further have:

$$\begin{aligned} \tilde{\ell}(\hat{\boldsymbol{\theta}}^{(k)}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) &= \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) + \left\langle \ln \left[\frac{\operatorname{pr}(\boldsymbol{y} \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}^{(k)}) \operatorname{pr}(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}^{(k)})}{\operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}) \operatorname{pr}(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}} = \\ &= \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}) + \underbrace{\left\langle \ln \left[\frac{\operatorname{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)})}{\operatorname{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(k)})} \right] \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}}}_{0} = \\ &= \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}). \end{aligned}$$

To summarize, the two important results we have found so far are:

- $\tilde{\ell}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) \leqslant \ell(\boldsymbol{\theta}; \boldsymbol{y})$,
- $\tilde{\ell}(\hat{\boldsymbol{\theta}}^{(k)}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) = \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y})$.

If we choose $\hat{\boldsymbol{\theta}}^{(k+1)}$ as

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} \left[\tilde{\ell}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y}) \right],$$

then

$$\ell(\hat{\boldsymbol{\theta}}^{(k+1)}; \boldsymbol{y}) \geqslant \ell(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y}),$$

where the last result follows from the previous two and by the definition of $\hat{\boldsymbol{\theta}}^{(k+1)}$ as the point $\boldsymbol{\theta}$ that maximizes $\tilde{\ell}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{y})$. Figure 3.1 shows an example of the basic steps of the MLEM algorithm.

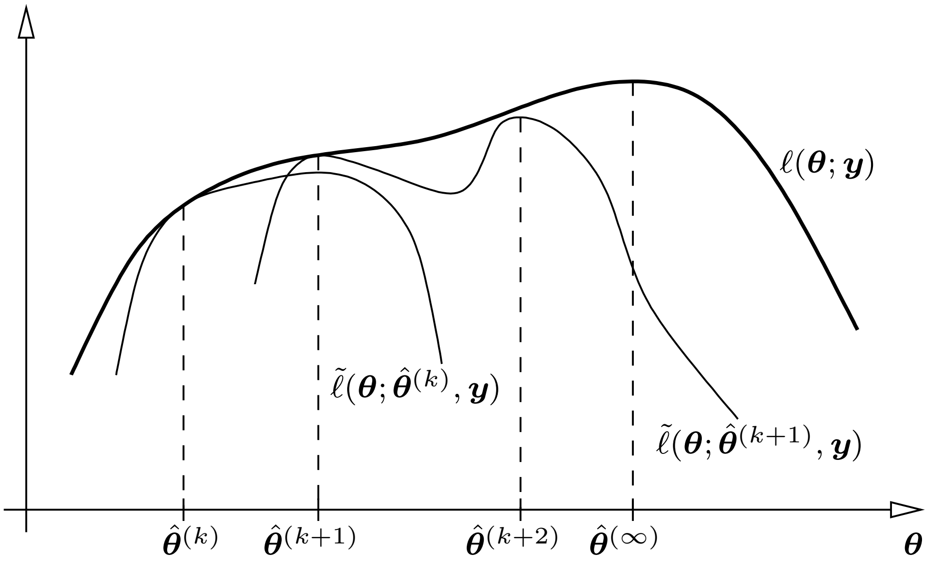

$\ell(\boldsymbol{\theta};\boldsymbol{y})$
$\tilde{\ell}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(k)},\boldsymbol{y})$
$\tilde{\ell}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(k+1)},\boldsymbol{y})$
$\hat{\boldsymbol{\theta}}^{(k)}$
$\hat{\boldsymbol{\theta}}^{(k+1)}$
$\hat{\boldsymbol{\theta}}^{(k+2)}$
$\hat{\boldsymbol{\theta}}^{(\infty)}$
$\boldsymbol{\theta}$

FIGURE 3.1. Illustration of the MLEM algorithm

Expanding the relationship $\hat{\boldsymbol{\theta}}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} \tilde{\ell}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(k)},\boldsymbol{y})$, yields

$$\begin{aligned}
\hat{\boldsymbol{\theta}}^{(k+1)} &= \arg\max_{\boldsymbol{\theta}} \left\{ \ell(\hat{\boldsymbol{\theta}}^{(k)};\boldsymbol{y}) + \left\langle \ln\left[\frac{\mathrm{pr}(\boldsymbol{y}\mid\boldsymbol{x},\boldsymbol{\theta})\,\mathrm{pr}(\boldsymbol{x}\mid\boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x}\mid\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)})\,\mathrm{pr}(\boldsymbol{y}\mid\hat{\boldsymbol{\theta}}^{(k)})}\right]\right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}} \right\} = \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ \left\langle \ln\left[\mathrm{pr}(\boldsymbol{y}\mid\boldsymbol{x},\boldsymbol{\theta})\,\mathrm{pr}(\boldsymbol{x}\mid\boldsymbol{\theta})\right]\right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}} \right\} = \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ \left\langle \ln\left[\frac{\mathrm{pr}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{x},\boldsymbol{\theta})}\frac{\mathrm{pr}(\boldsymbol{x},\boldsymbol{\theta})}{\mathrm{pr}(\boldsymbol{\theta})}\right]\right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}} \right\} = \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ \left\langle \ln\mathrm{pr}(\boldsymbol{x},\boldsymbol{y}\mid\boldsymbol{\theta})\right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}} \right\},
\end{aligned} \tag{3.6}$$

which shows that $\hat{\boldsymbol{\theta}}^{(k+1)}$ is obtained by first performing an estimation, denoted above as $\langle \ln\mathrm{pr}(\boldsymbol{x},\boldsymbol{y}\mid\boldsymbol{\theta})\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(k)}}$, and then by performing the maximization with respect to $\boldsymbol{\theta}$ of this quantity. Thus, each iteration of the algorithm involves two steps: the expectation step (which we will also call the "E-step") and the maximization step (the "M-step").

Some theoretical results about the MLEM algorithm can be found in [141,186,187]. For example, it is shown that successive iterations of the E- and M-steps always increase the likelihood and that if convergence to $\hat{\boldsymbol{\theta}}^{(\infty)}$ is reached, then $\hat{\boldsymbol{\theta}}^{(\infty)}$ is a stationary point of the likelihood. In [141], conditions for convergence and rate of convergence to a stationary point are discussed as well.

We will devote the rest of this section to the derivation of the MLEM algorithm in the form that is usually used in medical imaging. In order to do that, we first need to introduce some notation and describe the physical model that we will assume. In nuclear medical imaging, a radionuclide-bearing compound is introduced into the object being imaged. This compound forms an unknown gamma-ray emitter distribution that we will denote with the function $f(\boldsymbol{r})$, in which $\boldsymbol{r}$ is a 3D spatial variable that varies over the entire field of view (FOV). The units of $f(\boldsymbol{r})$ are number of emissions per unit time, per unit volume. We will divide the field of view into N non-overlapping voxels $B_1, \ldots, B_N$ and define:

$$f_n = \int_{B_n} f(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r},$$

for $n = 1, \ldots, N$. Collectively, the numbers $f_1, \ldots, f_N$ will be denoted as the vector $\boldsymbol{f}$. The expression above represents a particular case of a more general model:

$$f_n = \int_{\mathrm{FOV}} \varphi_n(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r},$$

where $\big\{\varphi_1(\boldsymbol{r}), \ldots, \varphi_N(\boldsymbol{r})\big\}$ is a set of expansion functions, usually an orthonormal basis. Given an estimate $\hat{\boldsymbol{f}}$ of the vector $\boldsymbol{f}$, we define the estimate $\hat{f}(\boldsymbol{r})$ of $f(\boldsymbol{r})$ as [69]

$$\hat{f}(\boldsymbol{r}) = \sum_{n=1}^{N} \hat{f}_n \phi_n(\boldsymbol{r}),$$

where functions $\phi_n(\boldsymbol{r})$ are not necessarily the same as $\varphi_n(\boldsymbol{r})$.

Gamma-ray photons emitted by the radiotracer distribution $f(\boldsymbol{r})$ are detected by one or more detectors. We will assume pixelated detectors with a total of M pixels, and we will denote the total count in pixel m as g_m, for $m = 1, \ldots, M$. Again, we will denote as $\boldsymbol{g}$ the vector of elements $g_1, \ldots, g_M$. Thus, $\boldsymbol{g}$ is the vector of the observed or incomplete data. We can further assume that g_m can be written as

$$g_m = \sum_{n=1}^{N} c_{m,n},$$

in which $c_{m,n}$ is the number of gamma-ray photons collected by the m^{th} detector pixel and emitted from voxel B_n in the field of view. From the physics, each $c_{m,n}$ is a random variable that follows a Poisson distribution with mean (parameter) $h_{m,n} f_n$, where $h_{m,n}$ can be interpreted as the probability that a gamma-ray photon emitted from voxel B_n in object space is detected at detector pixel m, multiplied by some exposure time τ. We will also assume that random variable $c_{m,n}$ for $m = 1, \ldots, M$ and $n = 1, \ldots, N$ are independent and we will collect all of them in the vector $\boldsymbol{c}$. Notice that we cannot directly measure $\boldsymbol{c}$; thus, in the language introduced in § 3.1, $\boldsymbol{c}$ represents the vector of complete data. From the assumption of independence of the random variables $c_{m,n}$ and from the fact that the random variables $c_{m,n}$ follow Poisson statistics, we obtain that g_m is a Poisson random variable with mean $\sum_{n=1}^{N} h_{m,n} f_n$. In image reconstruction, it is customary to interpret the randomness in g_m as noise. Within this framework, the goal of any reconstruction algorithm is to calculate an estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ given the noisy data $\boldsymbol{g}$ and for known values of $h_{m,n}$, with $m = 1, \ldots, M$ and $n = 1, \ldots, N$. For the case of maximum-likelihood, estimate $\hat{\boldsymbol{f}}$ is defined as any object distribution that maximizes the probability of measuring the data $\boldsymbol{g}$ [188,189]:

$$\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \big[L(\boldsymbol{f}; \boldsymbol{g})\big] = \arg\max_{\boldsymbol{f}} \big[\Pr(\boldsymbol{g} \mid \boldsymbol{f})\big].$$

For reasons that will become clear later, we will start by calculating the conditional probability $\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f})$. Notice first that if $g_m \neq \sum_{n=1}^{N} c_{m,n}$ for some m, then the probability $\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f})$ would be 0. If, on the other hand, $g_m = \sum_{n=1}^{N} c_{m,n}$ for all $m = 1, \ldots, M$, then we can use the rules of conditional probability to write

$$\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f}) = \prod_{m=1}^{M} \Pr(g_m, c_{m,1}, \ldots, c_{m,N} \mid \boldsymbol{f}) = \prod_{m=1}^{M} \frac{\Pr(g_m, c_{m,1}, \ldots, c_{m,N}, \boldsymbol{f})}{\Pr(\boldsymbol{f})} =$$

$$= \prod_{m=1}^{M} \left[\Pr(g_m \mid c_{m,1}, \ldots, c_{m,N}, \boldsymbol{f}) \times \Pr(c_{m,1}, \ldots, c_{m,N} \mid \boldsymbol{f}) \right] =$$
$$= \prod_{m=1}^{M} \left[\prod_{n=1}^{N} \frac{[h_{m,n} f_n]^{c_{m,n}}}{c_{m,n}!} e^{-h_{m,n} f_n} \right],$$

where we used the fact that $c_{m,n}$ follows a Poisson distribution with mean $h_{m,n} f_n$. With some abuse of notation, we can include the $\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f}) = 0$ case and the $\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f}) \neq 0$ case in the same expression and write

$$\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f}) = \prod_{m=1}^{M} \left[\delta_{\text{Kron}} \left(g_m - \sum_{n=1}^{N} c_{m,n} \right) \times \prod_{n=1}^{N} \frac{[h_{m,n} f_n]^{c_{m,n}}}{c_{m,n}!} e^{-h_{m,n} f_n} \right], \qquad (3.7)$$

in which $\delta_{\text{Kron}}(k)$ is the Kronecker delta function defined as

$$\delta_{\text{Kron}}(k) = \begin{cases} 1 & \text{if } k = 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (3.8)$$

Notice that in (3.7), we used the relationship $g_m = \sum_{n=1}^{N} c_{m,n}$ to express the probability $\Pr(g_m \mid c_{m,1}, \ldots, c_{m,N}, \boldsymbol{f})$ using the Kronecker delta function.

To calculate the likelihood for the problem at hand, we need the probability $\Pr(\boldsymbol{g} \mid \boldsymbol{f})$, which is obtained from $\Pr(\boldsymbol{g}, \boldsymbol{c} \mid \boldsymbol{f})$ by marginalizing over $\boldsymbol{c}$:

$$\Pr(\boldsymbol{g} \mid \boldsymbol{f}) = \prod_{m=1}^{M} \Pr(g_m \mid \boldsymbol{f}) = \prod_{m=1}^{M} \left[\sum_{c_{m,1}, \ldots, c_{m,N}} \Pr(g_m, c_{m,1}, \ldots, c_{m,N} \mid \boldsymbol{f}) \right] =$$
$$= \prod_{m=1}^{M} \left[\sum_{c_{m,1} + \cdots + c_{m,N} = g_m} \prod_{n=1}^{N} \frac{[h_{m,n} f_n]^{c_{m,n}}}{c_{m,n}!} e^{-h_{m,n} f_n} \right] =$$
$$= \prod_{m=1}^{M} \frac{e^{-\sum_{n=1}^{N} h_{m,n} f_n}}{g_m!} \left\{ \sum_{c_{m,1} + \cdots + c_{m,N} = g_m} \frac{g_m!}{c_{m,1}! \ldots c_{m,N}!} \times \right.$$
$$\left. \times [h_{m,1} f_1]^{c_{m,1}} \times \cdots \times [h_{m,N} f_N]^{c_{m,N}} \right\} =$$
$$= \prod_{m=1}^{M} \left\{ \frac{\left[\sum_{n=1}^{N} h_{m,n} f_n \right]^{g_m}}{g_m!} e^{-\sum_{n=1}^{N} h_{m,n} f_n} \right\} = \prod_{m=1}^{M} \Pr(g_m \mid \boldsymbol{f}),$$

in which the last step shows that $g_1, \ldots, g_M$ are independent Poisson random variables with means $\sum_{n=1}^{N} h_{1,n} f_n, \ldots, \sum_{n=1}^{N} h_{M,n} f_n$, as expected. From the expression above, it follows that the log-likelihood $\ell(\boldsymbol{f}; \boldsymbol{g})$ is

$$\ell(\boldsymbol{f}; \boldsymbol{g}) = \ln \Pr(\boldsymbol{g} \mid \boldsymbol{f}) = \sum_{m=1}^{M} \left\{ g_m \ln \left[\sum_{n=1}^{N} h_{m,n} f_n \right] - \sum_{n=1}^{N} h_{m,n} f_n - \ln(g_m!) \right\}.$$

It is show in [179] that the matrix of second derivatives of $\ell(\boldsymbol{f}; \boldsymbol{g})$ is negative semi-definite. Therefore, $\ell(\boldsymbol{f}; \boldsymbol{g})$ as a function of $\boldsymbol{f}$ is concave. To calculate the maximum-likelihood estimate, we can calculate the gradient of $\ell(\boldsymbol{f}; \boldsymbol{g})$ with respect to $\boldsymbol{f}$, set it to zero, and try to solve the resulting system of equations for $\boldsymbol{f}$. The gradient of $\ell(\boldsymbol{f}; \boldsymbol{g})$ is

$$\frac{\partial \ell(\boldsymbol{f}; \boldsymbol{g})}{\partial f_{n'}} = \sum_{m=1}^{M} \left\{ \frac{g_m h_{m,n'}}{\sum_{n=1}^{N} h_{m,n} f_n} - h_{m,n'} \right\}, \qquad n' = 1, \ldots, N,$$

leading to

$$\frac{1}{\sum_{m=1}^{M} h_{m,n'}} \sum_{m=1}^{M} \frac{g_m h_{m,n'}}{\sum_{n=1}^{N} h_{m,n} \hat{f}_n} = 1, \qquad n' = 1, \ldots, N.$$

The system of equations above, however, cannot be explicitly solved for $\hat{\boldsymbol{f}}$. On the other hand, the MLEM algorithm, as developed in (3.6), yields

$$\hat{\boldsymbol{f}}^{(k+1)} = \arg\max_{\boldsymbol{f}} \left\{ \big\langle \ln \Pr(\boldsymbol{c}, \boldsymbol{g} \mid \boldsymbol{f}) \big\rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}} \right\},$$

where

$$\begin{aligned} &\big\langle \ln \Pr(\boldsymbol{c}, \boldsymbol{g} \mid \boldsymbol{f}) \big\rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}} = \\ &\qquad = \sum_{m=1}^{M} \sum_{n=1}^{N} \Big[\langle c_{m,n} \rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}} \ln(h_{m,n} f_n) - \langle \ln(c_{m,n}!) \rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}} - h_{m,n} f_n \Big]. \end{aligned}$$

For the purpose of calculating the maximum with respect to $\boldsymbol{f}$, we can discard all the terms of $\big\langle \ln \Pr(\boldsymbol{c}, \boldsymbol{g} \mid \boldsymbol{f}) \big\rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}}$ that do not involve $\boldsymbol{f}$:

$$\hat{\boldsymbol{f}}^{(k+1)} = \arg\max_{\boldsymbol{f}} \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} \Big[\langle c_{m,n} \rangle_{\boldsymbol{c}|\boldsymbol{g}, \hat{\boldsymbol{f}}^{(k)}} \ln(h_{m,n} f_n) - h_{m,n} f_n \Big] \right\}. \tag{3.9}$$

To calculate $\langle c_{m,n} \rangle_{\boldsymbol{c}|\boldsymbol{g},\hat{\boldsymbol{f}}^{(k)}}$ we note that such a quantity is the fraction of $g_{m,n}$ that is coming from the n^{th} voxel, conditioned on $\hat{\boldsymbol{f}}^{(k)}$:

$$\langle c_{m,n} \rangle_{\boldsymbol{c}|\boldsymbol{g},\hat{\boldsymbol{f}}^{(k)}} = g_m \frac{h_{m,n} \hat{f}_n^{(k)}}{\sum_{n'=1}^{N} h_{m,n'} \hat{f}_{n'}^{(k)}}.$$

Define $Q(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \boldsymbol{g})$ as the quantity in $\{\ldots\}$ that appears in (3.9):

$$\begin{aligned} Q(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \boldsymbol{g}) &= \sum_{m=1}^{M} \sum_{n=1}^{N} \left[\langle c_{m,n} \rangle_{\boldsymbol{c}|\boldsymbol{g},\hat{\boldsymbol{f}}^{(k)}} \ln(h_{m,n} f_n) - h_{m,n} f_n \right] = \\ &= \sum_{m=1}^{M} \sum_{n=1}^{N} \left[g_m \frac{h_{m,n} \hat{f}_n^{(k)}}{\sum_{n'=1}^{N} h_{m,n'} \hat{f}_{n'}^{(k)}} \ln(h_{m,n} f_n) - h_{m,n} f_n \right]. \end{aligned}$$

The gradient is

$$\frac{\partial Q(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \boldsymbol{g})}{\partial f_n} = \frac{1}{f_n} \sum_{m=1}^{M} g_m \frac{h_{m,n} \hat{f}_n^{(k)}}{\sum_{n'=1}^{N} h_{m,n'} \hat{f}_{n'}^{(k)}} - \sum_{m=1}^{M} h_{m,n}.$$

Thus, the maximum of $Q(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \boldsymbol{g})$ is reached for $\boldsymbol{f} = \hat{\boldsymbol{f}}^{(k+1)}$, where

$$\hat{f}_n^{(k+1)} = C_n^{(k)} \hat{f}_n^{(k)} = \hat{f}_n^{(k)} \left\{ \frac{1}{\sum_{m=1}^{M} h_{m,n}} \sum_{m=1}^{M} \frac{g_m h_{m,n}}{\sum_{n'=1}^{N} h_{m,n'} \hat{f}_{n'}^{(k)}} \right\}, \tag{3.10}$$

in which the coefficients $C_n^{(k)}$ describe how the pixel intensities in the reconstructed image are updated throughout the iterations of the MLEM algorithm.

We conclude this section by commenting on some of the properties of the MLEM algorithm that follow directly from the relation just derived above. First of all, recall that the numbers $h_{m,n}$ were defined as the exposure time τ multiplied by the probability that a gamma-ray photon emitted from voxel B_n in object space is detected at detector pixel m. Thus, all the $h_{m,n}$ are non-negative and so are the numbers g_m, as they count the number of gamma-ray photons collected at the m^{th} detector pixel. If all the $\hat{f}_n^{(k)}$ for $n = 1, \ldots, N$ are non-negative, then also all the $\hat{f}_n^{(k+1)}$ are non-negative, as the quantity in $\{\ldots\}$ above is non-negative. In other words, iterations of the relation in (3.10) automatically ensures positivity of $\hat{\boldsymbol{f}}^{(k+1)}$, provided that $\hat{\boldsymbol{f}}^{(k)}$ satisfies positivity constraints as well. Notice further that the probability s_n that a gamma-ray photon emitted from voxel B_n satisfies $s_n = \sum_{m=1}^{M} h_{m,n} > 0$ for all $n = 1, \ldots, N$. This property, along with the positivity of $\hat{\boldsymbol{f}}^{(k)}$ ensures that, in (3.10), a positive number is never divided by zero [179]. Should zero divided by zero occur in (3.10), we will define such a quantity as zero [179].

An important property of the MLEM algorithm can be obtained by multiplying both sides of (3.10) by $s_n = \sum_{m=1}^{M} h_{m,n}$ and then by summing over n, for $n = 1, \ldots, N$. This gives

$$\sum_{n=1}^{N} s_n \hat{f}_n^{(k+1)} = \sum_{m=1}^{M} g_m \frac{\sum_{n=1}^{N} h_{m,n} \hat{f}_n^{(k)}}{\sum_{n'=1}^{N} h_{m,n'} \hat{f}_{n'}^{(k)}} = \sum_{m=1}^{M} g_m = \text{constant},$$

which implies that the MLEM algorithm conserves the quantity $\sum_{n=1}^{N} s_n \hat{f}_n^{(k)}$ throughout the iterations. Notice that $\sum_{m=1}^{M} g_m$ is the total number of photon counts collected by the hardware during the exposure time τ. Hence, we can interpret one iteration of the MLEM algorithm as a way to redistribute the total detectable activity $\sum_{n=1}^{N} s_n \hat{f}_n^{(k)}$ so that $\Pr(\boldsymbol{g} \mid \hat{\boldsymbol{f}}^{(k+1)}) \geqslant \Pr(\boldsymbol{g} \mid \hat{\boldsymbol{f}}^{(k)})$ for $k \geqslant 0$.

The relation in (3.10) is clearly non-linear in $\hat{\boldsymbol{f}}^{(k)}$. Instead, estimate $\hat{\boldsymbol{f}}^{(k+1)}$ is calculated by multiplying $\hat{\boldsymbol{f}}^{(k)}$ by a (non-negative) corrective factor $C_n^{(k)}$. In particular, we notice that if agreement between the data and the estimate is reached (in other words, $\sum_{n=1}^{N} h_{m,n} \hat{f}_n^{(k)} = g_m$ for all $m = 1, \ldots, M$) then all the corrective factors $C_n^{(k)}$ will be 1. An alternative way to state this is by saying that if $\sum_{n=1}^{N} h_{m,n} \hat{f}_n^{(k)} = g_m$ for all $m = 1, \ldots, M$, then $\hat{\boldsymbol{f}}^{(k)}$ is a fixed-point for the relationship in (3.10).

We noticed above that the data $\boldsymbol{g}$ we collected are necessarily noisy. Thus, any $\hat{\boldsymbol{f}}^{(k)}$ for which agreement with the data is enforce is necessarily noisy. The multiplicative nature of the MLEM algorithm as presented in (3.10) precludes the possibility for the noise in $\hat{\boldsymbol{f}}^{(k)}$ to be normally distributed. Indeed, if that were the case, we would have a non-zero probability to have one of the $\hat{f}_n^{(k)}$ take on a negative value, in violation of the positivity constraints automatically enforced by the MLEM algorithm. On the other hand, it has been observed that the probability density function for the grey level at a pixel in the image is well approximated by a log-normal probability law [190, 191].

We noted at the beginning of this chapter that the goal of MLE is to find a value of the parameter vector that maximizes the likelihood (or, equivalently, its logarithm). In a practical case, there might be more than one parameter vector that maximizes the likelihood. In the context of image reconstruction (in which the parameter vector we want to estimate is an image $\boldsymbol{f}$), we talk about *null functions* [192, 193]. A null function $\boldsymbol{f}_{\text{null}}$ for $\boldsymbol{f}$ is a function that when added to $\boldsymbol{f}$ produces the same mean data as $\boldsymbol{f}$ does. Here, "mean data" have to either be understood as the data $\boldsymbol{g}$ averaged over many realizations of the noise (so that the noise itself is averaged out), or as the expected data vector $\langle \boldsymbol{g} \rangle$. Hence, the interpretation of a null function is that if we take two data sets, say $\boldsymbol{g}$ and $\boldsymbol{g}'$, the former produced by imaging $\boldsymbol{f}$ and the latter produced by imaging $\boldsymbol{f} + \boldsymbol{f}_{\text{null}}$, any difference between $\boldsymbol{g}$ and $\boldsymbol{g}'$ is solely due to different realizations of the random noise. In other words, it is not possible to discern between $\boldsymbol{f}$ and $\boldsymbol{f} + \boldsymbol{f}_{\text{null}}$ by examining the two data sets. Null functions depend on the imaging system we are using. In a sense, they are objects that are "invisible" through the system. If the imaging system has null functions, there are many object estimates that give the same likelihood.

An important question that the relation in (3.10) does not answer concerns the number of iterations of (3.10) we need to perform to obtain an image suitable for the task of interest. As we saw above, the MLEM algorithm strives for agreement between the noisy measured data $\boldsymbol{g}$ and the image of the estimate [69]. If a large number of iterations is performed, agreement with the measured data often results in a virtually useless image consisting of a few bright pixels in a black uniform background ("night-sky" reconstruction). It can be shown [69] that if the MLEM algorithm is used to reconstruct a discretized object, then the algorithm will converge to a unique night-sky reconstruction, regardless of starting point. It has also been shown [194] that image quality perceived by a human observer reaches a peak after a small number of iterations and then it rapidly decreases due to the night-sky effect. Other artifacts, such as large distortions near the edges of the image, have been observed as well [195]. It has also been noticed that noise and reconstruction artifacts generally increase as successive estimates $\hat{\boldsymbol{f}}^{(k)}$ are calculated. Such artifacts are essentially due to the ill-posedness [22, 196] of the image-reconstruction process.

One way to control noise in the ML reconstruction is to introduce a stopping rule and to stop the algorithm before excessive noise amplification occurs. Some stopping rules are discussed in [197, 198]. For example, the rule suggested in [197] consists in stopping the reconstruction algorithm when the following condition is reached:

$$\chi^2 = \sum_{m=1}^{M} \frac{\left[g_m - \sum_{n=1}^{N} h_{m,n} \hat{f}_n^{(k)} \right]^2}{\sum_{n=1}^{N} h_{m,n} \hat{f}_n^{(k)}} \approx M,$$

in which we have recognized the expression for a χ^2 test [199, 200]. More elaborate stopping rules, based, for example, on the pixel updating coefficients $C_n^{(k)}$ introduced in (3.10) have been proposed as well [201, 202].

Whatever stopping rule is used, it should take into consideration the fact that images are often being viewed by a human observer to learn something about the object being imaged. This links the development of a stopping rule to the concept of task-based assessment of image quality, as discussed in Chapter 2. Some work has been done in that direction. For example, in [203] the authors compare different image reconstruction algorithms—including the MLEM algorithm—and stopping rules. A ranking of these methods based on the AUC for a signal detection task is then produced.

A second way to reduce edge artifacts and noise in the reconstructed image is to modify the iterative expression in (3.10) so that the estimate is intentionally a blurred version of the underlying intensity $\boldsymbol{f}$, rather than the intensity itself. The maximum-likelihood estimate can also be forced to belong to a set of vectors that satisfy some desirable properties, such as smoothness near the edges of the image.

This approach has been successfully pursued in [195].

Other modifications of the basic MLEM algorithm have been proposed and analyzed. This includes the addition to the model of prior knowledge of the statistical properties of $\boldsymbol{f}$ [184]. Another important modification to the MLEM algorithm consists in grouping the data into an ordered sequence of subsets and processing these subsets one after the other. This method has been shown to greatly speed up the reconstructing algorithm while maintaining the same reconstruction quality [204]. Finally, adaptation mechanisms [205] have been fruitfully used in conjunction with the MLEM algorithm, as shown in the preliminary results of [7].

CHAPTER 4
List-Mode Data and Maximum Likelihood Estimation

This chapter begins with a discussion of the basic setup of gamma-ray cameras. The physical phenomena responsible for the generation of a signal upon interaction of a gamma-ray photon with the camera's crystal are discussed in some detail. A formal description of the statistical processes occurring inside a photomultiplier tube (PMT) provides the theoretical framework for the remaining topics covered in the chapter, such as maximum-likelihood estimation (MLE) of position of interaction and image reconstruction from list-mode data. After a brief digression on Markov chain Monte Carlo (MCMC) methods, the chapter continues by presenting the theory of list-mode task performance for some relevant observers, namely the Hotelling and the ideal observers. Estimation from list-mode data is covered in more details towards the end of the chapter. We then apply the list-mode maximum-likelihood expectation-maximization (LMMLEM) algorithm to two different problems: estimation of a vector of parameters that characterize the object being imaged and image reconstruction from list-mode PET data.

4.1 The Gamma-Ray Camera

A gamma-ray camera is a device used in nuclear medicine to image gamma-ray radiation generated by radioisotopes. The first gamma-ray camera was developed in 1957 by Hal Oscar Anger (May 20, 1920–October 31, 2005). Anger's original design [206–208], often referred to as an "Anger camera," is still widely used today. The gamma-ray camera that Anger proposed [207] consisted of a lead housing with a pinhole aperture enclosing a 4-inch-diameter, 1/4-inch-thick circular scintillation crystal made of thallium-activated sodium iodine, coupled with a bank of seven 1.5-inch-diameter photomultiplier tubes (PMTs), arranged in a hexagonal configuration (six PMTs at the vertices of a hexagon plus a PMT at the center). One side of the scintillation crystal is facing the pinhole aperture, while the other side is facing the PMT bank. The space between the scintillation crystal and the PMTs is filled with a transparent optical fluid. The gamma-ray camera is connected via appropriate circuitry to a cathode-ray tube (CRT) for display. The circuitry, which included resistances, difference circuits, and amplifiers, used the PMT outputs to generate—using Anger arithmetic—signals approximatively corresponding to the 2D location of scintillation in the crystal:

$$\hat{X} = \frac{\sum_{k=1}^{K} g_k X_k}{\sum_{k=1}^{K} g_k}, \qquad \hat{Y} = \frac{\sum_{k=1}^{K} g_k Y_k}{\sum_{k=1}^{K} g_k},$$

where $g_1, \ldots, g_K$ are the PMT signals and $X_1, \ldots, X_K$ and $Y_1, \ldots, Y_K$ are the X and Y PMT locations, respectively. The expressions above correspond to centroid estimation. Variants on this basic idea have been proposed as well [209, 210].

Scintillation locations due to the gamma-rays that make it through the pinhole in the lead shield are reproduced as flashes of light on the CRT display. Over time, they form an image of the gamma-ray emitting source distribution. The camera was used for *in vivo* ^{131}I imaging of the human thyroid gland.

A diagram of a typical gamma-ray camera is provided as Figure 4.1. Many designs for gamma-ray cameras have been proposed in the past (see, for example, [211–219]). A relevant design is the one discussed in [211, 212], which is a modular gamma-ray camera featuring a large detector area and optimal position estimation based on maximum-likelihood. This camera can be used as building block for larger imaging systems [220].

When a gamma-ray interacts with the scintillation crystal, a shower of visible-light photons is produced. Some of these photons travel through the crystal and the light guide, and enter the PMT bank. When a photon enters a PMT and interacts

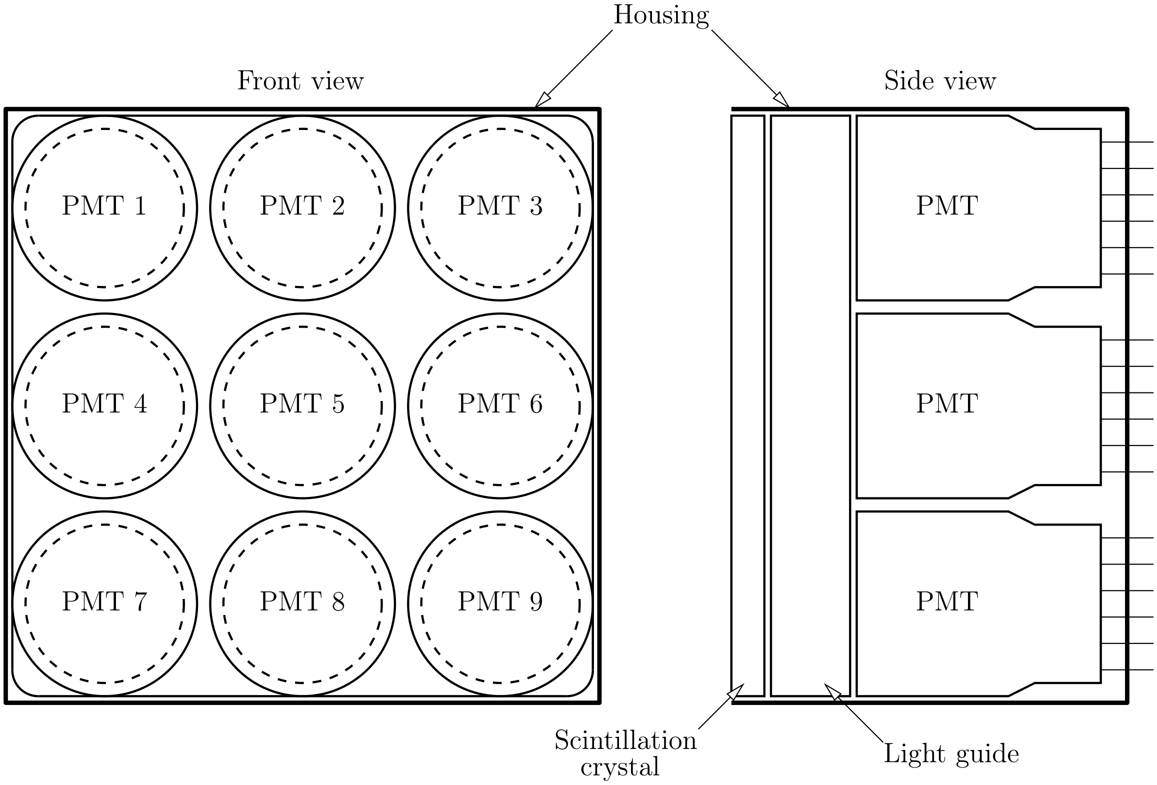


Figure 4.1. Diagram of a gamma-ray camera

with it, a measurable electric signal is produced. This signal is amplified by the PMT and processed by the camera's electronics. To provide a more detailed explanation of the whole process, we first need to describe how a gamma-ray photon interacts with matter, how the scintillation process takes place, and, finally, how a PMT converts optical photons into electric signals.

4.1.1 Linear Attenuation

A beam of gamma-ray radiation passing through matter gets attenuated according to Beer's Law [221, 222]:

$$I(z) = I_0 e^{-\mu z},$$

in which I_0 is the incident gamma-ray flux (measured in photons per unit area), $I(z)$ is the gamma-ray flux at depth z in the traversed medium, and μ is the linear attenuation coefficient, as shown in Figure 4.2.

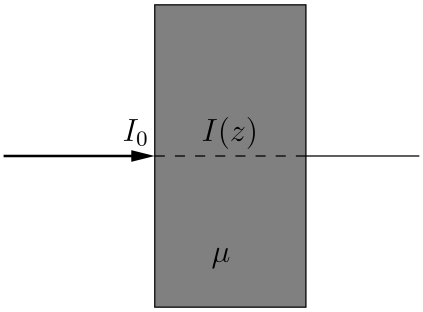


FIGURE 4.2. Diagram of absorption of a beam of light as it travels through the traversed medium

The linear attenuation coefficient μ measures the probability of a photon-matter interaction per unit length traveled in the medium. It depends on many factors, including photon energy E and density and effective atomic number Z_{eff} of the medium [221, 223]. A gamma-ray photon can interact with matter via three different processes. These are: photoelectric absorption, Compton scattering, and positron-electron pair production. The prevalence of each of these different processes depends on the atomic number of the chemical elements making up the transversed medium and on the energy of the gamma-ray photons. For example, positron-electron pair production (usually denoted as $\gamma \rightarrow e^- + \beta^+$) only occurs when the energy of the gamma-ray photon exceeds twice the rest energy of an electron. Usually, gamma-ray photons used in medical imaging have energies between 50 keV and 550 keV, well below the 1022 keV energy threshold for positron-electron pair production. Thus, only Compton scattering and photoelectric absorption are of interest in nuclear medicine.

4.1.2 Compton Scattering

Compton scattering [224, 225] is a phenomenon whereby an incoming gamma-ray photon interacts with a loosely-bound or free electron in an atom, and only a portion of the energy of the gamma-ray photon is transferred to the electron. The electron recoils and departs from the atom, which becomes ionized. The remaining energy is emitted as a scattered photon. If this photon has enough energy, the process just described may be repeated. Conservation of total energy and momentum and special relativity allow us to calculate scattering angles and energy of the scattered photon. The relationship between the shift in wavelength and the deflecting angle of the scattered photon is as follows [224]:

$$\lambda' - \lambda = \frac{h}{m_{\mathrm{e}} c}\bigl(1 - \cos\vartheta\bigr),$$

in which λ and λ' are the wavelengths of the incident and scattered photons, respectively, h is Planck's constant, m_{e} is the mass of the electron, c is the speed of light, and ϑ is the scattering angle of the scattered photon (see Figure 4.3). In terms of photon energies $E = hc/\lambda$ and $E' = hc/\lambda'$, the expression above becomes [224]

$$E' = \frac{E}{1 + \dfrac{E}{m_{\mathrm{e}} c^2}\bigl(1 - \cos\vartheta\bigr)}.$$

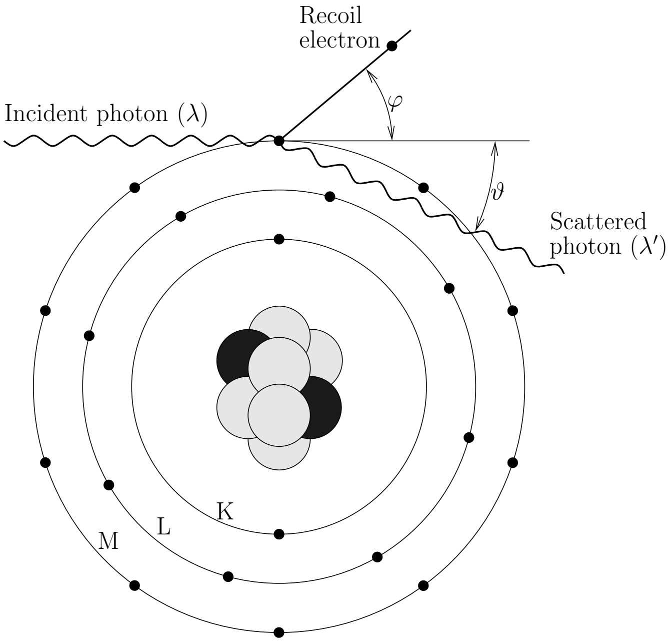


FIGURE 4.3. Diagram of Compton scattering interaction

The relationship between the scattering angle ϑ and the electron recoil angle φ is given by

$$\cot\varphi = \left(1 + \frac{h}{\lambda m_{\mathrm{e}} c}\right) \tan\left(\frac{\vartheta}{2}\right).$$

The angular distribution of the scattered gamma-ray photon is modeled by the *Klein-Nishina formula* [226]. This distribution takes the form:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = Zr_0^2 \left[\frac{1}{1+\alpha(1-\cos\vartheta)}\right]^2 \left(\frac{1+\cos^2\vartheta}{2}\right) \left\{1 + \frac{\alpha^2(1-\cos\vartheta)^2}{(1+\cos^2\vartheta)[1+\alpha(1-\cos\vartheta)]}\right\},$$

where $\mathrm{d}\sigma/\mathrm{d}\Omega$ is the differential scattering cross-section per solid angle, Z is the atomic number, r_0 is the electron radius, and we have set $\alpha = E/m_{\mathrm{e}}c^2$ for notational convenience. The differential scattering cross-section $\mathrm{d}\sigma/\mathrm{d}\Omega$ for different energies of the incident photon is plotted in Figure 4.4. For convenience, we assumed $Zr_0^2 = 1$ in these plots. Some methods for generating random scattering angles according to the distribution above have been proposed in [227–229].

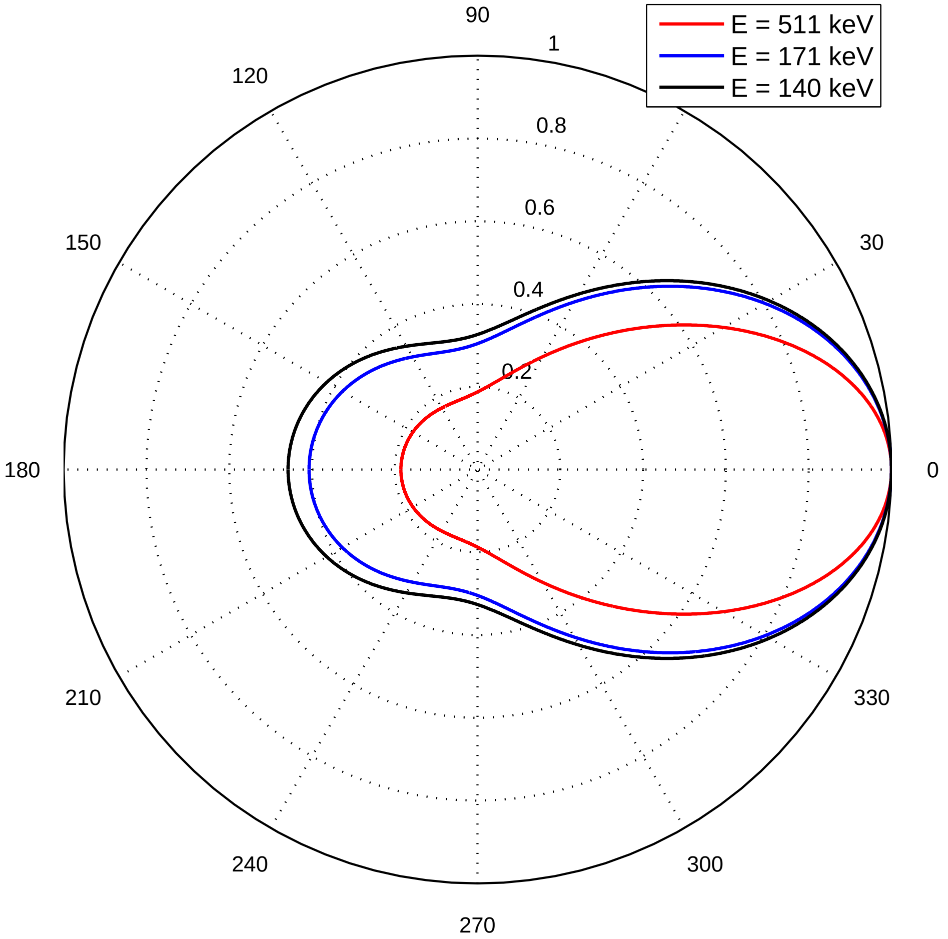


FIGURE 4.4. Differential scattering cross-sections for different energies E of the incident photon. In these plots, we assumed $Zr_0^2 = 1$

4.1.3 Photoelectric Absorption

In photoelectric absorption, an incoming gamma-ray photon is completely absorbed by an electron and it transfers all of its energy to the electron [10]. This process usually involves one of the most tightly bound electrons, in the inner shells of the atom. The electron, now referred to as a photoelectron, ejects from the atom, which, as in the case of Compton scattering, becomes ionized (see Figure 4.5). The recoil of the atom left behind conserves momentum. No scattered photon is emitted from the atom, as all the energy of the incident photon is absorbed by the electron. Again, by conservation of energy, the kinetic energy of the photoelectron equals the energy of the gamma-ray photon minus the electron's binding energy. The probability that a photoelectric interaction will occur is related to the atomic number Z of the nucleus and it is approximatively proportional to Z^3 [230]. For this reason, good scintillation materials have high atomic numbers [10]. In materials with low Z (such as soft tissues), photoelectric absorption accounts for a small fraction of photon interactions with matter [230].

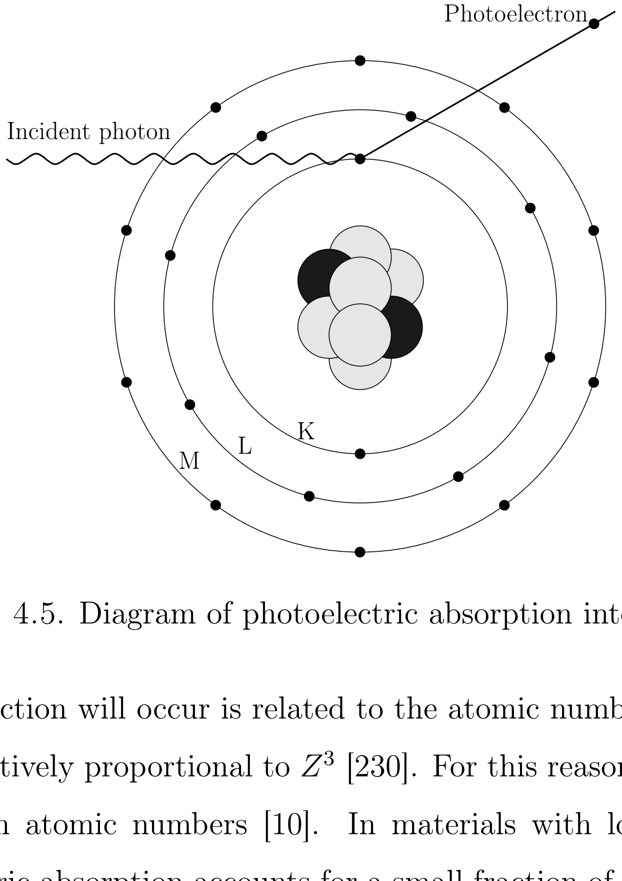


FIGURE 4.5. Diagram of photoelectric absorption interaction

4.1.4 Scintillators and the Scintillation Process

A scintillator is any material able to emit low-energy photons in the visible range upon absorption of high-energy photons, such as gamma-ray photons [230]. To describe how a scintillator converts gamma-ray photons into visible photons that can be used to produce a measurable electric signal, we have to make use of solid-state physics and describe the band structure [231, 232] of an inorganic scintillator. A diagram of the typical band structure of a scintillator [10, 233] is shown in Figure 4.6. The filled band corresponds to the lower energy levels that are usually filled by electrons. The next allowed energy levels are collectively referred to as the valence band, which is occupied by electrons that are bound to the crystalline lattice. The valence band is also the uppermost filled band at absolute zero. The filled band and the valence band are separated by a forbidden band, which, according to the laws of quantum mechanics, corresponds to energy levels that electrons are not allowed to have [230]. Finally, the conduction band contains electrons that have enough energy to move freely throughout the crystal. The energy gap between the valence band and the conduction band is called band gap and it is denoted as E_g.

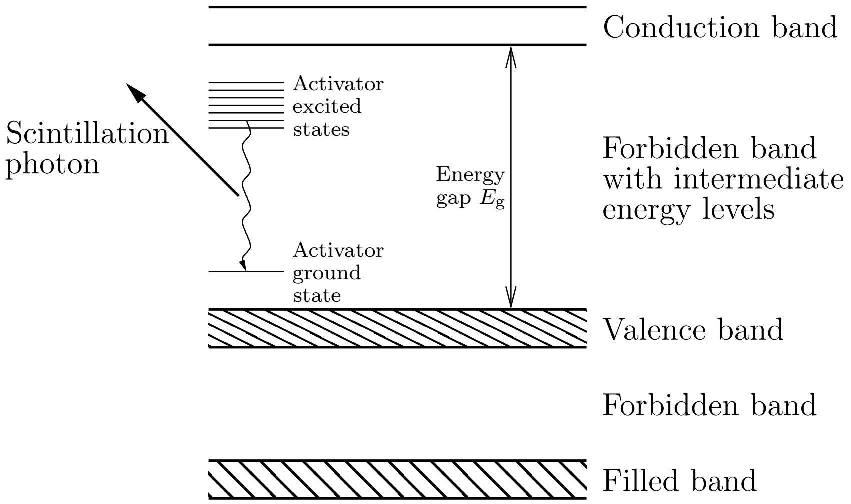


Figure 4.6. Diagram of the typical band structure of a scintillator

Upon light-matter interaction between a gamma-ray photon and an atom in the crystal, either Compton scattering or photoelectric absorption can occur. This results in a high-energy electron that in turn excites electrons so that their energy increases from the valence band to the conduction band. When this happens, holes are created in the valence band. In a pure crystal, electrons that have moved from the valence band to the conduction band can lose energy in a variety of ways. One of them is by emitting a photon. That would make the electron go back to the valence band and recombine with the hole. This emitted photon of light is usually re-absorbed by another valence electron before it can escape the crystal [10].

To deal with this problem, impurities (called *activators*) are intentionally introduced in highly diluted concentrations into the crystal to alter the band structure by creating lattice imperfections (called *luminescence centers*), thus allowing additional energy levels in the forbidden band, between the conduction band and the valence band. These impurities increase the efficiency for dissipating the absorbed gamma-ray energy in the crystal as scintillation photons and introduce intermediate energy levels in the band structure, so that emitted photons now have longer wavelength and are in or near the visible range of the electromagnetic spectrum [10, 230]. The wavelength of the visible photons emitted by the crystal depends on the energy differences between the new energy levels in the forbidden band. Such energy differences, in turn, depend on the dopant material used. Ideally, we would like to have emission energies that do not overlap with the crystal absorption band gaps, so that reabsorption of the emitted photons does not occur [10, 233].

By far, the most popular material for a scintillation crystal is NaI(Tl). For a detailed description of the scintillation process in NaI(Tl), the interested reader can consult [234] and the references cited therein. Advantages of NaI(Tl) include high light yield ($\eta_{\mathrm{NaI(Tl)}} = 37700$ photons/MeV, see [235]) resulting in good energy resolution, short decay time (230 ns) allowing a high detection rate, high effective atomic number ($Z_{\mathrm{eff}} = 51$, see [10]), high density (3.67 g/cm^3), no significant self absorption of the scintillation light, and low cost. The high density contributes to a large absorption coefficient (about 0.3384 cm^{-1} at 511 keV [236]), thus reducing the need for a very thick slab of material to obtain a high-absorbing crystal. Scintillation crystals made of NaI(Tl) can be fabricated in a wide variety of sizes and shapes. They can be as small as 1 cm in diameter up to 60 cm $\times$ 40 cm in size [230]. The absolute scintillation efficiency, which is defined as the percentage of absorbed gamma-ray energy that is emitted as visible photons, of NaI(Tl) has been reported to be about 10.8–13.5% [230,235,237]. This number puts NaI(Tl) among the most efficient scintillation materials.

However, NaI(Tl) also presents some disadvantages. For example, NaI(Tl) crystals are quite fragile, and may fracture under conditions of mechanical stress or rapid temperature change. In addition, NaI(Tl) is highly hygroscopic, meaning that it absorbs the moisture naturally present in the air. When this happens, the crystal turns yellow and starts absorbing scintillation photons [233]. To prevent this, NaI(Tl) scintillation crystals are hermetically sealed to prevent air or water coming into contact with the crystal. Important parameters of NaI(Tl) and other common scintillation materials [238] are summarized in Table 4.1. Further details can be found in [235,237,239–242].

4.1.5 Photomultiplier Tubes

We saw above that after a gamma-ray photon interacts with the crystal, a shower of visible-light photons is emitted in the crystal. These photons are converted to measurable electric signals by means of a bank of photomultiplier tubes (PMTs) coupled to the scintillator via a light guide. Features of these signals—such as amplitude and shape—are used to estimate parameters of the gamma-ray photons absorbed by the crystal. A diagram of a PMT is provided in Figure 4.7. A typical PMT consists of a hermetically sealed chamber enclosing a photocathode, a focusing grid, a series of

Material	**Density [g/cm^3]**	**Effective atomic number, Z_{eff}**	**Wavelength of maximum emission [nm]**	**Decay time [μs]**	**Index of refraction at maximum emission, n**	**Photons per MeV**	**Hygroscopic?**
NaI(Tl)	3.67	51	415	0.23	1.85	37700[a]	yes
CsI(Na)	4.51	54	420	0.63	1.84	38500[a]	slightly
CsI(Tl)	4.51	54	540	0.68	1.80	51800[a]	slightly
Lu_2SiO_5(Ce)	7.40	65	420	0.04	1.82	25000[b]	no
CaF_2(Eu)	3.19	17	435	0.9	1.44	23650[a]	no
^{6}LiI(Eu)	3.49	54	470	1.4	1.96	10680[c]	very
CsF	4.11	53	390	0.004	1.48	2000[d]	very
$CdWO_4$	7.90	64	480	5.0	2.20	15300[a]	no
$GdSiO_5$(Ce)	6.71	59	430	0.06	1.85	7540[e]	no

[a] [235]; [b] [239]; [c] [237]; [d] [240]; [e] [241].

Table 4.1. Physical parameters for common scintillation materials (adapted from [10])

terminals known as dynodes, an anode, and a bank of resistors.

The photocathode is the component of the PMT that is closest to the crystal. It consists of a thin layer of material (such as K_2CsSb or Na_2KSb [233]) that possesses photoemissive properties. When a scintillation photon enters the PMT through the entrance window and strikes the photocathode, a photoelectron is ejected from the photocathode with some probability. This probability, which is called quantum efficiency and is usually denoted as η, is defined as the ratio between the mean number of emitted photoelectrons to the mean number of photons incident on the PMT. Typical values for η are around 0.20–0.25. Vacuum must be maintained inside the PMT so that no electrons are absorbed by gas molecules and lost [233].

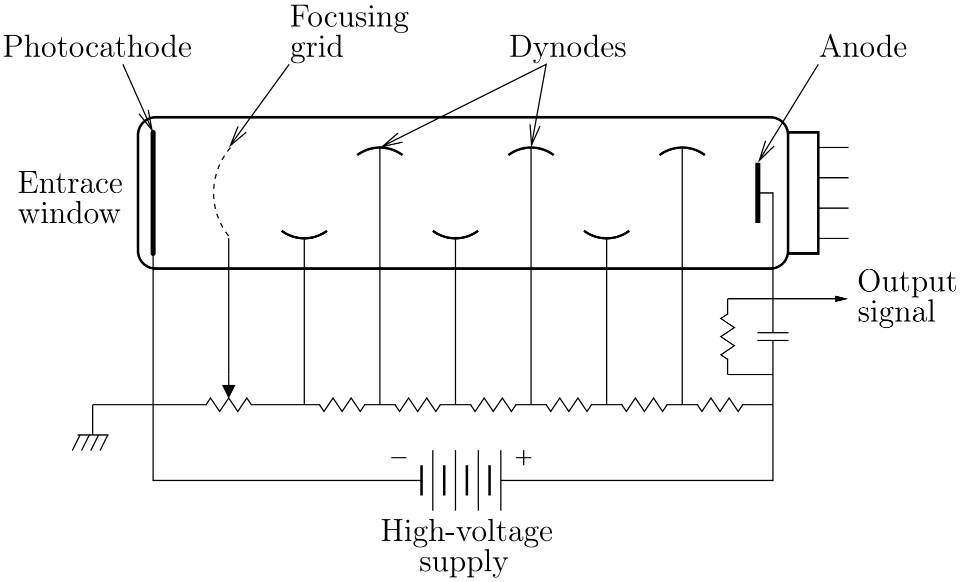


FIGURE 4.7. Diagram of a photomultiplier tube (adapted from [3])

The focusing grid and dynodes are driven by a high-voltage supply. The voltage at different ends of the PMT is on the order of 1000 V. This voltage gets split by a voltage divider—typically consisting of a bank of high-impedance resistors and sometimes a zener diode—so that the voltage differences between the focusing grid and the first dynode and adjacent dynodes are on the order of 100 V. The voltage divider can also include decoupling capacitors connected across the last three or four dynodes to provide transient signal charge. The voltage difference between one dynode and the next induces an electric field inside the PMT. A photoelectron emitted by the photocathode gets first focused by the focusing grid towards the dynodes. Along its way to the first dynode, it acquires kinetic energy and accelerates due to the presence of the electric field. An accelerated photoelectron colliding with the first dynode causes a few secondary electrons being emitted. These electrons are, in turn, accelerated by the electric field present between the first dynode and the next until they hit the second dynode. Again, each electron hitting the dynode causes emission of a few electrons, and the process outlined above continues until the bundle of electrons reaches the anode at the end of the PMT. If the PMT has d dynodes and each electron striking a dynode produces, on average, n electrons, then the overall multiplication factor of the PMT is n^d. For example, for $d = 10$ and $n = 5$, one electron ejected at the photocathode gives rise to almost 10^7 electrons at the anode. This multiplication factor can be fine-tuned by varying the voltages at the dynodes.

At the anode, all the electrons are collected and a measurable output signal—in the form of a narrow current pulse—is produced. This pulse, whose peak amplitude is of the order of 1 mA, is transmitted over a coaxial cable to amplifying electronics, so that it can be analyzed. A transimpedance amplifier is used to convert the current to a voltage pulse. A shaping amplifier further amplifies the signal and reshapes it, making it broader and smoother. A broad pulse is easier to sample via an analog-to-digital converter. The digitized samples are then scanned for events. Detected events are, in turn, fed to a computer for further processing [45].

Photomultiplier tubes are currently the best devices we can use for converting visible photons into a current signal. They, however, present some disadvantages. They are bulky and sensitive to changes in temperature, humidity, and magnetic fields [230]. Alternatives to PMTs do exist. These include solid-state photon detectors, such as avalanche photodiode arrays [243, 244].

4.1.6 Statistical Model

Our discussion of gamma-ray cameras would not be complete if we did not discuss how we can statistically describe all the random processes involved with the generation of a signal upon the interaction of a gamma-ray photon with the crystal. In this section, we will provide only the general idea; for a detailed treatment the interested reader can consult [69, 245].

As we saw above, an interaction between a gamma-ray photon at location $\boldsymbol{R}$ in the crystal produces a shower of lower-energy optical photons. The efficiency of this process is very low. As we mentioned earlier, for NaI(Tl), which is one of the most efficient scintillation materials, the absolute scintillation efficiency is about 10.8–13.5%: most of the gamma-ray energy gets dissipated as phonons (crystal lattice vibrations). The rarity with which the optical photons are generated satisfies the postulates for Poisson statistics [69] and it therefore allows us to define a Poisson random variable, N_{opt}, for the number of optical photons generated upon interaction of a gamma-ray photon with the crystal. We will denote the mean of N_{opt} as $\overline{N}_{\text{opt}}(E_{\text{int}})$, in which E_{int} is the energy deposited during interaction in the scintillator at location $\boldsymbol{R}$. Although there is evidence that N_{opt} might not follow Poisson statistics [246, 247] we will assume, for simplicity, that the statistics of N_{opt} are well approximated by the Poisson model.

As shown in Figure 4.1, some of the N_{opt} optical photons will propagate through the crystal and the light guide and will impinge upon the entrance faces of the PMTs. This process is mathematically formalized by introducing the probability $\beta_k(\boldsymbol{R})$ that an optical photon emitted at location $\boldsymbol{R}$ in the crystal will reach the k^{th} PMT. The probability $\beta_k(\boldsymbol{R})$ is a strong function of $\boldsymbol{R}$ and k, but it does not depend on E_{int}. Conditioned on the knowledge of N_{opt}, the number of photons reaching the k^{th} PMT follows a binomial distribution [127, 128] with parameters N_{opt} and $\beta_k(\boldsymbol{R})$. Furthermore, by the *binomial selection theorem* [69] and if we treat N_{opt} as random, the number of optical photons reaching the k^{th} PMT for an interaction at $\boldsymbol{R}$ is a Poisson random variable with mean $\beta_k(\boldsymbol{R})\overline{N}_{\text{opt}}(E_{\text{int}})$.

A photon that arrives at the k^{th} PMT has a probability η_k (which we called the quantum efficiency) of producing a photoelectron upon impinging on the photocathode. Once again, by the binomial selection theorem, the mean number of photoelectrons produced follows a Poisson distribution with mean [245]

$$\overline{n}_k(\boldsymbol{R}, E_{\text{int}}) = \eta_k \beta_k(\boldsymbol{R})\overline{N}_{\text{opt}}(E_{\text{int}}). \tag{4.1}$$

To complete the analysis, we have to consider the amplification process taking place inside each PMT. We saw before that when an electron collides with a dynode stage, a small random number of secondary electrons is emitted. Thus, we have to treat the whole multiplication process inside the k^{th} PMT as a random phenomena characterized by the probability density function of the gain G_k. It is argued in [245] that all the random processes that take place right after each scintillation event will result in PMT outputs $V_k(\boldsymbol{R}, E_{\text{int}})$ being normally distributed with means

$$\overline{V}_k(\boldsymbol{R}, E_{\text{int}}) = \overline{G}_k \overline{n}_k(\boldsymbol{R}, E_{\text{int}}) = \overline{G}_k \eta_k \beta_k(\boldsymbol{R})\overline{N}_{\text{opt}}(E_{\text{int}}),$$

where $\overline{G}_k$ is the mean gain of the k^{th} PMT and $k = 1, \ldots, K$. Notice that what happens inside one PMT is independent on what happens inside another PMT. For this reason, the covariance matrix of the vector of the $\overline{V}_k(\boldsymbol{R}, E_{\text{int}})$ for $k = 1, \ldots, K$ is diagonal. For good-quality PMTs, the distribution of G_k is sharply peaked (meaning that most of the randomness in $V_k(\boldsymbol{R}, E_{\text{int}})$ is due to the randomness in $n_k(\boldsymbol{R}, E_{\text{int}})$) and so we can estimate the number of electrons at the anode of the k^{th} PMT by dividing $V_k(\boldsymbol{R}, E_{\text{int}})$ by $\overline{G}_k$ and rounding to the nearest integer [69, 245]

$$U_k = \text{round}\left(\frac{V_k(\boldsymbol{R}, E_{\text{int}})}{\overline{G}_k}\right),$$

where the function round(x) rounds the value of x to its nearest integer. An excellent model for the statistics of $U_1, \ldots, U_K$ (and, in general, for the output of any photocounting detector [69]) is a multivariate Poisson probability [245]:

$$\Pr\bigl(\boldsymbol{U} \mid \boldsymbol{R}\bigr) = \prod_{k=1}^{K} \frac{[\overline{U}_k(\boldsymbol{R})]^{U_k}}{U_k!} e^{-\overline{U}_k(\boldsymbol{R})}, \tag{4.2}$$

where the $\overline{U}_1(\boldsymbol{R}), \ldots, \overline{U}_K(\boldsymbol{R})$ are the means of the random variables $U_1, \ldots, U_K$, which, in turn, are the components of the random vector $\boldsymbol{U}$. Notice that

$$\overline{U}_k(\boldsymbol{R}) \approx \frac{\overline{V}_k(\boldsymbol{R}, E_{\text{int}})}{\overline{G}_k} = \frac{\overline{G}_k \overline{n}_k(\boldsymbol{R}, E_{\text{int}})}{\overline{G}_k} = \overline{n}_k(\boldsymbol{R}, E_{\text{int}}),$$

where $k = 1, \ldots, K$ and the numbers $\overline{n}_k(\boldsymbol{R}, E_{\text{int}})$ were introduced in (4.1). Knowledge of the statistics of digitized PMT outputs $U_1, \ldots, U_K$ is important because it allows us to perform position estimation via more sophisticated algorithms—such as maximum-likelihood methods [45, 211]—than the simple centroid method of [206–208].

4.2 What is List-Mode Data?

Today's gamma-ray cameras are imaging devices able to deliver many parameters for each event they collect. For example, we can consider the raw PMT outputs and mathematically collect them in a vector, which we call *attribute vector*. Alternatively, from the collection of PMT data, we can—as we will discuss in the next section—estimate the location of gamma-ray photon interaction within the crystal. In the latter case, we can still define an attribute vector; this time the attribute vector might include the estimated location of interaction, the total gamma-ray energy, a time-stamp, and so on. In both cases, we will denote the attribute vector as $\hat{\boldsymbol{A}}^{(j)}$ with the superscript j running from 1 to the total number J of collected events. The "hat" symbol we used in defining the attribute vector emphasizes the fact that such attribute vector is an estimated quantity. In other words, $\hat{\boldsymbol{A}}^{(j)}$ is a random quantity, and its statistical properties will play a crucial role in the reminder of this work.

When an object is imaged, a collection of J attribute vectors is measured. There are at least two possible ways to organize the information we learn from the detectors.

The first way, which is suggested from the description above, is to create the list $\hat{\mathscr{A}}$ of the J attribute vectors:

$$\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\}.$$

Again, the hat notation we used in $\hat{\mathscr{A}}$ underlines the fact that the list $\hat{\mathscr{A}}$ is random. We will refer to these data as *list-mode data.* List-mode data are not restricted to medical imaging applications [248], but they are used in high-energy physics [249] and astronomy [250] as well.

An alternative to collecting data in list-mode is to perform *binning.* In binning, the attribute space is partitioned into bins. Bins do not overlap and cover the whole attribute space. The list $\hat{\mathscr{A}}$ is analyze and, for each bin, the number of events that fall within that bin is counted. At the end of this process, the counts for all the bins form a histogram of the data in $\hat{\mathscr{A}}$. The histogram is retained and the original list $\hat{\mathscr{A}}$ is discarded. For simplicity, the descriptions we have just presented makes use of the list $\hat{\mathscr{A}}$. In practice, such list is never produced. Instead, the attributes $\hat{\boldsymbol{A}}^{(j)}$ are processed one at a time as they are acquired by the camera. Binning can also be performed by the camera circuitry, and the bin counts are read from the camera when the exposure time has elapsed.

As an example, assume that we are imaging a planar self-luminous object through a lens onto a detector. The lens will produce an image on the detector. Because of the lens, the image will be inverted and reversed. Depending on the separation between the object, the lens, and the detector, the image may also be magnified (or minified). In any case, we would expect the image on the detector to be similar to the planar object. To each photon emitted from the object that impinges on the detector, we can associate an attribute vector, for example the 2D location on the detector face at which the photon was absorbed and detected. In the case of list-mode data, the image of the object is simply represented in the memory of the computer as the list of 2D photon locations. If, on the other hand, we use the binned data representation, the image will be represented as a 2D grid of gray-level pixel intensities. In the latter case, which is very convenient if the image needs to be printed or displayed on a computer screen, a gray-level intensity is calculated by counting the number of photons that were localized within a small, square region of the detector.

List-mode data have several advantages over binned data. First of all, list-mode data, especially at low counts, allow for huge amounts of savings in storage requirements. To show this point, let us consider an example. Consider again the case of the 2D planar self-luminous object we discussed above. For the case of list-mode data, we can assume the attribute $\hat{\boldsymbol{A}}^{(j)}$ stores the K PMT outputs, where K is the number of PMTs. For this example, we will assume $K = 9$. As we argued before, we also assume that the PMT outputs are integer numbers. If each integer number takes up two bytes to be represented and if the total number J of events collected is, say, 10^5, then storing the list $\hat{\mathscr{A}}$ takes $2 \times 9 \times 10^5$ bytes, which corresponds to, roughly, 1.72 MB. If, on the other hand, we opt for using binning, we first need to decide how to bin the attribute space. The simplest way consists of quantizing the K-dimensional attribute space along each dimension with, say, 256 bins. The attribute space will result partitioned into $256^K = 256^9 \approx 4.72 \cdot 10^{21}$ bins. Even if we would use just one byte for each bin count, storing the histogram would require about $4.29 \cdot 10^9$ TB. As a comparison, the largest hard disk for which a prototype has been demonstrated has a storage capacity of 4 TB; it would take over a billion such disks to store in binned-mode the data from the example described above.

The second advantage of list-mode data is that all the desired information about an event can be easily stored without information loss. For example, in some cases we might be interested in associating a time-stamp to every event, or we might want to characterize an event based on the total energy of the gamma-ray photon. The time-stamp turns out very useful when we want to perform dynamic studies. For example, if we are imaging the heart, we might be interested in reconstructing images of the heart at different stages of the cardiac cycle. Or, we might want to use the collected data to estimate the amount of blood pumped out by the heart at each beat. Energy information can be used to select events and use only those events whose energy falls within a given energy window of interest. If we were to use binned-data to memorize all this information, we would realize pretty soon that storage is possible only if we use large bins. And as the bin size increases, more and more events will fall in the same bin and they will contribute to the bin count. Events in the same bin might thus be fairly "far apart" but they will still give the same contribution to the bin count.

For a theoretical standpoint, the most important advantage of list-mode data is that they allow for an accurate representation of what we measured. On the other hand, we can interpret binning as replacing an event vector with the one at the center of the bin into which the original vector falls. Thus, by binning, a systematic error is inevitably introduced [251] into the data. This does not happen if list-mode data are used instead.

Detector technology also makes us favor list-mode data versus binned data. Indeed, if the detector output is just a sequence of events, the detector circuitry is very simple and less prone to failure or fatigue. List-mode data also scale well with detector technology. To illustrate this point, consider a 2D gamma-ray camera and assume that each event $\hat{\boldsymbol{A}}^{(j)}$ it outputs is a pair $(\hat{x}^{(j)}, \hat{y}^{(j)})$ of spatial coordinates. As scintillation technology advances, we would expect more and more accurate estimates $\hat{x}^{(j)}$ and $\hat{y}^{(j)}$. Any algorithm that uses list-mode data would not need be modified to process these better estimates. On the other hand, if we use binned data, it seems reasonable to use bins whose size is comparable to the standard deviation with which $\hat{x}^{(j)}$ and $\hat{y}^{(j)}$ can be estimated. To reiterate the concept, list-mode data allow for the hardware to be upgraded with minimal (if any) software upgrade; if binned data are used, a software update will likely be needed. The new data structures (with smaller bins) might no longer fit into the memory of the computer.

Another advantage of list-mode data is that, with list-mode data, we can start doing reconstruction as soon as we collected information about the first event. This advantage of list-mode data has been emphasized multiple times in the literature. For example, we can acquire data in list-mode format and perform a real-time reconstruction. Or we can record smaller lists—one after the other—and use them cyclically in the reconstruction. Many more variants are possible too. On the other hand, binned data would require to wait for the whole collection time to elapse before starting the image reconstruction step. Most of the flexibility that list-mode data allow is simply not possible with binned data.

4.3 2D and 3D Position Estimation

In the previous section, we considered an example of an imaging system, and we used it to collect list-mode data. The data we collected in the list $\hat{\mathscr{A}}$ were the attribute vectors $\hat{\boldsymbol{A}}^{(j)}$, and we used the "hat" symbol to mean that the components of $\hat{\boldsymbol{A}}^{(j)}$ were noisy quantities. These quantities corresponded to estimates of electric charges, which were obtained by integrating over time the current pulses at the PMT outputs. However, we also mentioned the case in which the PMT outputs are processed to estimate spatial coordinates. In the latter case, the estimated spatial coordinates will still make up attribute vectors, which we will keep denoting by $\hat{\boldsymbol{A}}^{(j)}$. In this section, we want to address the problem of estimating the spatial coordinates from PMT outputs. Various methods have been proposed to carry out this estimation step. Among them, we want to recall the ones described in [45, 53, 60, 174, 252–255]. The method we present here [45, 59, 60, 173, 174, 253, 256, 257] is based on maximum-likelihood estimation [138, 139]. Thus, it enjoys all the properties we briefly discussed in § 3.2.

Assume that the vector $\boldsymbol{g}$ represents noisy PMT outputs for a photon-crystal interaction that occurred at location $\boldsymbol{R}$ in the camera's crystal. In our treatment, $\boldsymbol{R}$ will either represent a 2D or a 3D location. According to our discussion about PMTs, we will assume that the K components $g_1, \ldots, g_K$ of $\boldsymbol{g}$ are independent and follow Poisson statistics with parameters $\overline{g}_1(\boldsymbol{R}), \ldots, \overline{g}_K(\boldsymbol{R})$, respectively. We will refer to the vector $\overline{\boldsymbol{g}}(\boldsymbol{R})$ of parameters as the mean detector response function (MDRF) vector for the location of interaction $\boldsymbol{R}$. As our notation implies, the MDRF vector $\overline{\boldsymbol{g}}(\boldsymbol{R})$ depends on the location of interaction $\boldsymbol{R}$ within the crystal. The ML estimation problem is mathematically formalized as

$$\hat{\boldsymbol{R}}_{\text{MLE}} = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \big[L(\boldsymbol{R}_0; \boldsymbol{g})\big] = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \big[\text{pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)\big],$$

in which D denotes the crystal space and $\text{pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)$ is the probability density function of the measured data $\boldsymbol{g}$ conditioned on the assumption that the location of interaction was $\boldsymbol{R}_0$. Equivalently

$$\hat{\boldsymbol{R}}_{\text{MLE}} = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \big[\ell(\boldsymbol{R}_0; \boldsymbol{g})\big] = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \big[\ln \text{pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)\big]. \tag{4.3}$$

In the problem outlined above, $\boldsymbol{R}$ is the parameter we want to estimate and $\boldsymbol{g}$ is the observed data. Thus, our goal is to use $\boldsymbol{g}$ to find $\boldsymbol{R}$.

From our discussion on gamma-ray cameras, we know that the K PMT outputs $g_1, \ldots, g_K$ obey Poisson statistics and that they are statistically independent [127, 128]. Thus, we can make use of (4.2) to replace the probability density function $\text{pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)$ in (4.3) with a probability:

$$\text{Pr}(\boldsymbol{g} \mid \boldsymbol{R}_0) = \prod_{k=1}^{K} \frac{[\overline{g}_k(\boldsymbol{R}_0)]^{g_k}}{g_k!} e^{-\overline{g}_k(\boldsymbol{R}_0)}.$$

If we take the logarithm of $\text{Pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)$, we get

$$\ln \text{Pr}(\boldsymbol{g} \mid \boldsymbol{R}_0) = \sum_{k=1}^{K} \Big\{ g_k \ln [\overline{g}_k(\boldsymbol{R}_0)] - \ln(g_k!) - \overline{g}_k(\boldsymbol{R}_0) \Big\}.$$

Inserting this expression in (4.3) gives [45]:

$$\hat{\boldsymbol{R}}_{\text{MLE}} = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \left[\sum_{k=1}^{K} \Big\{ g_k \ln [\overline{g}_k(\boldsymbol{R}_0)] - \ln(g_k!) - \overline{g}_k(\boldsymbol{R}_0) \Big\} \right] =$$

$$= \arg\max_{\boldsymbol{R}_0 \in D} \left[\sum_{k=1}^{K} \left\{ g_k \ln \left[\overline{g}_k(\boldsymbol{R}_0) \right] - \overline{g}_k(\boldsymbol{R}_0) \right\} \right], \tag{4.4}$$

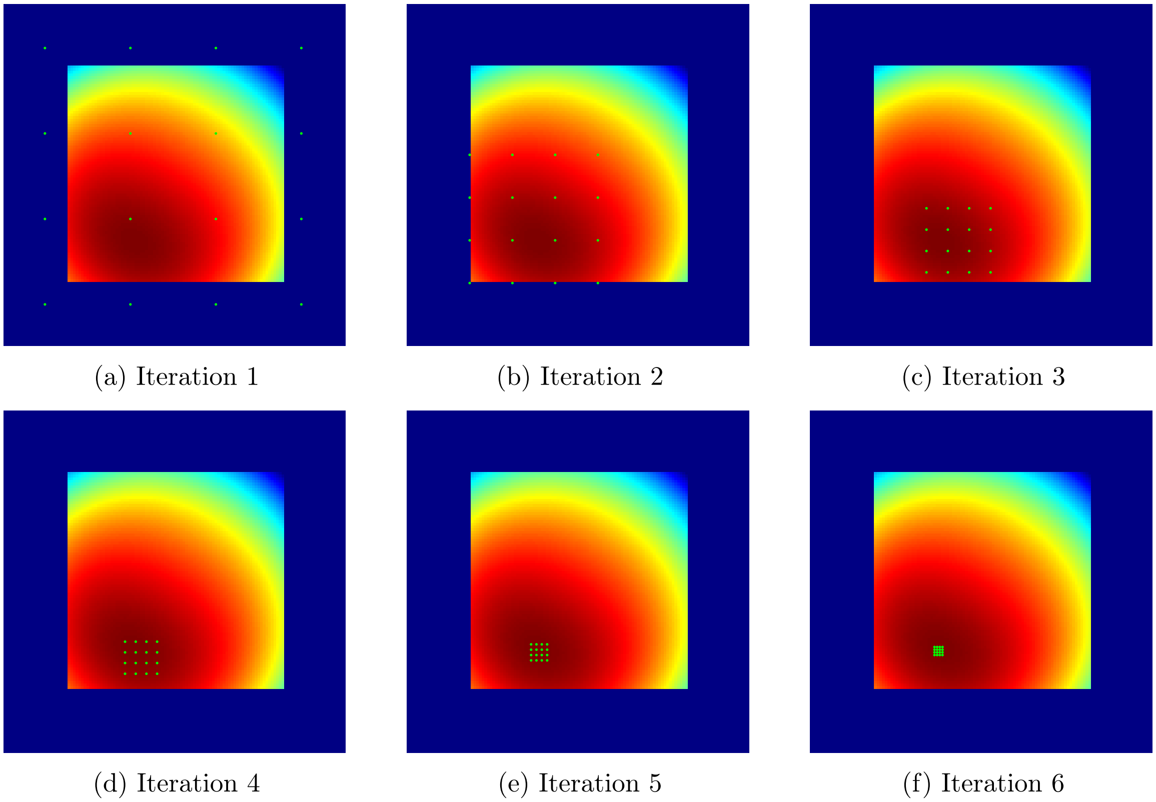
where the last form was obtained from the previous by discarding the $\ln(g_k!)$ term, which does not depend on $\boldsymbol{R}_0$.

In a practical case, we might have MDRF data $\overline{\boldsymbol{g}}(\boldsymbol{R}_0)$ only for a discrete set of points $\boldsymbol{R}_0$ in D. This would be the case in which MDRF data are obtained through simulation or experimentally measured [59, 60]. Typically, the detector D is partitioned into pixels (for a 2D estimation) or voxels (for a 3D estimation) and MDRF data $\overline{\boldsymbol{g}}(\boldsymbol{R}_0)$ are available for points $\boldsymbol{R}_0$ at which the pixels or voxels are centered.

As noted in [45], one possible way to solve the estimation problem formalized above, consists of using (4.4) to precompute $\hat{\boldsymbol{R}}_{\text{MLE}}$ for every possible value of $\boldsymbol{g}$, and use $\boldsymbol{g}$ as the address in a look-up table. However, the size of the look-up table grows exponentially with the number of bits needed to store the vector $\boldsymbol{g}$ in the memory of the computer. Even when the gamma-ray camera has only a few PMTs, this approach is not feasible [45].

Alternatively, we can take advantage of the fact that $\ln \text{pr}(\boldsymbol{g} \mid \boldsymbol{R}_0)$ is a smooth function of $\boldsymbol{R}_0$ for fixed $\boldsymbol{g}$, and consider an algorithm that performs multiple iterations to refine the estimate $\hat{\boldsymbol{R}}_{\text{MLE}}$. The algorithm we are going to use is the one we presented in § 3.3, but adapted for the particular problem at hand. Recall that the MDRF data $\overline{\boldsymbol{g}}(\boldsymbol{R}_0)$ is available only for a discrete set of points. This poses some limitations on the implementation of the algorithm. For example, we cannot perform an arbitrary number of iterations and make the final grid as small as we want: the separation between adjacent points in the grid cannot be smaller than one pixel/voxel. Similarly, we will assume that the separation between adjacent points is halved at each iteration of the algorithm. These assumptions are very convenient when $\overline{\boldsymbol{g}}(\boldsymbol{R}_0)$ is sampled on a pixel/voxel grid. Because the size of the grid and the number of iterations are fixed, this algorithm is well suited for hardware implementation [45]. Some robustness considerations are reported in [45] as well.

To clarify how the algorithm works, let us consider the example of a 2D estimation from PMT outputs. The steps of the algorithm are graphically show in Figure 4.8. The detector is divided into an array of 81×81 pixels. For this example, six iterations are enough to achieve convergence and to identify the pixel that attains the largest log-likelihood. For each pixel in the array, MDRF data $\overline{\boldsymbol{g}}(\boldsymbol{R}_0)$ are available [59, 60]. Thus, given noisy PMT outputs $\boldsymbol{g}$, we can calculate the log-likelihood $\ell(\boldsymbol{R}_0; \boldsymbol{g})$ for all the pixel locations $\boldsymbol{R}_0$.



(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 4

(e) Iteration 5

(f) Iteration 6

FIGURE 4.8. Illustration of the contracting-grid algorithm

Figure 4.8 shows plots of the log-likelihood data, and we marked the points of the grids with green dots. The green dots are in correspondence with the pixels at which the log-likelihood is considered during each iteration. Because we choose to halve the grid spacing at each iteration, the grid spacing will always be a power of 2. This means that, at the first iteration, some of the points of the grid might actually fall outside the detector. To cope with this problem, we zero-pad the log-likelihood data.

In § 5.5 and § 5.6 we will present two implementations of the algorithm for the cases of 2D and 3D estimation, and we will provide details for the implementation on two different parallel architectures.

4.4 The LMMLEM Algorithm

After mathematically formalizing the MLEM algorithm in § 3.4 and discussing list-mode data in § 4.2, we proceed by applying the theory of maximum-likelihood estimation to list-mode (LM) data to derive the list-mode MLEM algorithm (LMMLEM algorithm for short) [248, 258, 259]. Our derivation strictly follows the methods developed in § 3.4 and it is similar to the one of [258]. The main difference between our derivation and the one of [258] is that the former uses a different and—we believe—more intuitive definition of the complete data. Yet another different derivation is discussed in [260]. In it, the authors introduce a "complete data energy" objective function and alternatively perform coordinate descent with respect to the complete data and the estimate of the object. Event if their approach is different than the one of [258], it leads to the same final expressions.

The results derived below can be applied equally well to the case of SPECT or PET. In the former case, an entry $\hat{\boldsymbol{A}}^{(j)}$ in the list-mode data $\hat{\mathscr{A}}$ is an estimated 2D location of interaction within the scintillation crystal (we will assume a parallel-hole collimator), thus $\hat{\boldsymbol{A}}^{(j)} = \hat{\boldsymbol{R}}^{(j)}$, where the capital letter denotes a point in detector space. For PET, we will assume $\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)})$, in which the subscripts tell to which of the two detector spaces the point belongs to, and $\hat{\boldsymbol{R}}_1^{(j)}$ and $\hat{\boldsymbol{R}}_2^{(j)}$ are now 3D points in the scintillation crystals. For PET, it is customary to consider arrival times associated with points $\hat{\boldsymbol{R}}_1^{(j)}$ and $\hat{\boldsymbol{R}}_1^{(j)}$ and use these arrival times to perform coincidence time-windowing. However, for our case, we will assume that coincidence time-windowing has already been performed and that coincidental events in the detectors have already been paired.

As in § 3.4 we will assume that the object $f(\boldsymbol{r})$ (where $\boldsymbol{r}$ denotes a 3D spatial variable in the field of view) is discretized over a set of N voxels B_n to obtain a vector $\boldsymbol{f}$ with components

$$f_n = \int_{B_n} f(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r},$$

for $n = 1, \ldots, N$. It is worth recalling that the number f_n represents the mean number of counts emitted from voxel B_n per unit time. Thus, the units of f_n are s^{-1}. For each voxel B_n, we can also introduce the probability that a count emitted from that voxel gets actually detected by the hardware. We will denote such probability as s_n for $n = 1, \ldots, N$. The numbers s_n do not depend on the object $\boldsymbol{f}$ being imaged. On the contrary, they depend on the system's geometry and detectors. If τ denotes the total exposure time, then the mean number of events that are emitted from voxel B_n and get detected is given by $\lambda_n = \tau s_n f_n$. The actual number of events that are emitted from voxel B_n and get detected is a random variable, which we denote as c_n. Conditioned on $\boldsymbol{f}$, c_n follows Poisson statistics [127, 128] with mean $\lambda_n = \tau s_n f_n$. In symbols $c_n \sim \mathcal{P}(\tau s_n f_n)$ and, as we did for $\boldsymbol{f}$ and $\boldsymbol{s}$, we introduce the vector of counts $\boldsymbol{c}$ with components $c_1, \ldots, c_N$. A consequence of our model is that c_n and $c_{n'}$ are (conditionally) independent for $n \neq n'$.

As we pointed out in § 4.2, list-mode data are represented as a list

$$\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\}$$

of J random attribute vectors $\hat{\boldsymbol{A}}^{(j)}$ for $j = 1, \ldots, J$. Because the j^{th} collected event must have been generated from some voxel B_n in the field of view, we can introduce discrete random variables I_j for $j = 1, \ldots, J$ which tell from which voxel the j^{th} collected event was generated. In other words, I_j takes values in $\{1, \ldots, N\}$ and if $\hat{\boldsymbol{A}}^{(j)}$ was measured for an event that was generated from voxel B_n, then $I_j = n$. The probability law on I_j conditioned on $\boldsymbol{f}$ is given by [258, 261]

$$\Pr(I_j = n \mid \boldsymbol{f}) = \Pr(n \mid \boldsymbol{f}) = \frac{s_n f_n}{\sum_{n'=1}^{N} f_{n'} s_{n'}}, \tag{4.5}$$

in which the intermediate form has to be understood as the probability of an event occurring in voxel B_n, given the object $\boldsymbol{f}$. The fact that $\Pr(I_j = n \mid \boldsymbol{f})$ does not depend on j can be explained by noticing that the actual order of the elements of the list $\hat{\mathscr{A}}$ does not matter. Mathematically speaking, $\hat{\mathscr{A}}$ should be interpreted as a set rather than a list of attribute vectors. Given I_j for $j = 1, \ldots, J$, we can write any of the c_n as

$$c_n = \sum_{j=1}^{J} \delta_{\mathrm{Kron}}(n - I_j),$$

where $\delta_{\mathrm{Kron}}(k)$ is the function that was introduced in (3.8).

To apply the MLEM algorithm to the list-mode data $\hat{\mathscr{A}}$, notice first that, with the notation developed prematurely in § 3.4, the iterative algorithm takes the form

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ \langle \ln \mathrm{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) \rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(k)}} \right\},$$

where $\hat{\boldsymbol{\theta}}^{(k)}$ is the estimate of $\boldsymbol{\theta}$ at the k^{th} iteration, $\boldsymbol{x}$ is the vector of the complete data, and $\boldsymbol{y}$ is the vector of the incomplete (measured) data. Rewriting the algorithm above for list-mode data yields

$$\hat{\boldsymbol{f}}^{(k+1)} = \arg\max_{\boldsymbol{f}} \left\{ \langle \ln \mathrm{pr}(\boldsymbol{c}, \{\hat{\mathscr{A}}, \tau\} \mid \boldsymbol{f}) \rangle_{\boldsymbol{c} \mid \{\hat{\mathscr{A}}, \tau\}, \hat{\boldsymbol{f}}^{(k)}} \right\},$$

which was obtained by setting $\boldsymbol{x} = \boldsymbol{c}$, $\boldsymbol{y} = \{\hat{\mathscr{A}}, \tau\}$, and $\hat{\boldsymbol{\theta}}^{(k)} = \hat{\boldsymbol{f}}^{(k)}$. Our notation for the incomplete data implies that the exposure time τ is part of the measured data. That would be the case of, for example, collecting a specified number J of events. This is known in the literature as "preset-count" case [258]. The alternative to preset-count is "preset-time" [258], in which τ is a specified amount of time and J is a random number. In the present-count case, the exposure time τ is part of the measured data, hence the need to combine it with the list of events as in $\{\hat{\mathscr{A}}, \tau\}$.

By repeated application of Bayes' theorem and taking advantage of the independence (conditioned on $\boldsymbol{f}$) of $\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}$, we can write

$$\begin{aligned}\operatorname{pr}(\{\hat{\mathscr{A}}, \tau\}, \boldsymbol{c} \mid \boldsymbol{f}) = \operatorname{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) \operatorname{pr}(\hat{\mathscr{A}} \mid \boldsymbol{c}, \boldsymbol{f}) \Pr(\boldsymbol{c} \mid \boldsymbol{f}) = \\ = \operatorname{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) \left[\prod_{j=1}^{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{c}, \boldsymbol{f})\right] \Pr(\boldsymbol{c} \mid \boldsymbol{f}).\end{aligned}$$

The expression above can be made more useful by writing $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{c}, \boldsymbol{f})$ as a marginal density

$$\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{c}, \boldsymbol{f}) = \sum_{n=1}^{N} \frac{\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)}, n, \boldsymbol{c}, \boldsymbol{f})}{\operatorname{pr}(\boldsymbol{c}, \boldsymbol{f})} = \sum_{n=1}^{N} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \Pr(n \mid \boldsymbol{c}, \boldsymbol{f}),$$

where $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ is the probability density function for the attribute vector $\hat{\boldsymbol{A}}^{(j)}$ conditioned on the voxel B_n, and we used the fact that $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n, \boldsymbol{c}, \boldsymbol{f}) = \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$. Moreover, the last term in the expression above can be written as

$$\Pr(n \mid \boldsymbol{c}, \boldsymbol{f}) = \frac{c_n}{\sum_{n'=1}^{N} c_{n'}} = \frac{c_n}{J},$$

where the last form was obtained by noting that each count in any of the $c_1, \ldots, c_N$ corresponds to one and only one attribute vector in the list $\hat{\mathscr{A}}$, so that $\sum_{n'=1}^{N} c_{n'} = J$. If we use the result above, we get

$$\operatorname{pr}(\{\hat{\mathscr{A}}, \tau\}, \boldsymbol{c} \mid \boldsymbol{f}) = \operatorname{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) \prod_{j=1}^{J} \left[\sum_{n=1}^{N} \frac{c_n}{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\right] \Pr(\boldsymbol{c} \mid \boldsymbol{f}),$$

and its logarithm is

$$\ln \operatorname{pr}(\{\hat{\mathscr{A}}, \tau\}, \boldsymbol{c} \mid \boldsymbol{f}) = \ln \operatorname{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) + \sum_{j=1}^{J} \ln \left[\sum_{n=1}^{N} \frac{c_n}{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\right] + \ln \Pr(\boldsymbol{c} \mid \boldsymbol{f}). \quad (4.6)$$

To write an expression for $\operatorname{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f})$, recall first that, for the case of preset time, the random variables $c_1, \ldots, c_N$ are independent and follow Poisson statistics: $c_n \sim \mathcal{P}(\lambda_n)$, for $\lambda_n = \tau s_n f_n$. Thanks to their independence, the sum of the c_n's is also a Poisson random variable but with parameter $\tau \sum_{n=1}^{N} s_n f_n$. Alternatively [258], we can model the measurements of attribute vectors $\hat{\boldsymbol{A}}^{(j)}$ as a Poisson process with rate $\sum_{n=1}^{N} s_n f_n$. A fundamental result that holds for Poisson processes is that the waiting time between two consecutive events follows an exponential distribution with parameter equal to the rate of the Poisson process [127, 128]. Going back to the case of preset count, notice that the random variable τ now measures the waiting time needed to collect $\sum_{n=1}^{N} c_n = J$ measurements of the attribute vectors. A random variable defined in this way is said to follow an Erlang distribution [262, 263] whose density is

$$\begin{aligned}\mathrm{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) &= \mathrm{pr}\left(\tau \,\middle|\, \sum_{n=1}^{N} c_n, \boldsymbol{f}\right) = \\ &= \frac{1}{\tau \left(\sum_{n=1}^{N} c_n - 1\right)!} \left(\tau \sum_{n=1}^{N} s_n f_n\right)^{\sum_{n=1}^{N} c_n} \exp\left(-\tau \sum_{n=1}^{N} s_n f_n\right), \qquad (4.7)\end{aligned}$$

in which quantities $\sum_{n=1}^{N} s_n f_n$ and $\sum_{n=1}^{N} c_n$ are called "rate" and "shape," respectively. Although we will not use it, we notice that, as $\sum_{n=1}^{N} c_n \to \infty$, the probability density function $\mathrm{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f})$ is well approximated by the probability density function of a Gaussian random variable

$$\begin{aligned}&\mathrm{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) \approx \\ &\approx \frac{1}{\sqrt{2\pi \sum_{n=1}^{N} c_n \Big/ \left(\sum_{n=1}^{N} s_n f_n\right)^2}} \exp\left(-\frac{1}{2} \frac{\left(\tau - \sum_{n=1}^{N} c_n \Big/ \sum_{n=1}^{N} s_n f_n\right)^2}{\sum_{n=1}^{N} c_n \Big/ \left(\sum_{n=1}^{N} s_n f_n\right)^2}\right),\end{aligned}$$

and

$$\langle \tau \rangle = \frac{\sum_{n=1}^{N} c_n}{\sum_{n=1}^{N} s_n f_n}, \qquad \sigma_\tau^2 = \frac{\sum_{n=1}^{N} c_n}{\left(\sum_{n=1}^{N} s_n f_n\right)^2}.$$

If we take the logarithm of $\mathrm{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f})$ in (4.7), we get

$$\ln \mathrm{pr}(\tau \mid \boldsymbol{c}, \boldsymbol{f}) = \left(\sum_{n=1}^{N} c_n\right) \ln\left(\tau \sum_{n=1}^{N} s_n f_n\right) - \tau \sum_{n=1}^{N} s_n f_n - \ln\left[\tau \left(\sum_{n=1}^{N} c_n - 1\right)!\right].$$

Finding an expression for the probability $\Pr(\boldsymbol{c} \mid \boldsymbol{f})$ needed in (4.6) requires some reasoning. We first need to recall that the sum of the counts $c_1, \ldots, c_N$ must equal the number of attribute vectors in $\hat{\mathscr{A}}$: $\sum_{n=1}^{N} c_n = J$. Second, the probability that a collected event was generated in voxel B_n is $\Pr(n \mid \boldsymbol{f}) = s_n f_n / \sum_{n'=1}^{N} s_{n'} f_{n'}$ [258]. From this, it follows that $\boldsymbol{c}$ conditioned on the knowledge of $\boldsymbol{f}$ follows a multinomial distribution [127, 128]:

$$\Pr(\boldsymbol{c} \mid \boldsymbol{f}) = \left[\left(\sum_{n=1}^{N} c_n\right)!\right] \prod_{n=1}^{N} \frac{1}{c_n!} \left(\frac{s_n f_n}{\sum_{n'=1}^{N} s_{n'} f_{n'}}\right)^{c_n}.$$

With simple algebraic manipulations, we obtain

$$\begin{aligned} \ln \operatorname{pr}(\{\hat{\mathscr{A}}, \tau\}, \boldsymbol{c} \mid \boldsymbol{f}) = {} & \left(\sum_{n=1}^{N} c_n\right) \ln\left(\tau \sum_{n=1}^{N} s_n f_n\right) - \tau \sum_{n=1}^{N} s_n f_n + \\ & - \ln\left[\tau \left(\sum_{n=1}^{N} c_n - 1\right)!\right] + \sum_{j=1}^{J} \ln\left[\sum_{n=1}^{N} \frac{c_n}{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\right] + \\ & + \ln\left[\left(\sum_{n=1}^{N} c_n\right)!\right] + \sum_{n=1}^{N} \left[c_n \ln(s_n f_n) - c_n \ln\left(\sum_{n'=1}^{N} s_{n'} f_{n'}\right) - \ln(c_n!)\right]. \end{aligned}$$

If we take the expectation, we get

$$\begin{aligned} \langle \ln \operatorname{pr}(\{\hat{\mathscr{A}}, \tau\}, \boldsymbol{c} \mid \boldsymbol{f}) \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} = {} & \left(\sum_{n=1}^{N} \langle c_n \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}}\right) \ln\left(\tau \sum_{n=1}^{N} s_n f_n\right) - \tau \sum_{n=1}^{N} s_n f_n + \\ - \left\langle \ln\left[\tau \left(\sum_{n=1}^{N} c_n - 1\right)!\right]\right\rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} & + \sum_{j=1}^{J} \left\langle \ln\left[\sum_{n=1}^{N} \frac{c_n}{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\right]\right\rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} + \\ + \left\langle \ln\left[\left(\sum_{n=1}^{N} c_n\right)!\right]\right\rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} & + \sum_{n=1}^{N} \langle c_n \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} \ln(s_n f_n) + \\ - \sum_{n=1}^{N} \langle c_n \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} & \ln\left(\sum_{n'=1}^{N} s_{n'} f_{n'}\right) - \sum_{n=1}^{N} \langle \ln(c_n!) \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}}. \end{aligned}$$

When we maximize $\langle \ln \operatorname{pr}(\boldsymbol{c}, \{\hat{\mathscr{A}}, \tau\} \mid \boldsymbol{f}) \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}}$ to find $\hat{\boldsymbol{f}}^{(k+1)}$ from $\hat{\boldsymbol{f}}^{(k)}$, we can discard the terms that are constant with respect to $\boldsymbol{f}$. Thus we can write:

$$\hat{\boldsymbol{f}}^{(k+1)} = \arg\max_{\boldsymbol{f}} \left\{\sum_{n=1}^{N} \langle c_n \rangle_{\boldsymbol{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} \ln(s_n f_n) - \tau \sum_{n=1}^{N} s_n f_n\right\} =$$

$$= \arg\max_{\boldsymbol{f}} \Big\{ Q\big(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \{\hat{\mathscr{A}}, \tau\}\big) \Big\}, \tag{4.8}$$

where we defined

$$Q\big(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \{\hat{\mathscr{A}}, \tau\}\big) = \sum_{n=1}^{N} \langle c_n \rangle_{\mathbf{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} \ln(s_n f_n) - \tau \sum_{n=1}^{N} s_n f_n.$$

To find the vector $\hat{\boldsymbol{f}}^{(k+1)}$ that maximizes the quantity $Q\big(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \{\hat{\mathscr{A}}, \tau\}\big)$ in (4.8) above, we calculate the gradient of $Q\big(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \{\hat{\mathscr{A}}, \tau\}\big)$ with respect to $\boldsymbol{f}$, we set it to zero, and we solve the resulting equations for $\boldsymbol{f}$. The components of the gradient are

$$\frac{\partial Q\big(\boldsymbol{f}; \hat{\boldsymbol{f}}^{(k)}, \{\hat{\mathscr{A}}, \tau\}\big)}{\partial f_n} = \frac{\langle c_n \rangle_{\mathbf{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}}}{f_n} - \tau s_n. \tag{4.9}$$

Note that

$$\begin{aligned}\langle c_n \rangle_{\mathbf{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} &= \sum_{j=1}^{J} \langle \delta_{\text{Kron}}(n - I_j) \rangle_{c_n|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} = \sum_{j=1}^{J} \Pr(n \mid \hat{\boldsymbol{f}}^{(k)}, \hat{\boldsymbol{A}}^{(j)}) = \\ &= \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n, \hat{\boldsymbol{f}}^{(k)})\, \text{pr}(n, \hat{\boldsymbol{f}}^{(k)})}{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \hat{\boldsymbol{f}}^{(k)})\, \text{pr}(\hat{\boldsymbol{f}}^{(k)})} = \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\, \Pr(n \mid \hat{\boldsymbol{f}}^{(k)})}{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \hat{\boldsymbol{f}}^{(k)})}.\end{aligned}$$

Furthermore

$$\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \hat{\boldsymbol{f}}^{(k)}) = \sum_{n=1}^{N} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\, \Pr(n \mid \hat{\boldsymbol{f}}^{(k)}),$$

which, upon substitution, gives

$$\begin{aligned}\langle c_n \rangle_{\mathbf{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}} &= \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)\, \Pr(n \mid \hat{\boldsymbol{f}}^{(k)})}{\sum_{n'=1}^{N} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n')\, \Pr(n' \mid \hat{\boldsymbol{f}}^{(k)})} = \\ &= \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) s_n \hat{f}_n^{(k)} \Big/ \sum_{n''=1}^{N} s_{n''} \hat{f}_{n''}^{(k)}}{\sum_{n'=1}^{N} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)} \Big/ \sum_{n''=1}^{N} s_{n''} \hat{f}_{n''}^{(k)}} = \\ &= s_n \hat{f}_n^{(k)} \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)}}.\end{aligned}$$

When we set the partial derivatives in (4.9) to zero to find $\hat{f}_n^{(k+1)}$ and use the fact that $s_n \neq 0$, we get [258, 264–268]:

$$\hat{f}_n^{(k+1)} = \frac{\langle c_n \rangle_{\mathbf{c}|\{\hat{\mathscr{A}},\tau\},\hat{\boldsymbol{f}}^{(k)}}}{\tau s_n} = \hat{f}_n^{(k)} \left\{ \frac{1}{\tau} \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)}} \right\}. \tag{4.10}$$

The expression above represents one step of the LMMLEM algorithm. As for the case of (3.10), the LMMLEM algorithm (4.10) is a multiplicative algorithm, so it automatically enforces positivity constraints on the estimates $\hat{\boldsymbol{f}}^{(k)}$. Similarly, it preserves the quantity $\sum_{n=1}^{N} s_n \hat{f}_n^{(k)}$. Indeed:

$$\sum_{n=1}^{N} s_n \hat{f}_n^{(k+1)} = \frac{1}{\tau} \sum_{j=1}^{J} \frac{\sum_{n=1}^{N} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) s_n \hat{f}_n^{(k)}}{\sum_{n'=1}^{N} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)}} = \frac{J}{\tau} = \text{constant.} \tag{4.11}$$

Some convergence results for the LMMLEM algorithm are derived in [258]. For example, under broad conditions, the list-mode likelihood is a strictly concave function which ensures that estimates $\hat{\boldsymbol{f}}_n^{(1)}, \hat{\boldsymbol{f}}_n^{(2)}, \ldots$ obtained according to (4.10) converge to the global maximum of the list-mode likelihood function. Accuracy of reconstruction has also been studied [269]. As concluding remark, we note that (4.10) is just a particular instance of the MLEM method, so it enjoys all the properties of the MLEM algorithm we listed in § 3.2.

4.5 An Introduction to Markov Chain Monte Carlo

4.5.1 Markov Chains

A *Markov chain* [270–273] is a discrete-time stochastic process $X_0, X_1, \ldots$ that satisfies the *Markov property*:

$$\Pr(X_n \in A \mid X_0, X_1, \ldots, X_{n-1}) = \Pr(X_n \in A \mid X_{n-1}),$$

where A is a subset of the state space S (the set of values each random variable X_n can take on). The Markov property states that the distribution of the random variable X_n given all previous ones, only depends on X_{n-1}. For simplicity, we will start our discussion with the case of discrete state spaces S; the extension to the general case of continuous state spaces requires a different notation [273, 274] but all the concepts developed for the case of discrete state spaces carry over to the case of continuous state spaces, as we will discuss at the end of this section.

Without loss of generality, we can identify all the elements of the state space with positive integer numbers, so $S = \{1, 2, 3, \dots\}$. A Markov chain is said to be a *time-homogeneous* (or *stationary*) Markov chain if $\Pr(X_n = j \mid X_{n-1} = i) = \Pr(X_{n+1} = j \mid X_n = i)$ for all n and all i and j in S. In other words, in a time-homogeneous Markov chain, the probability of transition from a state i to a state j does not depends on time n and we can simply use the notation $p_{i\to j}$ to mean $\Pr(X_n = j \mid X_{n-1} = i)$. If S is finite and the chain is time-homogeneous, we can graphically represent a Markov chain with a directed graph in which nodes correspond to states in S and an arc from node i to node j means $p_{i\to j} > 0$. The arc is further labeled with this probability.

A Markov chain is called *irreducible* if, for all i and j in S, there exists $n > 0$ such that $\Pr(X_n = j \mid X_0 = i) > 0$. In other words, a Markov chain is irreducible if it is possible to reach any state j from any state i in any finite number of steps. Figure 4.9a shows an example of a Markov chain that is not irreducible (from state 2, it is not possible to reach state 1, no matter how many steps we take). On the other hand, the Markov chain in Figure 4.9b is irreducible. Furthermore, an irreducible Markov chain is said to be *recurrent* if:

$$\sum_{n=0}^{\infty} \Pr(X_n = j \mid X_0 = i) = \infty.$$

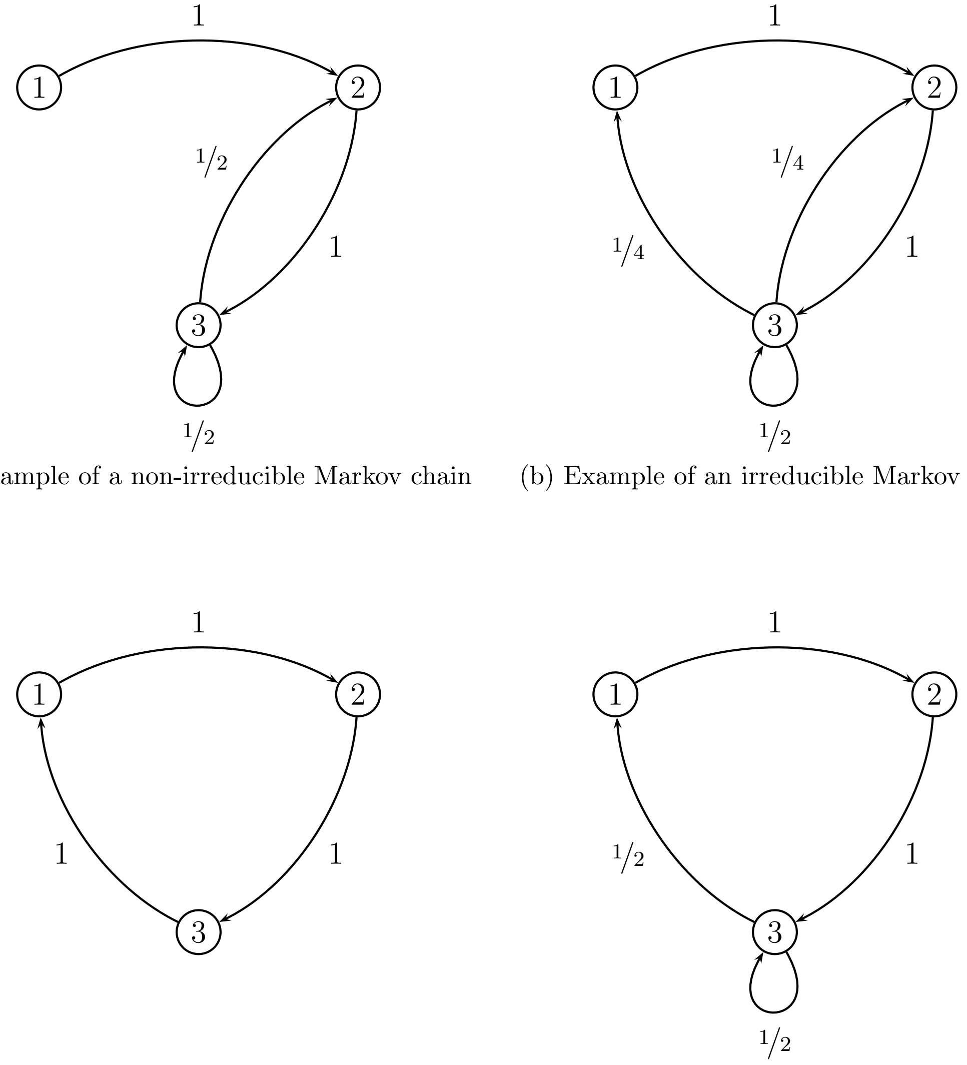
Given a recurrent irreducible Markov chain, we say that it is *positive recurrent* if it admits a *stationary probability distribution* $\boldsymbol{\pi} = \{\pi_i \geqslant 0, i \in S\}$:

$$\sum_{j\in S} \pi_j \Pr(X_n = j \mid X_0 = i) = \pi_i,$$

for all $i \in S$ and $n \geqslant 0$. Finally, an irreducible Markov chain is said to be *aperiodic* if, for all $i \in S$,

$$\gcd\{n > 0 \text{ such that } \Pr(X_n = i \mid X_0 = i) > 0\} = 1,$$

where $\gcd\{\dots\}$ denotes the greatest common divisor of a set of numbers. As an example, the Markov chain shown in Figure 4.9c is periodic because, starting from a



(a) Example of a non-irreducible Markov chain

(b) Example of an irreducible Markov chain

(c) Example of a periodic Markov chain

(d) Example of an aperiodic Markov chain

FIGURE 4.9. Some examples of time-homogeneous Markov chains

state, we can only return to the same state by performing a number of steps that is a multiple of 3. On the other hand, the Markov chain shown in Figure 4.9d is aperiodic.

For a time-homogeneous Markov chain, the Markov property and repeated applications of Bayes' theorem give

$$\begin{aligned}\Pr(X_2 = j \mid X_0 = i) &= \sum_{k\in S} \Pr(X_2 = j \mid X_1 = k) \Pr(X_1 = k \mid X_0 = i) = \\ &= \sum_{k\in S} p_{k\to j} p_{i\to k} = \sum_{k\in S} p_{i\to k} p_{k\to j} = \left[\mathbf{P}^2\right]_{i,j}.\end{aligned}$$

where we have recognized the expression for the $(i, j)^{\text{th}}$ element of the square of the transition matrix $\mathbf{P}$, where $[\mathbf{P}]_{i,j} = p_{i\to j}$. Generalizing:

$$\Pr(X_n = j \mid X_0 = i) = \left[\mathbf{P}^n\right]_{i,j}.$$

Given an irreducible, positive recurrent, aperiodic, time-homogeneous Markov chain, we can prove that the stationary distribution is unique and satisfy

$$\sum_{i\in S} \pi_i = 1, \qquad \boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}.$$

The reason why $\boldsymbol{\pi}$ is called the "stationary" distribution has now become clear: once the Markov chain reaches it, this distribution does not change as the chain evolves over time. The system of linear equations above can be solved for $\boldsymbol{\pi}$ to find the stationary distribution given the transition matrix $\mathbf{P}$. A stronger condition on $\boldsymbol{\pi}$ which automatically ensures its stationarity is the *detailed balance* condition:

$$\pi_i p_{i\to j} = \pi_j p_{j\to i}. \tag{4.12}$$

Indeed

$$\pi_i = \pi_i \underbrace{\sum_{j\in S} p_{i\to j}}_{=1} = \sum_{j\in S} \pi_i p_{i\to j} = \sum_{j\in S} \pi_j p_{j\to i},$$

which, for all $i \in S$, gives $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$.

All the expressions given here for the case of a discrete state space S easily generalize to the of a continuous state space. For example, instead of a stationary probability distribution $\pi_i \geqslant 0$ indexed by $i \in S$, we will have a stationary probability density function $\pi(\boldsymbol{\theta}) \geqslant 0$, where $\boldsymbol{\theta} \in S$. Similarly, the conditions given above for a stationary distribution become

$$\int_S \pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = 1, \qquad \pi(\boldsymbol{\theta}) = \int_S \pi(\boldsymbol{\theta}')[\mathcal{P}](\boldsymbol{\theta}', \boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}',$$

where $[\mathcal{P}](\boldsymbol{\theta}', \boldsymbol{\theta}) = \mathrm{pr}(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = p_{\boldsymbol{\theta}' \to \boldsymbol{\theta}}$ is the integral kernel of the operator $\mathcal{P}$. For the continuous state space, the operator $\mathcal{P}$ plays the same role as the matrix $\mathbf{P}$ in the discrete state space.

4.5.2 Markov-Chain Monte Carlo Integration

In a nutshell, Markov-chain Monte Carlo (MCMC) integration can be defined as Monte Carlo calculation of integrals of the form:

$$\langle f(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}} = \int_S f(\boldsymbol{\theta})\mathrm{pr}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}$$

via Monte Carlo methods

$$\langle f(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}} \approx \frac{1}{N} \sum_{n=1}^{N} f\big(\boldsymbol{\theta}^{(n)}\big), \tag{4.13}$$

where samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$ are generated using a Markov chain whose stationary distribution is $\pi(\boldsymbol{\theta}) = \mathrm{pr}(\boldsymbol{\theta})$. We refer to N as the length of the chain. The necessity of Markov chain Monte Carlo integration arises every time we have to calculate an expectation involving a high-dimensional complicated probability density function or when the expectation does not admit an analytic closed form.

In the previous section, we have considered the problem of finding the stationary probability distribution (or the stationary probability density function) given a Markov chain. Now, the problem consists in doing the opposite: given a desired probability density function, we want to construct a Markov chain that has such probability density function as its (unique) stationary probability density function.

We will present here the algorithm proposed in [168, 275], which is known in the literature as the *Metropolis-Hastings algorithm* [276]. This iterative algorithm starts from some initial state $\boldsymbol{\theta}^{(n-1)}$ for $n = 1$ and produces a *candidate* state $\boldsymbol{\theta}' \in S$ by drawing a sample according to a *proposal* distribution $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$. The candidate state $\boldsymbol{\theta}'$ is accepted with probability

$$P_{\text{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(n-1)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})}\right\}. \tag{4.14}$$

Given $P_{\text{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$, the standard way to accept $\boldsymbol{\theta}'$ with such probability is to generate a realization u of the uniformly distributed random variable $U \sim \mathcal{U}([0,1])$ and to compare $P_{\text{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$ with u: if $u < P_{\text{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$, then $\boldsymbol{\theta}'$ is accepted and we set $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}'$. If $u \geqslant P_{\text{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$, the candidate is rejected and we set $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}$. The value of n is subsequently incremented and the algorithm repeats, as shown in Figure 4.10. Pseudocode of the Metropolis-Hastings algorithm is reported below [4]:

```
Initialize $\boldsymbol{\theta}^{(0)}$
$n = 0$
loop
  Sample $\boldsymbol{\theta}'$ from $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$
  $P_{\text{acc}} = \min\{1, \pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}')/\pi(\boldsymbol{\theta}^{(n-1)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})\}$
  Sample $u$ from $U \sim \mathcal{U}([0,1])$
  if $u < P_{\text{acc}}$ then
    $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}'$
  else
    $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}$
  end if
  $n = n + 1$
end loop
```

In this way, a sequence $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ is produced, and such a sequence of samples is used in (4.13) to calculate $\langle f(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}}$. The attentive reader must have surely noticed that the procedure of Figure 4.10 does not generate samples that are statistically

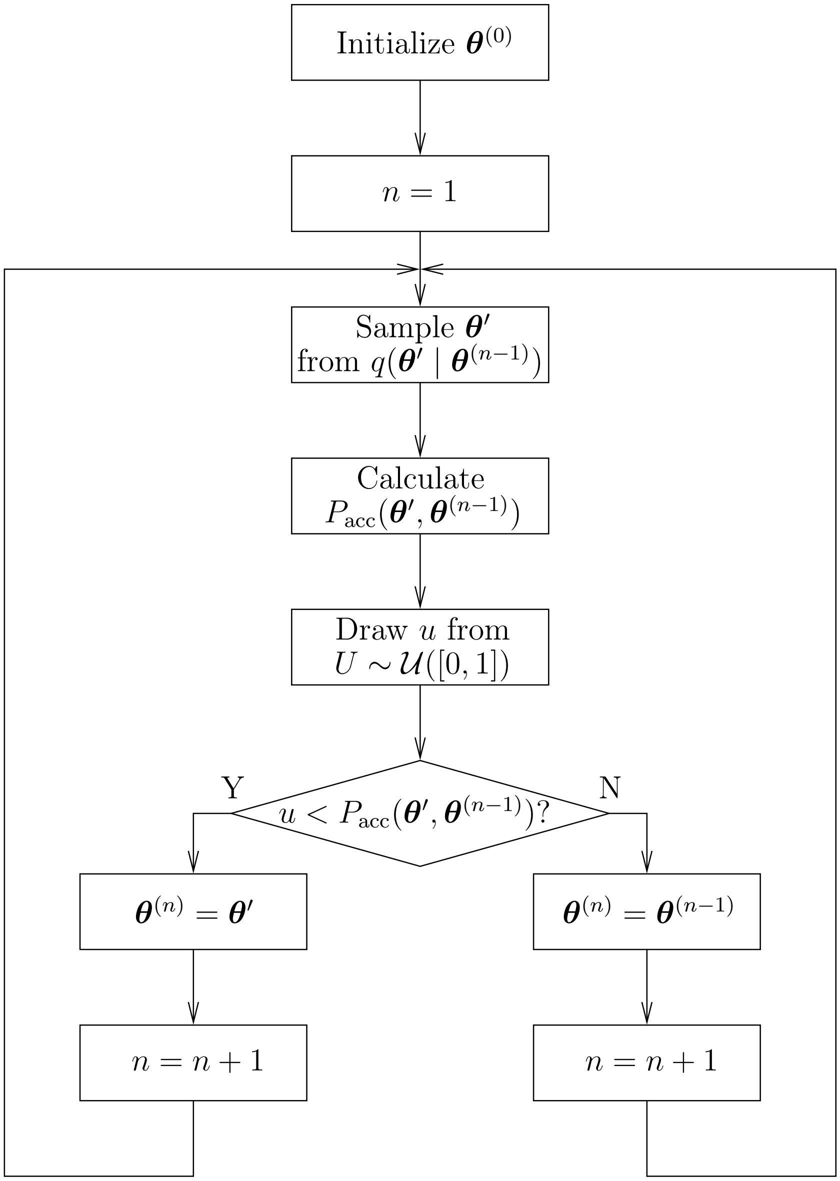

Initialize $\boldsymbol{\theta}^{(0)}$
$n = 1$
Sample $\boldsymbol{\theta}'$ from $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$
Calculate $P_{\mathrm{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$
Draw u from $U \sim \mathcal{U}([0,1])$
$u < P_{\mathrm{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)})$?
Y
N
$\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}'$
$\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}$
$n = n + 1$
$n = n + 1$

Figure 4.10. Flux diagram of the Metropolis-Hastings algorithm

independent. On the contrary, and because we are defining a Markov chain, the statistics of $\boldsymbol{\theta}^{(n)}$ depend on $\boldsymbol{\theta}^{(n-1)}$. However, independence of the samples in the $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ sequence is not a requirement; all we need is a method to draw samples throughout the support of $\mathrm{pr}(\boldsymbol{\theta})$ in the correct proportions.

To show that the Markov chain just defined has $\pi(\boldsymbol{\theta}) = \mathrm{pr}(\boldsymbol{\theta})$ as stationary distribution, we start by calculating the probability density function $\mathrm{pr}(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)})$:

$$\begin{aligned}\mathrm{pr}(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}) = p_{\boldsymbol{\theta}^{(n-1)} \to \boldsymbol{\theta}^{(n)}} = q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}) P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n-1)}) + \\ + I\left(\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}\right) \left[1 - \int_S q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) P_{\mathrm{acc}}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n-1)})\, \mathrm{d}\boldsymbol{\theta}\right], \quad (4.15)\end{aligned}$$

where

$$I\left(\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}\right) = \begin{cases} 1 & \text{if } \boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}, \\ 0 & \text{otherwise.} \end{cases}$$

To continue the proof, we need to distinguish two cases. If $P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n-1)}) = 1$, then

$$\pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)}) \geqslant \pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}),$$

which implies

$$P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n-1)}, \boldsymbol{\theta}^{(n)}) = \frac{\pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)})}{\pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)})} \leqslant 1.$$

Thus

$$\pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}) P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n-1)}) = \pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)}) P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n-1)}, \boldsymbol{\theta}^{(n)}).$$

Vice-versa, if

$$P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n-1)}) = \frac{\pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)})}{\pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)})}$$

then

$$\pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)}) \leqslant \pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}),$$

which leads to $P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n-1)}, \boldsymbol{\theta}^{(n)}) = 1$. As before:

$$\pi(\boldsymbol{\theta}^{(n-1)}) q(\boldsymbol{\theta}^{(n)} \mid \boldsymbol{\theta}^{(n-1)}) P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n-1)}) = \pi(\boldsymbol{\theta}^{(n)}) q(\boldsymbol{\theta}^{(n-1)} \mid \boldsymbol{\theta}^{(n)}) P_{\mathrm{acc}}(\boldsymbol{\theta}^{(n-1)}, \boldsymbol{\theta}^{(n)}).$$

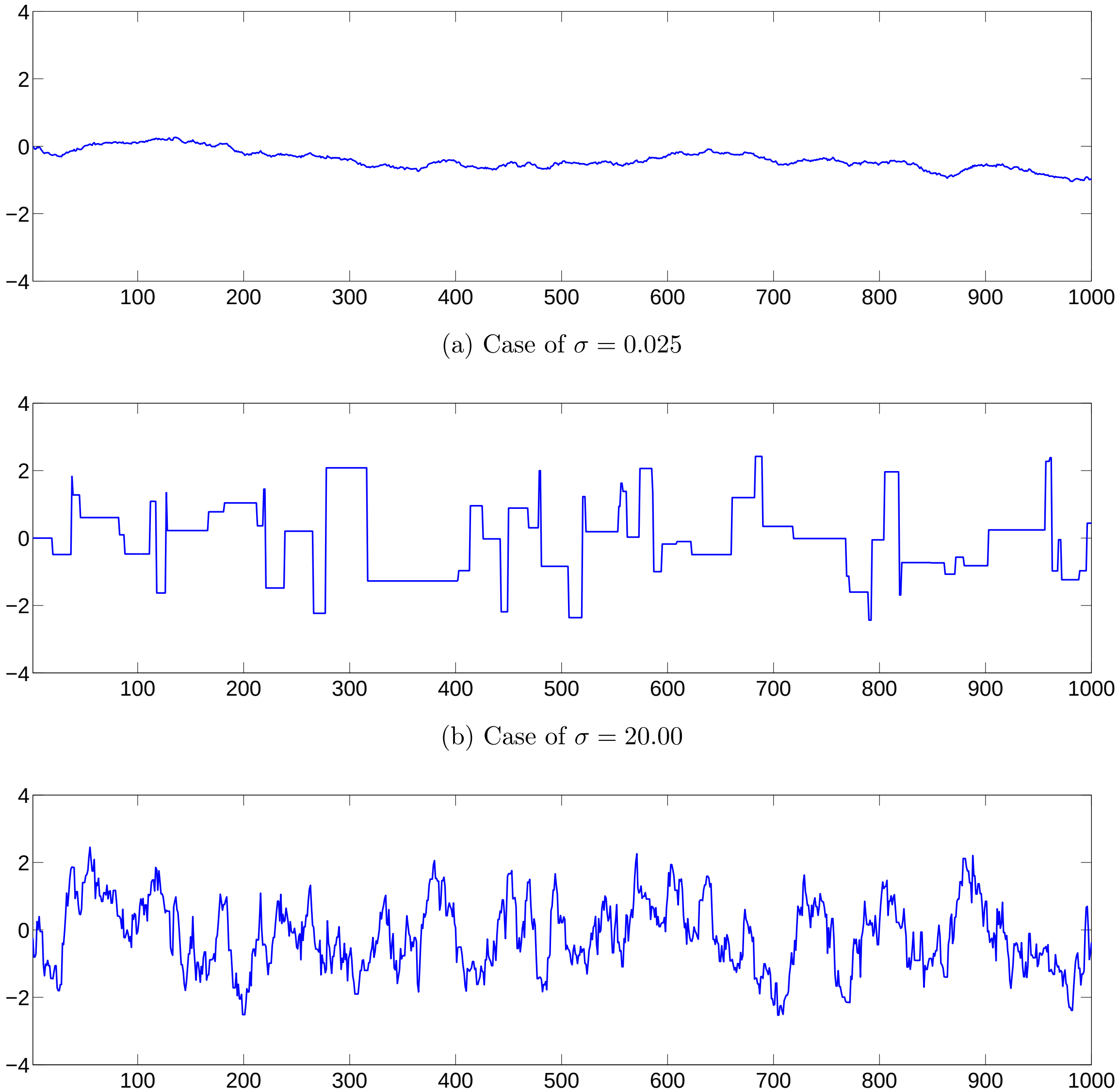
If we multiply both sides of (4.15) by $\pi(\boldsymbol{\theta}^{(n-1)})$ and use the result above, we get

$$\pi(\boldsymbol{\theta}^{(n-1)})p_{\boldsymbol{\theta}^{(n-1)}\to\boldsymbol{\theta}^{(n)}} = \pi(\boldsymbol{\theta}^{(n)})p_{\boldsymbol{\theta}^{(n)}\to\boldsymbol{\theta}^{(n-1)}},$$

which is the continuous equivalent of the detailed balance condition of (4.12). Because $\pi(\boldsymbol{\theta})$ satisfies the detailed balance condition, $\pi(\boldsymbol{\theta})$ is the stationary distribution of the Markov chain. One point to note is that we did not make any assumption on the form of the proposal distribution $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$. From a theoretical point of view, it can be chosen freely, and the stationary distribution will still be the same. In general, a density $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ from which it is easy to draw samples is desirable. Furthermore, we can restrict our attention to densities that satisfy $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)}) = q(\boldsymbol{\theta}'^{(n-1)} \mid \boldsymbol{\theta}')$, so that

$$P_{\mathrm{acc}}(\boldsymbol{\theta}', \boldsymbol{\theta}^{(n-1)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(n-1)})}\right\}.$$

The choice of $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ is critical to ensure fast convergence of the sum in (4.13) to the quantity to estimate. Practical aspects are discussed, for example, in [277,278]. Instead of providing a formal mathematical treatment of the convergence properties of Markov chain Monte Carlo integration, we will use some examples to discuss what a "good" proposal distribution $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ should look like. Intuitively, the spread of $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ should neither be too small, nor too large. Indeed, if $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ is narrow, large jumps of $\boldsymbol{\theta}'$ will have low probability and the Markov chain will not be able to sample the whole support of $\pi(\boldsymbol{\theta})$ in a convenient number of steps (see Figure 4.11a). On the other hand, if $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ is broad, very large jumps will be allowed, leading to small acceptance probabilities for the samples $\boldsymbol{\theta}'$. Small acceptance probabilities show up as horizontal segments in Figure 4.11b. A good choice of $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(n-1)})$ may be one that gives plots similar to the one of Figure 4.11c. The plots in Figure 4.11 also lend themselves to discussing *burn-in* time in the context of Markov chain Monte Carlo integration. In order to obtain samples distributed according to the target distribution $\pi(\boldsymbol{\theta})$ and that do not depend on the choice of $\boldsymbol{\theta}^{(0)}$ (see



(a) Case of $\sigma = 0.025$

(b) Case of $\sigma = 20.00$

(c) Case of $\sigma = 0.50$

FIGURE 4.11. Plots of 1000 samples obtained via Markov chain Monte Carlo. For all cases, the stationary distribution $\pi(\theta)$ is the probability density function of a univariate normally-distributed random variable with mean 0 and standard deviation 1. The proposal distribution $q(\theta' \mid \theta^{(n-1)})$ was chosen to be the probability density function of a univariate normally-distributed random variable with mean $\mu = \theta^{(n-1)}$ and standard deviation σ, for three different values of σ (adapted from [4])

Figure 4.10), some of the first samples might need be discarded and excluded from the sum in (4.13).

4.6 Task Performance with List-Mode Data

In this section, we analyze one of the main topics of the dissertation, namely how the framework of list-mode data is used to study task performance. In some cases and for a simple imaging system, analytical results are possible. Whenever analytical results cannot be found, we turned to simulation codes and Markov chain Monte Carlo integration, which were briefly introduced in the previous section. In all the cases analyzed below, we will assume that the signal we want to detect is known in location and shape. We will refer to this hypothesis as "signal known exactly," or SKE for short. The background (defined as everything but the signal to detect) will either be assumed known exactly ("background known exactly," or BKE) or random, but its statistical characterization known ("background known statistically," or BKS).

4.6.1 Information Content of a Photon: SKE/BKE Case

In this section, we turn our attention to an objective way to measure how much "information" a detected photon conveys for the case of the estimation of a generic $P \times 1$ parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)^\mathsf{T}$ from noisy list-mode data $\hat{\mathscr{A}}$. We have said numerous times that, because of measurement noise, the data $\hat{\mathscr{A}}$ that are used to perform an estimation are random, and so is any non-constant quantity deterministically calculated from $\hat{\mathscr{A}}$. Generally speaking, the performance of an estimator can be quantified by calculating bias and variance (or covariance matrix) of the estimates $\hat{\boldsymbol{\theta}}$ it produces. The bias at $\boldsymbol{\theta}$ for the random vector $\hat{\boldsymbol{\theta}}$ is written as

$$\boldsymbol{b}(\boldsymbol{\theta}) = \left\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \right\rangle_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} = \left\langle \hat{\boldsymbol{\theta}} \right\rangle_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} - \boldsymbol{\theta}.$$

An estimator for $\boldsymbol{\theta}$ is said to be *unbiased* if, for such estimator, $\boldsymbol{b}(\boldsymbol{\theta}) = \mathbf{0}$ for all $\boldsymbol{\theta}$. More important, however, is knowledge of the $P \times P$ covariance matrix of the estimates $\hat{\boldsymbol{\theta}}$ at $\boldsymbol{\theta}$:

$$\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} = \left\langle \left[\hat{\boldsymbol{\theta}} - \langle\hat{\boldsymbol{\theta}}\rangle_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}}\right] \left[\hat{\boldsymbol{\theta}} - \langle\hat{\boldsymbol{\theta}}\rangle_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}}\right]^{\mathsf{T}} \right\rangle_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}}.$$

The concept of "information" contained in the list $\hat{\mathscr{A}}$ is numerically quantified via the Fisher information matrix (FIM) $\mathbf{F}_{\boldsymbol{\theta}}$ [69, 279, 280] which, for an unbiased estimator, assumes the form

$$\begin{aligned}[\mathbf{F}_{\boldsymbol{\theta}}]_{p,q} &= \left\langle \left[\frac{\partial}{\partial\theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\right] \left[\frac{\partial}{\partial\theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\right] \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_{\boldsymbol{\theta}},\tau} = \\ &= \int_{\infty} \left[\frac{\partial}{\partial\theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\right] \left[\frac{\partial}{\partial\theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\right] \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\, \mathrm{d}\hat{\mathscr{A}}, \quad (4.16)\end{aligned}$$

in which $\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)$ denotes the likelihood of $\hat{\mathscr{A}}$, and we have explicitly included the dependence of the object $\boldsymbol{f}_{\boldsymbol{\theta}}$ on the parameter vector $\boldsymbol{\theta}$. In this section, we will assume that the vector $\boldsymbol{\theta}$ fully characterizes the object $\boldsymbol{f}_{\boldsymbol{\theta}}$. To clarify any possible confusion that might arise from the expression above, with $\frac{\partial}{\partial\theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)$ we mean the partial derivative of $\ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)$ with respect to the p^{th} component of $\boldsymbol{\theta}$. The resulting function is then evaluated at $\boldsymbol{\theta}$, giving an expression that depends on the random list $\hat{\mathscr{A}}$ only. Finally, the quantity in $\langle\ldots\rangle$ above is averaged over $\hat{\mathscr{A}}$, statistically conditioned on $\boldsymbol{f}_{\boldsymbol{\theta}}$ and τ. The Fisher information matrix $\mathbf{F}_{\boldsymbol{\theta}}$ is also the covariance matrix of $\nabla_{\boldsymbol{\theta}} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)$, which is often called *score*, provided that we can show that the mean of the score is zero:

$$\begin{aligned}\left\langle \frac{\partial}{\partial\theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_{\boldsymbol{\theta}},\tau} &= \int_{\infty} \frac{\frac{\partial}{\partial\theta_p}\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)}{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)} \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\, \mathrm{d}\hat{\mathscr{A}} = \\ &= \frac{\partial}{\partial\theta_p} \underbrace{\int_{\infty} \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau)\, \mathrm{d}\hat{\mathscr{A}}}_{=1} = 0.\end{aligned}$$

The result above also allows us to calculate the Fisher information matrix in a different way. Indeed, from

$$\frac{\partial}{\partial\theta_p} \left\langle \frac{\partial}{\partial\theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_{\boldsymbol{\theta}}, \tau) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_{\boldsymbol{\theta}},\tau} = 0,$$

we obtain [280, 281]

$$\begin{aligned}
0 &= \frac{\partial}{\partial \theta_p} \int_\infty \left[\frac{\partial}{\partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \, \mathrm{d}\hat{\mathscr{A}} = \\
&= \int_\infty \left[\frac{\partial^2}{\partial \theta_p \partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \, \mathrm{d}\hat{\mathscr{A}} + \\
&\quad + \int_\infty \left[\frac{\partial}{\partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \left[\frac{\partial}{\partial \theta_p} \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \mathrm{d}\hat{\mathscr{A}} = \\
&= \int_\infty \left[\frac{\partial^2}{\partial \theta_p \partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \, \mathrm{d}\hat{\mathscr{A}} + \\
&\quad + \int_\infty \left[\frac{\partial}{\partial \theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \left[\frac{\partial}{\partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right] \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \, \mathrm{d}\hat{\mathscr{A}}.
\end{aligned}$$

Upon rearrangement and using (4.16), we get

$$[\mathbf{F}_{\boldsymbol{\theta}}]_{p,q} = - \left\langle \frac{\partial^2}{\partial \theta_p \partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) \right\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau}.$$

For an unbiased estimator and if $\mathbf{F}_{\boldsymbol{\theta}}$ is non-singular, the relationship between the covariance matrix $\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}}$ and the Fisher information matrix $\mathbf{F}_{\boldsymbol{\theta}}$ can be summarized as

$$\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} \geqslant \mathbf{F}_{\boldsymbol{\theta}}^{-1},$$

which has to be understood as an alternative way to say that $\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} - \mathbf{F}_{\boldsymbol{\theta}}^{-1}$ is a positive-semidefinite matrix. A consequence of $\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}} - \mathbf{F}_{\boldsymbol{\theta}}^{-1}$ being positive-semidefinite is that the variances $\sigma^2_{\hat{\theta}_p}$ of the components of any unbiased estimator of $\boldsymbol{\theta}$ satisfy [114, 282]

$$\left[\mathbf{K}_{\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}}\right]_{p,p} = \sigma^2_{\hat{\theta}_p} \geqslant \left[\mathbf{F}_{\boldsymbol{\theta}}^{-1}\right]_{p,p}.$$

The inequality above, which is known in the literature as the *Cramér-Rao lower bound* [144, 145], provides an upper bound on the performance of any estimator, in the sense that no unbiased estimator can produce estimates with a variance lower than $\left[\mathbf{F}_{\boldsymbol{\theta}}^{-1}\right]_{p,p}$. Hence, the Fisher information matrix quantifies how statistically accurate an estimate of $\boldsymbol{\theta}$ can be.

If we want to calculate the Fisher information matrix for the estimation of $\boldsymbol{\theta}$ from the list-mode data $\hat{\mathscr{A}}$, we first need an expression for the likelihood [5]:

$$\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) = \mathrm{Pr}(J \mid \boldsymbol{f_\theta}, \tau) \prod_{j=1}^{J} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) =$$

$$= \frac{\tau^J}{J!} \exp\left(-\tau \int_{\mathrm{FOV}} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right) \prod_{j=1}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r},$$

in which we have used the fact that J conditioned on $\boldsymbol{f_\theta}$ and τ is a Poisson random variable and the following results:

$$\overline{J}(\boldsymbol{f_\theta}, \tau) = \tau \int_{\mathrm{FOV}} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}, \tag{4.17}$$

and

$$\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) = \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}.$$

Taking the derivative of the logarithm of $\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau)$ gives

$$\begin{aligned} \frac{\partial}{\partial \theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau) = -\tau \int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \\ + \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}. \end{aligned}$$

From the fact that $\left\langle \frac{\partial}{\partial \theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau)\right\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau} = 0$ and using the result above, we immediately have

$$\left\langle \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau} = \tau \int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}. \tag{4.18}$$

Because the entries in $\hat{\mathscr{A}}$ are independent and, on average, there are $\overline{J}(\boldsymbol{f_\theta}, \tau)$ of them in $\hat{\mathscr{A}}$, we can also write

$$\overline{J}(\boldsymbol{f_\theta}, \tau) \left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}} = \frac{\partial}{\partial \theta_p} \overline{J}(\boldsymbol{f_\theta}, \tau).$$

We now have all the ingredients to calculate the Fisher information matrix corresponding to the estimation of the parameter $\boldsymbol{\theta}$ given the list $\hat{\mathscr{A}}$:

$$\begin{aligned} [\mathbf{F}_{\boldsymbol{\theta}}]_{p,q} &= \left\langle \left[\frac{\partial}{\partial \theta_p} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau)\right] \left[\frac{\partial}{\partial \theta_q} \ln \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau)\right] \right\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau} = \\ &= \left\langle \left[\sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} - \tau \int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \right] \times \right. \end{aligned}$$

$$\times \left[\sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} - \tau \int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right]\Bigg\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau} =$$

$$= \Bigg\langle \left[\sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}\right] \times$$

$$\times \left[\sum_{j'=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}\right]\Bigg\rangle_{\hat{\mathscr{A}} \mid \boldsymbol{f_\theta}, \tau} +$$

$$- \tau^2 \left[\int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right] \left[\int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right],$$

in which the last form follows from (4.18). By carrying out the calculations, we can easily recognize that the product of the two summations above gives rise to $J^2 - J$ terms in which $j \neq j'$ and J terms for the case $j = j'$. Furthermore, we can make use of the independence assumption of the elements in $\hat{\mathscr{A}}$ to write

$$[\mathbf{F}_{\boldsymbol{\theta}}]_{p,q} = \langle J^2 - J \rangle_{J \mid \boldsymbol{f_\theta}, \tau} \times$$

$$\times \left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}} \left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}} +$$

$$+ \langle J \rangle_{J \mid \boldsymbol{f_\theta}, \tau} \left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \cdot \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}} +$$

$$- \tau^2 \left[\int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right] \left[\int_{\mathrm{FOV}} \frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right].$$

If we use the properties [127, 128] of the Poisson random variable J (conditioned on $\boldsymbol{f_\theta}$ and τ) we can easily show

$$\langle J^2 - J \rangle_{J \mid \boldsymbol{f_\theta}} = \sigma^2_{J \mid \boldsymbol{f_\theta}} + \langle J \rangle^2_{J \mid \boldsymbol{f_\theta}} - \langle J \rangle_{J \mid \boldsymbol{f_\theta}} =$$
$$= \langle J \rangle_{J \mid \boldsymbol{f_\theta}} + \langle J \rangle^2_{J \mid \boldsymbol{f_\theta}} - \langle J \rangle_{J \mid \boldsymbol{f_\theta}} = \left[\overline{J}(\boldsymbol{f_\theta}, \tau)\right]^2.$$

When we use the identity above, many cancellations occur, and the expression for $[\mathbf{F}_{\boldsymbol{\theta}}]_{p,q}$ reduces to

$$[\mathbf{F}_{\boldsymbol{\theta}}]_{p,q} = \tau \left[\int_{\mathrm{FOV}} f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}\right] \times$$

$$\times \left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \big[\frac{\partial}{\partial \theta_p} f_{\boldsymbol{\theta}}(\boldsymbol{r})\big] s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \cdot \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) \big[\frac{\partial}{\partial \theta_q} f_{\boldsymbol{\theta}}(\boldsymbol{r})\big] s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}} \mid \boldsymbol{f_\theta}},$$

in which we have used (4.17). The expression above shows that $\mathbf{F}_{\boldsymbol{\theta}}$ is proportional to the exposure time τ. Furthermore, using (4.17), we can numerically quantify the average contribution to the Fisher information matrix for each element of the list $\hat{\mathscr{A}}$ as the expectation above.

4.6.2 Hotelling Observer's Task Performance: SKE/BKS Case

In this section we want to evaluate the performance of the Hotelling observer for the case of list-mode data and as the exposure time is varied. From the list-mode data $\hat{\mathscr{A}}$, we can construct a Poisson point process [5, 283]

$$u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) = \sum_{\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}} \delta_{\mathrm{Dir}}\big(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}^{(j)}\big), \tag{4.19}$$

where the attribute vectors $\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}$ form the list of attributes $\hat{\mathscr{A}}$ collected during the fixed exposure time τ, and $\delta_{\mathrm{Dir}}(\ldots)$ denotes the Dirac delta function [284]. It is important to note that the Poisson point process $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ contains all the information of the list $\hat{\mathscr{A}}$ and it allows us to calculate means over realizations of $\hat{\mathscr{A}}$ [5]:

$$\begin{aligned}\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau) &= \sum_{J=0}^{\infty} J \mathrm{Pr}(J \mid \boldsymbol{f}, \tau) \int_{\infty} \mathrm{pr}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}') \, \mathrm{d}\hat{\boldsymbol{A}}' = \\ &= \overline{J}(\boldsymbol{f}, \tau) \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}).\end{aligned}$$

In the expression above, the quantity $\overline{J}(\boldsymbol{f}, \tau)$ denotes the mean number of attributes in the list $\hat{\mathscr{A}}$ when the object being imaged is $f(\boldsymbol{r})$, and the exposure time is τ. For simplicity, we will assume that $f(\boldsymbol{r})$ is a 2D object, but our treatment can be extended to more general cases. The function $f(\boldsymbol{r})$ represents the mean number of photons emitted by the object per unit time and per unit area. Thus, $f(\boldsymbol{r})$ has units of $\mathrm{s}^{-1}\mathrm{m}^{-2}$. If we introduce the sensitivity function $s(\boldsymbol{r})$ as the probability that an event (such as a positron-electron annihilation for the case of PET imaging) occurring at location $\boldsymbol{r}$ in the field of view (FOV) gives rise to an attribute $\hat{\boldsymbol{A}}^{(j)}$ being recorded in the list $\hat{\mathscr{A}}$, then

$$\overline{J}(\boldsymbol{f}, \tau) = \tau \int_{\mathrm{FOV}} s(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}^2\boldsymbol{r}. \tag{4.20}$$

If we rewrite $\mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})$ as

$$\mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) = \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \mathrm{pr}(\boldsymbol{r} \mid \boldsymbol{f}) \, \mathrm{d}^2\boldsymbol{r} \tag{4.21}$$

and use the equivalent of (4.5) but for the continuous case

$$\mathrm{pr}(\boldsymbol{r} \mid \boldsymbol{f}) = \frac{s(\boldsymbol{r}) f(\boldsymbol{r})}{\int_{\mathrm{FOV}} s(\boldsymbol{r}') f(\boldsymbol{r}') \, \mathrm{d}^2\boldsymbol{r}'}, \tag{4.22}$$

we get

$$\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau) = \tau \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) s(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}^2\boldsymbol{r}.$$

The form above allows us to introduce the operator $\mathcal{L}$ acting on the object $\boldsymbol{f}$ and producing a function defined in attribute space

$$(\mathcal{L}\boldsymbol{f})(\hat{\boldsymbol{A}}) = \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) s(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}^2\boldsymbol{r},$$

where $\mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r})$ is the probability density function for the estimated attribute $\hat{\boldsymbol{A}}$ given the 2D location of interaction $\boldsymbol{r}$. Because we want to study how the SNR^2 for the Hotelling observer changes with the exposure time τ, we have not concealed τ by including it in the definition of the operator $\mathcal{L}$. The integral kernel of $\mathcal{L}$ is [5]

$$[\mathcal{L}](\hat{\boldsymbol{A}}, \boldsymbol{r}) = \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) s(\boldsymbol{r}),$$

where we used the notation $[\mathcal{L}](\ldots)$ to explicity refer to the integral kernel of $\mathcal{L}$. hence, if we apply $\mathcal{L}$ to $\boldsymbol{f}$, we get

$$\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau) = (\tau\mathcal{L}\boldsymbol{f})(\hat{\boldsymbol{A}}). \tag{4.23}$$

Any decision algorithm that operates on list-mode data, receives as input the list $\hat{\mathscr{A}}$ or, equivalently, the Poisson point process $\boldsymbol{u}$ constructed from the list $\hat{\mathscr{A}}$. Therefore, by (2.11), the expression of SNR^2 takes the form [5]

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \Delta\overline{\overline{\boldsymbol{u}}}^{\dagger}\mathcal{K}_{\boldsymbol{u}}^{-1}\Delta\overline{\overline{\boldsymbol{u}}},$$

in which the symbol "†" denotes the adjoint of a process or operator, $\mathcal{K}_{\boldsymbol{u}} = \frac{1}{2}[\mathcal{K}_{\boldsymbol{u}|H_0} + \mathcal{K}_{\boldsymbol{u}|H_1}]$ is the mean covariance operator of the process $\boldsymbol{u}$, and

$$\begin{aligned}\Delta\overline{\overline{u}}(\hat{\boldsymbol{A}}) = \overline{\overline{u}}_1(\hat{\boldsymbol{A}}) - \overline{\overline{u}}_0(\hat{\boldsymbol{A}}) = \big(\tau\mathcal{L}\overline{\boldsymbol{f}}_1\big)(\hat{\boldsymbol{A}}) - \big(\tau\mathcal{L}\overline{\boldsymbol{f}}_0\big)(\hat{\boldsymbol{A}}) = \\ = \tau\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r})s(\boldsymbol{r})\left[\overline{f}_1(\boldsymbol{r}) - \overline{f}_0(\boldsymbol{r})\right]\,\mathrm{d}^2\boldsymbol{r} = (\tau\mathcal{L}\Delta\boldsymbol{f})(\hat{\boldsymbol{A}}),\end{aligned}$$

where $\overline{\boldsymbol{f}}_k$ denotes the mean of $\boldsymbol{f}$ under the hypothesis H_k: $\overline{\boldsymbol{f}}_k = \langle\boldsymbol{f}\rangle_{\boldsymbol{f}|H_k}$. The double bars appearing in the expression above denote double averages: average over list-mode data first and then over object. By writing $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ we have made it clear that the SNR^2 for the Hotelling observer is a function of the exposure time τ. To calculate $\mathcal{K}_{\boldsymbol{u}}$, we start by calculating $\mathcal{K}_{\boldsymbol{u}|H_k}$. By definition:

$$[\mathcal{K}_{\boldsymbol{u}|H_k}](\hat{\boldsymbol{A}},\hat{\boldsymbol{A}}') = \left\langle\left\langle\left[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}})\right]\left[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}}')\right]\right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}}\right\rangle_{\boldsymbol{f}|H_k}.$$

If we add and subtract the quantities $\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})$ and $\overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f})$ inside the first and second pair of square brackets of the right hand side of the expression above [66, 69, 124], we obtain

$$\begin{aligned}[\mathcal{K}_{\boldsymbol{u}|H_k}](\hat{\boldsymbol{A}},\hat{\boldsymbol{A}}') = \Bigg\langle\Bigg\langle \left[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) + \overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}})\right] \times \\ \times \left[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) + \overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}}')\right] \Bigg\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}}\Bigg\rangle_{\boldsymbol{f}|H_k}.\end{aligned}$$

When we carry out the products, we get

$$\begin{aligned}[\mathcal{K}_{\boldsymbol{u}|H_k}](\hat{\boldsymbol{A}},\hat{\boldsymbol{A}}') = \Bigg\langle\Bigg\langle \left[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})\right]\left[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f})\right] \times \\ \times \left[\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}})\right]\left[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f})\right] \times\end{aligned}$$

$$\times \Big[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})\Big] \Big[\overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}}')\Big] \times$$
$$\times \Big[\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}})\Big] \Big[\overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}}')\Big] \Big\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}} \Big\rangle_{\boldsymbol{f}|H_k}.$$

Notice that

$$\Big\langle\Big\langle\Big[\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}})\Big] \Big[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f})\Big]\Big\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}}\Big\rangle_{\boldsymbol{f}|H_k} = 0,$$
$$\Big\langle\Big\langle\Big[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})\Big] \Big[\overline{u}(\hat{\boldsymbol{A}}' \mid \boldsymbol{f}) - \overline{\overline{u}}_k(\hat{\boldsymbol{A}}')\Big]\Big\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}}\Big\rangle_{\boldsymbol{f}|H_k} = 0,$$

which allow us to rewrite $[\mathcal{K}_{\boldsymbol{u}|H_k}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')$ as

$$\begin{aligned}
[\mathcal{K}_{\boldsymbol{u}|H_k}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') &= \Big\langle[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\rangle_{\boldsymbol{f}|H_k} + \\
&\quad + \Big\langle\Big[(\tau\mathcal{L}\boldsymbol{f})(\hat{\boldsymbol{A}}) - (\tau\mathcal{L}\overline{\boldsymbol{f}}_k)(\hat{\boldsymbol{A}})\Big]\Big[(\tau\mathcal{L}\boldsymbol{f})(\hat{\boldsymbol{A}}') - (\tau\mathcal{L}\overline{\boldsymbol{f}}_k)(\hat{\boldsymbol{A}}')\Big]\Big\rangle_{\boldsymbol{f}|H_k} = \\
&= \Big\langle[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\rangle_{\boldsymbol{f}|H_k} + \tau^2[\mathcal{L}\langle(\boldsymbol{f} - \overline{\boldsymbol{f}}_k)(\boldsymbol{f} - \overline{\boldsymbol{f}}_k)^\dagger\rangle_{\boldsymbol{f}|H_k}\mathcal{L}^\dagger](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') = \\
&= \Big\langle[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\rangle_{\boldsymbol{f}|H_k} + \tau^2[\mathcal{L}\mathcal{K}_{\boldsymbol{f}|H_k}\mathcal{L}^\dagger](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}').
\end{aligned}$$

The quantity $[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')$ is the covariance at $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')$ of the stochastic process $\boldsymbol{u}$ conditioned on the knowledge of $\boldsymbol{f}$ [69]

$$[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') = \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}')\,\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}),$$

where, as before, $\delta_{\mathrm{Dir}}(\ldots)$ denotes the Dirac [284] delta function. If we take the expectation of $[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')$ we get

$$\Big\langle[\mathcal{K}_{\boldsymbol{u}|\boldsymbol{f}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\rangle_{\boldsymbol{f}|H_k} = \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}')\Big\langle\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f})\Big\rangle_{\boldsymbol{f}|H_k} = \tau\delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}')(\mathcal{L}\overline{\boldsymbol{f}}_k)(\hat{\boldsymbol{A}}),$$

which leads to

$$\begin{aligned}
[\mathcal{K}_{\boldsymbol{u}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') &= \frac{1}{2}\Big\{[\mathcal{K}_{\boldsymbol{u}|H_0}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') + [\mathcal{K}_{\boldsymbol{u}|H_1}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\} = \\
&= \frac{1}{2}\Big\{\tau\delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}')(\mathcal{L}\overline{\boldsymbol{f}}_0)(\hat{\boldsymbol{A}}) + \tau^2[\mathcal{L}\mathcal{K}_{\boldsymbol{f}|H_0}\mathcal{L}^\dagger](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') + \\
&\quad + \tau\delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}')(\mathcal{L}\overline{\boldsymbol{f}}_1)(\hat{\boldsymbol{A}}) + \tau^2[\mathcal{L}\mathcal{K}_{\boldsymbol{f}|H_1}\mathcal{L}^\dagger](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}')\Big\} =
\end{aligned}$$

$$= \tau \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}') \mathcal{L}\big\{\big(\overline{\boldsymbol{f}}_0 + \overline{\boldsymbol{f}}_1\big)/2\big\}(\hat{\boldsymbol{A}}) + \tau^2 [\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'),$$

where we used the fact that, for the SKE case, $\mathcal{K}_{\boldsymbol{f}|H_0} = \mathcal{K}_{\boldsymbol{f}|H_1}$. Finally [5]

$$\mathcal{K}_{\boldsymbol{u}} = \tau(\mathcal{L}\overline{\overline{\boldsymbol{f}}})\mathcal{I}_{\hat{\boldsymbol{A}}} + \tau^2 \mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger.$$

In the expression above, $\overline{\overline{f}}(\boldsymbol{r}) = \frac{1}{2}[\overline{f}_0(\boldsymbol{r}) + \overline{f}_1(\boldsymbol{r})]$, $\mathcal{K}_{\boldsymbol{f}} = \mathcal{K}_{\boldsymbol{f}|H_0} = \mathcal{K}_{\boldsymbol{f}|H_1}$, and $\mathcal{I}_{\hat{\boldsymbol{A}}}$ is the identity operator defined by the integral kernel

$$[\mathcal{I}_{\hat{\boldsymbol{A}}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') = \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}').$$

We remark that $(\mathcal{L}\overline{\overline{\boldsymbol{f}}})\mathcal{I}_{\hat{\boldsymbol{A}}}$ has to be interpreted as the operator whose kernel is

$$[(\mathcal{L}\overline{\overline{\boldsymbol{f}}})\mathcal{I}_{\hat{\boldsymbol{A}}}](\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') = \delta_{\mathrm{Dir}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}') \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) s(\boldsymbol{r}) \overline{\overline{f}}(\boldsymbol{r}) \, \mathrm{d}^2\boldsymbol{r}.$$

Using the quantities calculated above, we can write [5]

$$\begin{aligned}\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \Delta\overline{\overline{\boldsymbol{u}}}^\dagger \mathcal{K}_{\boldsymbol{u}}^{-1} \Delta\overline{\overline{\boldsymbol{u}}} = \tau^2 (\mathcal{L}\Delta\boldsymbol{f})^\dagger \mathcal{K}_{\boldsymbol{u}}^{-1} (\mathcal{L}\Delta\boldsymbol{f}) = \\ = \tau(\mathcal{L}\Delta\boldsymbol{f})^\dagger \left\{ (\mathcal{L}\overline{\overline{\boldsymbol{f}}})\mathcal{I}_{\hat{\boldsymbol{A}}} + \tau\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger \right\}^{-1} (\mathcal{L}\Delta\boldsymbol{f}).\end{aligned}$$

Application to a Simple Imaging System

To make this discussion more practical and, at the same time, to make calculations tractable, we will consider the simple case of a 2D object $f(\boldsymbol{r})$ imaged onto a 2D detector by a simple imaging system, as shown in Figure 4.12. The attribute vectors $\hat{\boldsymbol{A}}^{(j)}$ will be taken as the 2D location at which photons emitted from the object and propagating through the imaging system impinges on the detector. For this reason, we will use the more familiar notation $\hat{\boldsymbol{R}}^{(j)}$ to mean $\hat{\boldsymbol{A}}^{(j)}$.

We will assume that the signal we want to detect is known and buried in a random lumpy background [285, 286]. The lumps, whose expression is

$$\ell(\boldsymbol{r}) = \frac{b_0}{2\pi r_b^2} \exp\left(-\frac{|\boldsymbol{r}|^2}{2r_b^2}\right), \tag{4.24}$$

are located at $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K$, in which K is a Poisson random variable with mean $\overline{K}$. Points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K$ are 2D points uniformly distributed over some area $[-W/2, W/2] \times$

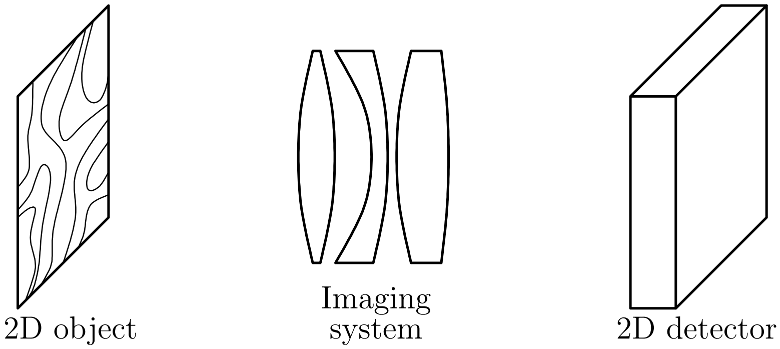


FIGURE 4.12. Schematic diagram of a 2D object imaged onto a 2D detector

$[-W/2, W/2]$. From the definition of $\ell(\boldsymbol{r})$, we see that each lump has an amplitude given by b_0 and width given by r_b. The quantity b_0 represents the mean number of photons emitted by a single lump per unit time. Therefore, the units of b_0 are s^{-1}. With this notation, the lumpy background can be written as [285, 286]

$$b_{\boldsymbol{\theta}}(\boldsymbol{r}) = \left[\sum_{k=1}^{K} \delta_{\text{Dir}}(\boldsymbol{r} - \boldsymbol{r}_k)\right] * \ell(\boldsymbol{r}) = \sum_{k=1}^{K} \ell(\boldsymbol{r} - \boldsymbol{r}_k) = b_0 \sum_{k=1}^{K} \frac{1}{2\pi r_b^2} \exp\left(-\frac{|\boldsymbol{r} - \boldsymbol{r}_k|^2}{2r_b^2}\right), \tag{4.25}$$

where the vector $\boldsymbol{\theta}$ completely characterizes the background: $\boldsymbol{\theta} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K\}$ and the symbol "$*$" denotes convolution. We could optionally include the integer number K in $\boldsymbol{\theta}$, but that number can always be found by counting the number of elements $\boldsymbol{r}_k$ in $\boldsymbol{\theta}$. We finally define the spatial (mean) lump density as $p = \overline{K}/W^2$. Some realizations of the 2D lumpy background $b_{\boldsymbol{\theta}}(\boldsymbol{r})$ are shown in Figure 4.13. For all these images, $W = 0.064$ m, $p = 2 \cdot 10^4$ m^{-2}, $r_b = 0.005$ m, and $b_0 = 1000$ s^{-1}.

To make calculations easier, we will assume $W \gg r_b$. Under this assumption, we can show that the background is a wide-sense stationary random process with mean and covariance given by [285, 286]

$$\langle b_{\boldsymbol{\theta}}(\boldsymbol{r}) \rangle_{\boldsymbol{\theta}} = p b_0,$$

and

$$K_{b_{\boldsymbol{\theta}}}(\boldsymbol{r}, \boldsymbol{r}') = K_{b_{\boldsymbol{\theta}}}(\boldsymbol{r} - \boldsymbol{r}', \boldsymbol{0}) = \frac{p b_0^2}{4\pi r_b^2} \exp\left(-\frac{|\boldsymbol{r} - \boldsymbol{r}'|^2}{4r_b^2}\right).$$

FIGURE 4.13. Examples of 2D lumpy backgrounds obtained for $W = 0.064$ m, $p = 2 \cdot 10^4$ m^{-2}, $r_b = 0.005$ m, and $b_0 = 1000$ s^{-1}

Because the signal is known, the covariance of the object $f(\boldsymbol{r})$ is the same under the hypotheses signal present and signal absent

$$\mathcal{K}_{\boldsymbol{f}|H_0} = \mathcal{K}_{\boldsymbol{f}|H_1} = \mathcal{K}_{\boldsymbol{f}},$$

and the integral kernel for the object covariance operator $\mathcal{K}_{\boldsymbol{f}}$ is simply [285, 286]

$$[\mathcal{K}_{\boldsymbol{f}}](\boldsymbol{r}, \boldsymbol{r}') = K_{b_{\boldsymbol{\theta}}}(\boldsymbol{r}, \boldsymbol{r}') = \frac{pb_0^2}{4\pi r_b^2} \exp\left(-\frac{|\boldsymbol{r} - \boldsymbol{r}'|^2}{4r_b^2}\right).$$

We will further assume

$$\mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) = \mathrm{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{r}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \boldsymbol{r}|^2}{2\sigma^2}\right), \tag{4.26}$$

for some σ^2 characterizing the imaging system and the detector, and we will set $s(\boldsymbol{r}) = 1$ for all $\boldsymbol{r}$ in the field of view. Using this and the expression for $\mathcal{K}_{\boldsymbol{f}}$ derived above, it follows [5]

$$[\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}}, \hat{\boldsymbol{R}}') = \frac{pb_0^2}{4\pi(r_b^2 + \sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'|^2}{4(r_b^2 + \sigma^2)}\right).$$

The result above, which was made possible by the fact that $\mathrm{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{r})$ only depends on $\hat{\boldsymbol{R}} - \boldsymbol{r}$ and $s(\boldsymbol{r}) = 1$,proves that $\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$ is stationary. Therefore

$$[\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}}, \hat{\boldsymbol{R}}') = [\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}', \boldsymbol{0}).$$

Concerning the signal $\Delta\boldsymbol{f}$, we will assume that it has a Gaussian shape and that it is centered at the origin:

$$\Delta f(\boldsymbol{r}) = \frac{b_s}{2\pi r_s^2} \exp\left(-\frac{|\boldsymbol{r}|^2}{2r_s^2}\right). \tag{4.27}$$

Similarly to b_0, b_s is the mean number of photons emitted by the signal per unit time. Thus, the units of b_s are s^{-1}. With this choice for the signal $\Delta\boldsymbol{f}$

$$\Delta\overline{\overline{u}}(\hat{\boldsymbol{R}}) = (\tau\mathcal{L}\Delta\boldsymbol{f})(\hat{\boldsymbol{R}}) = \frac{b_s\tau}{2\pi(r_s^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right),$$

and

$$(\mathcal{L}\overline{\overline{\boldsymbol{f}}})(\hat{\boldsymbol{R}}) = pb_0 + \frac{1}{2}\frac{b_s}{2\pi(r_s^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right).$$

If we define the operator $\mathcal{B}$ with integral kernel

$$[\mathcal{B}](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}') = \delta_{\mathrm{Dir}}(\hat{\boldsymbol{R}}-\hat{\boldsymbol{R}}')\left[pb_0 + \frac{1}{2}\frac{b_s}{2\pi(r_s^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right)\right].$$

we can write [5]

$$\begin{aligned}\mathcal{K}_{\boldsymbol{u}}^{-1} &= \left\{\tau\mathcal{B} + \tau^2\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right\}^{-1} = \left\{\tau\left[\mathcal{B}+\tau\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right]\right\}^{-1} = \\ &= \left\{\tau\mathcal{B}\left[\mathcal{I}_{\hat{\boldsymbol{R}}} + \tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right]\right\}^{-1} = \frac{1}{\tau}\left[\mathcal{I}_{\hat{\boldsymbol{R}}} + \tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right]^{-1}\mathcal{B}^{-1}.\end{aligned}$$

Recall that, at least formally [69]

$$[\mathcal{I}-\mathcal{X}]^{-1} = \sum_{n=0}^{\infty}\mathcal{X}^n.$$

We can use the identity above to calculate $\mathcal{K}_{\boldsymbol{u}}^{-1}$ provided that we set $\mathcal{X} = -\tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$

$$\mathcal{K}_{\boldsymbol{u}}^{-1} = \frac{1}{\tau}\left[\sum_{n=0}^{\infty}(-\tau)^n(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n\right]\mathcal{B}^{-1}. \tag{4.28}$$

Conditions under which the summation above converges are related to the eigenvalues $\lambda_{\boldsymbol{\rho}}$ of the operator $\mathcal{X} = -\tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$. Using the definition of $\mathcal{B}$, we get

$$[\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}') = \int_\infty [\mathcal{B}^{-1}](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}'')[\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}}'',\hat{\boldsymbol{R}}')\,\mathrm{d}^2\hat{\boldsymbol{R}}'' =$$

$$= \frac{\dfrac{pb_0^2}{4\pi(r_b^2+\sigma^2)} \exp\left(-\dfrac{|\hat{\boldsymbol{R}}-\hat{\boldsymbol{R}}'|^2}{4(r_b^2+\sigma^2)}\right)}{pb_0 + \dfrac{1}{2}\dfrac{b_s}{2\pi(r_s^2+\sigma^2)} \exp\left(-\dfrac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right)}. \tag{4.29}$$

In order to proceed, it is convenient to make the following assumption on the signal strength with respect to the background mean

$$\frac{1}{2}\frac{b_s}{2\pi(r_s^2+\sigma^2)} \ll pb_0,$$

or, equivalently

$$\frac{b_s}{4\pi pb_0(r_s^2+\sigma^2)} \ll 1. \tag{4.30}$$

It is worth noting that similar assumptions, which have been used in other circumstances in the literature [107, 124, 286], have usually been referred to as “weak-signal approximations” to mean that the signal intensity is much smaller than the background intensity. We will still refer to (4.30) as the weak-signal approximation even though that condition depends on other quantities—such as p, r_s, and σ—as well. Similarly to [107, 124, 286], the weak-signal approximation is used to make the covariance matrix of the data (understood here as the Poisson point process $\boldsymbol{u}$) the same under both hypotheses. When this happens, we talk about homoscedasticity, as we first pointed out in § 2.4. For the case at hand [5]

$$\mathcal{K}_{\boldsymbol{u}|H_k} = \tau\big(\mathcal{L}\overline{\boldsymbol{f}}_k\big)\mathcal{I}_{\hat{\boldsymbol{R}}} + \tau^2\mathcal{L}\mathcal{K}_{\boldsymbol{f}|H_k}\mathcal{L}^\dagger,$$

where

$$\begin{aligned} \overline{f}_0(\boldsymbol{r}) &= \langle b_{\boldsymbol{\theta}}(\boldsymbol{r})\rangle_{\boldsymbol{\theta}} = pb_0, \\ \overline{f}_1(\boldsymbol{r}) &= \langle b_{\boldsymbol{\theta}}(\boldsymbol{r})\rangle_{\boldsymbol{\theta}} + \Delta f(\boldsymbol{r}) = pb_0 + \frac{b_s}{2\pi r_s^2}\exp\left(-\frac{|\boldsymbol{r}|^2}{2r_s^2}\right). \end{aligned}$$

Therefore

$$[\mathcal{K}_{\boldsymbol{u}}](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}') = \frac{1}{2}\Big[\mathcal{K}_{\boldsymbol{u}|H_0} + \mathcal{K}_{\boldsymbol{u}|H_1}\Big](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}') =$$

$$= \tau p b_0 \left[1 + \frac{b_s}{4\pi p b_0 (r_s^2 + \sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right)\right] \delta_{\mathrm{Dir}}(\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}') + \\ + \tau^2 \frac{p b_0^2}{4\pi(r_b^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'|^2}{4(r_b^2+\sigma^2)}\right).$$

With the weak-signal approximation:

$$[\mathcal{K}_{\boldsymbol{u}}](\hat{\boldsymbol{R}}, \hat{\boldsymbol{R}}') \approx \tau p b_0 \left[\delta_{\mathrm{Dir}}(\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}') + \frac{\tau b_0}{4\pi(r_s^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'|^2}{4(r_b^2+\sigma^2)}\right)\right].$$

The assumption in (4.30) allows us to write $\mathcal{B}^{-1} \approx (p b_0)^{-1} \mathcal{I}_{\hat{\boldsymbol{R}}}$. Using this and (4.29), we obtain

$$[\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}}, \hat{\boldsymbol{R}}') \approx \frac{b_0}{4\pi(r_b^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'|^2}{4(r_b^2+\sigma^2)}\right).$$

With this approximation, we can easily calculate the eigenvalues of $-\tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$ by Fourier transform diagonalization:

$$\lambda_{\boldsymbol{\rho}} = -b_0 \tau e^{-4\pi^2(r_b^2+\sigma^2)|\boldsymbol{\rho}|^2}.$$

Thus, for the weak-signal case, the series $\sum_{n=0}^{\infty}(-\tau)^n(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n$ converges if and only if $|\lambda_{\boldsymbol{\rho}}| < 1$, which requires $b_0\tau < 1$. Notice that this condition is dimensionally correct because the units of b_0 are s^{-1} and so $b_0\tau$ is unitless. To calculate powers of $\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$, we notice that the integral kernel of such operator is well approximated by b_0 times the probability density function of a normally distributed random vector, evaluated at $\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'$. Using

$$\mathop{\Large\ast}_{n=1}^{N} \frac{1}{2\pi\sigma_n^2} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2\sigma_n^2}\right) = \frac{1}{2\pi\sum_{n=1}^{N}\sigma_n^2} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2\sum_{n=1}^{N}\sigma_n^2}\right),$$

we immediately get

$$[(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n](\hat{\boldsymbol{R}}, \hat{\boldsymbol{R}}') \approx \frac{b_0^n}{4\pi n(r_b^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}'|^2}{4n(r_b^2+\sigma^2)}\right).$$

With the results found above, we can write

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \frac{1}{\tau}\sum_{n=0}^{\infty}(-\tau)^n \Delta\overline{\overline{\boldsymbol{u}}}^\dagger (\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n \mathcal{B}^{-1}\Delta\overline{\overline{\boldsymbol{u}}},$$

where

$$(\mathcal{B}^{-1}\Delta\overline{\overline{\boldsymbol{u}}})(\hat{\boldsymbol{R}}) = \frac{b_s\tau}{2\pi(r_s^2+\sigma^2)}\exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right)\times$$

$$\times\left[pb_0+\frac{1}{2}\frac{b_s}{2\pi(r_s^2+\sigma^2)}\exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right)\right]^{-1}\approx$$

$$\approx\frac{b_s\tau}{pb_0}\frac{1}{2\pi(r_s^2+\sigma^2)}\exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right),$$

in which we used $\mathcal{B}^{-1}\approx(pb_0)^{-1}\mathcal{I}_{\hat{\boldsymbol{R}}}$. Notice that

$$\big(\Delta\overline{\overline{\boldsymbol{u}}}^\dagger(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n\big)(\hat{\boldsymbol{R}}) = \int_\infty \Delta\overline{\overline{u}}(\hat{\boldsymbol{R}}')[(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n](\hat{\boldsymbol{R}}',\hat{\boldsymbol{R}})\,\mathrm{d}^2\hat{\boldsymbol{R}}' =$$

$$= \frac{b_s b_0^n\tau}{2\pi[r_s^2+\sigma^2+2n(r_b^2+\sigma^2)]}\exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2[r_s^2+\sigma^2+2n(r_b^2+\sigma^2)]}\right).$$

Thus

$$\Delta\overline{\overline{\boldsymbol{u}}}^\dagger(\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger)^n\mathcal{B}^{-1}\Delta\overline{\overline{\boldsymbol{u}}} = \frac{b_s^2 b_0^{n-1}\tau^2}{p}\frac{1}{4\pi[nr_b^2+r_s^2+(n+1)\sigma^2]}.$$

Finally

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \frac{b_s^2\tau}{4\pi pb_0}\sum_{n=0}^{\infty}(-1)^n\frac{(b_0\tau)^n}{nr_b^2+r_s^2+(n+1)\sigma^2}.$$

Because the series above is of the form $\sum_{n=0}^\infty(-1)^n a_n$ with $a_n>0$, we have that it converges if and only if $a_{n+1}<a_n$ and $\lim_{n\to\infty}a_n=0$. Not surprisingly, that happens when $b_0\tau<1$, which is the same conditions we had on the eigenvalues $\lambda_{\boldsymbol{\rho}}$ of $\mathcal{X}=-\tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$. We also notice that this expression for $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ cannot be used to study what happens when the acquisition time τ increases. Indeed, we would eventually get $b_0\tau\geqslant 1$.

To fix the problem, we will follow another approach. Recall that [5]

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)\approx\tau(\mathcal{L}\Delta\boldsymbol{f})^\dagger\left\{\mathcal{I}_{\hat{\boldsymbol{R}}}+\tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right\}^{-1}\mathcal{B}^{-1}(\mathcal{L}\Delta\boldsymbol{f}),$$

where

$$[\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger](\hat{\boldsymbol{R}},\hat{\boldsymbol{R}}')\approx\frac{b_0}{4\pi(r_b^2+\sigma^2)}\exp\left(-\frac{|\hat{\boldsymbol{R}}-\hat{\boldsymbol{R}}'|^2}{4(r_b^2+\sigma^2)}\right),$$

$$(\mathcal{L}\Delta\boldsymbol{f})(\hat{\boldsymbol{R}}) = \frac{b_s}{2\pi(r_s^2+\sigma^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}}|^2}{2(r_s^2+\sigma^2)}\right).$$

Because $\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger$ is shift invariant, we can write

$$\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger = \mathcal{U}\mathcal{D}\mathcal{U}^\dagger, \tag{4.31}$$

provided that

$$\begin{aligned}
[\mathcal{U}](\hat{\boldsymbol{R}},\boldsymbol{\rho}) &= e^{i2\pi\hat{\boldsymbol{R}}\cdot\boldsymbol{\rho}},\\
[\mathcal{D}](\boldsymbol{\rho},\boldsymbol{\rho}') &= b_0\delta_{\mathrm{Dir}}(\boldsymbol{\rho}-\boldsymbol{\rho}')e^{-4\pi^2(r_b^2+\sigma^2)|\boldsymbol{\rho}|^2}.
\end{aligned}$$

The form above can be verified by substituting and performing the integrals. Then

$$\begin{aligned}
\left\{\mathcal{I}_{\hat{\boldsymbol{R}}} + \tau\mathcal{B}^{-1}\mathcal{L}\mathcal{K}_{\boldsymbol{f}}\mathcal{L}^\dagger\right\}^{-1} &= \left\{\mathcal{I}_{\hat{\boldsymbol{R}}} + \tau\mathcal{U}\mathcal{D}\mathcal{U}^\dagger\right\}^{-1} = \left\{\mathcal{U}\mathcal{U}^\dagger + \tau\mathcal{U}\mathcal{D}\mathcal{U}^\dagger\right\}^{-1} =\\
&= \left\{\mathcal{U}\left[\mathcal{I}_{\boldsymbol{\rho}}+\tau\mathcal{D}\right]\mathcal{U}^\dagger\right\}^{-1} = \mathcal{U}\left[\mathcal{I}_{\boldsymbol{\rho}}+\tau\mathcal{D}\right]^{-1}\mathcal{U}^\dagger,
\end{aligned}$$

in which we have used the fact that $\mathcal{U}$ is unitary ($\mathcal{U}^\dagger\mathcal{U} = \mathcal{U}\mathcal{U}^\dagger = \mathcal{I}$). Notice that $\mathcal{I}_{\boldsymbol{\rho}} + \tau\mathcal{D}$ is diagonal, thus its inverse is readily available. We have obtained [5]

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \approx \tau(\mathcal{L}\Delta\boldsymbol{f})^\dagger\mathcal{U}\left\{\mathcal{I}_{\boldsymbol{\rho}}+\tau\mathcal{D}\right\}^{-1}\mathcal{U}^\dagger\mathcal{B}^{-1}(\mathcal{L}\Delta\boldsymbol{f}),$$

where

$$\begin{aligned}
\big((\mathcal{L}\Delta\boldsymbol{f})^\dagger\mathcal{U}\big)(\boldsymbol{\rho}) &= b_s e^{-2\pi^2(r_s^2+\sigma^2)|\boldsymbol{\rho}|^2},\\
(\mathcal{U}^\dagger\mathcal{B}^{-1}\mathcal{L}\Delta\boldsymbol{f})(\boldsymbol{\rho}) &= \frac{b_s}{pb_0}e^{-2\pi^2(r_s^2+\sigma^2)|\boldsymbol{\rho}|^2}.
\end{aligned}$$

Therefore

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \approx \frac{b_s^2}{pb_0^2}\int_\infty \frac{b_0\tau e^{-4\pi^2(r_s^2+\sigma^2)|\boldsymbol{\rho}|^2}}{1+b_0\tau e^{-4\pi^2(r_b^2+\sigma^2)|\boldsymbol{\rho}|^2}}\,\mathrm{d}^2\boldsymbol{\rho}. \tag{4.32}$$

A dimensional analysis of the quantity on the right hand side shows that $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ is dimensionless, as it should be. For the integral above,

$$\int_\infty \frac{\alpha e^{-a|\boldsymbol{\rho}|^2}}{1+\alpha e^{-b|\boldsymbol{\rho}|^2}}\,\mathrm{d}^2\boldsymbol{\rho} = 2\pi\int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x,$$

with $\alpha = b_0\tau > 0$, $a = 4\pi^2(r_s^2 + \sigma^2) > 0$, and $b = 4\pi^2(r_b^2 + \sigma^2) > 0$. Assume first $a \neq b$. Then [287–290]

$$
\begin{aligned}
2\pi \int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x = \pi \int_0^\infty \frac{e^{-au}}{\frac{1}{\alpha} + e^{-bu}}\,\mathrm{d}u = \frac{\pi}{b}\int_0^1 \frac{y^{\frac{a}{b}-1}}{\frac{1}{\alpha}+y}\,\mathrm{d}y = \\
= \frac{\pi\alpha}{a}\,{}_2F_1\left(1, \frac{a}{b}; 1+\frac{a}{b}; -\alpha\right) = \\
= \frac{\pi}{a}\frac{\alpha}{1+\alpha}\,{}_2F_1\left(1, 1; 1+\frac{a}{b}; \frac{\alpha}{1+\alpha}\right),
\end{aligned}
$$

in which ${}_2F_1(\alpha, \beta; \gamma; z)$ is the hypergeometric function defined as the analytic continuation to the whole complex plane of the hypergeometric series [287]:

$$
{}_2F_1(\alpha, \beta; \gamma; z) = \sum_{n=0}^{\infty} \frac{(\alpha)_n(\beta)_n}{(\gamma)_n}\frac{z^n}{n!},
$$

where we defined

$$
(\alpha)_n = \begin{cases} \alpha(\alpha+1)(\alpha+2)\dots(\alpha+n-1) & \text{if } n \geqslant 1, \\ 1 & \text{if } n = 0. \end{cases}
$$

Notice that the integrals above converge when $\frac{a}{b} > 0$, which is the case as both a and b are positive. Hence, we can rewrite $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ as [5]

$$
\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \approx \frac{b_s^2}{pb_0^2}\frac{1}{4\pi(r_s^2+\sigma^2)}\frac{b_0\tau}{1+b_0\tau}\,{}_2F_1\left(1, 1; 1+\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}; \frac{b_0\tau}{1+b_0\tau}\right). \tag{4.33}
$$

Figure 4.14 shows plots of $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ as a function of time τ for three different cases, namely $r_s < r_b$, $r_s > r_b$, and $r_s = r_b$. Other parameters we have used are: $b_s = 100\ \mathrm{s}^{-1}$, $b_0 = 1000\ \mathrm{s}^{-1}$, $p = 20000\ \mathrm{m}^{-2}$, $\sigma = 0.001$ m, $r_b = 0.005$ m, and $\tau \in [0, 0.20]$. Being the ratio $b_s/4\pi p b_0(r_s^2+\sigma^2) < 0.02$, the $b_s/4\pi p b_0(r_s^2+\sigma^2) \ll 1$ condition of (4.30) is satisfied for all the three cases considered.

As a special case, if $r_s = r_b$, the integral in (4.32) can be solved with elementary methods, resulting in

$$
\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \approx \frac{b_s^2}{pb_0^2}\frac{1}{4\pi(r_s^2+\sigma^2)}\ln(1+b_0\tau).
$$

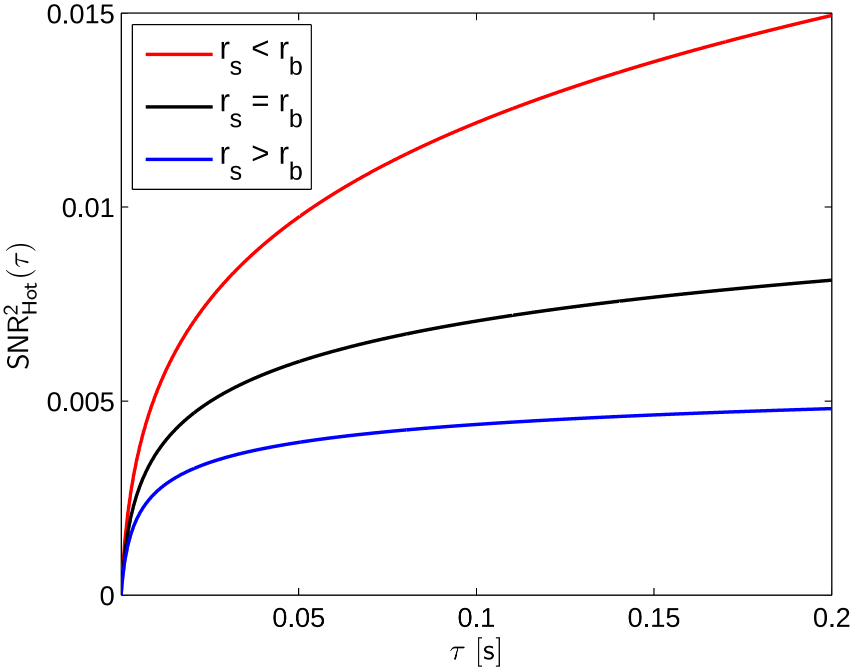


FIGURE 4.14. Plots of $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ for three different cases. For all plots, $b_s = 100\ \mathrm{s}^{-1}$, $b_0 = 1000\ \mathrm{s}^{-1}$, $p = 20000\ \mathrm{m}^{-2}$, $\sigma = 0.001$ m, $r_b = 0.005$ m, and $\tau \in [0, 0.20]$. For the red curve, $r_s = 0.0045$ m; for the blue curve $r_s = 0.0055$ m; and, finally, for the black curve $r_s = r_b = 0.005$ m (adapted from [5])

We want to study now the asymptotic behavior of $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ as $\tau \to 0$ and $\tau \to \infty$. For the $\tau \to 0$ case, we use a result found above and the definition of hypergeometric series to write

$$2\pi \int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x = \frac{\pi\alpha}{a}\,{}_2F_1\left(1, \frac{a}{b}; 1+\frac{a}{b}; -\alpha\right) = \pi \sum_{k=0}^{\infty} (-1)^k \frac{\alpha^{k+1}}{a+kb},$$

which is the Taylor series expansion at $\alpha = 0$. The series converges when $|\alpha| < 1$. Thus, for small values of τ, we can retain the first term of the Taylor expansion and get

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \overset{\tau \ll 1}{\approx} \frac{b_s^2}{p b_0^2} \frac{b_0 \tau}{4\pi(r_s^2+\sigma^2)}.$$

For the $\tau \to \infty$ case, let us first set $m = 1 - \frac{a}{b}$ and assume that m is not an integer. We can perform a Taylor series expansion to the second order of

$$\frac{\alpha}{1+\alpha}\,{}_2F_1\left(1, 1; 1+\frac{a}{b}; \frac{\alpha}{1+\alpha}\right) = \frac{\alpha}{1+\alpha}\,{}_2F_1\left(1, 1; 2-m; \frac{\alpha}{1+\alpha}\right)$$

for $1/\alpha \to 0^+$:

$$2\pi \int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x = \csc(\pi m)\frac{\pi^2}{a}\Big\{\alpha^m \left[\nu_1 + o\big(\alpha^{-2}\big)\right] - \left[\nu_2 + \nu_3 \alpha^{-1} + o\big(\alpha^{-2}\big)\right]\Big\}, \tag{4.34}$$

where $\csc(x) = \frac{1}{\sin(x)}$ and

$$\nu_1 = 1 - m, \qquad \nu_2 = \frac{\nu_1}{\Gamma(1-m)\Gamma(1+m)}, \qquad \nu_3 = -\frac{m}{m+1}\nu_2.$$

In the expressions above, the Gamma function $\Gamma(x)$ is the function defined as

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\,\mathrm{d}t.$$

Further manipulation of the expressions found above is possible. Indeed, by using $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1-x)\Gamma(x) = \pi\csc(\pi x) = \frac{\pi}{\sin(\pi x)}$ (for a proof, see, for example, [291, 292]), we get that, for $\alpha \gg 1$, (4.34) above can be rewritten as

$$2\pi \int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x \approx \csc\left(\pi\frac{a}{b}\right)\frac{\pi^2}{b}\alpha^{1-\frac{a}{b}} + \frac{\pi}{a-b}\left[1 - \frac{b-a}{2b-a}\alpha^{-1}\right].$$

If $m = 1 - \frac{a}{b}$ is an integer, the result above is not valid, because $\sin(\pi\frac{a}{b}) = 0$ and so $\csc(\pi\frac{a}{b})$ would not be defined. However [288]:

$$\begin{aligned} 2\pi\int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x &= \frac{\pi}{b}\int_0^1 \frac{y^{\frac{a}{b}-1}}{\frac{1}{\alpha}+y}\,\mathrm{d}y = \frac{\pi}{b}\int_0^1 \frac{y^n}{\frac{1}{\alpha}+y}\,\mathrm{d}y = \\ &= \frac{\pi}{b}\left\{\sum_{k=0}^{n-1}\left(-\frac{1}{\alpha}\right)^k \frac{1}{n-k} + \left(-\frac{1}{\alpha}\right)^n \ln(1+\alpha)\right\}, \end{aligned}$$

in which $n = -m = \frac{a}{b} - 1 \geqslant 0$ and integer. Equivalently, we can require $a \geqslant b$ with $\frac{a}{b}$ integer. Thus, for large values of α and $\frac{a}{b}$ integer, the quantity above behaves according to

$$2\pi\int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x \overset{\alpha\gg 1}{\approx} \begin{cases} \dfrac{\pi}{b}\ln(1+\alpha) & \text{if } a = b, \\ \dfrac{\pi}{a-b} & \text{if } \frac{a}{b} \in \mathbb{N}\setminus\{1\}, \end{cases}$$

where $\mathbb{N} = \{1, 2, 3, \dots\}$ is the set of positive integer numbers, and $\mathbb{N}\setminus\{1\}$ denotes the set $\mathbb{N}$ with the element 1 removed. To summarize, we have [5]

$$2\pi\int_0^\infty \frac{\alpha x e^{-ax^2}}{1+\alpha e^{-bx^2}}\,\mathrm{d}x \overset{\alpha\gg 1}{\approx} \begin{cases} \dfrac{\pi}{b}\ln(1+\alpha) & \text{if } a = b, \\ \dfrac{\pi}{a-b} & \text{if } \frac{a}{b} \in \mathbb{N}\setminus\{1\}, \\ \csc\left(\pi\dfrac{a}{b}\right)\dfrac{\pi^2}{b}\alpha^{1-\frac{a}{b}} + & \\ \qquad + \dfrac{\pi}{a-b}\left[1 - \dfrac{b-a}{2b-a}\alpha^{-1}\right] & \text{if } \frac{a}{b} \notin \mathbb{N}\setminus\{1\}. \end{cases}$$

Substituting the definitions of a, b, and α gives [5]

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) \overset{\tau\gg 1}{\approx} \frac{b_s^2}{pb_0^2} \begin{cases} \dfrac{1}{4\pi(r_s^2+\sigma^2)}\ln(1+b_0\tau) & \text{if } r_s = r_b, \\ \dfrac{1}{4\pi(r_s^2-r_b^2)} & \text{if } \frac{r_s^2+\sigma^2}{r_b^2+\sigma^2} \in \mathbb{N}\setminus\{1\}, \\ \csc\left(\pi\dfrac{r_s^2+\sigma^2}{r_b^2+\sigma^2}\right)\dfrac{1}{4(r_b^2+\sigma^2)}\left(b_0\tau\right)^{1-\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}} + & \\ \qquad + \dfrac{1}{4\pi(r_s^2-r_b^2)}\left[1 - \dfrac{r_b^2-r_s^2}{2r_b^2-r_s^2+\sigma^2}\dfrac{1}{b_0\tau}\right] & \text{if } \frac{r_s^2+\sigma^2}{r_b^2+\sigma^2} \notin \mathbb{N}\setminus\{1\}. \end{cases} \tag{4.35}$$

Although complicated, the expression above covers all possible cases. Indeed, if $r_s = r_b$, the first case gives the asymptotic behavior for $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ for large values of τ.

If $r_s < r_b$, then $\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}$ must necessarily be a real number in the open interval $(0,1)$. In such a case, the third case would apply. Finally, if $r_s > r_b$, then $\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}$ is strictly greater than 1 and either the second or the third case would apply.

We will now study what happens to the value of $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ as $\tau \to \infty$. From the previous equation and for the case $r_s = r_b$,

$$\lim_{\tau\to\infty} \mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \lim_{\tau\to\infty} \frac{b_s^2}{pb_0^2}\frac{1}{4\pi(r_b^2+\sigma^2)}\ln(1+b_0\tau) = \infty.$$

Similarly, for $r_s < r_b$,

$$\lim_{\tau\to\infty} \big(b_0\tau\big)^{1-\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}} = \infty,$$

because $1 - \frac{r_s^2+\sigma^2}{r_b^2+\sigma^2} > 0$. Therefore, if $r_s < r_b$,

$$\lim_{\tau\to\infty} \mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \infty.$$

If $r_s > r_b$ and $\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}$ is an integer, then, trivially,

$$\lim_{\tau\to\infty} \mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \frac{b_s^2}{pb_0^2}\frac{1}{4\pi(r_s^2-r_b^2)}.$$

If, $r_s > r_b$ but now $\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}$ is not an integer, we have

$$\lim_{\tau\to\infty} \big(b_0\tau\big)^{1-\frac{r_s^2+\sigma^2}{r_b^2+\sigma^2}} = 0,$$

which leads to

$$\lim_{\tau\to\infty} \mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \frac{b_s^2}{pb_0^2}\frac{1}{4\pi(r_s^2-r_b^2)}.$$

Thus, we can summarize our findings as follows [5]

$$\lim_{\tau\to\infty} \mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \begin{cases} \infty & \text{if } r_s \leqslant r_b, \\ \dfrac{b_s^2}{pb_0^2}\dfrac{1}{4\pi(r_s^2-r_b^2)} & \text{if } r_s > r_b. \end{cases}$$

The result above and the plots in Figure 4.14 show that the $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ keeps growing as the exposure time increases only for the $r_s \leqslant r_b$ case. Otherwise, $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ is limited. In the latter case, it grows as the signal intensity grows, and it decreases if the background intensity and/or the lump density increases. The peculiar behavior we have in the $r_s > r_b$ case can be intuitively explained by noting that the signal has a Gaussian shape, so do the lumps in the background. If the signal is wider that the lumps, it is possible to have a false positive when a few lumps cluster at the signal location in such a way that they actually look like the signal we are looking for. On the other hand, when the signal is narrower that the lumps ($r_s < r_b$), false positives are reduced because no lump arrangement will look like the signal we want to detect.

The $r_s = r_b$ case deserves a different discussion. One could argue that because $r_s = r_b$, a background lump placed at $\boldsymbol{r} = (0, 0)$ could mistakenly be interpreted as the signal $\Delta\boldsymbol{f}$ we want to detect, which in our case is indeed centered at $(0, 0)$ and it is a scaled version of the lump function $\ell(\boldsymbol{r})$. Thus, the $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ should be bounded when $r_s = r_b$. However, if we recall that the lump centers $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K$ are 2D points uniformly distributed over $[-W/2, W/2] \times [-W/2, W/2]$, we immediately realize that the probability of having a lump centered at $(0, 0)$ is zero. Thus, with probability 1, a lump centered at $(0, 0)$ must be understood as the signal $\Delta\boldsymbol{f}$.

We can show that the two expressions we have found for $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ are the same for the case $b_0\tau < 1$ and $r_s = r_b$. Using the singular value decomposition (4.31) and for finite time

$$\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) = \frac{b_s^2}{p b_0^2} \frac{\ln(1 + b_0\tau)}{4\pi(r_s^2 + \sigma^2)}.$$

Using the Neumann series expansion of (4.28), we can write

$$\begin{aligned}\mathrm{SNR}^2_{\mathrm{Hot}}(\tau) &= \frac{b_s^2\tau}{4\pi p b_0} \sum_{n=0}^{\infty} (-1)^n \frac{(b_0\tau)^n}{(n+1)(r_s^2 + \sigma^2)} = \frac{b_s^2}{p b_0^2} \frac{\sum_{n=0}^{\infty} (-1)^n \frac{(b_0\tau)^{n+1}}{n+1}}{4\pi(r_s^2 + \sigma^2)} = \\ &= \frac{b_s^2}{p b_0^2} \frac{\sum_{n=1}^{\infty} (-1)^{n+1} \frac{(b_0\tau)^n}{n}}{4\pi(r_s^2 + \sigma^2)} = \frac{b_s^2}{p b_0^2} \frac{\ln(1 + b_0\tau)}{4\pi(r_s^2 + \sigma^2)},\end{aligned}$$

in which we have recognized the Taylor series expansion of $\ln(1 + b_0\tau)$.

Study of Detective Quantum Efficiency

The development of a performance figure of merit for detection performance—such as the $\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)$ in (4.33)—lends itself to a way to assess the detector and system's performance by introducing the *detective quantum efficiency* or DQE [293–295]

$$\mathrm{DQE} = \frac{\mathrm{SNR}^2}{\mathrm{SNR}^2_{\text{ideal system}}}. \tag{4.36}$$

In other words, the DQE describes the degree to which the SNR^2 has been degraded by the system, in terms of observer performance [69]. The DQE was first introduced in [296] to study the efficiency of the human eye as a photoreceptor. In recent years, the concept of DQE has also been extended to observer models, as a way to measure the relative performance of two model observers on the same task [69] or to compare the performance of the human observer relative to the ideal observer [297]. In our treatment, we use the definition in (4.36) to evaluate the DQE as a function of exposure time τ using (4.33) as figure of merit for the detection task

$$\mathrm{DQE}_{\mathrm{Hot}}(\tau) = \frac{\mathrm{SNR}^2_{\mathrm{Hot}}(\tau)}{\mathrm{SNR}^2_{\text{Hot, ideal system}}(\tau)},$$

in which $\mathrm{SNR}^2_{\text{Hot, ideal system}}(\tau)$ is calculated by assuming $\sigma \to 0$ in (4.26) and by carrying out the same derivation that led to (4.33). It is interesting to analyze how $\mathrm{DQE}_{\mathrm{Hot}}(\tau)$ is affected by the exposure time τ and for the same cases that were considered in Figure 4.14; plots of $\mathrm{DQE}_{\mathrm{Hot}}(\tau)$ are reported in Figure 4.15.

We do not have have a justification for the different behaviors of $\mathrm{DQE}_{\mathrm{Hot}}(\tau)$ as $\tau \to \infty$ that Figure 4.15 suggests. Intuitively, and as τ increases, $\mathrm{DQE}_{\mathrm{Hot}}(\tau)$ should not decrease because the longer the exposure time, the more (on average) events are collected and so the more "information" about the object will be available, hence "compensating" for the detector's non-ideal behavior (modeled by $\sigma > 0$). However, as Figure 4.15 shows, this does not happen for the $r_s < r_b$ case, which leads us to suspect that the Hotelling observer for a list-mode SKE/BKS detection problem

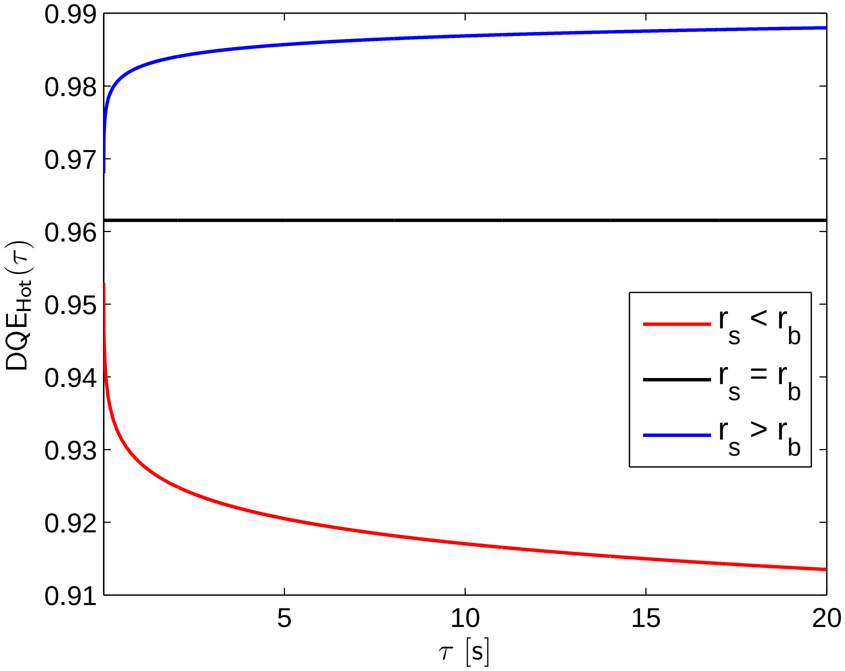


FIGURE 4.15. Plots of $\mathrm{DQE}_{\mathrm{Hot}}(\tau)$ for three different cases. For all plots, $b_s = 100$ s^{-1}, $b_0 = 1000$ s^{-1}, $p = 20000$ m^{-2}, $\sigma = 0.001$ m, $r_b = 0.005$ m, and $\tau \in [0, 20]$. For the red curve, $r_s = 0.0045$ m; for the blue curve $r_s = 0.0055$ m; and, finally, for the black curve $r_s = r_b = 0.005$ m

should not be used to assess system performance, as it does not always behave as common sense would suggest.

4.6.3 Ideal Observer's Task Performance: SKE/BKS Case

We begin this section by describing a way to calculate $\mathrm{SNR}^2_\lambda(\tau)$ where $\lambda(\hat{\mathscr{A}}) = \ln \Lambda(\hat{\mathscr{A}})$ is the list-mode log-likelihood ratio:

$$\lambda(\hat{\mathscr{A}}) = \ln \left[\frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \right]$$

and τ is the exposure time. The main goal of this section is to compare the performance of the list-mode log-likelihood ratio—quantified via the $\mathrm{SNR}^2_\lambda(\tau)$—with the performance of the list-mode Hotelling observer, for which analytical results were derived in § 4.6.2.

Even for the simple imaging system of Figure 4.12, an expression for the $\mathrm{SNR}^2_\lambda(\tau)$ was not found. Therefore, we turned to Markov chain Monte Carlo (MCMC) integration to calculate expectations and, ultimately, the $\mathrm{SNR}^2_\lambda(\tau)$. A basic introduction to MCMC was given in § 4.5.2. We also remark that our procedure for the calculation of $\mathrm{SNR}^2_\lambda(\tau)$ is based on the one detailed in [298], with the necessary adaptations.

As in [298], we characterize the background with the matrix $\boldsymbol{\Phi}$ defined as

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{r}_1 & \beta_1 \\ \vdots & \vdots \\ \boldsymbol{r}_{K_{\max}} & \beta_{K_{\max}} \end{pmatrix}, \tag{4.37}$$

where $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{K_{\max}}$ are 2D points independently and uniformly distributed over the whole field of view $[-W/2, W/2] \times [-W/2, W/2]$, and the dimensionless quantities $\beta_1, \ldots, \beta_{K_{\max}}$ are independent Bernoulli random variables [127, 128] with parameter P: $\Pr(\beta_k = 1) = P$ and $\mathrm{pr}(\beta_k = 0) = 1 - P$. Later in this section we will assume $K_{\max} \to \infty$ and we will modify the expressions accordingly. The interpretation of $\boldsymbol{\Phi}$ is that if $\beta_k = 1$, then there is an "active" lump at location $\boldsymbol{r}_k$. If, on the other hand,

$\beta_k = 0$, then no lump is present at $\boldsymbol{r}_k$. Thus we can write

$$b_{\boldsymbol{\Phi}}(\boldsymbol{r}) = \sum_{k=1}^{K_{\max}} \beta_k \ell(\boldsymbol{r} - \boldsymbol{r}_k),$$

where $\ell(\boldsymbol{r})$ is the lump function given in (4.24). Notice that the number of active lumps (i.e., $\beta_k = 1$) in $\boldsymbol{\Phi}$ is $K_1 = \sum_{k=1}^{K_{\max}} \beta_k$ and it follows a binomial distribution [127, 128] with parameters $K_{\max}$ and P. Imposing that the mean number of active lumps $\overline{K} = K_{\max} P$ equates pW^2 (where p is the mean spatial lump density; see § 4.6.2) yields

$$P = \frac{\overline{K}}{K_{\max}} = \frac{pW^2}{K_{\max}}. \tag{4.38}$$

Following [298],

$$\begin{aligned}
\Lambda(\hat{\mathscr{A}}) &= \int_\infty \frac{\mathrm{pr}(\hat{\mathscr{A}}, \boldsymbol{\Phi} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \, \mathrm{d}\boldsymbol{\Phi} = \int_\infty \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_1) \, \mathrm{pr}(\boldsymbol{\Phi}) \, \mathrm{Pr}(H_1)}{\mathrm{Pr}(H_1) \, \mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \, \mathrm{d}\boldsymbol{\Phi} = \\
&= \int_\infty \underbrace{\left[\frac{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_0)} \right]}_{\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi})} \frac{\mathrm{pr}(\hat{\mathscr{A}}, \boldsymbol{\Phi}, H_0)}{\mathrm{pr}(\boldsymbol{\Phi}, H_0)} \frac{\mathrm{pr}(\boldsymbol{\Phi}) \mathrm{Pr}(H_0)}{\mathrm{pr}(\hat{\mathscr{A}}, H_0)} \, \mathrm{d}\boldsymbol{\Phi} = \\
&= \int_\infty \Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}) \, \mathrm{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_0) \, \mathrm{d}\boldsymbol{\Phi},
\end{aligned} \tag{4.39}$$

where we used $\mathrm{pr}(\boldsymbol{\Phi}, H_0) = \mathrm{pr}(\boldsymbol{\Phi}) \, \mathrm{Pr}(H_0)$ and we introduced the background-known-exactly (BKE) likelihood ratio $\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi})$. The expression above is used to estimate $\Lambda(\hat{\mathscr{A}})$ as follows

$$\hat{\Lambda}(\hat{\mathscr{A}}) = \frac{1}{N} \sum_{n=1}^{N} \Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}^{(n)}),$$

where $\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(N)}$ are samples distributed according to $\mathrm{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_0)$ and they are obtained via MCMC methods.

The implementation of MCMC integration requires the calculation of the acceptance probability for a candidate background matrix $\boldsymbol{\Phi}'$ given the current background matrix $\mathbf{\Phi}$ (here, $\mathbf{\Phi}$ is one of the $\mathbf{\Phi}^{(1)}, \ldots, \mathbf{\Phi}^{(N)}$). Using (4.14), the acceptance probability for the problem at hand, is written as

$$P_{\text{acc}}(\mathbf{\Phi}', \mathbf{\Phi}; \hat{\mathscr{A}}, H_0) = \min\left\{1, \frac{\text{pr}(\mathbf{\Phi}' \mid \hat{\mathscr{A}}, H_0)q(\mathbf{\Phi} \mid \mathbf{\Phi}')}{\text{pr}(\mathbf{\Phi} \mid \hat{\mathscr{A}}, H_0)q(\mathbf{\Phi}' \mid \mathbf{\Phi})}\right\}. \tag{4.40}$$

By application of the definition of conditional density,

$$\text{pr}(\mathbf{\Phi} \mid \hat{\mathscr{A}}, H_0) = \frac{\text{pr}(\hat{\mathscr{A}} \mid \mathbf{\Phi}, H_0)\,\text{pr}(\mathbf{\Phi})}{\text{pr}(\mathscr{A})}.$$

The factor $\text{pr}(\mathscr{A})$ cancels when the fraction in (4.40) is evaluated, leading to

$$\frac{\text{pr}(\mathbf{\Phi}' \mid \hat{\mathscr{A}}, H_0)}{\text{pr}(\mathbf{\Phi} \mid \hat{\mathscr{A}}, H_0)} = \frac{\text{pr}(\hat{\mathscr{A}} \mid \mathbf{\Phi}', H_0)}{\text{pr}(\hat{\mathscr{A}} \mid \mathbf{\Phi}, H_0)} \frac{\text{pr}(\mathbf{\Phi}')}{\text{pr}(\mathbf{\Phi})}.$$

Writing an expression for $q(\mathbf{\Phi}' \mid \mathbf{\Phi})$ first requires deciding how a proposal matrix $\mathbf{\Phi}'$ is generated from the current matrix $\mathbf{\Phi}$. Recall that $\mathbf{\Phi}$ and $\mathbf{\Phi}'$ have the form given in (4.37) and, in particular, the random variables $\beta_1, \ldots, \beta_{K_{\max}}$ indicate the presence or absence of lumps at locations $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{K_{\max}}$, respectively. Equivalently, we will say that if $\beta_k = 1$, there is an "active" lump centered at $\boldsymbol{r}_k$; if $\beta_k = 0$, the lump at $\boldsymbol{r}_k$ is "inactive." Generating $\mathbf{\Phi}'$ from $\mathbf{\Phi}$ is implemented by perturbing one of the active lump centers in $\mathbf{\Phi}$ and by randomly flipping the active/inactive flags of the remaining lump centers. Conditions we want to ensure when generating $\mathbf{\Phi}'$ from $\mathbf{\Phi}$ are the following:

- exactly one lump that is active in both $\mathbf{\Phi}$ and $\mathbf{\Phi}'$ has been "perturbed" in $\mathbf{\Phi}'$;
- on average, the number of flips from active lumps to inactive lumps ("$1 \to 0$") is the same as the number of flips from inactive lumps to active lumps ("$0 \to 1$").

To enforce the first condition, we randomly select an active lump center $\boldsymbol{r}_k$ in $\mathbf{\Phi}$, we perturb it to get $\boldsymbol{r}'_k$, and set it as an active lump center in $\mathbf{\Phi}'$. More specifically, $\boldsymbol{r}'_k$ is obtained from $\boldsymbol{r}_k$ as $\boldsymbol{r}'_k = \boldsymbol{r}_k + \Delta\boldsymbol{r}$ where $\Delta\boldsymbol{r} = (\Delta x, \Delta y)$ follows a 2D normal distribution:

$$\text{pr}(\Delta\boldsymbol{r}) = \frac{1}{2\pi\sigma^2_{\text{pert.}}} \exp\left(-\frac{(\Delta x)^2 + (\Delta y)^2}{2\sigma^2_{\text{pert.}}}\right).$$

For the second condition, we introduce the probability $\eta_{0\to1}$ of an inactive-to-active flip and the probability $\eta_{1\to0}$ of an active-to-inactive lump flip. Let $K_0 = \sum_{k=1}^{K_{\max}}(1 - \beta_k)$ denote the number of inactive lumps in $\mathbf{\Phi}$ and recall that $K_1 = \sum_{k=1}^{K_{\max}} \beta_k$ is the number of active lumps in $\mathbf{\Phi}$. Clearly, $K_0 + K_1 = K_{\max}$. The number of inactive-to-active flips follows a binomial distribution with parameters K_0 and $\eta_{0\to1}$, while the number of active-to-inactive flips follows a binomial distribution but with parameters $K_1 - 1$ and $\eta_{1\to0}$. The reason why we have $K_1 - 1$ instead of K_1 is because at least one active lumps must remain active (more precisely, the one we force to remain active is the one we slightly perturb). Thus the expected number of inactive-to-active flips is $K_0\eta_{0\to1}$ and the expected number of active-to-inactive flips is $(K_1 - 1)\eta_{1\to0}$. We want these two number to be equal (so that, for any k, the probability of increasing the number of lumps by k is the same as the probability of decreasing the number of lumps by k). Hence, the expected total number of flips is $K_0\eta_{0\to1} + (K_1 - 1)\eta_{1\to0}$. If we force the expected total number of flips be equal to $(K_0 + K_1)\eta = K_{\max}\eta$, where η is defined as the probability of a flip, we obtain

$$\eta_{0\to1} = \frac{K_{\max}\eta}{2K_0}, \qquad \eta_{1\to0} = \frac{K_{\max}\eta}{2(K_1 - 1)},$$

where we also imposed $K_0\eta_{0\to1} = (K_1 - 1)\eta_{1\to0}$. We can now write

$$q(\mathbf{\Phi}' \mid \mathbf{\Phi}) = (\eta_{0\to1})^{f_{0\to1}}(1 - \eta_{0\to1})^{K_0 - f_{0\to1}}(\eta_{1\to0})^{f_{1\to0}}(1 - \eta_{1\to0})^{K_1 - 1 - f_{1\to0}} \times$$
$$\times \frac{1}{K_1}\sum_{k=1}^{K_{\max}} \beta_k\beta'_k \left[\prod_{\substack{k''=1\\k''\neq k}}^{K_{\max}} \delta_{\mathrm{Dir}}(\boldsymbol{r}_{k''} - \boldsymbol{r}'_{k''})\right] \frac{1}{2\pi\sigma^2_{\mathrm{pert.}}} \exp\left(-\frac{|\boldsymbol{r}_k - \boldsymbol{r}'_k|^2}{2\sigma^2_{\mathrm{pert.}}}\right),$$

where $f_{0\to1}$ and $f_{1\to0}$ denote the number of inactive-to-active and active-to-inactive flips, respectively:

$$f_{0\to1} = \left|\left\{k \text{ such that } \beta_k = 0 \text{ and } \beta'_k = 1\right\}\right|,$$
$$f_{1\to0} = \left|\left\{k \text{ such that } \beta_k = 1 \text{ and } \beta'_k = 0\right\}\right|,$$

and $|S|$ stands for the number elements (cardinality) of a set S. The factor $\eta_{0\to 1}^{f_{0\to 1}}(1-\eta_{0\to 1})^{K_0-f_{0\to 1}}$ in the expression of $q(\mathbf{\Phi}' \mid \mathbf{\Phi})$ refers to the probability of turning $f_{0\to 1}$ of the K_0 inactive lumps of $\mathbf{\Phi}$ into active lumps in $\mathbf{\Phi}'$. Similarly, $\eta_{1\to 0}^{f_{1\to 0}}(1-\eta_{1\to 0})^{K_1-1-f_{1\to 0}}$ is the probability of deactivating $f_{1\to 0}$ lumps that were active in $\mathbf{\Phi}$, and we have excluded one of the active lumps of $\mathbf{\Phi}$. Finally, the second line in the expression for $q(\mathbf{\Phi}' \mid \mathbf{\Phi})$ accounts for the probability density function of the perturbed lump. Because only one active lump is perturbed, we have used the Dirac delta function [284] to write the probability density function that expresses this fact. In other words, the integral over $\boldsymbol{r}'_1, \ldots, \boldsymbol{r}'_{K_{\max}}$ of the expression in $[\ldots]$ evaluates to 1 only when $\boldsymbol{r}_k$ was the lump center in $\mathbf{\Phi}$ that was perturbed to get $\boldsymbol{r}'_k$ in $\mathbf{\Phi}'$, otherwise it evaluates to 0.

If we exchange $\mathbf{\Phi}$ and $\mathbf{\Phi}'$ and use

$$K'_0 = K_0 + f_{1\to 0} - f_{0\to 1}, \qquad K'_1 = K_1 + f_{0\to 1} - f_{1\to 0},$$
$$f'_{0\to 1} = f_{1\to 0}, \qquad f'_{1\to 0} = f_{0\to 1},$$

we obtain

$$q(\mathbf{\Phi} \mid \mathbf{\Phi}') = (\eta'_{0\to 1})^{f_{1\to 0}}(1-\eta'_{0\to 1})^{K_0-f_{0\to 1}}(\eta'_{1\to 0})^{f_{0\to 1}}(1-\eta'_{1\to 0})^{K_1-1-f_{1\to 0}} \times$$
$$\times \frac{1}{K_1 + f_{0\to 1} - f_{1\to 0}} \sum_{k=1}^{K_{\max}} \beta_k \beta'_k \left[\prod_{\substack{k''=1 \\ k''\neq k}}^{K_{\max}} \delta_{\mathrm{Dir}}(\boldsymbol{r}_{k''} - \boldsymbol{r}'_{k''}) \right] \frac{1}{2\pi\sigma^2_{\text{pert.}}} \exp\left(-\frac{|\boldsymbol{r}_k - \boldsymbol{r}'_k|^2}{2\sigma^2_{\text{pert.}}} \right).$$

The ratio $q(\mathbf{\Phi} \mid \mathbf{\Phi}')/q(\mathbf{\Phi}' \mid \mathbf{\Phi})$ thus is

$$\frac{q(\mathbf{\Phi} \mid \mathbf{\Phi}')}{q(\mathbf{\Phi}' \mid \mathbf{\Phi})} = \frac{K_1}{K_1 + f_{0\to 1} - f_{1\to 0}} \left[\frac{\eta'_{1\to 0}}{\eta_{0\to 1}}\right]^{f_{0\to 1}} \left[\frac{\eta'_{0\to 1}}{\eta_{1\to 0}}\right]^{f_{1\to 0}} \times$$
$$\times \left[\frac{1-\eta'_{0\to 1}}{1-\eta_{0\to 1}}\right]^{K_0-f_{0\to 1}} \left[\frac{1-\eta'_{1\to 0}}{1-\eta_{1\to 0}}\right]^{K_1-1-f_{1\to 0}},$$

where

$$\frac{\eta'_{1\to 0}}{\eta_{0\to 1}} = \frac{K_0}{K'_1 - 1},$$
$$\frac{\eta'_{0\to 1}}{\eta_{1\to 0}} = \frac{K_1 - 1}{K'_0},$$

$$\begin{aligned}\frac{1-\eta'_{0\to 1}}{1-\eta_{0\to 1}} &= \frac{2K'_0 - K_{\max}\eta}{2K_0 - K_{\max}\eta}\cdot\frac{K_0}{K'_0},\\ \frac{1-\eta'_{1\to 0}}{1-\eta_{1\to 0}} &= \frac{2(K'_1-1) - K_{\max}\eta}{2(K_1-1) - K_{\max}\eta}\cdot\frac{K_1-1}{K'_1-1}.\end{aligned}$$

Finally, the density $\mathrm{pr}(\mathbf{\Phi})$ is

$$\mathrm{pr}(\mathbf{\Phi}) = \frac{P^{K_1}(1-P)^{K_{\max}-K_1}}{K_1!\,(K_{\max}-K_1)!}\left(\frac{1}{W^2}\right)^{K_{\max}},$$

where the factorial terms in the denominator account for the fact that we can permute the sets of active and inactive lump centers, whereas P is the probability that any given lump is active. The ratio $\mathrm{pr}(\mathbf{\Phi'})/\mathrm{pr}(\mathbf{\Phi})$ thus is

$$\frac{\mathrm{pr}(\mathbf{\Phi'})}{\mathrm{pr}(\mathbf{\Phi})} = \frac{\dfrac{P^{K'_1}(1-P)^{K'_{\max}-K'_1}}{K'_1!\,(K_{\max}-K'_1)!}}{\dfrac{P^{K_1}(1-P)^{K_{\max}-K_1}}{K_1!\,(K_{\max}-K_1)!}} = \frac{\dbinom{K_{\max}}{K'_1}\left(\dfrac{pW^2}{K_{\max}}\right)^{K'_1}\left(1-\dfrac{pW^2}{K_{\max}}\right)^{K_{\max}-K'_1}}{\dbinom{K_{\max}}{K_1}\left(\dfrac{pW^2}{K_{\max}}\right)^{K_1}\left(1-\dfrac{pW^2}{K_{\max}}\right)^{K_{\max}-K_1}},$$

where we used (4.38) to rewrite P in terms of other quantities and we multiplied both the numerator and the denominator by $K_{\max}!$ so that binomial coefficients can be introduced. As $K_{\max}$ become large:

$$\lim_{K_{\max}\to\infty}\binom{K_{\max}}{K_1}\left(\frac{pW^2}{K_{\max}}\right)^{K_1}\left(1-\frac{pW^2}{K_{\max}}\right)^{K_{\max}-K_1} = \frac{(pW^2)^{K_1}}{K_1!}e^{-pW^2} > 0.$$

By the theorem in Appendix C:

$$\begin{aligned}\lim_{K_{\max}\to\infty}\frac{\mathrm{pr}(\mathbf{\Phi'})}{\mathrm{pr}(\mathbf{\Phi})} &= \frac{\displaystyle\lim_{K_{\max}\to\infty}\binom{K_{\max}}{K'_1}\left(\frac{pW^2}{K_{\max}}\right)^{K'_1}\left(1-\frac{pW^2}{K_{\max}}\right)^{K_{\max}-K'_1}}{\displaystyle\lim_{K_{\max}\to\infty}\binom{K_{\max}}{K_1}\left(\frac{pW^2}{K_{\max}}\right)^{K_1}\left(1-\frac{pW^2}{K_{\max}}\right)^{K_{\max}-K_1}} = \\ &= \frac{K_1!}{K'_1!}\left(pW^2\right)^{K'_1-K_1}.\end{aligned}$$

To complete the calculation of $P_{\mathrm{acc}}(\mathbf{\Phi'},\mathbf{\Phi};\hat{\mathscr{A}},H_0)$, we need an expression for $\mathrm{pr}(\hat{\mathscr{A}}\mid\mathbf{\Phi},H_0)$. We will assume that the number of attribute vectors J in the list $\hat{\mathscr{A}}$ is not set in advance. Therefore, conditioned on $\mathbf{\Phi}$ and H_0, J follows Poisson statistics:

$$\Pr(J\mid\mathbf{\Phi},H_0) = \frac{1}{J!}\left[\tau\int_{\mathrm{FOV}} b_{\mathbf{\Phi}}(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right]^J \exp\left(-\tau\int_{\mathrm{FOV}} b_{\mathbf{\Phi}}(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right),$$

where τ is the exposure time. We can now write

$$\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_0) = \mathrm{Pr}(J \mid \boldsymbol{\Phi}, H_0) \prod_{j=1}^{J} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{\Phi}, H_0),$$

where

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{\Phi}, H_0) = \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \mathrm{pr}(\boldsymbol{r} \mid \boldsymbol{\Phi}, H_0)\, \mathrm{d}\boldsymbol{r} = \\ = \frac{1}{\int_{\mathrm{FOV}} b_{\boldsymbol{\Phi}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) b_{\boldsymbol{\Phi}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}.\end{aligned}$$

Thus

$$\begin{aligned}\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_0) = \frac{\tau^J}{J!} \exp\left(-\tau \int_{\mathrm{FOV}} b_{\boldsymbol{\Phi}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}\right) \times \\ \times \prod_{j=1}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) b_{\boldsymbol{\Phi}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}.\end{aligned} \tag{4.41}$$

In a similar fashion

$$\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_1) = \mathrm{Pr}(J \mid \boldsymbol{\Phi}, H_1) \prod_{j=1}^{J} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{\Phi}, H_1),$$

where

$$\begin{aligned}\mathrm{Pr}(J \mid \boldsymbol{\Phi}, H_1) = \frac{1}{J!} \left[\tau \int_{\mathrm{FOV}} \Big\{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\Big\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}\right]^J \times \\ \times \exp\left(-\tau \int_{\mathrm{FOV}} \Big\{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\Big\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}\right),\end{aligned}$$

and

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{\Phi}, H_1) = \frac{1}{\int_{\mathrm{FOV}} \{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \times \\ \times \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Big\{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\Big\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}.\end{aligned}$$

The expressions above allow us to write

$$\begin{aligned}\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_1) = \frac{\tau^J}{J!} \exp\left(-\tau \int_{\mathrm{FOV}} \Big\{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\Big\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}\right) \times \\ \times \prod_{j=1}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Big\{b_{\boldsymbol{\Phi}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})\Big\} s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}.\end{aligned} \tag{4.42}$$

Using (4.41) and (4.42), we can write an expression for $\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi})$:

$$\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}) = \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}, H_0)} = \exp\left(-\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}\right) \times$$
$$\times \prod_{j=1}^{J} \left[1 + \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) b_{\boldsymbol{\Phi}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}\right].$$

Application to a Simple Imaging System

As a concrete example, we consider again the imaging system of Figure 4.12 along with the same assumptions we made in § 4.6.2, which we repeat below for convenience

$$\mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) = \mathrm{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{r}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \boldsymbol{r}|^2}{2\sigma^2}\right),$$
$$b_{\boldsymbol{\Phi}}(\boldsymbol{r}) = \frac{b_0}{2\pi r_b^2} \sum_{k=1}^{K_{\max}} \beta_k \exp\left(-\frac{|\boldsymbol{r} - \boldsymbol{r}_k|^2}{2r_b^2}\right),$$
$$\Delta f(\boldsymbol{r}) = \frac{b_s}{2\pi r_s^2} \exp\left(-\frac{|\boldsymbol{r}|^2}{2r_s^2}\right),$$

and $s(\boldsymbol{r}) = 1$. We will further assume that the field of view is $\mathrm{FOV} = [a, b] \times [c, d]$, for $a = c = -W/2$ and $b = d = W/2$. For the problem at hand:

$$\frac{\mathrm{pr}(\boldsymbol{\Phi}' \mid \hat{\mathscr{A}}, H_0)}{\mathrm{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_0)} = \frac{(pW^2)^{K_1'}}{K_1'!} \exp\left\{ - \frac{b_0\tau}{4} \sum_{k=1}^{K_{\max}} \beta_k' \left[\mathrm{erf}\left(\frac{b - x_k'}{\sqrt{2} r_b}\right) - \mathrm{erf}\left(\frac{a - x_k'}{\sqrt{2} r_b}\right)\right] \times \right.$$
$$\left. \times \left[\mathrm{erf}\left(\frac{d - y_k'}{\sqrt{2} r_b}\right) - \mathrm{erf}\left(\frac{c - y_k'}{\sqrt{2} r_b}\right)\right] \right\} \prod_{j=1}^{J} \sum_{k=1}^{K_{\max}} \beta_k' \exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}_k'|^2}{2(\sigma^2 + r_b^2)}\right) \times$$
$$\times \left[\frac{(pW^2)^{K_1}}{K_1!}\right]^{-1} \exp\left\{ \frac{b_0\tau}{4} \sum_{k=1}^{K_{\max}} \beta_k \left[\mathrm{erf}\left(\frac{b - x_k}{\sqrt{2} r_b}\right) - \mathrm{erf}\left(\frac{a - x_k}{\sqrt{2} r_b}\right)\right] \times \right.$$
$$\left. \times \left[\mathrm{erf}\left(\frac{d - y_k}{\sqrt{2} r_b}\right) - \mathrm{erf}\left(\frac{c - y_k}{\sqrt{2} r_b}\right)\right] \right\} \left[\prod_{j=1}^{J} \sum_{k=1}^{K_{\max}} \beta_k \exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}_k|^2}{2(\sigma^2 + r_b^2)}\right)\right]^{-1},$$

while the background-known-exactly (BKE) likelihood ratio takes the form

$$\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}) = \prod_{j=1}^{J} \left[1 + \frac{\dfrac{b_s}{2\pi(\sigma^2 + r_s^2)} \exp\left(-\dfrac{|\hat{\boldsymbol{R}}^{(j)}|^2}{2(\sigma^2 + r_s^2)}\right)}{\dfrac{b_0}{2\pi(\sigma^2 + r_b^2)} \displaystyle\sum_{k=1}^{K_{\max}} \beta_k \exp\left(-\dfrac{|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}_k|^2}{2(\sigma^2 + r_b^2)}\right)}\right] \times$$

$$\times \exp\left\{-\frac{b_s\tau}{4}\left[\mathrm{erf}\left(\frac{b}{\sqrt{2}r_s}\right)-\mathrm{erf}\left(\frac{a}{\sqrt{2}r_s}\right)\right]\left[\mathrm{erf}\left(\frac{d}{\sqrt{2}r_s}\right)-\mathrm{erf}\left(\frac{c}{\sqrt{2}r_s}\right)\right]\right\},$$

where we assumed a signal of the form given in (4.27).

Pseudocode for the generation of T samples $\Lambda^{(1)}_{|H_0},\ldots,\Lambda^{(T)}_{|H_0}$ of $\Lambda(\hat{\mathscr{A}})$ for different realizations of $\hat{\mathscr{A}}$ under the hypothesis H_0 and for the MCMC estimation of $\hat{\Lambda}(\hat{\mathscr{A}})$ is summarized below:

```
$t = 1$
for $r = 1$ to $R$ do
  Generate $\mathbf{\Phi}^{(r)}_{\mathrm{true}}$
  for $s = 1$ to $S$ do
    Generate $\hat{\mathscr{A}}^{(r,s)}$ from $\boldsymbol{f} = \boldsymbol{b}_{\mathbf{\Phi}^{(r)}_{\mathrm{true}}}$
    Initialize $\mathbf{\Phi}^{(0)}$
    for $n = 1$ to $N$ do
      Sample $\mathbf{\Phi}'$ from $q(\mathbf{\Phi}' \mid \mathbf{\Phi}^{(n)})$
      $P_{\mathrm{acc}} = \min\{1, \mathrm{pr}(\mathbf{\Phi}' \mid \hat{\mathscr{A}}^{(r,s)}, H_0)q(\mathbf{\Phi} \mid \mathbf{\Phi}')/\mathrm{pr}(\mathbf{\Phi} \mid \hat{\mathscr{A}}^{(r,s)}, H_0)q(\mathbf{\Phi}' \mid \mathbf{\Phi})\}$
      Sample $u$ from $U \sim \mathcal{U}([0,1])$
      if $u < P_{\mathrm{acc}}$ then
        $\mathbf{\Phi}^{(n)} = \mathbf{\Phi}'$
      else
        $\mathbf{\Phi}^{(n)} = \mathbf{\Phi}^{(n-1)}$
      end if
    end for
    $\Lambda^{(t)}_{|H_0} = \frac{1}{N}\sum_{n=1}^{N} \Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}}^{(r,s)} \mid \mathbf{\Phi}^{(n)})$
    $t = t + 1$
  end for
end for
```

In the pseudocode above, R random lumpy backgrounds $\boldsymbol{b}_{\mathbf{\Phi}^{(1)}_{\mathrm{true}}},\ldots,\boldsymbol{b}_{\mathbf{\Phi}^{(R)}_{\mathrm{true}}}$ are generated by generating $\mathbf{\Phi}^{(1)}_{\mathrm{true}},\ldots,\mathbf{\Phi}^{(R)}_{\mathrm{true}}$. For each $\boldsymbol{b}_{\mathbf{\Phi}^{(r)}_{\mathrm{true}}}$, S samples $\hat{\mathscr{A}}^{(r,1)},\ldots,\hat{\mathscr{A}}^{(r,S)}$ of $\hat{\mathscr{A}}$ are generated. From each $\hat{\mathscr{A}}^{(r,s)}$, a Markov chain is constructed and samples $\mathbf{\Phi}^{(1)},\ldots,\mathbf{\Phi}^{(N)}$ are used to calculate $\Lambda^{(t)}_{|H_0}$ according to (4.39). Index t goes from 1 to T, where $T = RS$. Hence, the pseudocode above implements

$$\langle\Lambda(\hat{\mathscr{A}})\rangle_{\hat{\mathscr{A}}|H_0} \approx \frac{1}{T}\sum_{t=1}^{T}\Lambda^{(t)}_{|H_0} \approx \frac{1}{RS}\sum_{r=1}^{R}\sum_{s=1}^{S}\underbrace{\left[\frac{1}{N}\sum_{n=1}^{N}\Lambda_{\mathrm{BKE}}(\hat{\mathscr{A}}^{(r,s)} \mid \mathbf{\Phi}^{(n)})\right]}_{\text{use MCMC}}.$$

Realizations of $\Lambda(\hat{\mathscr{A}})$ under the hypothesis H_1 could, in theory, be calculated using (4.39) and the same code with the only difference that, this time, list-mode data $\hat{\mathscr{A}}^{(r,s)}$ would be generated assuming $\boldsymbol{f} = \boldsymbol{b}_{\boldsymbol{\Phi}^{(r)}_{\text{true}}} + \Delta\boldsymbol{f}$, in which $\Delta\boldsymbol{f}$ is the signal to detect. However, it has been remarked [299] that numerical inaccuracies might arise if the same MCMC code we provided above is used to generate realizations of $\Lambda(\hat{\mathscr{A}})$ under the hypothesis H_1. Indeed, in the code above, acceptance probabilities are calculated using conditional densities of the form $\text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}^{(r,s)}, H_0)$ and, if the same code is used to generate realizations of $\Lambda(\hat{\mathscr{A}})$ under the hypothesis H_1, such densities will be evaluated for list-mode data $\hat{\mathscr{A}}^{(r,s)}$ calculate assuming the signal present. A more numerically accurate algorithm will use $\text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}^{(r,s)}, H_1)$ instead, along with the necessary modifications to the code above.

By the same process that led to (4.39), we can show that [299]

$$\frac{1}{\Lambda(\hat{\mathscr{A}})} = \int_\infty \frac{1}{\Lambda_{\text{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi})} \text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_1)\, \text{d}\boldsymbol{\Phi}, \tag{4.43}$$

so that

$$\hat{\Lambda}(\hat{\mathscr{A}}) = \left[\frac{1}{N} \sum_{n=1}^{N} \frac{1}{\Lambda_{\text{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{\Phi}^{(n)})}\right]^{-1},$$

where $\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(N)}$ are samples distributed according to $\text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_1)$ and they are obtained via MCMC methods. The acceptance probability now assumes the form:

$$P_{\text{acc}}(\boldsymbol{\Phi}', \boldsymbol{\Phi}; \hat{\mathscr{A}}, H_1) = \min\left\{1, \frac{\text{pr}(\boldsymbol{\Phi}' \mid \hat{\mathscr{A}}, H_1) q(\boldsymbol{\Phi} \mid \boldsymbol{\Phi}')}{\text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_1) q(\boldsymbol{\Phi}' \mid \boldsymbol{\Phi})}\right\}, \tag{4.44}$$

where, for the case of the signal given in (4.27):

$$\frac{\text{pr}(\boldsymbol{\Phi}' \mid \hat{\mathscr{A}}, H_1)}{\text{pr}(\boldsymbol{\Phi} \mid \hat{\mathscr{A}}, H_1)} = \frac{(pW^2)^{K_1'}}{K_1'!} \left[\frac{(pW^2)^{K_1}}{K_1!}\right]^{-1} \times$$

$$\times \exp\left\{-\frac{b_0\tau}{4} \sum_{k=1}^{K_{\max}} \beta_k' \left[\text{erf}\left(\frac{b - x_k'}{\sqrt{2}r_b}\right) - \text{erf}\left(\frac{a - x_k'}{\sqrt{2}r_b}\right)\right] \left[\text{erf}\left(\frac{d - y_k'}{\sqrt{2}r_b}\right) - \text{erf}\left(\frac{c - y_k'}{\sqrt{2}r_b}\right)\right]\right\} \times$$

$$\times \prod_{j=1}^{J} \left[\frac{b_0}{\sigma^2 + r_b^2} \sum_{k=1}^{K_{\max}} \beta_k' \exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}_k'|^2}{2(\sigma^2 + r_b^2)}\right) + \frac{b_s}{\sigma^2 + r_s^2} \exp\left(\frac{|\hat{\boldsymbol{R}}^{(j)}|^2}{2(\sigma^2 + r_s^2)}\right)\right] \times$$

$$\times \exp\left\{\frac{b_0\tau}{4}\sum_{k=1}^{K_{\max}}\beta_k\left[\operatorname{erf}\left(\frac{b-x_k}{\sqrt{2}r_b}\right)-\operatorname{erf}\left(\frac{a-x_k}{\sqrt{2}r_b}\right)\right]\left[\operatorname{erf}\left(\frac{d-y_k}{\sqrt{2}r_b}\right)-\operatorname{erf}\left(\frac{c-y_k}{\sqrt{2}r_b}\right)\right]\right\}\times$$
$$\times\left\{\prod_{j=1}^{J}\left[\frac{b_0}{\sigma^2+r_b^2}\sum_{k=1}^{K_{\max}}\beta_k\exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)}-\boldsymbol{r}_k|^2}{2(\sigma^2+r_b^2)}\right)+\frac{b_s}{\sigma^2+r_s^2}\exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)}|^2}{2(\sigma^2+r_s^2)}\right)\right]\right\}^{-1}.$$

Consistency Checks of MCMC Simulation

The particular mathematical form of the likelihood ratio $\Lambda(\hat{\mathscr{A}})$ makes it possible to derive many relevant theoretical results in the form of equalities and inequalities. These results, in turn, can be used to test any algorithm for the generation of samples of $\Lambda(\hat{\mathscr{A}})$. We begin by defining moments $M_k(\beta)$ of the random variable $\Lambda(\hat{\mathscr{A}})$ under the hypothesis H_k [65]

$$M_k(\beta) = \langle e^{\lambda\beta}\rangle_{H_k} = \langle \Lambda^\beta\rangle_{H_k} = \int_\infty \Lambda^\beta(\hat{\mathscr{A}})\mathrm{pr}(\hat{\mathscr{A}} \mid H_k)\,\mathrm{d}\hat{\mathscr{A}}, \tag{4.45}$$

in which β can also be a complex number [65]. By recalling that

$$\Lambda(\hat{\mathscr{A}}) = \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)}, \tag{4.46}$$

we can easily relate moments of $\lambda(\hat{\mathscr{A}})$ under H_0 to its moments under H_1 [65]

$$\begin{aligned} M_0(\beta+1) = \int_\infty \Lambda^{\beta+1}(\hat{\mathscr{A}})\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)\,\mathrm{d}\hat{\mathscr{A}} = \\ = \int_\infty \Lambda^{\beta}(\hat{\mathscr{A}})\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)\,\mathrm{d}\hat{\mathscr{A}} = \langle\Lambda^\beta\rangle_{H_1} = M_1(\beta). \end{aligned}$$

Special cases are obtained for $\beta = 0$ and $\beta = -1$ [65, 300]:

$$1 = \langle\Lambda^0\rangle_{H_1} = M_1(0) = M_0(1) = \langle\Lambda\rangle_{H_0}, \tag{4.47a}$$
$$1 = \langle\Lambda^0\rangle_{H_0} = M_0(0) = M_1(-1) = \langle\tfrac{1}{\Lambda}\rangle_{H_1}. \tag{4.47b}$$

The two identities above already give two consistency checks on samples $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$

$$\frac{1}{T}\sum_{t=1}^{T}\Lambda_{|H_0}^{(t)} \approx 1, \qquad \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\Lambda_{|H_1}^{(t)}} \approx 1.$$

Two more consistency checks are obtained by taking the second derivative of $M_0(\beta)$ and $M_1(\beta)$ [300]

$$\frac{\mathrm{d}^2 M_k(\beta)}{\mathrm{d}\beta^2} = \langle \lambda^2 \Lambda^\beta \rangle_{H_k} > 0.$$

Moments $M_k(\beta)$ can be written in terms of the so-called "likelihood generating function" $G(\beta)$ [65]:

$$\begin{aligned} \langle \Lambda^\beta \rangle_{H_0} &= M_0(\beta) = \exp\left[\beta(\beta-1)G\left(\beta - \tfrac{1}{2}\right)\right], \\ \langle \Lambda^\beta \rangle_{H_1} &= M_1(\beta) = \exp\left[\beta(\beta+1)G\left(\beta + \tfrac{1}{2}\right)\right], \end{aligned} \tag{4.48}$$

where [65]

$$G(\beta) = \frac{\ln M_0\left(\beta+\frac{1}{2}\right)}{\left(\beta-\frac{1}{2}\right)\left(\beta+\frac{1}{2}\right)} = \frac{\ln M_1\left(\beta-\frac{1}{2}\right)}{\left(\beta-\frac{1}{2}\right)\left(\beta+\frac{1}{2}\right)}, \qquad \beta \neq \pm\tfrac{1}{2}.$$

The expression above does not allow to calculate $G(\pm\frac{1}{2})$. However, we can consider

$$\lim_{\beta\to\frac{1}{2}} G(\beta) = \lim_{\beta\to 0} G\left(\beta+\tfrac{1}{2}\right) = \lim_{\beta\to 0}\frac{\ln M_1(\beta)}{\beta(\beta+1)},$$

and

$$\lim_{\beta\to-\frac{1}{2}} G(\beta) = \lim_{\beta\to 0} G\left(\beta-\tfrac{1}{2}\right) = \lim_{\beta\to 0}\frac{\ln M_0(\beta)}{\beta(\beta-1)}.$$

By application of l'Hôpital's rule:

$$\lim_{\beta\to\frac{1}{2}} G(\beta) = \lim_{\beta\to 0}\frac{\int_\infty \lambda(\hat{\mathscr{A}})\Lambda^\beta(\hat{\mathscr{A}})\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)\,\mathrm{d}\hat{\mathscr{A}}}{2\beta+1} = \langle\lambda\rangle_{H_1},$$

and

$$\lim_{\beta\to-\frac{1}{2}} G(\beta) = \lim_{\beta\to 0}\frac{\int_\infty \lambda(\hat{\mathscr{A}})\Lambda^\beta(\hat{\mathscr{A}})\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)\,\mathrm{d}\hat{\mathscr{A}}}{2\beta-1} = -\langle\lambda\rangle_{H_0},$$

respectively. We can thus extend $G(\beta)$ to a continuous function defined on the real line by setting [65]

$$G\left(-\tfrac{1}{2}\right) = -\langle\lambda\rangle_{H_0}, \qquad G\left(\tfrac{1}{2}\right) = \langle\lambda\rangle_{H_1}. \tag{4.49}$$

In a similar way

$$G'\left(-\tfrac{1}{2}\right) = -\tfrac{1}{2}\sigma^2_{\lambda|H_0} - \langle\lambda\rangle_{H_0}, \qquad G'\left(\tfrac{1}{2}\right) = \tfrac{1}{2}\sigma^2_{\lambda|H_1} - \langle\lambda\rangle_{H_1},$$

where $G'(\beta)$ denotes the first derivative of the function $G(\beta)$. We have now all the ingredients needed to write an exact expression for SNR^2_λ in terms of $G(\beta)$. By (2.4):

$$\mathrm{SNR}^2_\lambda = \frac{\left[G\left(\frac{1}{2}\right) + G\left(-\frac{1}{2}\right)\right]^2}{G\left(\frac{1}{2}\right) + G\left(-\frac{1}{2}\right) + G'\left(\frac{1}{2}\right) - G'\left(-\frac{1}{2}\right)}.$$

If $G'(\frac{1}{2}) \approx G'(-\frac{1}{2})$ (or if $G'(\frac{1}{2})$ and $G'(-\frac{1}{2})$ are both very small when compared to $G(\frac{1}{2})$ and $G(-\frac{1}{2})$), an excellent approximation is [65, 301]

$$\mathrm{SNR}^2_\lambda \approx 2G(0) = -8 \ln M_0\left(\tfrac{1}{2}\right) = -8 \ln \left\langle \sqrt{\Lambda} \right\rangle_{H_0}. \tag{4.50}$$

By Jensen's inequality [185] applied to (4.48) and using (4.49):

$$\begin{aligned} \left(\beta - \tfrac{1}{2}\right) G(\beta) &\geqslant -G\left(-\tfrac{1}{2}\right), & \beta &\geqslant -\tfrac{1}{2}, \\ \left(\beta + \tfrac{1}{2}\right) G(\beta) &\geqslant G\left(\tfrac{1}{2}\right), & \beta &\geqslant \tfrac{1}{2}. \end{aligned}$$

Similarly [65] and from (4.45):

$$\ln M_k(\beta) \geqslant \beta \langle \lambda \rangle_{H_k}, \qquad \beta \geqslant 0.$$

As special cases

$$\langle \lambda \rangle_{H_0} \leqslant \ln M_0(1) = \ln 1 = 0, \tag{4.51a}$$

$$\langle \lambda \rangle_{H_1} \leqslant \ln M_1(1) = \ln e^{2G(3/2)} = 2G\left(\tfrac{3}{2}\right). \tag{4.51b}$$

Calculation of the AUC from realizations of $\Lambda(\hat{\mathscr{A}})$ (or $\lambda(\hat{\mathscr{A}})$) under hypotheses H_0 and H_1 can be performed in a variety of different ways. We will use Roman numeral subscripts to denote values of the AUC calculate via different relationship, as in $\mathrm{AUC}_{\mathrm{I}}, \ldots, \mathrm{AUC}_{\mathrm{VI}}$.

By definition, the AUC is given by the area under the ROC curve [65, 69, 74, 85]

$$\mathrm{AUC}_{\mathrm{I}} = \int_0^1 \mathrm{TPF}(\mathrm{FPF})\, \mathrm{dFPF},$$

where

$$\mathrm{TPF}(\Lambda) = \int_\Lambda^\infty \mathrm{pr}(\Lambda' \mid H_1)\, \mathrm{d}\Lambda',$$

$$\mathrm{FPF}(\Lambda) = \int_{\Lambda}^{\infty} \mathrm{pr}(\Lambda' \mid H_0)\, \mathrm{d}\Lambda'.$$

Quantities $\mathrm{TPF}(\Lambda)$ and $\mathrm{FPF}(\Lambda)$ above are estimated from the sets of realizations $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$. An alternative method for the calculation of AUC uses the fact that, for the likelihood ratio [69, 301, 302]

$$\mathrm{AUC}_{\mathrm{II}} = 1 - \frac{1}{2} \int_{0}^{\infty} \big[\mathrm{FPF}(\Lambda)\big]^2 \, \mathrm{d}\Lambda.$$

Further elaboration on the identity above gives an expression for the AUC in terms of $M_0(\beta)$ [65, 300]:

$$\mathrm{AUC}_{\mathrm{III}} = 1 - \frac{1}{2\pi} \int_{0}^{\infty} \Big| M_0\left(\tfrac{1}{2} + i\alpha\right) \Big|^2 \frac{1}{\alpha^2 + \frac{1}{4}} \, \mathrm{d}\alpha. \tag{4.52}$$

The integrand in the expression above goes to zero as $\alpha \to \infty$ [300]. The identity above can also be used to derive a lower bound on the AUC [65]

$$\begin{aligned} \big| M_0\left(\tfrac{1}{2} + i\alpha\right) \big| &= \big| \langle \Lambda^{1/2} \Lambda^{i\alpha} \rangle_{H_0} \big| \leqslant \langle |\Lambda^{1/2} \Lambda^{i\alpha}| \rangle_{H_0} = \langle \Lambda^{1/2} |\Lambda^{i\alpha}| \rangle_{H_0} = \\ &= \langle \Lambda^{1/2} \rangle_{H_0} = M_0\left(\tfrac{1}{2}\right) = \exp\left[-\tfrac{1}{4} G(0)\right], \end{aligned}$$

which implies [65]

$$\mathrm{AUC} \geqslant 1 - \tfrac{1}{2} \exp\left[-\tfrac{1}{2} G(0)\right].$$

The AUC can also be calculated via a two-alternative forced-choice (2AFC) study [65, 74, 92, 114]. Indeed, by recalling that the AUC is also the probability of correct classification in a 2AFC experiment [74, 92]

$$\mathrm{AUC} = \Pr\big(\Lambda_{|H_1} \geqslant \Lambda_{|H_0}\big),$$

and so we can estimate the AUC from realizations $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$ as [303, 304]

$$\mathrm{AUC}_{\mathrm{IV}} = \frac{\big|\{(t_0, t_1) \text{ such that } \Lambda_{|H_0}^{(t_0)} \leqslant \Lambda_{|H_1}^{(t_1)} \text{ for } t_0 = 1, \ldots, T \text{ and } t_1 = 1, \ldots, T\}\big|}{T^2} =$$

$$= \frac{1}{T^2} \sum_{t_0=1}^{T} \sum_{t_1=1}^{T} \text{step}\left(\Lambda_{|H_1}^{(t_1)} - \Lambda_{|H_0}^{(t_0)}\right),$$

where

$$\text{step}(x) = \begin{cases} 1 & \text{if } x \geqslant 0, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we can use $\text{SNR}_\lambda^2 \approx 2G(0)$ and (2.5) to write [65]

$$\text{AUC}_\text{V} = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\sqrt{2G(0)}}{2}\right),$$

of which a more accurate version is [65]

$$\text{AUC}_\text{VI} = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\sqrt{2G(0)}}{2}\right) - \frac{G''(0)}{16\sqrt{2\pi}} \exp\left(-\frac{G(0)}{2}\right) \left[G(0)\right]^{-3/2},$$

in which $G''(0)$ is the second derivative of $G(\beta)$ evaluated at $\beta = 0$:

$$G''(0) = 4 \left[\left(\frac{\langle \lambda \sqrt{\Lambda} \rangle_{H_0}}{M_0\left(\frac{1}{2}\right)} \right)^2 - \frac{\langle \lambda^2 \sqrt{\Lambda} \rangle_{H_0}}{M_0\left(\frac{1}{2}\right)} + 2G(0) \right], \tag{4.53}$$

In theory, quantities $\text{AUC}_\text{I}, \ldots, \text{AUC}_\text{IV}$ all represent the same numerical value [65], as they all are exact expressions for the same quantity—namely, the AUC for the likelihood ratio test statistic—but calculated in different ways. On the other hand, AUC_V and AUC_VI represent approximate quantities, and become exact expression for the case of normally distributed log-likelihoods [65].

A few bounds on the AUC, which we will list without providing any proof, can be summarized as [302]:

$$1 - \tfrac{1}{2} \exp\left[-\tfrac{1}{2} G(0)\right] \leqslant \text{AUC} \leqslant 1 - \tfrac{1}{2} \exp\left[-\tfrac{1}{2} G(0) - \sqrt{G(0) - \tfrac{1}{8} G''(0)}\right], \tag{4.54a}$$

$$1 - \tfrac{1}{2} \exp\left[-\tfrac{1}{4} G(0)\right] \leqslant \text{AUC} \leqslant 1 - \tfrac{1}{2} \exp\left[-\tfrac{1}{2} G(0) - \sqrt{2G(0) + \tfrac{1}{16}\left[G'(0)\right]^2 - \tfrac{1}{4} G''(0)}\right], \tag{4.54b}$$

and [305]

$$1 - P_e \leqslant \text{AUC} \leqslant 1 - 2P_e^2, \tag{4.54c}$$

where P_e is the minimum error probability for equally likely hypotheses [302, 305]:

$$\begin{aligned} P_e &= \tfrac{1}{2}\min_{\Lambda\geqslant 0}\Big\{\mathrm{FPF}(\Lambda)+\mathrm{FNF}(\Lambda)\Big\} = \\ &= \tfrac{1}{2}\min_{\Lambda\geqslant 0}\Big\{1+\mathrm{FPF}(\Lambda)-\mathrm{TPF}(\Lambda)\Big\} = \\ &= \tfrac{1}{2}\Big[1+\mathrm{FPF}(1)-\mathrm{TPF}(1)\Big]. \end{aligned}$$

With some simple but tedious calculations, the argument of the square root in (4.54a) turns out to be

$$G(0)-\tfrac{1}{8}G''(0)=\tfrac{1}{2}\left[\langle\lambda^2\rangle_{H_{1/2}}-\Big(\langle\lambda\rangle_{H_{1/2}}\Big)^2\right]=\tfrac{1}{2}\sigma^2_{\lambda|H_{1/2}},$$

where we have denoted as $\langle\ldots\rangle_{H_\beta}$ the expectation with respect the probability density function [306, 307]

$$\mathrm{pr}(\hat{\mathscr{A}}\mid H_\beta)=\frac{\mathrm{pr}^{1-\beta}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{pr}^{\beta}(\hat{\mathscr{A}}\mid H_1)}{M_0(\beta)}.$$

We remark that $\mathrm{pr}(\hat{\mathscr{A}}\mid H_\beta)$ above is indeed a probability density function because it is a continuous and non-negative function and it satisfies $\int_\infty \mathrm{pr}(\hat{\mathscr{A}}\mid H_\beta)\,\mathrm{d}\hat{\mathscr{A}}=1$:

$$\begin{aligned} \int_\infty \mathrm{pr}(\hat{\mathscr{A}}\mid H_\beta)\,\mathrm{d}\hat{\mathscr{A}} &= \frac{1}{M_0(\beta)}\int_\infty \mathrm{pr}^{1-\beta}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{pr}^{\beta}(\hat{\mathscr{A}}\mid H_1)\,\mathrm{d}\hat{\mathscr{A}} = \\ &= \frac{\int_\infty \mathrm{pr}^{1-\beta}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{pr}^{\beta}(\hat{\mathscr{A}}\mid H_1)\,\mathrm{d}\hat{\mathscr{A}}}{\int_\infty \mathrm{pr}^{1-\beta}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{pr}^{\beta}(\hat{\mathscr{A}}\mid H_1)\,\mathrm{d}\hat{\mathscr{A}}} = 1, \end{aligned}$$

where we have used (4.45) and (4.46) to get

$$\begin{aligned} M_0(\beta) &= \int_\infty \left[\frac{\mathrm{pr}(\hat{\mathscr{A}}\mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}}\mid H_0)}\right]^{\beta}\mathrm{pr}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{d}\hat{\mathscr{A}} = \\ &= \int_\infty \mathrm{pr}^{1-\beta}(\hat{\mathscr{A}}\mid H_0)\,\mathrm{pr}^{\beta}(\hat{\mathscr{A}}\mid H_1)\,\mathrm{d}\hat{\mathscr{A}}. \end{aligned}$$

The argument of the square root in (4.54b) simplifies to

$$2G(0)+\tfrac{1}{16}\big[G'(0)\big]^2-\tfrac{1}{4}G''(0)=\langle\lambda^2\rangle_{H_{1/2}}.$$

Simulation Results

Armed with the theoretical background we just developed, we wrote and ran simulation code to calculate realizations $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$ of the likelihood ratio via MCMC sampling (§ 4.5.2). We considered the same cases we had in the calculation of $\mathrm{SNR}_{\mathrm{Hot}}^2$ (see § 4.6.2 and Figure 4.14). In other words, we run our code three times but with $r_s = 4.50$ mm, $r_s = 5.00$ mm, and $r_s = 5.50$ mm, respectively. Other relevant parameters are listed in Table 4.2.

For each value of r_s, the exposure time τ was varied between 0.002 s to 0.040 s. Exposure time and mean total number $\overline{J}(\boldsymbol{f}, \tau)$ of attributes in the list $\hat{\mathscr{A}}$ are proportionally related with each other as follows:

$$\overline{J}(\boldsymbol{f}, \tau) = \tau \int_{\mathrm{FOV}} f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r},$$

where $f(\boldsymbol{r})$ is the object being imaged. Hence, the average size of the list $\hat{\mathscr{A}}$ increases linearly with the exposure time τ. Simulation parameters were tweaked in a way that the mean acceptance probability P_{acc} as calculated during execution of the MCMC code was neither too small not too large. Indeed, a small value of the acceptance probability means that the chain is not exploring much of the space and that just a few points will represent much of the integration space. Conversely, a large value of the acceptance probability can imply that the proposals were made too close to the current value, so that the ratio of densities in the acceptance function is close to 1 (see (4.40) and (4.44)). In practice, the expected total number of flips $K_{\max}\eta$ performed to calculate $\boldsymbol{\Phi}'$ from $\boldsymbol{\Phi}$ was adjusted so that the mean acceptance probability was around 35 %, as shown in Table 4.3.

Samples $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$ can be checked for consistency using some of the results derived above. In particular, Table 4.4 shows the results of consistency checks (4.47) and (4.51) for the $r_s = 4.50$ mm case. Similarly, Table 4.5 corresponds to the $r_s = 5.00$ mm case and Table 4.6 to the $r_s = 5.50$ mm case. In these tables, a checkmark symbol ("✓") indicates whether the consistency check was

Parameter	**Value**
Number of lumpy background images (R)	100
Number of lists $\hat{\mathscr{A}}$ per image (S)	50
Chain length (N)	10000
Signal width (r_s)	4.50, 5.00, or 5.50 mm
Signal amplitude (b_s)	1000 s^{-1}
Lump width (r_b)	5.00 mm
Lump amplitude (b_0)	1000 s^{-1}
Mean lump density (p)	20000 m^{-2}
Mean number of lumps (pW^2)	81.92
Maximum number of lumps (K_{max})	1000
FOV size ($W \times W$)	64 mm × 64 mm
Uncertainty in estimating $\hat{\boldsymbol{R}}$ (σ)	1.00 mm
Acquisition time (τ)	See Table 4.3
Expected number of flips ($K_{\mathrm{max}}\eta$)	See Table 4.3
Perturbation standard deviation ($\sigma_{\mathrm{pert.}}$)	2.50 mm
Number of bootstrap estimates for the calculation of AUC	512
Number of bootstrap estimates for the calculation of SNR_λ^2	8192

Table 4.2. Parameters used in the MCMC simulations

		$r_s = 4.50$ mm		**$r_s = 5.00$ mm**		**$r_s = 5.50$ mm**	
τ [s]	**$K_{\max}\eta$**	**$\%_{H_0}$**	**$\%_{H_1}$**	**$\%_{H_0}$**	**$\%_{H_1}$**	**$\%_{H_0}$**	**$\%_{H_1}$**
0.002	22.00	34.24	34.25	34.24	34.29	34.24	34.33
0.004	11.00	35.33	35.54	35.33	35.56	35.33	35.57
0.006	7.80	34.76	34.99	34.76	35.01	34.76	35.02
0.008	5.95	35.35	35.61	35.29	35.53	35.29	35.54
0.010	5.10	34.40	34.67	34.40	34.66	34.40	34.67
0.012	4.20	35.63	35.86	35.63	35.87	35.63	35.88
0.014	3.90	34.40	34.64	34.40	34.65	34.40	34.65
0.016	3.40	35.35	35.57	35.35	35.57	35.35	35.57
0.018	3.05	35.90	36.14	35.90	36.13	35.90	36.12
0.020	2.89	35.44	35.64	35.35	35.54	35.35	35.55
0.022	2.70	35.36	35.58	35.36	35.57	35.36	35.57
0.024	2.50	35.80	36.00	35.80	36.01	35.80	36.00
0.026	2.38	35.66	35.84	35.66	35.85	35.66	35.86
0.028	2.28	35.60	35.78	35.50	35.69	35.50	35.69
0.030	2.19	35.37	35.54	35.37	35.54	35.37	35.54
0.032	2.11	35.33	35.50	35.24	35.42	35.24	35.42
0.034	2.04	35.21	35.37	35.12	35.27	35.12	35.28
0.036	1.95	35.44	35.61	35.34	35.50	35.34	35.50
0.038	1.87	35.55	35.72	35.55	35.71	35.55	35.71
0.040	1.81	35.57	35.73	35.57	35.72	35.57	35.74

Table 4.3. Exposure times (τ), expected number of flips ($K_{\max}\eta$) and acceptance probabilities ($\%_{H_0}$, $\%_{H_1}$) for our MCMC simulation code for the $r_s = 4.50$ mm, $r_s = 5.00$ mm, and $r_s = 5.50$ mm cases

passed, whereas an "✗" mark denotes that the data failed the test. For consistency checks (4.47), we assumed a ±0.05 tolerance within the expected value. A tolerance of a few hundredths was also used in [300]. For consistency checks (4.51), we required the data to satisfy the inequality exactly. Some plots of $M_0(\beta)$ are reported in Figure 4.16. In general, we see that our MCMC code fails consistency checks (4.47) for small values of the signal width r_s and for calculation of samples of Λ under hypothesis H_1. The reason for this is likely to rely on the estimation of $\Lambda(\hat{\mathscr{A}})$ via the harmonic mean relation in (4.43).

Values of SNR_λ^2 can be calculated from the samples of the likelihood ratio, as (4.50) shows. Results for the $r_s = 4.50$ mm, $r_s = 5.00$ mm, and $r_s = 5.50$ mm cases are reported in Table 4.7, along with the values of the standard deviations estimated using a bootstrapping procedure (see Appendix B). Later in this section, we will comment on the behavior of SNR_λ^2 as a function of the exposure time τ.

Samples $\Lambda_{|H_0}^{(1)}, \ldots, \Lambda_{|H_0}^{(T)}$ and $\Lambda_{|H_1}^{(1)}, \ldots, \Lambda_{|H_1}^{(T)}$ also allow to estimate the AUC for the signal detection task. We have discussed above six different ways, $\mathrm{AUC}_{\mathrm{I}}, \ldots, \mathrm{AUC}_{\mathrm{VI}}$, to calculate the AUC from the samples of Λ under hypotheses H_0 and H_1. Table 4.8 reports the values of the AUC for the $r_s = 4.50$ mm case, Table 4.9 for the $r_s = 5.00$ mm case, and Table 4.10 for the $r_s = 5.50$ mm case. Some plots of ROC curves (which were used to calculate $\mathrm{AUC}_{\mathrm{I}}$) are shown in Figure 4.17, while plots of the integrand that appears in (4.52) are reported in Figure 4.18. The plots in Figure 4.18 show that, indeed, the quantity $|M_0(\frac{1}{2} + i\alpha)|^2 \frac{1}{\alpha^2 + \frac{1}{4}}$ goes to zero very rapidly as $\alpha \to \infty$ and that numerical integration for $\alpha \in [0, 5]$—instead of the whole set of positive real numbers—would give an excellent approximation for the calculation of $\mathrm{AUC}_{\mathrm{III}}$ in (4.52).

Consistency checks on the values of $\mathrm{AUC}_{\mathrm{I}}, \ldots, \mathrm{AUC}_{\mathrm{VI}}$ have been performed as well. For the $r_s = 4.50$ mm case, Table 4.11 reports consistency results for the bounds in (4.54a); Table 4.12 for the bounds in (4.54b), and Table 4.13 for the bounds in (4.54c). Similarly, Table 4.14 through Table 4.16 refer to the $r_s = 5.00$ mm case and Table 4.17 through Table 4.19 to the $r_s = 5.50$ mm case. As before, appropriate marks ("✓" or "✗") indicates whether or not bounds are satisfied. In general, we see that the estimated values of the AUC almost always satisfy theoretical bounds. Exceptions are for $\mathrm{AUC_{II}}$ (calculation of AUC as integral of the ROC curve) and for $\mathrm{AUC_{VI}}$ (higher order asymptotic expansion of AUC). The latter is probably due to numerical inaccuracies in estimation $G''(0)$ as in (4.53).

Similarly to SNR^2_λ, $\mathrm{AUC_I}, \ldots, \mathrm{AUC_{IV}}$ are estimated from realizations of the likelihood, so they are random quantities. Therefore, it makes sense to pair such estimates with their standard deviations. Standard deviations $s_{\mathrm{AUC_I}}, \ldots, s_{\mathrm{AUC_{IV}}}$ for $\mathrm{AUC_I}, \ldots, \mathrm{AUC_{IV}}$ have been calculated as byproduct of the bootstrapping procedure used to calculate the AUCs (see Appendix B), and have been reported in Table 4.20 for the $r_s = 4.50$ mm case, in Table 4.21 for the $r_s = 5.00$ mm case, and in Table 4.22 for the $r_s = 5.50$ mm case. An approximate theoretical expression for the variance of the AUC estimated via a 2AFC experiment has been reported in [92, 308]

$$\begin{aligned}\sigma^2_{\mathrm{2AFC}} &= \\ &= \frac{1}{T_0 T_1}\left\{A_z(1-A_z) + (T_1-1)A_z\left[\frac{1}{2-A_z} - A_z\right] + (T_1-1)A_z^2\left[\frac{2}{1+A_z} - 1\right]\right\},\end{aligned}$$

in which T_0 and T_1 denotes, respectively, the number of samples of Λ under the hypotheses H_0 and H_1, and we set $A_z = \mathrm{AUC_{IV}}$ for convenience. In our case, $T_0 = T_1 = T = RS = 5000$, as reported in Table 4.2. The value of σ^2_{2AFC} does depend on the underlining distribution of Λ under H_0 and H_1, but it has been found that σ^2_{2AFC} is almost entirely determined by A_z [92]. The values of σ_{2AFC} we reported in the last column of Table 4.20 through Table 4.22 are comparable to the corresponding values of $s_{\mathrm{AUC_{IV}}}$.

Other methods for the calculation of the standard deviation of AUC have been derived. For example, in [309, 310], the authors use the multiple-reader, multiple-case (MRMC) paradigm [311] and the Wilcoxon statistic [303, 304] as an unbiased estimator of $\mathrm{AUC}_{\mathrm{IV}}$, to derive an exact expression for the standard deviation of $\mathrm{AUC}_{\mathrm{IV}}$:

$$\sigma^2_{\mathrm{AUC}_{\mathrm{IV}}} = \frac{\alpha_1}{T_0} + \frac{\alpha_2}{T_1} + \frac{\alpha_3}{T_0 T_1} + \frac{\alpha_4}{T_r} + \frac{\alpha_5}{T_r T_0} + \frac{\alpha_6}{T_r T_1} + \frac{\alpha_7}{T_r T_0 T_1},$$

in which T_r denotes the number of different "readers" (a reader could be, for example, one of the radiologists examining images). Expressions for $\alpha_1, \ldots, \alpha_7$ are derived in [309] and practical techniques for their evaluation are discussed in [310, 312].

Finally, values of d_A^2 have been calculated from $\mathrm{AUC}_{\mathrm{I}}, \ldots, \mathrm{AUC}_{\mathrm{VI}}$ according to (2.6). The inverse $x = \mathrm{erf}^{-1}(y)$ of the $\mathrm{erf}(x)$ function that appears in (2.6) was first approximated according to [313]

$$\widetilde{x} = \mathrm{sgn}(y)\sqrt{\sqrt{\left(\frac{2}{\pi a} + \frac{\ln(1-y^2)}{2}\right)^2 - \frac{\ln(1-y^2)}{a}} - \left(\frac{2}{\pi a} + \frac{\ln(1-y^2)}{2}\right)},$$

where

$$a = \frac{8(\pi - 3)}{3\pi(4 - \pi)} \approx 0.14001228,$$

and

$$\mathrm{sgn}(y) = \begin{cases} +1 & \text{if } y > 0, \\ 0 & \text{if } y = 0, \\ -1 & \text{if } y < 0. \end{cases}$$

The solution $\widetilde{x}$ to $\mathrm{erf}(x) = y$ was further refined by performing one iteration of the Newton's method [154]:

$$x = \widetilde{x} - \frac{\mathrm{erf}\left(\widetilde{x}\right) - y}{\frac{2}{\sqrt{\pi}} \exp\left(-\widetilde{x}^2\right)}.$$

Table 4.23 reports values of d_A^2 for the $r_s = 4.50$ mm case, Table 4.24 for the $r_s = 5.00$ mm case and, finally, Table 4.25 for the $r_s = 5.50$ mm case. Values in these tables are found to be in accordance with those reported in Table 4.7.

$\boldsymbol{\tau}$ [s]	$\boldsymbol{\langle \Lambda \rangle_{H_0} = 1?}$	$\boldsymbol{\langle \frac{1}{\Lambda} \rangle_{H_1} = 1?}$	$\boldsymbol{\langle \lambda \rangle_{H_0} \leqslant 0?}$	$\boldsymbol{\langle \lambda \rangle_{H_1} \leqslant 2G\left(\frac{3}{2}\right)?}$
0.002	$0.9930 \approx 1$ ✓	$0.9803 \approx 1$ ✓	$-0.1227 \leqslant 0$ ✓	$0.1777 \leqslant 0.2050$ ✓
0.004	$0.9961 \approx 1$ ✓	$0.9836 \approx 1$ ✓	$-0.1955 \leqslant 0$ ✓	$0.2258 \leqslant 0.3311$ ✓
0.006	$1.0026 \approx 1$ ✓	$0.9565 \approx 1$ ✓	$-0.2487 \leqslant 0$ ✓	$0.2965 \leqslant 0.4287$ ✓
0.008	$1.0195 \approx 1$ ✓	$1.0217 \approx 1$ ✓	$-0.3080 \leqslant 0$ ✓	$0.2909 \leqslant 0.5621$ ✓
0.010	$1.0133 \approx 1$ ✓	$0.9760 \approx 1$ ✓	$-0.3290 \leqslant 0$ ✓	$0.2970 \leqslant 0.5804$ ✓
0.012	$1.0190 \approx 1$ ✓	$0.9697 \approx 1$ ✓	$-0.3714 \leqslant 0$ ✓	$0.3277 \leqslant 0.6916$ ✓
0.014	$1.0127 \approx 1$ ✓	$0.9668 \approx 1$ ✓	$-0.4187 \leqslant 0$ ✓	$0.3493 \leqslant 0.7557$ ✓
0.016	$1.0018 \approx 1$ ✓	$0.9960 \approx 1$ ✓	$-0.4727 \leqslant 0$ ✓	$0.3781 \leqslant 0.7706$ ✓
0.018	$1.0100 \approx 1$ ✓	$1.0149 \approx 1$ ✓	$-0.5068 \leqslant 0$ ✓	$0.4066 \leqslant 0.8297$ ✓
0.020	$1.0187 \approx 1$ ✓	$1.0689 \approx 1$ ✗	$-0.6285 \leqslant 0$ ✓	$0.3816 \leqslant 1.5845$ ✓
0.022	$1.0117 \approx 1$ ✓	$0.9672 \approx 1$ ✓	$-0.5940 \leqslant 0$ ✓	$0.4407 \leqslant 1.0606$ ✓
0.024	$1.0194 \approx 1$ ✓	$0.9675 \approx 1$ ✓	$-0.6540 \leqslant 0$ ✓	$0.4735 \leqslant 1.4835$ ✓
0.026	$1.0160 \approx 1$ ✓	$0.9744 \approx 1$ ✓	$-0.6929 \leqslant 0$ ✓	$0.4922 \leqslant 1.6111$ ✓
0.028	$0.9983 \approx 1$ ✓	$1.1215 \approx 1$ ✗	$-0.8660 \leqslant 0$ ✓	$0.4565 \leqslant 1.4346$ ✓
0.030	$1.0046 \approx 1$ ✓	$0.9634 \approx 1$ ✓	$-0.8098 \leqslant 0$ ✓	$0.5530 \leqslant 1.2537$ ✓
0.032	$0.9894 \approx 1$ ✓	$1.1090 \approx 1$ ✗	$-0.9923 \leqslant 0$ ✓	$0.5151 \leqslant 1.7561$ ✓
0.034	$1.0085 \approx 1$ ✓	$1.0460 \approx 1$ ✓	$-1.0441 \leqslant 0$ ✓	$0.5514 \leqslant 1.9427$ ✓
0.036	$0.9853 \approx 1$ ✓	$1.1979 \approx 1$ ✗	$-1.1292 \leqslant 0$ ✓	$0.5635 \leqslant 2.1728$ ✓
0.038	$0.9763 \approx 1$ ✓	$0.9708 \approx 1$ ✓	$-1.0629 \leqslant 0$ ✓	$0.6786 \leqslant 1.6042$ ✓
0.040	$0.9645 \approx 1$ ✓	$1.0007 \approx 1$ ✓	$-1.1203 \leqslant 0$ ✓	$0.7023 \leqslant 1.7292$ ✓

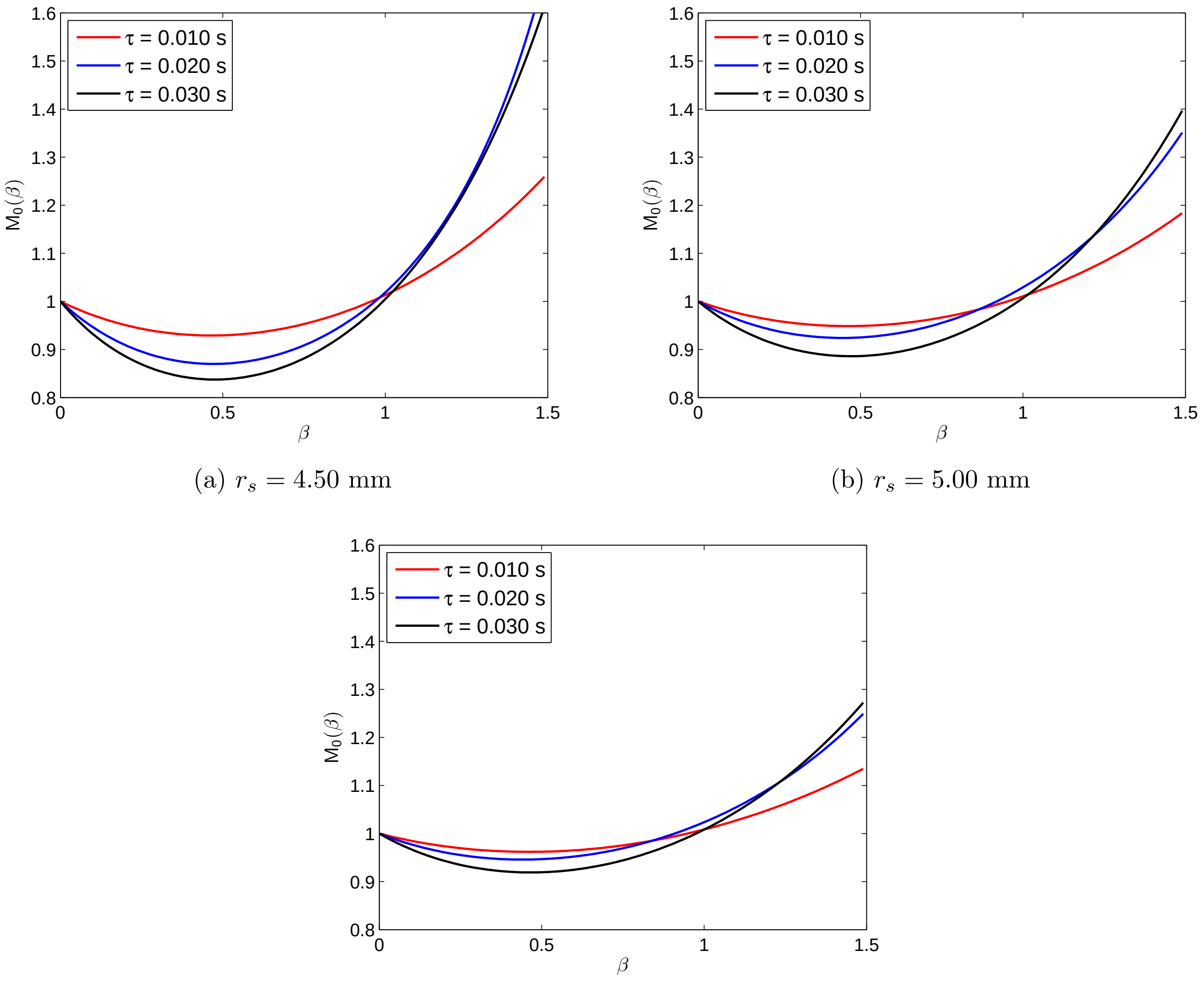
Table 4.4. Results of consistency checks for the $r_s = 4.50$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{\langle\Lambda\rangle_{H_0} = 1?}$	$\boldsymbol{\langle\frac{1}{\Lambda}\rangle_{H_1} = 1?}$	$\boldsymbol{\langle\lambda\rangle_{H_0} \leqslant 0?}$	$\boldsymbol{\langle\lambda\rangle_{H_1} \leqslant 2G\left(\frac{3}{2}\right)?}$
0.002	$0.9940 \approx 1$ ✓	$0.9782 \approx 1$ ✓	$-0.0975 \leqslant 0$ ✓	$0.1435 \leqslant 0.1595$ ✓
0.004	$0.9958 \approx 1$ ✓	$0.9844 \approx 1$ ✓	$-0.1502 \leqslant 0$ ✓	$0.1710 \leqslant 0.2480$ ✓
0.006	$1.0007 \approx 1$ ✓	$0.9595 \approx 1$ ✓	$-0.1864 \leqslant 0$ ✓	$0.2239 \leqslant 0.3172$ ✓
0.008	$1.0036 \approx 1$ ✓	$0.9473 \approx 1$ ✗	$-0.2115 \leqslant 0$ ✓	$0.2547 \leqslant 0.3636$ ✓
0.010	$1.0106 \approx 1$ ✓	$0.9799 \approx 1$ ✓	$-0.2362 \leqslant 0$ ✓	$0.2216 \leqslant 0.4230$ ✓
0.012	$1.0149 \approx 1$ ✓	$0.9776 \approx 1$ ✓	$-0.2630 \leqslant 0$ ✓	$0.2385 \leqslant 0.4891$ ✓
0.014	$1.0101 \approx 1$ ✓	$0.9709 \approx 1$ ✓	$-0.2947 \leqslant 0$ ✓	$0.2584 \leqslant 0.5284$ ✓
0.016	$1.0017 \approx 1$ ✓	$1.0007 \approx 1$ ✓	$-0.3315 \leqslant 0$ ✓	$0.2657 \leqslant 0.5391$ ✓
0.018	$1.0082 \approx 1$ ✓	$1.0175 \approx 1$ ✓	$-0.3526 \leqslant 0$ ✓	$0.2818 \leqslant 0.5859$ ✓
0.020	$1.0293 \approx 1$ ✓	$0.9969 \approx 1$ ✓	$-0.3753 \leqslant 0$ ✓	$0.3035 \leqslant 0.7948$ ✓
0.022	$1.0101 \approx 1$ ✓	$0.9862 \approx 1$ ✓	$-0.4080 \leqslant 0$ ✓	$0.3115 \leqslant 0.7054$ ✓
0.024	$1.0128 \approx 1$ ✓	$0.9702 \approx 1$ ✓	$-0.4504 \leqslant 0$ ✓	$0.3382 \leqslant 0.9276$ ✓
0.026	$1.0178 \approx 1$ ✓	$0.9870 \approx 1$ ✓	$-0.4740 \leqslant 0$ ✓	$0.3456 \leqslant 1.1615$ ✓
0.028	$1.0135 \approx 1$ ✓	$1.0072 \approx 1$ ✓	$-0.5117 \leqslant 0$ ✓	$0.3677 \leqslant 0.9724$ ✓
0.030	$1.0069 \approx 1$ ✓	$0.9509 \approx 1$ ✓	$-0.5558 \leqslant 0$ ✓	$0.3923 \leqslant 0.8719$ ✓
0.032	$1.0093 \approx 1$ ✓	$0.9759 \approx 1$ ✓	$-0.5986 \leqslant 0$ ✓	$0.4067 \leqslant 1.5004$ ✓
0.034	$1.0098 \approx 1$ ✓	$0.9949 \approx 1$ ✓	$-0.6473 \leqslant 0$ ✓	$0.4078 \leqslant 1.5367$ ✓
0.036	$1.0373 \approx 1$ ✓	$0.9570 \approx 1$ ✓	$-0.6732 \leqslant 0$ ✓	$0.4591 \leqslant 1.6601$ ✓
0.038	$0.9898 \approx 1$ ✓	$0.9924 \approx 1$ ✓	$-0.7347 \leqslant 0$ ✓	$0.4574 \leqslant 1.1558$ ✓
0.040	$0.9854 \approx 1$ ✓	$1.0331 \approx 1$ ✓	$-0.7742 \leqslant 0$ ✓	$0.4784 \leqslant 1.3200$ ✓

TABLE 4.5. Results of consistency checks for the $r_s = 5.00$ mm case

$\boldsymbol{\tau}$ [s]	$\langle\boldsymbol{\Lambda}\rangle_{\boldsymbol{H_0}} = \mathbf{1}?$	$\langle\frac{\mathbf{1}}{\boldsymbol{\Lambda}}\rangle_{\boldsymbol{H_1}} = \mathbf{1}?$	$\langle\boldsymbol{\lambda}\rangle_{\boldsymbol{H_0}} \leqslant \mathbf{0}?$	$\langle\boldsymbol{\lambda}\rangle_{\boldsymbol{H_1}} \leqslant \mathbf{2}\boldsymbol{G}\big(\frac{\mathbf{3}}{\mathbf{2}}\big)?$
0.002	$0.9949 \approx 1$ ✓	$0.9773 \approx 1$ ✓	$-0.0787 \leqslant 0$ ✓	$0.1180 \leqslant 0.1270$ ✓
0.004	$0.9959 \approx 1$ ✓	$0.9822 \approx 1$ ✓	$-0.1176 \leqslant 0$ ✓	$0.1371 \leqslant 0.1916$ ✓
0.006	$0.9997 \approx 1$ ✓	$0.9596 \approx 1$ ✓	$-0.1427 \leqslant 0$ ✓	$0.1799 \leqslant 0.2413$ ✓
0.008	$1.0024 \approx 1$ ✓	$0.9532 \approx 1$ ✓	$-0.1583 \leqslant 0$ ✓	$0.1975 \leqslant 0.2731$ ✓
0.010	$1.0086 \approx 1$ ✓	$0.9863 \approx 1$ ✓	$-0.1736 \leqslant 0$ ✓	$0.1633 \leqslant 0.3167$ ✓
0.012	$1.0119 \approx 1$ ✓	$0.9861 \approx 1$ ✓	$-0.1907 \leqslant 0$ ✓	$0.1748 \leqslant 0.3573$ ✓
0.014	$1.0078 \approx 1$ ✓	$0.9776 \approx 1$ ✓	$-0.2125 \leqslant 0$ ✓	$0.1904 \leqslant 0.3754$ ✓
0.016	$1.0014 \approx 1$ ✓	$0.9940 \approx 1$ ✓	$-0.2380 \leqslant 0$ ✓	$0.1977 \leqslant 0.3861$ ✓
0.018	$1.0063 \approx 1$ ✓	$1.0070 \approx 1$ ✓	$-0.2511 \leqslant 0$ ✓	$0.2125 \leqslant 0.4222$ ✓
0.020	$1.0236 \approx 1$ ✓	$0.9770 \approx 1$ ✓	$-0.2647 \leqslant 0$ ✓	$0.2257 \leqslant 0.5949$ ✓
0.022	$1.0095 \approx 1$ ✓	$0.9753 \approx 1$ ✓	$-0.2865 \leqslant 0$ ✓	$0.2329 \leqslant 0.4942$ ✓
0.024	$1.0095 \approx 1$ ✓	$0.9666 \approx 1$ ✓	$-0.3167 \leqslant 0$ ✓	$0.2479 \leqslant 0.6020$ ✓
0.026	$1.0158 \approx 1$ ✓	$0.9855 \approx 1$ ✓	$-0.3309 \leqslant 0$ ✓	$0.2499 \leqslant 0.7846$ ✓
0.028	$1.0126 \approx 1$ ✓	$0.9978 \approx 1$ ✓	$-0.3570 \leqslant 0$ ✓	$0.2713 \leqslant 0.6657$ ✓
0.030	$1.0083 \approx 1$ ✓	$0.9835 \approx 1$ ✓	$-0.3891 \leqslant 0$ ✓	$0.2765 \leqslant 0.6207$ ✓
0.032	$1.0086 \approx 1$ ✓	$0.9599 \approx 1$ ✓	$-0.4188 \leqslant 0$ ✓	$0.2982 \leqslant 1.0560$ ✓
0.034	$1.0086 \approx 1$ ✓	$0.9768 \approx 1$ ✓	$-0.4524 \leqslant 0$ ✓	$0.3043 \leqslant 1.0216$ ✓
0.036	$1.0337 \approx 1$ ✓	$0.9625 \approx 1$ ✓	$-0.4695 \leqslant 0$ ✓	$0.3203 \leqslant 1.2080$ ✓
0.038	$0.9972 \approx 1$ ✓	$0.9797 \approx 1$ ✓	$-0.5168 \leqslant 0$ ✓	$0.3363 \leqslant 0.8380$ ✓
0.040	$0.9962 \approx 1$ ✓	$0.9731 \approx 1$ ✓	$-0.5440 \leqslant 0$ ✓	$0.3535 \leqslant 0.9869$ ✓

Table 4.6. Results of consistency checks for the $r_s = 5.50$ mm case



(a) $r_s = 4.50$ mm

(b) $r_s = 5.00$ mm

(c) $r_s = 5.50$ mm

FIGURE 4.16. Plots of $M_0(\beta)$ for different values of τ. Theoretical results show that these plots should pass through points $(0, 1)$ and $(1, 1)$

	$\boldsymbol{r_s = 4.50}$ **mm**		$\boldsymbol{r_s = 5.00}$ **mm**		$\boldsymbol{r_s = 5.50}$ **mm**	
$\boldsymbol{\tau}$ [s]	$\mathbf{SNR}^2_{\boldsymbol{\lambda}}$	$s_{\mathbf{SNR}^2_{\boldsymbol{\lambda}}}$	$\mathbf{SNR}^2_{\boldsymbol{\lambda}}$	$s_{\mathbf{SNR}^2_{\boldsymbol{\lambda}}}$	$\mathbf{SNR}^2_{\boldsymbol{\lambda}}$	$s_{\mathbf{SNR}^2_{\boldsymbol{\lambda}}}$
0.002	0.2563	0.0276	0.2040	0.0244	0.1651	0.0218
0.004	0.3856	0.0354	0.2984	0.0308	0.2354	0.0270
0.006	0.4673	0.0405	0.3529	0.0347	0.2727	0.0302
0.008	0.5320	0.0454	0.3910	0.0372	0.2943	0.0321
0.010	0.5848	0.0471	0.4188	0.0396	0.3078	0.0339
0.012	0.6486	0.0509	0.4568	0.0424	0.3303	0.0359
0.014	0.7422	0.0533	0.5214	0.0441	0.3765	0.0372
0.016	0.8535	0.0560	0.6004	0.0462	0.4335	0.0389
0.018	0.8965	0.0587	0.6244	0.0482	0.4467	0.0403
0.020	1.1110	0.0660	0.6231	0.0512	0.4372	0.0428
0.022	1.0492	0.0639	0.7165	0.0519	0.5005	0.0431
0.024	1.1588	0.0673	0.7946	0.0542	0.5568	0.0447
0.026	1.2377	0.0702	0.8348	0.0571	0.5760	0.0472
0.028	1.5309	0.0760	0.8934	0.0584	0.6179	0.0483
0.030	1.4156	0.0741	0.9648	0.0599	0.6707	0.0495
0.032	1.7633	0.0827	1.0771	0.0640	0.7487	0.0524
0.034	1.8381	0.0866	1.1702	0.0681	0.8134	0.0554
0.036	2.0474	0.0900	1.1558	0.0697	0.7912	0.0572
0.038	1.9219	0.0866	1.3213	0.0700	0.9251	0.0577
0.040	2.0524	0.0888	1.4081	0.0719	0.9824	0.0592

Table 4.7. Values of SNR^2_λ calculated according to (4.50) and their estimated standard deviations for the $r_s = 4.50$ mm, $r_s = 5.00$ mm, and $r_s = 5.50$ mm cases and for different values of the exposure time τ

$\boldsymbol{\tau}$ **[s]**	**AUC$_{\mathbf{I}}$**	**AUC$_{\mathbf{II}}$**	**AUC$_{\mathbf{III}}$**	**AUC$_{\mathbf{IV}}$**	**AUC$_{\mathbf{V}}$**	**AUC$_{\mathbf{VI}}$**
0.002	0.6424	0.7189	0.6348	0.6426	0.6401	0.5879
0.004	0.6722	0.7016	0.6667	0.6724	0.6702	0.6550
0.006	0.6987	0.6988	0.6853	0.6988	0.6861	0.6944
0.008	0.6991	0.7054	0.7015	0.6991	0.6963	0.7491
0.010	0.7119	0.7111	0.7077	0.7120	0.7064	0.7325
0.012	0.7225	0.7193	0.7177	0.7226	0.7162	0.7465
0.014	0.7310	0.7292	0.7284	0.7313	0.7294	0.7431
0.016	0.7422	0.7405	0.7399	0.7423	0.7438	0.7412
0.018	0.7490	0.7476	0.7473	0.7491	0.7489	0.7553
0.020	0.7459	0.7679	0.7681	0.7460	0.7714	0.7781
0.022	0.7618	0.7633	0.7629	0.7620	0.7659	0.7702
0.024	0.7741	0.7738	0.7736	0.7741	0.7772	0.7823
0.026	0.7795	0.7811	0.7809	0.7795	0.7847	0.7864
0.028	0.7739	0.8028	0.8032	0.7741	0.8084	0.8035
0.030	0.7949	0.7971	0.7969	0.7949	0.8003	0.7978
0.032	0.7914	0.8189	0.8191	0.7915	0.8260	0.8167
0.034	0.8005	0.8257	0.8261	0.8007	0.8307	0.8279
0.036	0.8090	0.8360	0.8364	0.8095	0.8435	0.8352
0.038	0.8276	0.8306	0.8304	0.8277	0.8369	0.8265
0.040	0.8325	0.8376	0.8373	0.8324	0.8449	0.8323

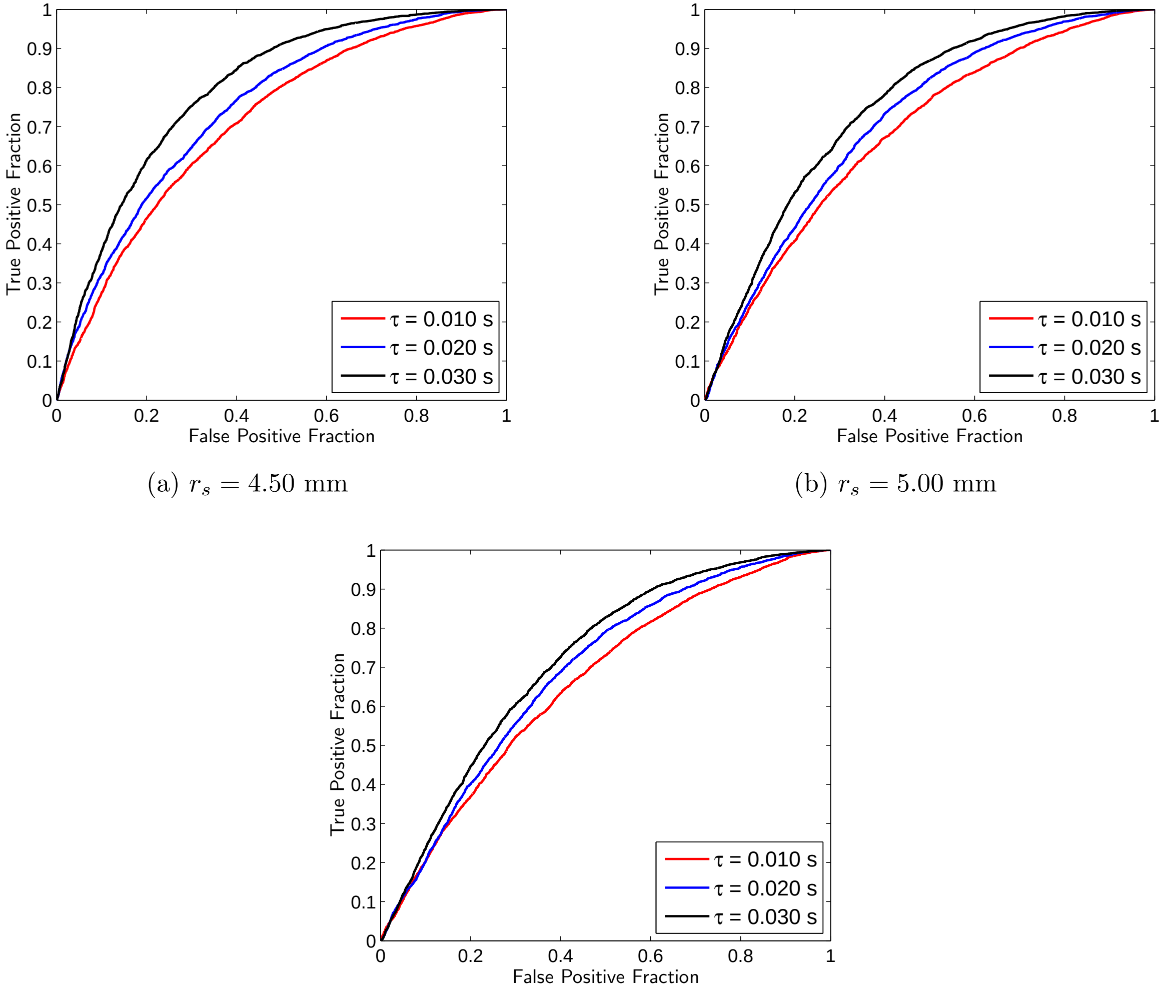
TABLE 4.8. Values of AUC$_{\text{I}}, \ldots,$ AUC$_{\text{VI}}$ for the $r_s = 4.50$ mm case

$\boldsymbol{\tau}$ [s]	$\mathbf{AUC_{I}}$	$\mathbf{AUC_{II}}$	$\mathbf{AUC_{III}}$	$\mathbf{AUC_{IV}}$	$\mathbf{AUC_{V}}$	$\mathbf{AUC_{VI}}$
0.002	0.6298	0.7117	0.6200	0.6301	0.6256	0.5621
0.004	0.6530	0.6903	0.6466	0.6532	0.6508	0.6261
0.006	0.6772	0.6803	0.6616	0.6772	0.6634	0.6676
0.008	0.6919	0.6821	0.6703	0.6919	0.6715	0.6849
0.010	0.6836	0.6853	0.6785	0.6836	0.6772	0.7144
0.012	0.6898	0.6897	0.6863	0.6901	0.6844	0.7288
0.014	0.6998	0.6968	0.6950	0.7001	0.6958	0.7176
0.016	0.7056	0.7060	0.7048	0.7057	0.7088	0.7075
0.018	0.7109	0.7115	0.7108	0.7110	0.7124	0.7237
0.020	0.7159	0.7162	0.7154	0.7162	0.7123	0.7614
0.022	0.7221	0.7240	0.7233	0.7224	0.7257	0.7359
0.024	0.7359	0.7332	0.7328	0.7359	0.7363	0.7448
0.026	0.7377	0.7391	0.7387	0.7378	0.7414	0.7527
0.028	0.7463	0.7462	0.7460	0.7464	0.7482	0.7571
0.030	0.7534	0.7537	0.7535	0.7535	0.7568	0.7571
0.032	0.7637	0.7631	0.7630	0.7641	0.7688	0.7666
0.034	0.7674	0.7719	0.7718	0.7677	0.7780	0.7764
0.036	0.7773	0.7747	0.7745	0.7776	0.7766	0.7929
0.038	0.7802	0.7858	0.7855	0.7803	0.7923	0.7813
0.040	0.7893	0.7922	0.7919	0.7892	0.7998	0.7867

Table 4.9. Values of $\mathrm{AUC_{I}}, \ldots, \mathrm{AUC_{VI}}$ for the $r_s = 5.00$ mm case

$\boldsymbol{\tau}$ [s]	$\mathbf{AUC_{I}}$	$\mathbf{AUC_{II}}$	$\mathbf{AUC_{III}}$	$\mathbf{AUC_{IV}}$	$\mathbf{AUC_{V}}$	$\mathbf{AUC_{VI}}$
0.002	0.6199	0.7084	0.6077	0.6202	0.6134	0.5391
0.004	0.6392	0.6837	0.6300	0.6394	0.6348	0.6005
0.006	0.6627	0.6674	0.6421	0.6627	0.6447	0.6440
0.008	0.6710	0.6657	0.6486	0.6709	0.6501	0.6650
0.010	0.6582	0.6673	0.6548	0.6584	0.6534	0.7045
0.012	0.6622	0.6675	0.6609	0.6625	0.6586	0.7212
0.014	0.6719	0.6716	0.6679	0.6723	0.6685	0.6995
0.016	0.6768	0.6786	0.6760	0.6770	0.6799	0.6807
0.018	0.6840	0.6823	0.6807	0.6842	0.6823	0.6993
0.020	0.6872	0.6860	0.6843	0.6875	0.6806	0.7545
0.022	0.6916	0.6924	0.6906	0.6918	0.6921	0.7135
0.024	0.7023	0.6995	0.6987	0.7024	0.7017	0.7165
0.026	0.7030	0.7041	0.7034	0.7030	0.7049	0.7296
0.028	0.7127	0.7099	0.7096	0.7129	0.7110	0.7301
0.030	0.7149	0.7168	0.7164	0.7152	0.7192	0.7265
0.032	0.7280	0.7249	0.7247	0.7283	0.7300	0.7321
0.034	0.7333	0.7330	0.7328	0.7338	0.7383	0.7412
0.036	0.7362	0.7351	0.7349	0.7363	0.7355	0.7692
0.038	0.7448	0.7460	0.7457	0.7448	0.7523	0.7426
0.040	0.7553	0.7517	0.7513	0.7551	0.7589	0.7475

TABLE 4.10. Values of $\mathrm{AUC}_{\mathrm{I}}, \ldots, \mathrm{AUC}_{\mathrm{VI}}$ for the $r_s = 5.50$ mm case

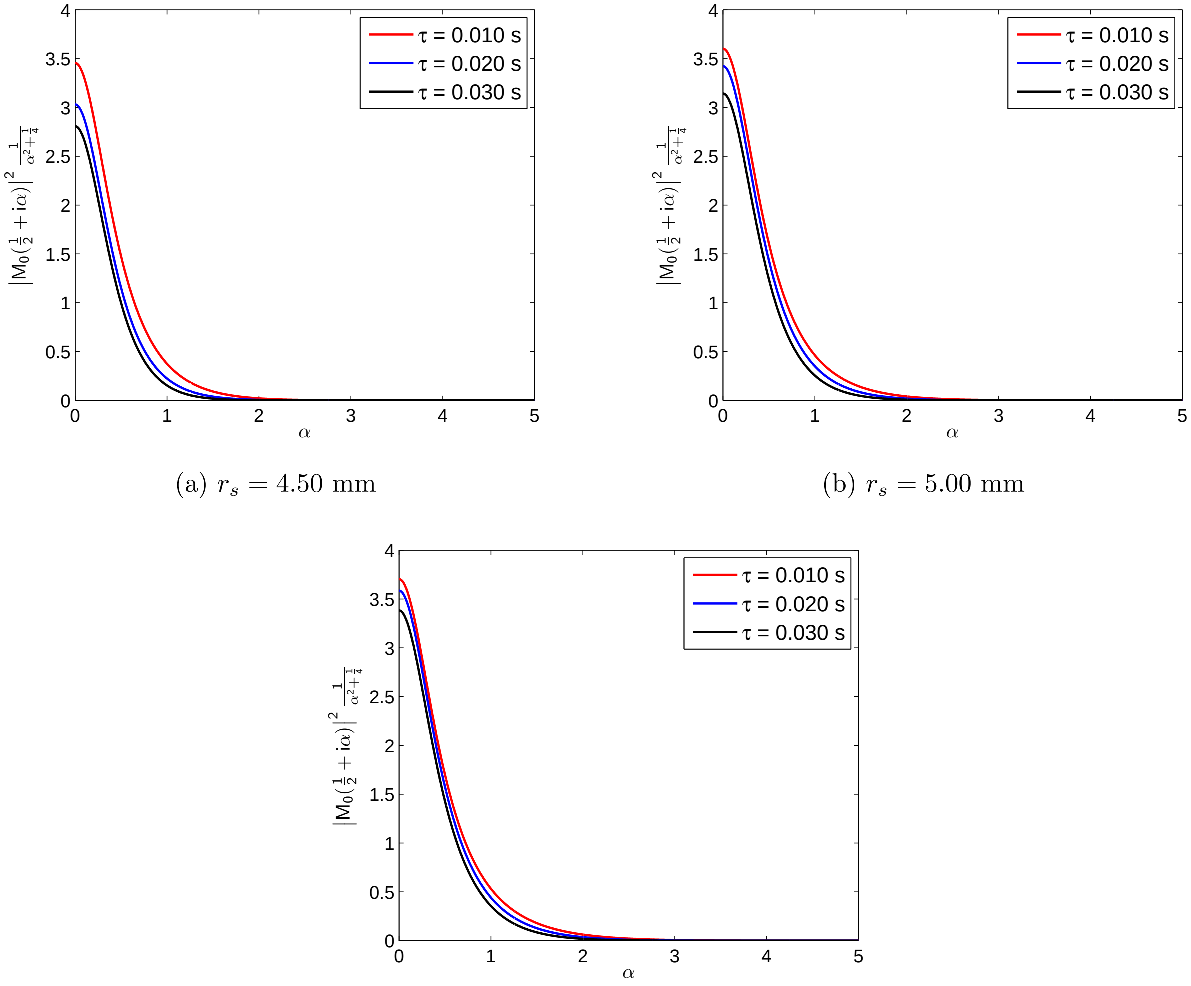


(a) $r_s = 4.50$ mm

(b) $r_s = 5.00$ mm

(c) $r_s = 5.50$ mm

FIGURE 4.17. Plots of ROC curves for different values of τ and r_s



(a) $r_s = 4.50$ mm

(b) $r_s = 5.00$ mm

(c) $r_s = 5.50$ mm

FIGURE 4.18. Plots of $|M_0(\frac{1}{2}+i\alpha)|^2\frac{1}{\alpha^2+\frac{1}{4}}$ for different values of τ and r_s

τ [s]	**AUC Check**	**AUC$_{\mathrm{I}}$?**	**AUC$_{\mathrm{II}}$?**	**AUC$_{\mathrm{III}}$?**	**AUC$_{\mathrm{IV}}$?**	**AUC$_{\mathrm{V}}$?**	**AUC$_{\mathrm{VI}}$?**
0.002	$0.5312 \leqslant \mathrm{AUC} \leqslant 0.6656$	✓	✗	✓	✓	✓	✓
0.004	$0.5463 \leqslant \mathrm{AUC} \leqslant 0.7048$	✓	✓	✓	✓	✓	✓
0.006	$0.5555 \leqslant \mathrm{AUC} \leqslant 0.7271$	✓	✓	✓	✓	✓	✓
0.008	$0.5619 \leqslant \mathrm{AUC} \leqslant 0.7471$	✓	✓	✓	✓	✓	✗
0.010	$0.5686 \leqslant \mathrm{AUC} \leqslant 0.7541$	✓	✓	✓	✓	✓	✓
0.012	$0.5754 \leqslant \mathrm{AUC} \leqslant 0.7664$	✓	✓	✓	✓	✓	✓
0.014	$0.5852 \leqslant \mathrm{AUC} \leqslant 0.7777$	✓	✓	✓	✓	✓	✓
0.016	$0.5967 \leqslant \mathrm{AUC} \leqslant 0.7893$	✓	✓	✓	✓	✓	✓
0.018	$0.6009 \leqslant \mathrm{AUC} \leqslant 0.7974$	✓	✓	✓	✓	✓	✓
0.020	$0.6208 \leqslant \mathrm{AUC} \leqslant 0.8214$	✓	✓	✓	✓	✓	✓
0.022	$0.6158 \leqslant \mathrm{AUC} \leqslant 0.8149$	✓	✓	✓	✓	✓	✓
0.024	$0.6263 \leqslant \mathrm{AUC} \leqslant 0.8270$	✓	✓	✓	✓	✓	✓
0.026	$0.6336 \leqslant \mathrm{AUC} \leqslant 0.8336$	✓	✓	✓	✓	✓	✓
0.028	$0.6582 \leqslant \mathrm{AUC} \leqslant 0.8548$	✓	✓	✓	✓	✓	✓
0.030	$0.6495 \leqslant \mathrm{AUC} \leqslant 0.8478$	✓	✓	✓	✓	✓	✓
0.032	$0.6782 \leqslant \mathrm{AUC} \leqslant 0.8696$	✓	✓	✓	✓	✓	✓
0.034	$0.6837 \leqslant \mathrm{AUC} \leqslant 0.8770$	✓	✓	✓	✓	✓	✓
0.036	$0.6995 \leqslant \mathrm{AUC} \leqslant 0.8861$	✓	✓	✓	✓	✓	✓
0.038	$0.6912 \leqslant \mathrm{AUC} \leqslant 0.8792$	✓	✓	✓	✓	✓	✓
0.040	$0.7013 \leqslant \mathrm{AUC} \leqslant 0.8853$	✓	✓	✓	✓	✓	✓

TABLE 4.11. AUC checks for the $r_s = 4.50$ mm case and bounds in (4.54a)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_{\mathbf{I}}$?**	**AUC$_{\mathbf{II}}$?**	**AUC$_{\mathbf{III}}$?**	**AUC$_{\mathbf{IV}}$?**	**AUC$_{\mathbf{V}}$?**	**AUC$_{\mathbf{VI}}$?**
0.002	$0.5159 \leqslant \text{AUC} \leqslant 0.7093$	✓	✗	✓	✓	✓	✓
0.004	$0.5237 \leqslant \text{AUC} \leqslant 0.7529$	✓	✓	✓	✓	✓	✓
0.006	$0.5286 \leqslant \text{AUC} \leqslant 0.7771$	✓	✓	✓	✓	✓	✓
0.008	$0.5320 \leqslant \text{AUC} \leqslant 0.7987$	✓	✓	✓	✓	✓	✓
0.010	$0.5356 \leqslant \text{AUC} \leqslant 0.8053$	✓	✓	✓	✓	✓	✓
0.012	$0.5393 \leqslant \text{AUC} \leqslant 0.8177$	✓	✓	✓	✓	✓	✓
0.014	$0.5446 \leqslant \text{AUC} \leqslant 0.8284$	✓	✓	✓	✓	✓	✓
0.016	$0.5509 \leqslant \text{AUC} \leqslant 0.8391$	✓	✓	✓	✓	✓	✓
0.018	$0.5533 \leqslant \text{AUC} \leqslant 0.8470$	✓	✓	✓	✓	✓	✓
0.020	$0.5646 \leqslant \text{AUC} \leqslant 0.8693$	✓	✓	✓	✓	✓	✓
0.022	$0.5617 \leqslant \text{AUC} \leqslant 0.8633$	✓	✓	✓	✓	✓	✓
0.024	$0.5677 \leqslant \text{AUC} \leqslant 0.8743$	✓	✓	✓	✓	✓	✓
0.026	$0.5720 \leqslant \text{AUC} \leqslant 0.8800$	✓	✓	✓	✓	✓	✓
0.028	$0.5866 \leqslant \text{AUC} \leqslant 0.8982$	✓	✓	✓	✓	✓	✓
0.030	$0.5814 \leqslant \text{AUC} \leqslant 0.8924$	✓	✓	✓	✓	✓	✓
0.032	$0.5989 \leqslant \text{AUC} \leqslant 0.9103$	✓	✓	✓	✓	✓	✓
0.034	$0.6023 \leqslant \text{AUC} \leqslant 0.9169$	✓	✓	✓	✓	✓	✓
0.036	$0.6124 \leqslant \text{AUC} \leqslant 0.9238$	✓	✓	✓	✓	✓	✓
0.038	$0.6071 \leqslant \text{AUC} \leqslant 0.9181$	✓	✓	✓	✓	✓	✓
0.040	$0.6136 \leqslant \text{AUC} \leqslant 0.9228$	✓	✓	✓	✓	✓	✓

TABLE 4.12. AUC checks for the $r_s = 4.50$ mm case and bounds in (4.54b)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_{\mathbf{I}}$?**	**AUC$_{\mathbf{II}}$?**	**AUC$_{\mathbf{III}}$?**	**AUC$_{\mathbf{IV}}$?**	**AUC$_{\mathbf{V}}$?**	**AUC$_{\mathbf{VI}}$?**
0.002	$0.6022 \leqslant \text{AUC} \leqslant 0.6834$	✓	✗	✓	✓	✓	✗
0.004	$0.6219 \leqslant \text{AUC} \leqslant 0.7141$	✓	✓	✓	✓	✓	✓
0.006	$0.6439 \leqslant \text{AUC} \leqslant 0.7463$	✓	✓	✓	✓	✓	✓
0.008	$0.6459 \leqslant \text{AUC} \leqslant 0.7491$	✓	✓	✓	✓	✓	✓
0.010	$0.6552 \leqslant \text{AUC} \leqslant 0.7623$	✓	✓	✓	✓	✓	✓
0.012	$0.6682 \leqslant \text{AUC} \leqslant 0.7798$	✓	✓	✓	✓	✓	✓
0.014	$0.6741 \leqslant \text{AUC} \leqslant 0.7876$	✓	✓	✓	✓	✓	✓
0.016	$0.6763 \leqslant \text{AUC} \leqslant 0.7904$	✓	✓	✓	✓	✓	✓
0.018	$0.6898 \leqslant \text{AUC} \leqslant 0.8075$	✓	✓	✓	✓	✓	✓
0.020	$0.6800 \leqslant \text{AUC} \leqslant 0.7951$	✓	✓	✓	✓	✓	✓
0.022	$0.6938 \leqslant \text{AUC} \leqslant 0.8124$	✓	✓	✓	✓	✓	✓
0.024	$0.7065 \leqslant \text{AUC} \leqslant 0.8277$	✓	✓	✓	✓	✓	✓
0.026	$0.7108 \leqslant \text{AUC} \leqslant 0.8327$	✓	✓	✓	✓	✓	✓
0.028	$0.6990 \leqslant \text{AUC} \leqslant 0.8188$	✓	✓	✓	✓	✓	✓
0.030	$0.7262 \leqslant \text{AUC} \leqslant 0.8501$	✓	✓	✓	✓	✓	✓
0.032	$0.7184 \leqslant \text{AUC} \leqslant 0.8414$	✓	✓	✓	✓	✓	✓
0.034	$0.7259 \leqslant \text{AUC} \leqslant 0.8497$	✓	✓	✓	✓	✓	✓
0.036	$0.7372 \leqslant \text{AUC} \leqslant 0.8619$	✓	✓	✓	✓	✓	✓
0.038	$0.7531 \leqslant \text{AUC} \leqslant 0.8781$	✓	✓	✓	✓	✓	✓
0.040	$0.7580 \leqslant \text{AUC} \leqslant 0.8828$	✓	✓	✓	✓	✓	✓

TABLE 4.13. AUC checks for the $r_s = 4.50$ mm case and bounds in (4.54c)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_{\mathrm{I}}$?**	**AUC$_{\mathrm{II}}$?**	**AUC$_{\mathrm{III}}$?**	**AUC$_{\mathrm{IV}}$?**	**AUC$_{\mathrm{V}}$?**	**AUC$_{\mathrm{VI}}$?**
0.002	$0.5251 \leqslant \mathrm{AUC} \leqslant 0.6481$	✓	✗	✓	✓	✓	✓
0.004	$0.5363 \leqslant \mathrm{AUC} \leqslant 0.6812$	✓	✗	✓	✓	✓	✓
0.006	$0.5426 \leqslant \mathrm{AUC} \leqslant 0.6998$	✓	✓	✓	✓	✓	✓
0.008	$0.5471 \leqslant \mathrm{AUC} \leqslant 0.7109$	✓	✓	✓	✓	✓	✓
0.010	$0.5503 \leqslant \mathrm{AUC} \leqslant 0.7212$	✓	✓	✓	✓	✓	✓
0.012	$0.5545 \leqslant \mathrm{AUC} \leqslant 0.7312$	✓	✓	✓	✓	✓	✓
0.014	$0.5616 \leqslant \mathrm{AUC} \leqslant 0.7409$	✓	✓	✓	✓	✓	✓
0.016	$0.5702 \leqslant \mathrm{AUC} \leqslant 0.7511$	✓	✓	✓	✓	✓	✓
0.018	$0.5727 \leqslant \mathrm{AUC} \leqslant 0.7580$	✓	✓	✓	✓	✓	✓
0.020	$0.5727 \leqslant \mathrm{AUC} \leqslant 0.7657$	✓	✓	✓	✓	✓	✓
0.022	$0.5825 \leqslant \mathrm{AUC} \leqslant 0.7728$	✓	✓	✓	✓	✓	✓
0.024	$0.5906 \leqslant \mathrm{AUC} \leqslant 0.7841$	✓	✓	✓	✓	✓	✓
0.026	$0.5947 \leqslant \mathrm{AUC} \leqslant 0.7905$	✓	✓	✓	✓	✓	✓
0.028	$0.6003 \leqslant \mathrm{AUC} \leqslant 0.7972$	✓	✓	✓	✓	✓	✓
0.030	$0.6077 \leqslant \mathrm{AUC} \leqslant 0.8041$	✓	✓	✓	✓	✓	✓
0.032	$0.6184 \leqslant \mathrm{AUC} \leqslant 0.8158$	✓	✓	✓	✓	✓	✓
0.034	$0.6270 \leqslant \mathrm{AUC} \leqslant 0.8255$	✓	✓	✓	✓	✓	✓
0.036	$0.6258 \leqslant \mathrm{AUC} \leqslant 0.8299$	✓	✓	✓	✓	✓	✓
0.038	$0.6412 \leqslant \mathrm{AUC} \leqslant 0.8366$	✓	✓	✓	✓	✓	✓
0.040	$0.6490 \leqslant \mathrm{AUC} \leqslant 0.8431$	✓	✓	✓	✓	✓	✓

TABLE 4.14. AUC checks for the $r_s = 5.00$ mm case and bounds in (4.54a)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_{\mathbf{I}}$?**	**AUC$_{\mathbf{II}}$?**	**AUC$_{\mathbf{III}}$?**	**AUC$_{\mathbf{IV}}$?**	**AUC$_{\mathbf{V}}$?**	**AUC$_{\mathbf{VI}}$?**
0.002	$0.5127 \leqslant \text{AUC} \leqslant 0.6893$	✓	✗	✓	✓	✓	✓
0.004	$0.5185 \leqslant \text{AUC} \leqslant 0.7271$	✓	✓	✓	✓	✓	✓
0.006	$0.5218 \leqslant \text{AUC} \leqslant 0.7479$	✓	✓	✓	✓	✓	✓
0.008	$0.5241 \leqslant \text{AUC} \leqslant 0.7599$	✓	✓	✓	✓	✓	✓
0.010	$0.5258 \leqslant \text{AUC} \leqslant 0.7714$	✓	✓	✓	✓	✓	✓
0.012	$0.5281 \leqslant \text{AUC} \leqslant 0.7820$	✓	✓	✓	✓	✓	✓
0.014	$0.5318 \leqslant \text{AUC} \leqslant 0.7916$	✓	✓	✓	✓	✓	✓
0.016	$0.5364 \leqslant \text{AUC} \leqslant 0.8016$	✓	✓	✓	✓	✓	✓
0.018	$0.5378 \leqslant \text{AUC} \leqslant 0.8088$	✓	✓	✓	✓	✓	✓
0.020	$0.5378 \leqslant \text{AUC} \leqslant 0.8175$	✓	✓	✓	✓	✓	✓
0.022	$0.5431 \leqslant \text{AUC} \leqslant 0.8235$	✓	✓	✓	✓	✓	✓
0.024	$0.5476 \leqslant \text{AUC} \leqslant 0.8344$	✓	✓	✓	✓	✓	✓
0.026	$0.5499 \leqslant \text{AUC} \leqslant 0.8406$	✓	✓	✓	✓	✓	✓
0.028	$0.5530 \leqslant \text{AUC} \leqslant 0.8470$	✓	✓	✓	✓	✓	✓
0.030	$0.5571 \leqslant \text{AUC} \leqslant 0.8531$	✓	✓	✓	✓	✓	✓
0.032	$0.5632 \leqslant \text{AUC} \leqslant 0.8638$	✓	✓	✓	✓	✓	✓
0.034	$0.5682 \leqslant \text{AUC} \leqslant 0.8726$	✓	✓	✓	✓	✓	✓
0.036	$0.5675 \leqslant \text{AUC} \leqslant 0.8775$	✓	✓	✓	✓	✓	✓
0.038	$0.5764 \leqslant \text{AUC} \leqslant 0.8821$	✓	✓	✓	✓	✓	✓
0.040	$0.5811 \leqslant \text{AUC} \leqslant 0.8876$	✓	✓	✓	✓	✓	✓

TABLE 4.15. AUC checks for the $r_s = 5.00$ mm case and bounds in (4.54b)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_{\mathbf{I}}$?**	**AUC$_{\mathbf{II}}$?**	**AUC$_{\mathbf{III}}$?**	**AUC$_{\mathbf{IV}}$?**	**AUC$_{\mathbf{V}}$?**	**AUC$_{\mathbf{VI}}$?**
0.002	$0.5922 \leqslant \text{AUC} \leqslant 0.6673$	✓	✗	✓	✓	✓	✗
0.004	$0.6063 \leqslant \text{AUC} \leqslant 0.6901$	✓	✗	✓	✓	✓	✓
0.006	$0.6259 \leqslant \text{AUC} \leqslant 0.7200$	✓	✓	✓	✓	✓	✓
0.008	$0.6381 \leqslant \text{AUC} \leqslant 0.7380$	✓	✓	✓	✓	✓	✓
0.010	$0.6358 \leqslant \text{AUC} \leqslant 0.7347$	✓	✓	✓	✓	✓	✓
0.012	$0.6440 \leqslant \text{AUC} \leqslant 0.7465$	✓	✓	✓	✓	✓	✓
0.014	$0.6517 \leqslant \text{AUC} \leqslant 0.7573$	✓	✓	✓	✓	✓	✓
0.016	$0.6501 \leqslant \text{AUC} \leqslant 0.7551$	✓	✓	✓	✓	✓	✓
0.018	$0.6602 \leqslant \text{AUC} \leqslant 0.7690$	✓	✓	✓	✓	✓	✓
0.020	$0.6625 \leqslant \text{AUC} \leqslant 0.7722$	✓	✓	✓	✓	✓	✓
0.022	$0.6633 \leqslant \text{AUC} \leqslant 0.7733$	✓	✓	✓	✓	✓	✓
0.024	$0.6790 \leqslant \text{AUC} \leqslant 0.7939$	✓	✓	✓	✓	✓	✓
0.026	$0.6774 \leqslant \text{AUC} \leqslant 0.7919$	✓	✓	✓	✓	✓	✓
0.028	$0.6865 \leqslant \text{AUC} \leqslant 0.8034$	✓	✓	✓	✓	✓	✓
0.030	$0.6916 \leqslant \text{AUC} \leqslant 0.8098$	✓	✓	✓	✓	✓	✓
0.032	$0.6989 \leqslant \text{AUC} \leqslant 0.8187$	✓	✓	✓	✓	✓	✓
0.034	$0.7011 \leqslant \text{AUC} \leqslant 0.8213$	✓	✓	✓	✓	✓	✓
0.036	$0.7127 \leqslant \text{AUC} \leqslant 0.8349$	✓	✓	✓	✓	✓	✓
0.038	$0.7123 \leqslant \text{AUC} \leqslant 0.8344$	✓	✓	✓	✓	✓	✓
0.040	$0.7217 \leqslant \text{AUC} \leqslant 0.8451$	✓	✓	✓	✓	✓	✓

TABLE 4.16. AUC checks for the $r_s = 5.00$ mm case and bounds in (4.54c)

$\boldsymbol{\tau}$ [s]	**AUC Check**	**AUC$_\text{I}$?**	**AUC$_\text{II}$?**	**AUC$_\text{III}$?**	**AUC$_\text{IV}$?**	**AUC$_\text{V}$?**	**AUC$_\text{VI}$?**
0.002	$0.5204 \leqslant \text{AUC} \leqslant 0.6334$	✓	✗	✓	✓	✓	✓
0.004	$0.5289 \leqslant \text{AUC} \leqslant 0.6616$	✓	✗	✓	✓	✓	✓
0.006	$0.5334 \leqslant \text{AUC} \leqslant 0.6770$	✓	✓	✓	✓	✓	✓
0.008	$0.5359 \leqslant \text{AUC} \leqslant 0.6854$	✓	✓	✓	✓	✓	✓
0.010	$0.5375 \leqslant \text{AUC} \leqslant 0.6938$	✓	✓	✓	✓	✓	✗
0.012	$0.5401 \leqslant \text{AUC} \leqslant 0.7017$	✓	✓	✓	✓	✓	✗
0.014	$0.5454 \leqslant \text{AUC} \leqslant 0.7097$	✓	✓	✓	✓	✓	✓
0.016	$0.5518 \leqslant \text{AUC} \leqslant 0.7185$	✓	✓	✓	✓	✓	✓
0.018	$0.5533 \leqslant \text{AUC} \leqslant 0.7242$	✓	✓	✓	✓	✓	✓
0.020	$0.5523 \leqslant \text{AUC} \leqslant 0.7311$	✓	✓	✓	✓	✓	✗
0.022	$0.5593 \leqslant \text{AUC} \leqslant 0.7365$	✓	✓	✓	✓	✓	✓
0.024	$0.5654 \leqslant \text{AUC} \leqslant 0.7464$	✓	✓	✓	✓	✓	✓
0.026	$0.5676 \leqslant \text{AUC} \leqslant 0.7520$	✓	✓	✓	✓	✓	✓
0.028	$0.5718 \leqslant \text{AUC} \leqslant 0.7580$	✓	✓	✓	✓	✓	✓
0.030	$0.5777 \leqslant \text{AUC} \leqslant 0.7648$	✓	✓	✓	✓	✓	✓
0.032	$0.5857 \leqslant \text{AUC} \leqslant 0.7755$	✓	✓	✓	✓	✓	✓
0.034	$0.5922 \leqslant \text{AUC} \leqslant 0.7848$	✓	✓	✓	✓	✓	✓
0.036	$0.5900 \leqslant \text{AUC} \leqslant 0.7893$	✓	✓	✓	✓	✓	✓
0.038	$0.6037 \leqslant \text{AUC} \leqslant 0.7963$	✓	✓	✓	✓	✓	✓
0.040	$0.6095 \leqslant \text{AUC} \leqslant 0.8026$	✓	✓	✓	✓	✓	✓

TABLE 4.17. AUC checks for the $r_s = 5.50$ mm case and bounds in (4.54a)

τ [s]	**AUC Check**	**AUC$_{\mathrm{I}}$?**	**AUC$_{\mathrm{II}}$?**	**AUC$_{\mathrm{III}}$?**	**AUC$_{\mathrm{IV}}$?**	**AUC$_{\mathrm{V}}$?**	**AUC$_{\mathrm{VI}}$?**
0.002	$0.5103 \leqslant \text{AUC} \leqslant 0.6721$	✓	✗	✓	✓	✓	✓
0.004	$0.5147 \leqslant \text{AUC} \leqslant 0.7049$	✓	✓	✓	✓	✓	✓
0.006	$0.5170 \leqslant \text{AUC} \leqslant 0.7226$	✓	✓	✓	✓	✓	✓
0.008	$0.5183 \leqslant \text{AUC} \leqslant 0.7322$	✓	✓	✓	✓	✓	✓
0.010	$0.5191 \leqslant \text{AUC} \leqslant 0.7419$	✓	✓	✓	✓	✓	✓
0.012	$0.5205 \leqslant \text{AUC} \leqslant 0.7507$	✓	✓	✓	✓	✓	✓
0.014	$0.5232 \leqslant \text{AUC} \leqslant 0.7590$	✓	✓	✓	✓	✓	✓
0.016	$0.5266 \leqslant \text{AUC} \leqslant 0.7679$	✓	✓	✓	✓	✓	✓
0.018	$0.5274 \leqslant \text{AUC} \leqslant 0.7742$	✓	✓	✓	✓	✓	✓
0.020	$0.5269 \leqslant \text{AUC} \leqslant 0.7824$	✓	✓	✓	✓	✓	✓
0.022	$0.5306 \leqslant \text{AUC} \leqslant 0.7871$	✓	✓	✓	✓	✓	✓
0.024	$0.5339 \leqslant \text{AUC} \leqslant 0.7972$	✓	✓	✓	✓	✓	✓
0.026	$0.5350 \leqslant \text{AUC} \leqslant 0.8031$	✓	✓	✓	✓	✓	✓
0.028	$0.5373 \leqslant \text{AUC} \leqslant 0.8091$	✓	✓	✓	✓	✓	✓
0.030	$0.5405 \leqslant \text{AUC} \leqslant 0.8155$	✓	✓	✓	✓	✓	✓
0.032	$0.5449 \leqslant \text{AUC} \leqslant 0.8258$	✓	✓	✓	✓	✓	✓
0.034	$0.5485 \leqslant \text{AUC} \leqslant 0.8349$	✓	✓	✓	✓	✓	✓
0.036	$0.5472 \leqslant \text{AUC} \leqslant 0.8402$	✓	✓	✓	✓	✓	✓
0.038	$0.5549 \leqslant \text{AUC} \leqslant 0.8454$	✓	✓	✓	✓	✓	✓
0.040	$0.5582 \leqslant \text{AUC} \leqslant 0.8512$	✓	✓	✓	✓	✓	✓

TABLE 4.18. AUC checks for the $r_s = 5.50$ mm case and bounds in (4.54b)

$\boldsymbol{\tau}$ **[s]**	**AUC Check**	**AUC**$_{\mathbf{I}}$**?**	**AUC**$_{\mathbf{II}}$**?**	**AUC**$_{\mathbf{III}}$**?**	**AUC**$_{\mathbf{IV}}$**?**	**AUC**$_{\mathbf{V}}$**?**	**AUC**$_{\mathbf{VI}}$**?**
0.002	$0.5852 \leqslant \text{AUC} \leqslant 0.6558$	✓	✗	✓	✓	✓	✗
0.004	$0.5974 \leqslant \text{AUC} \leqslant 0.6759$	✓	✗	✓	✓	✓	✓
0.006	$0.6156 \leqslant \text{AUC} \leqslant 0.7044$	✓	✓	✓	✓	✓	✓
0.008	$0.6220 \leqslant \text{AUC} \leqslant 0.7143$	✓	✓	✓	✓	✓	✓
0.010	$0.6164 \leqslant \text{AUC} \leqslant 0.7057$	✓	✓	✓	✓	✓	✓
0.012	$0.6210 \leqslant \text{AUC} \leqslant 0.7126$	✓	✓	✓	✓	✓	✗
0.014	$0.6310 \leqslant \text{AUC} \leqslant 0.7277$	✓	✓	✓	✓	✓	✓
0.016	$0.6309 \leqslant \text{AUC} \leqslant 0.7274$	✓	✓	✓	✓	✓	✓
0.018	$0.6384 \leqslant \text{AUC} \leqslant 0.7385$	✓	✓	✓	✓	✓	✓
0.020	$0.6442 \leqslant \text{AUC} \leqslant 0.7467$	✓	✓	✓	✓	✓	✗
0.022	$0.6422 \leqslant \text{AUC} \leqslant 0.7439$	✓	✓	✓	✓	✓	✓
0.024	$0.6535 \leqslant \text{AUC} \leqslant 0.7598$	✓	✓	✓	✓	✓	✓
0.026	$0.6551 \leqslant \text{AUC} \leqslant 0.7620$	✓	✓	✓	✓	✓	✓
0.028	$0.6575 \leqslant \text{AUC} \leqslant 0.7654$	✓	✓	✓	✓	✓	✓
0.030	$0.6599 \leqslant \text{AUC} \leqslant 0.7686$	✓	✓	✓	✓	✓	✓
0.032	$0.6742 \leqslant \text{AUC} \leqslant 0.7876$	✓	✓	✓	✓	✓	✓
0.034	$0.6762 \leqslant \text{AUC} \leqslant 0.7903$	✓	✓	✓	✓	✓	✓
0.036	$0.6783 \leqslant \text{AUC} \leqslant 0.7930$	✓	✓	✓	✓	✓	✓
0.038	$0.6849 \leqslant \text{AUC} \leqslant 0.8014$	✓	✓	✓	✓	✓	✓
0.040	$0.6981 \leqslant \text{AUC} \leqslant 0.8177$	✓	✓	✓	✓	✓	✓

TABLE 4.19. AUC checks for the $r_s = 5.50$ mm case and bounds in (4.54c)

$\boldsymbol{\tau}$ [s]	$\boldsymbol{s_{\mathrm{AUC_I}}}$	$\boldsymbol{s_{\mathrm{AUC_{II}}}}$	$\boldsymbol{s_{\mathrm{AUC_{III}}}}$	$\boldsymbol{s_{\mathrm{AUC_{IV}}}}$	$\boldsymbol{s_{\mathrm{AUC_V}}}$	$\boldsymbol{s_{\mathrm{AUC_{VI}}}}$	$\boldsymbol{\sigma_{\mathrm{2AFC}}}$
0.002	0.0027	0.0040	0.0002	0.0039	0.0070	0.0424	0.0055
0.004	0.0025	0.0041	0.0002	0.0036	0.0073	0.0306	0.0053
0.006	0.0026	0.0050	0.0002	0.0037	0.0074	0.0274	0.0052
0.008	0.0026	0.0038	0.0003	0.0036	0.0078	0.0303	0.0052
0.010	0.0027	0.0036	0.0003	0.0038	0.0075	0.0241	0.0051
0.012	0.0027	0.0037	0.0003	0.0039	0.0076	0.0222	0.0051
0.014	0.0028	0.0037	0.0003	0.0039	0.0073	0.0176	0.0050
0.016	0.0029	0.0038	0.0003	0.0039	0.0070	0.0117	0.0049
0.018	0.0029	0.0037	0.0003	0.0037	0.0069	0.0114	0.0049
0.020	0.0029	0.0038	0.0003	0.0037	0.0068	0.0155	0.0049
0.022	0.0031	0.0039	0.0003	0.0037	0.0069	0.0104	0.0048
0.024	0.0030	0.0037	0.0003	0.0036	0.0065	0.0122	0.0047
0.026	0.0032	0.0037	0.0003	0.0036	0.0064	0.0114	0.0046
0.028	0.0030	0.0038	0.0003	0.0036	0.0061	0.0056	0.0047
0.030	0.0032	0.0039	0.0003	0.0035	0.0063	0.0052	0.0045
0.032	0.0033	0.0037	0.0003	0.0034	0.0060	0.0051	0.0045
0.034	0.0033	0.0034	0.0003	0.0036	0.0055	0.0057	0.0044
0.036	0.0033	0.0033	0.0003	0.0034	0.0053	0.0057	0.0043
0.038	0.0031	0.0034	0.0003	0.0032	0.0052	0.0037	0.0041
0.040	0.0032	0.0034	0.0003	0.0032	0.0054	0.0036	0.0041

TABLE 4.20. Values of AUC standard deviations for the $r_s = 4.50$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{s}_{\mathbf{AUC_I}}$	$\boldsymbol{s}_{\mathbf{AUC_{II}}}$	$\boldsymbol{s}_{\mathbf{AUC_{III}}}$	$\boldsymbol{s}_{\mathbf{AUC_{IV}}}$	$\boldsymbol{s}_{\mathbf{AUC_V}}$	$\boldsymbol{s}_{\mathbf{AUC_{VI}}}$	$\boldsymbol{\sigma}_{\mathbf{2AFC}}$
0.002	0.0026	0.0057	0.0002	0.0039	0.0070	0.0539	0.0055
0.004	0.0024	0.0068	0.0002	0.0037	0.0074	0.0401	0.0054
0.006	0.0026	0.0061	0.0002	0.0038	0.0076	0.0381	0.0053
0.008	0.0026	0.0035	0.0002	0.0038	0.0074	0.0342	0.0052
0.010	0.0026	0.0034	0.0002	0.0039	0.0078	0.0383	0.0053
0.012	0.0025	0.0036	0.0003	0.0040	0.0079	0.0357	0.0053
0.014	0.0026	0.0035	0.0003	0.0040	0.0076	0.0285	0.0052
0.016	0.0027	0.0036	0.0003	0.0040	0.0074	0.0195	0.0052
0.018	0.0029	0.0036	0.0003	0.0039	0.0072	0.0195	0.0051
0.020	0.0031	0.0037	0.0003	0.0040	0.0075	0.0275	0.0051
0.022	0.0030	0.0038	0.0003	0.0038	0.0073	0.0173	0.0051
0.024	0.0031	0.0036	0.0003	0.0039	0.0069	0.0170	0.0050
0.026	0.0031	0.0037	0.0003	0.0038	0.0070	0.0190	0.0049
0.028	0.0034	0.0037	0.0003	0.0040	0.0071	0.0143	0.0049
0.030	0.0034	0.0039	0.0003	0.0038	0.0069	0.0096	0.0048
0.032	0.0033	0.0036	0.0003	0.0036	0.0065	0.0139	0.0047
0.034	0.0035	0.0037	0.0003	0.0037	0.0067	0.0123	0.0047
0.036	0.0034	0.0038	0.0003	0.0036	0.0070	0.0141	0.0046
0.038	0.0035	0.0036	0.0003	0.0036	0.0059	0.0056	0.0046
0.040	0.0034	0.0036	0.0003	0.0036	0.0062	0.0062	0.0045

Table 4.21. Values of AUC standard deviations for the $r_s = 5.00$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{s}_{\mathbf{AUC_I}}$	$\boldsymbol{s}_{\mathbf{AUC_{II}}}$	$\boldsymbol{s}_{\mathbf{AUC_{III}}}$	$\boldsymbol{s}_{\mathbf{AUC_{IV}}}$	$\boldsymbol{s}_{\mathbf{AUC_V}}$	$\boldsymbol{s}_{\mathbf{AUC_{VI}}}$	$\boldsymbol{\sigma}_{\mathbf{2AFC}}$
0.002	0.0025	0.0079	0.0001	0.0039	0.0070	0.0676	0.0056
0.004	0.0024	0.0093	0.0002	0.0037	0.0074	0.0518	0.0055
0.006	0.0025	0.0070	0.0002	0.0038	0.0076	0.0514	0.0054
0.008	0.0025	0.0045	0.0002	0.0038	0.0075	0.0483	0.0054
0.010	0.0024	0.0033	0.0002	0.0039	0.0080	0.0578	0.0054
0.012	0.0024	0.0035	0.0002	0.0040	0.0081	0.0553	0.0054
0.014	0.0026	0.0035	0.0002	0.0041	0.0078	0.0434	0.0053
0.016	0.0027	0.0034	0.0002	0.0041	0.0076	0.0303	0.0053
0.018	0.0028	0.0033	0.0002	0.0040	0.0075	0.0307	0.0053
0.020	0.0030	0.0035	0.0003	0.0041	0.0078	0.0443	0.0053
0.022	0.0030	0.0036	0.0003	0.0039	0.0077	0.0286	0.0052
0.024	0.0031	0.0034	0.0003	0.0041	0.0072	0.0248	0.0052
0.026	0.0032	0.0035	0.0003	0.0040	0.0074	0.0295	0.0052
0.028	0.0035	0.0036	0.0003	0.0042	0.0075	0.0233	0.0051
0.030	0.0036	0.0038	0.0003	0.0040	0.0074	0.0175	0.0051
0.032	0.0035	0.0036	0.0003	0.0038	0.0070	0.0208	0.0050
0.034	0.0036	0.0037	0.0003	0.0039	0.0072	0.0178	0.0050
0.036	0.0035	0.0038	0.0003	0.0038	0.0077	0.0245	0.0050
0.038	0.0036	0.0035	0.0003	0.0038	0.0064	0.0095	0.0049
0.040	0.0037	0.0036	0.0003	0.0038	0.0068	0.0109	0.0048

TABLE 4.22. Values of AUC standard deviations for the $r_s = 5.50$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{d^2_{A,\mathrm{I}}}$	$\boldsymbol{d^2_{A,\mathrm{II}}}$	$\boldsymbol{d^2_{A,\mathrm{III}}}$	$\boldsymbol{d^2_{A,\mathrm{IV}}}$	$\boldsymbol{d^2_{A,\mathrm{V}}}$	$\boldsymbol{d^2_{A,\mathrm{VI}}}$
0.002	0.2662	0.6722	0.2373	0.2670	0.2573	0.0988
0.004	0.3979	0.5596	0.3714	0.3988	0.3878	0.3183
0.006	0.5424	0.5426	0.4656	0.5425	0.4701	0.5169
0.008	0.5446	0.5831	0.5589	0.5446	0.5278	0.9022
0.010	0.6248	0.6195	0.5974	0.6255	0.5893	0.7700
0.012	0.6966	0.6747	0.6634	0.6979	0.6533	0.8804
0.014	0.7587	0.7453	0.7394	0.7608	0.7467	0.8526
0.016	0.8454	0.8317	0.8274	0.8464	0.8584	0.8370
0.018	0.9010	0.8899	0.8869	0.9023	0.9003	0.9555
0.020	0.8752	1.0719	1.0731	0.8765	1.1054	1.1730
0.022	1.0143	1.0278	1.0247	1.0164	1.0527	1.0934
0.024	1.1325	1.1297	1.1276	1.1320	1.1636	1.2171
0.026	1.1870	1.2042	1.2017	1.1871	1.2429	1.2605
0.028	1.1301	1.4505	1.4558	1.1322	1.5210	1.4599
0.030	1.3562	1.3820	1.3797	1.3567	1.4203	1.3904
0.032	1.3159	1.6609	1.6639	1.3179	1.7611	1.6298
0.034	1.4223	1.7577	1.7626	1.4252	1.8307	1.7891
0.036	1.5290	1.9130	1.9199	1.5350	2.0361	1.9004
0.038	1.7850	1.8292	1.8263	1.7868	1.9271	1.7690
0.040	1.8595	1.9383	1.9344	1.8579	2.0596	1.8564

TABLE 4.23. Values of $d^2_{A,\mathrm{I}}, \ldots, d^2_{A,\mathrm{VI}}$ for the $r_s = 4.50$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{d^2_{A,\mathrm{I}}}$	$\boldsymbol{d^2_{A,\mathrm{II}}}$	$\boldsymbol{d^2_{A,\mathrm{III}}}$	$\boldsymbol{d^2_{A,\mathrm{IV}}}$	$\boldsymbol{d^2_{A,\mathrm{V}}}$	$\boldsymbol{d^2_{A,\mathrm{VI}}}$
0.002	0.2197	0.6236	0.1865	0.2205	0.2050	0.0489
0.004	0.3095	0.4934	0.2830	0.3104	0.3005	0.2068
0.006	0.4229	0.4392	0.3475	0.4231	0.3557	0.3757
0.008	0.5025	0.4486	0.3886	0.5024	0.3945	0.4635
0.010	0.4563	0.4656	0.4296	0.4567	0.4228	0.6413
0.012	0.4906	0.4900	0.4711	0.4921	0.4609	0.7423
0.014	0.5485	0.5308	0.5206	0.5504	0.5252	0.6627
0.016	0.5842	0.5870	0.5797	0.5854	0.6044	0.5963
0.018	0.6181	0.6222	0.6174	0.6191	0.6278	0.7056
0.020	0.6513	0.6533	0.6480	0.6535	0.6274	1.0109
0.022	0.6941	0.7074	0.7024	0.6962	0.7199	0.7956
0.024	0.7957	0.7752	0.7724	0.7958	0.7986	0.8664
0.026	0.8096	0.8205	0.8178	0.8106	0.8393	0.9334
0.028	0.8786	0.8780	0.8767	0.8793	0.8945	0.9714
0.030	0.9390	0.9419	0.9397	0.9402	0.9689	0.9721
0.032	1.0320	1.0262	1.0248	1.0352	1.0803	1.0589
0.034	1.0663	1.1107	1.1096	1.0697	1.1715	1.1559
0.036	1.1646	1.1385	1.1366	1.1673	1.1579	1.3337
0.038	1.1945	1.2539	1.2510	1.1953	1.3262	1.2057
0.040	1.2930	1.3260	1.3225	1.2918	1.4147	1.2639

TABLE 4.24. Values of $d^2_{A,\mathrm{I}}, \ldots, d^2_{A,\mathrm{VI}}$ for the $r_s = 5.00$ mm case

$\boldsymbol{\tau}$ [s]	$\boldsymbol{d^2_{A,\mathrm{I}}}$	$\boldsymbol{d^2_{A,\mathrm{II}}}$	$\boldsymbol{d^2_{A,\mathrm{III}}}$	$\boldsymbol{d^2_{A,\mathrm{IV}}}$	$\boldsymbol{d^2_{A,\mathrm{V}}}$	$\boldsymbol{d^2_{A,\mathrm{VI}}}$
0.002	0.1864	0.6023	0.1494	0.1873	0.1660	0.0193
0.004	0.2539	0.4571	0.2203	0.2546	0.2374	0.1298
0.006	0.3523	0.3747	0.2653	0.3526	0.2753	0.2725
0.008	0.3920	0.3665	0.2909	0.3915	0.2975	0.3634
0.010	0.3322	0.3739	0.3174	0.3331	0.3113	0.5777
0.012	0.3504	0.3750	0.3441	0.3516	0.3338	0.6877
0.014	0.3965	0.3948	0.3769	0.3985	0.3796	0.5471
0.016	0.4207	0.4300	0.4171	0.4222	0.4368	0.4409
0.018	0.4586	0.4498	0.4409	0.4596	0.4497	0.5460
0.020	0.4764	0.4693	0.4603	0.4779	0.4408	0.9488
0.022	0.5006	0.5052	0.4952	0.5017	0.5036	0.6356
0.024	0.5642	0.5470	0.5424	0.5643	0.5600	0.6552
0.026	0.5681	0.5749	0.5708	0.5684	0.5798	0.7481
0.028	0.6303	0.6117	0.6098	0.6316	0.6192	0.7521
0.030	0.6447	0.6575	0.6551	0.6465	0.6742	0.7256
0.032	0.7361	0.7140	0.7125	0.7385	0.7514	0.7666
0.034	0.7761	0.7735	0.7721	0.7794	0.8145	0.8372
0.036	0.7976	0.7895	0.7880	0.7988	0.7928	1.0841
0.038	0.8663	0.8766	0.8739	0.8666	0.9294	0.8489
0.040	0.9556	0.9241	0.9212	0.9539	0.9882	0.8890

TABLE 4.25. Values of $d^2_{A,\mathrm{I}}, \ldots, d^2_{A,\mathrm{VI}}$ for the $r_s = 5.50$ mm case

Theoretical Explanation of SNR_λ^2 *Behavior*

It is now time to discuss the behavior of SNR_λ^2 as the exposure time τ is changed. We will start this treatment by deriving an approximate expression that justifies our simulation results for large values of τ. For smaller values of τ, our approximate expression breaks down and we will rely on simulation studies to draw conclusions.

Consider first the SKE/BKE case, for which both the signal and the background are known. We will denote with $\boldsymbol{f}_0$ the object under the signal-absent hypothesis H_0 and with $\boldsymbol{f}_1 = \boldsymbol{f}_0 + \Delta\boldsymbol{f}$ the object under the signal-present hypothesis H_1. The SKE/BKE log-likelihood ratio can thus be written as [5]

$$\begin{aligned}\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) &= \ln\left[\frac{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0 + \Delta\boldsymbol{f})}{\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)}\right] = \\ &= \ln\left[\frac{\frac{\tau^J}{J!}\exp\left[-\tau\int_{\mathrm{FOV}}\big(f_0+\Delta f\big)(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right]}{\frac{\tau^J}{J!}\exp\left[-\tau\int_{\mathrm{FOV}}f_0(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right]}\right] + \\ &\quad + \ln\left[\prod_{j=1}^{J}\frac{\int_{\mathrm{FOV}}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\big(f_0+\Delta f\big)(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})f_0(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}\right] = \\ &= -\int_{\mathrm{FOV}}\Delta f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \sum_{j=1}^{J}\ln\left[1 + \frac{\int_{\mathrm{FOV}}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\Delta f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})f_0(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}\right].\end{aligned}$$

The last line in the expression above emphasizes the fact that $\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)$ is the sum of a large number of independent and identically distributed random variables. Therefore, by the central limit theorem [126–128], the statistics of $\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)$ approximate those of normally-distributed random variables.

Normally distributed log-likelihoods enjoy many interesting properties [65], one of which can be summarized as

$$\sigma^2_{\lambda_{\mathrm{BKE}}|\boldsymbol{f}_0} = \sigma^2_{\lambda_{\mathrm{BKE}}|\boldsymbol{f}_1} = 2\left\langle\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)\right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_1} = -2\left\langle\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)\right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_0}.$$

To clarify our notation, recall first that $\hat{\mathscr{A}}$ statistically conditioned on the known object $\boldsymbol{f}_k$, is random. Hence, the function $\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)$ applied to random $\hat{\mathscr{A}}$ is a random variable, and we can calculate its mean and variance. In other words, with $\sigma^2_{\lambda_{\mathrm{BKE}}|\boldsymbol{f}_k}$ we denote the standard deviation of $\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)$ for $\hat{\mathscr{A}}$ statistically conditioned on the object $\boldsymbol{f}_k$. Similarly, $\left\langle \lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_k}$ denotes the mean of $\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0)$ for $\hat{\mathscr{A}}$ conditioned on object $\boldsymbol{f}_k$. The result above shows that all we need to calculate $\mathrm{SNR}^2_{\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}}|\boldsymbol{f}_0)}$ according to (2.4) is $\left\langle \lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_0}$:

$$\begin{aligned}\mathrm{SNR}^2_{\lambda_{\mathrm{BKE}}(\hat{\mathscr{A}}|\boldsymbol{f}_0)} &= \frac{\left[\left\langle \lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_1} - \left\langle \lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_0}\right]^2}{\frac{1}{2}\left(\sigma^2_{\lambda_{\mathrm{BKE}}|\boldsymbol{f}_1} + \sigma^2_{\lambda_{\mathrm{BKE}}|\boldsymbol{f}_0}\right)} = \\ &= -2\left\langle \lambda_{\mathrm{BKE}}(\hat{\mathscr{A}} \mid \boldsymbol{f}_0) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}_0}.\end{aligned}$$

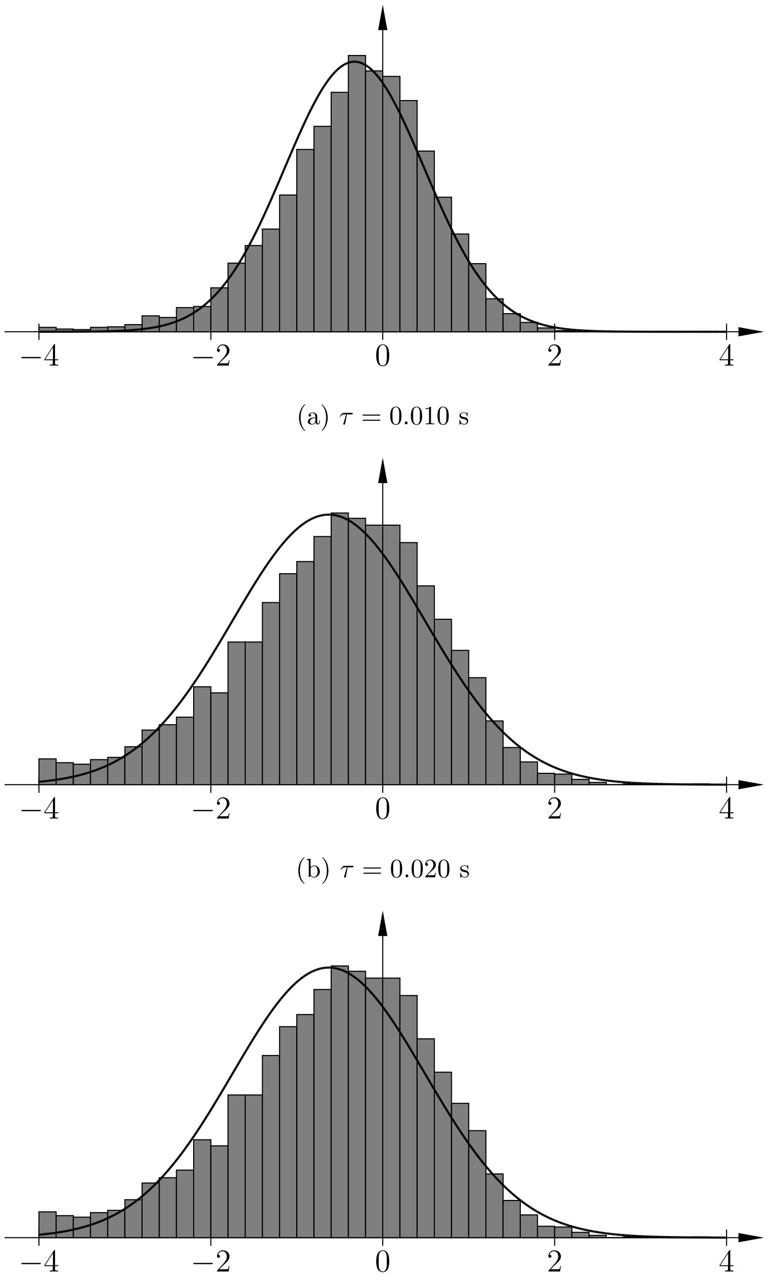
We now want to modify the result above and calculate the SNR^2_λ for the SKE/BKS case. The key difference is that the background randomness introduces statistical correlations between pairs of elements of the list $\hat{\mathscr{A}}$ [5]

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}, \hat{\boldsymbol{A}}^{(j')} \mid H_k) &= \int_\infty \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f})\mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{f})\mathrm{pr}(\boldsymbol{f} \mid H_k)\,\mathrm{d}\boldsymbol{f} \neq \\ &\neq \left[\int_\infty \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f})\mathrm{pr}(\boldsymbol{f} \mid H_k)\,\mathrm{d}\boldsymbol{f}\right]\left[\int_\infty \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{f})\mathrm{pr}(\boldsymbol{f} \mid H_k)\,\mathrm{d}\boldsymbol{f}\right] = \\ &= \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid H_k)\mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid H_k),\end{aligned}$$

so that the SKE/BKS log-likelihood

$$\lambda(\hat{\mathscr{A}}) = \ln\left[\frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)}\right] = \ln\left[\frac{\prod_{j=1}^J \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid H_1)}{\prod_{j=1}^J \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid H_0)}\right] = \sum_{j=1}^J \ln\left[\frac{\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid H_1)}{\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid H_0)}\right]$$

is no longer the sum of a large number of independent random quantities. However, though the central-limit theorem is usually stated for the sum of a large number of independent random variables, strict independence is not required [126]. In other words, although the elements of $\hat{\mathscr{A}}$ are no longer independent, the statistics of $\lambda(\hat{\mathscr{A}})$ still approximate those of normally-distributed random variables, as shown in Figure 4.19 through Figure 4.21. Data for these plots were obtained using our MCMC simulation code we discussed earlier in this section.

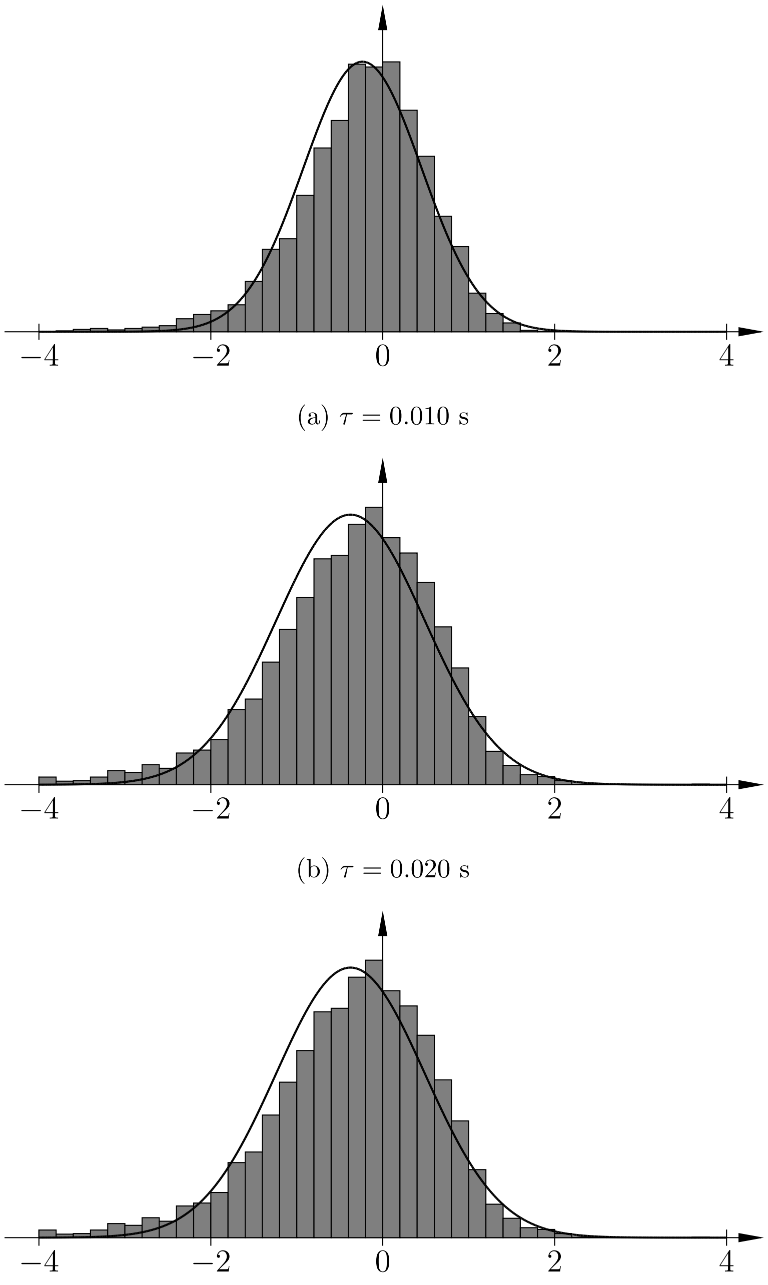


(a) $\tau = 0.010$ s

(b) $\tau = 0.020$ s

(c) $\tau = 0.030$ s

FIGURE 4.19. Histograms of $\lambda(\hat{\mathscr{A}})$ superimposed on the probability density function of normally-distributed random variables for the case $r_s = 4.50$ mm and different values of τ

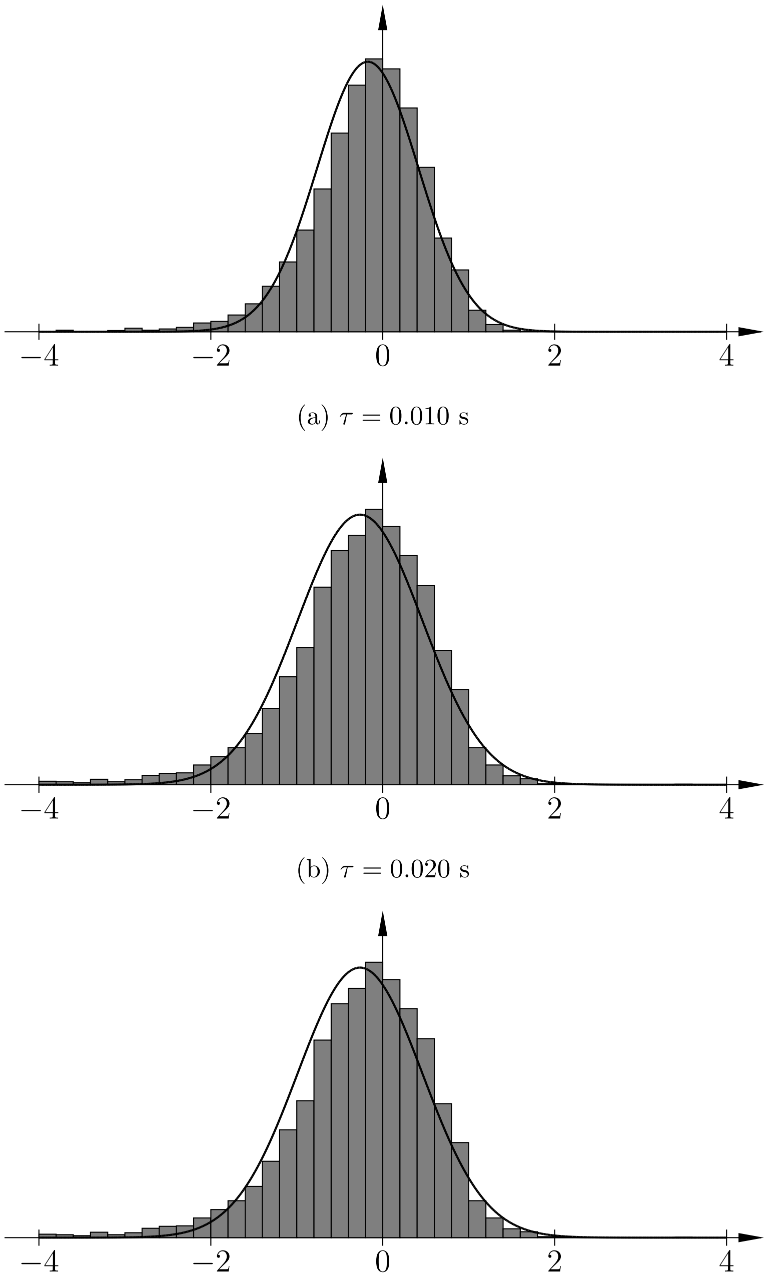


(a) $\tau = 0.010$ s

(b) $\tau = 0.020$ s

(c) $\tau = 0.030$ s

FIGURE 4.20. Histograms of $\lambda(\hat{\mathscr{A}})$ superimposed on the probability density function of normally-distributed random variables for the case $r_s = 5.00$ mm and different values of τ



(a) $\tau = 0.010$ s

(b) $\tau = 0.020$ s

(c) $\tau = 0.030$ s

FIGURE 4.21. Histograms of $\lambda(\hat{\mathscr{A}})$ superimposed on the probability density function of normally-distributed random variables for the case $r_s = 5.50$ mm and different values of τ

By the approximate normality of the statistics of $\lambda(\hat{\mathscr{A}})$ under hypothesis H_k, we can write

$$\sigma^2_{\lambda(\hat{\mathscr{A}})|H_0} \approx \sigma^2_{\lambda(\hat{\mathscr{A}})|H_1} \approx 2\left\langle \lambda(\hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|H_1} \approx -2\left\langle \lambda(\hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|H_0},$$

so that [5]:

$$\mathrm{SNR}^2_\lambda = \frac{\left[\left\langle \lambda(\hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|H_1} - \left\langle \lambda(\hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|H_0}\right]^2}{\frac{1}{2}\left(\sigma^2_{\lambda(\hat{\mathscr{A}})|H_1} + \sigma^2_{\lambda(\hat{\mathscr{A}})|H_0}\right)} \approx -2\left\langle \lambda(\hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|H_0}. \tag{4.55}$$

To evaluate the expectation above, we first go back to $\Lambda(\hat{\mathscr{A}})$ for a moment. From the definition of list-mode likelihood and using $\mathrm{pr}(\hat{\mathscr{A}} \mid H_k) = \langle \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) \rangle_{\boldsymbol{f}|H_k}$, we get [5]

$$\Lambda(\hat{\mathscr{A}}) = \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_1)}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} = \frac{\langle \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f} + \Delta\boldsymbol{f}) \rangle_{\boldsymbol{f}|H_0}}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)}.$$

Because the signal $\Delta\boldsymbol{f}$ is weak, we can expand $\Lambda(\hat{\mathscr{A}})$ in Taylor series around $\boldsymbol{f}$:

$$\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f} + \Delta\boldsymbol{f}) = \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) + \Delta\boldsymbol{f}^\dagger \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} + \frac{1}{2}\Delta\boldsymbol{f}^\dagger \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \Delta\boldsymbol{f} + \ldots,$$

in which $\Delta\boldsymbol{f}^\dagger$ stands for the adjoint of $\Delta\boldsymbol{f}$, and $\frac{\partial}{\partial \boldsymbol{f}}$ denotes the Fréchet derivative [69, 314], commonly used to introduce the concept of functional derivative. If we keep the terms up to second order in $\Delta\boldsymbol{f}$, we get

$$\begin{aligned}\langle \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f} + \Delta\boldsymbol{f}) \rangle_{\boldsymbol{f}|H_0} \approx \langle \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) \rangle_{\boldsymbol{f}|H_0} + \Delta\boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} + \\ + \frac{1}{2}\Delta\boldsymbol{f}^\dagger \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \Delta\boldsymbol{f},\end{aligned}$$

which allows us to write [5]

$$\begin{aligned}\Lambda(\hat{\mathscr{A}}) \approx 1 + \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta\boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} + \\ + \frac{1}{2}\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta\boldsymbol{f}^\dagger \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \Delta\boldsymbol{f}.\end{aligned}$$

If we take the logarithm of $\Lambda(\hat{\mathscr{A}})$ and use $\ln(1+\varepsilon) = \varepsilon - \frac{1}{2}\varepsilon^2 + \ldots$, we get

$$\lambda(\hat{\mathscr{A}}) =$$

$$= \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \left(\Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} + \frac{1}{2} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \Delta \boldsymbol{f} \right) +$$

$$-\frac{1}{2} \left[\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \left(\Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} + \frac{1}{2} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \Delta \boldsymbol{f} \right) \right]^2 + \ldots,$$

which can be further approximated as [5]

$$\begin{aligned} \lambda(\hat{\mathscr{A}}) \approx & \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} + \\ & + \frac{1}{2} \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \Delta \boldsymbol{f} + \\ & - \frac{1}{2} \left[\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} \right]^2 . \end{aligned} \tag{4.56}$$

The first two terms, when averaged over $\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)$, evaluate to zero [5]:

$$\begin{aligned} \left\langle \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} \right\rangle_{\hat{\mathscr{A}}|H_0} &= \int_\infty \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} \mathrm{d}\hat{\mathscr{A}} = \\ &= \left\langle \frac{\partial}{\partial \boldsymbol{f}} \int_\infty \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) \, \mathrm{d}\hat{\mathscr{A}} \right\rangle_{\boldsymbol{f}|H_0} = 0, \end{aligned}$$

and

$$\begin{aligned} \left\langle \frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \right\rangle_{\hat{\mathscr{A}}|H_0} &= \int_\infty \left\langle \frac{\partial^2 \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \right\rangle_{\boldsymbol{f}|H_0} \mathrm{d}\hat{\mathscr{A}} = \\ &= \left\langle \frac{\partial^2}{\partial \boldsymbol{f} \partial \boldsymbol{f}^\dagger} \int_\infty \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) \, \mathrm{d}\hat{\mathscr{A}} \right\rangle_{\boldsymbol{f}|H_0} = 0. \end{aligned}$$

Hence, from (4.55),

$$\mathrm{SNR}_\lambda^2 = \left\langle \left[\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} \right]^2 \right\rangle_{\hat{\mathscr{A}}|H_0} . \tag{4.57}$$

To explicitly evaluate the Fréchet derivative in (4.56), we can make use of (4.20), (4.21), and (4.22) to write

$$\begin{aligned}\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) &= \Pr(J \mid \boldsymbol{f}) \prod_{j=1}^{J} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f}) = \\ &= \frac{\tau^J}{J!} e^{-\overline{J}(\boldsymbol{f},\tau)} \prod_{j=1}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) s(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r},\end{aligned}$$

so that

$$\Delta \boldsymbol{f}^\dagger \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} = -\tau \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} +$$
$$+ \frac{\tau^J}{J!} e^{-\overline{J}(\boldsymbol{f},\tau)} \sum_{j=1}^{J} \left\{ \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \prod_{\substack{j'=1 \\ j' \neq j}}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{r}) f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \right\}.$$

Taking expectation with respect to $\boldsymbol{f}$ conditioned on H_0 yields

$$\Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} = -\tau \mathrm{pr}(\hat{\mathscr{A}} \mid H_0) \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} +$$
$$+ \frac{\tau^J}{J!} \left\langle e^{-\overline{J}(\boldsymbol{f},\tau)} \sum_{j=1}^{J} \left\{ \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \prod_{\substack{j'=1 \\ j' \neq j}}^{J} \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{r}) f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \right\} \right\rangle_{\boldsymbol{f}|H_0}.$$

If we divide the expression above by $\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)$ and use (4.20) and

$$\begin{aligned}\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) &= \Pr(J \mid \boldsymbol{f}) \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J) = \\ &= \frac{[\overline{J}(\boldsymbol{f},\tau)]^J}{J!} e^{-\overline{J}(\boldsymbol{f},\tau)} \prod_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}},\end{aligned}$$

we obtain [5]

$$\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^\dagger \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} = -\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} +$$

$$
+\frac{\left\langle [\overline{J}(\boldsymbol{f},\tau)]^J e^{-\overline{J}(\boldsymbol{f},\tau)} \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}|\boldsymbol{r})\Delta f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}|\boldsymbol{r}) f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \prod_{j'=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')}|\boldsymbol{r}) f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\boldsymbol{f}|H_0}}{\left\langle [\overline{J}(\boldsymbol{f},\tau)]^J e^{-\overline{J}(\boldsymbol{f},\tau)} \prod_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}|\boldsymbol{r}) f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\boldsymbol{f}|H_0}} =
$$

$$
-\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \frac{\left\langle [\overline{J}(\boldsymbol{f},\tau)]^J e^{-\overline{J}(\boldsymbol{f},\tau)} \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J) \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}|\boldsymbol{r})\Delta f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)}|\boldsymbol{r}) f(\boldsymbol{r})s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\boldsymbol{f}|H_0}}{\left\langle [\overline{J}(\boldsymbol{f},\tau)]^J e^{-\overline{J}(\boldsymbol{f},\tau)} \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J) \right\rangle_{\boldsymbol{f}|H_0}}. \tag{4.58}
$$

At this point, it is worth recalling that $\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J)$ is the probability density function of the list-mode data $\hat{\mathscr{A}}$ for the object $\boldsymbol{f}$ and when the number of entries in $\hat{\mathscr{A}}$ is J. As J increases, we would expect that such list would convey more and more "knowledge" about the object $\boldsymbol{f}$ being imaged [315]. If the imaging system does not have null functions, we can make this concept more formal by writing

$$
\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J) \approx C\delta\big(\boldsymbol{f} - \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}\big), \qquad |\hat{\mathscr{A}}| \gg 1, \tag{4.59}
$$

in which $\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}$ is the object that maximizes $\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J)$ as a function of $\boldsymbol{f}$, $|\hat{\mathscr{A}}|$ denotes the number of entries in the list $\hat{\mathscr{A}}$, and $\delta(\boldsymbol{f} - \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}})$ has to be understood as 1 if $\boldsymbol{f} \equiv \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}$, 0 otherwise. In other words, $\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}$ is the maximum-likelihood estimate of $\boldsymbol{f}$ given the data $\hat{\mathscr{A}}$ and the model $\mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}, J)$. The constant C that appears in (4.59) is given by

$$
C \approx \frac{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0, J)}{\mathrm{pr}(\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0)}.
$$

By the properties of the delta function, the expectations that appear in (4.58) are trivially calculated, and we obtain [5]

$$
\frac{1}{\mathrm{pr}(\hat{\mathscr{A}} \mid H_0)} \Delta \boldsymbol{f}^{\dagger} \left\langle \frac{\partial \mathrm{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})}{\partial \boldsymbol{f}} \right\rangle_{\boldsymbol{f}|H_0} =
$$

$$
= -\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}.
$$

By inserting the expression above in (4.57), we get

$$\begin{aligned}
\mathrm{SNR}_\lambda^2 &= \left\langle \left[-\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} + \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right]^2 \right\rangle_{\hat{\mathscr{A}} \mid H_0} = \\
&= \left[-\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \right]^2 + \\
&\quad - 2\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \left\langle \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\mathscr{A}} \mid H_0} + \\
&\quad + \left\langle \sum_{j=1}^{J} \sum_{j'=1}^{J} \left[\frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right] \left[\frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j')} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right] \right\rangle_{\hat{\mathscr{A}} \mid H_0}.
\end{aligned}$$

Let us now evaluate the expectations that appear in the expression above:

$$\begin{aligned}
&\left\langle \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\mathscr{A}} \mid H_0} = \\
&\qquad = \left\langle \left\langle \sum_{j=1}^{J} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}}^{(j)} \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0} = \\
&= \left\langle \sum_{J=0}^{\infty} \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) \sum_{j=1}^{J} \int_{\infty} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}})\, \mathrm{d}\hat{\boldsymbol{A}}^{(j)} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0} = \\
&\qquad = \left\langle \sum_{J=0}^{\infty} \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) \sum_{j=1}^{J} \int_{\infty} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \times \right. \\
&\qquad\qquad \left. \times \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}\, \mathrm{d}\hat{\boldsymbol{A}}^{(j)} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0} = \\
&\qquad = \left\langle \sum_{J=0}^{\infty} \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) \sum_{j=1}^{J} \int_{\infty} \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}\, \mathrm{d}\hat{\boldsymbol{A}}^{(j)} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0} = \\
&\qquad = \left\langle \sum_{J=0}^{\infty} J \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) \frac{\int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}} \mid H_0} = \tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r},
\end{aligned}$$

in which we used $\int_{\infty} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r})\, \mathrm{d}\hat{\boldsymbol{A}}^{(j)} = 1$ and

$$\overline{J}(\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}, \tau) = \sum_{J=0}^{\infty} J \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) = \tau \int_{\mathrm{FOV}} \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}. \tag{4.60}$$

Similarly

$$\left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\mathscr{A}}|H_0} = \frac{\tau \int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\overline{J}(\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}, \tau)}. \tag{4.61}$$

Using the results found above and the independence of the entries in $\hat{\mathscr{A}}$, we can now rewrite the SNR_{λ}^2 as [5]

$$\begin{aligned}
\mathrm{SNR}_{\lambda}^2 = -\tau^2 \left[\int_{\mathrm{FOV}} \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \right]^2 + \\
+ \left\langle \sum_{J=0}^{\infty} \mathrm{Pr}(J \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}) \left\{ (J^2 - J) \left[\left\langle \frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right\rangle_{\hat{\boldsymbol{A}}|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} \right]^2 + \right.\right. \\
\left.\left. + J \left\langle \left[\frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right]^2 \right\rangle_{\hat{\boldsymbol{A}}|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} \right\} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}|H_0},
\end{aligned}$$

in which $\hat{\boldsymbol{A}}$ is randomly drawn according to $\mathrm{pr}(\hat{\boldsymbol{A}} \mid \hat{\boldsymbol{f}}_{\hat{\mathscr{A}}})$. Because J conditioned on $\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}$ is a Poisson random variable [127, 128], we obtain

$$\begin{aligned}
\langle J^2 - J \rangle_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} &= \sigma^2_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} + \langle J \rangle^2_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} - \langle J \rangle_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} = \\
&= \langle J \rangle_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} + \langle J \rangle^2_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} - \langle J \rangle_{J|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} = \left[\overline{J}(\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}, \tau) \right]^2.
\end{aligned}$$

Using the result above and with the help of (4.61), the SNR_{λ}^2 is rewritten as [5]

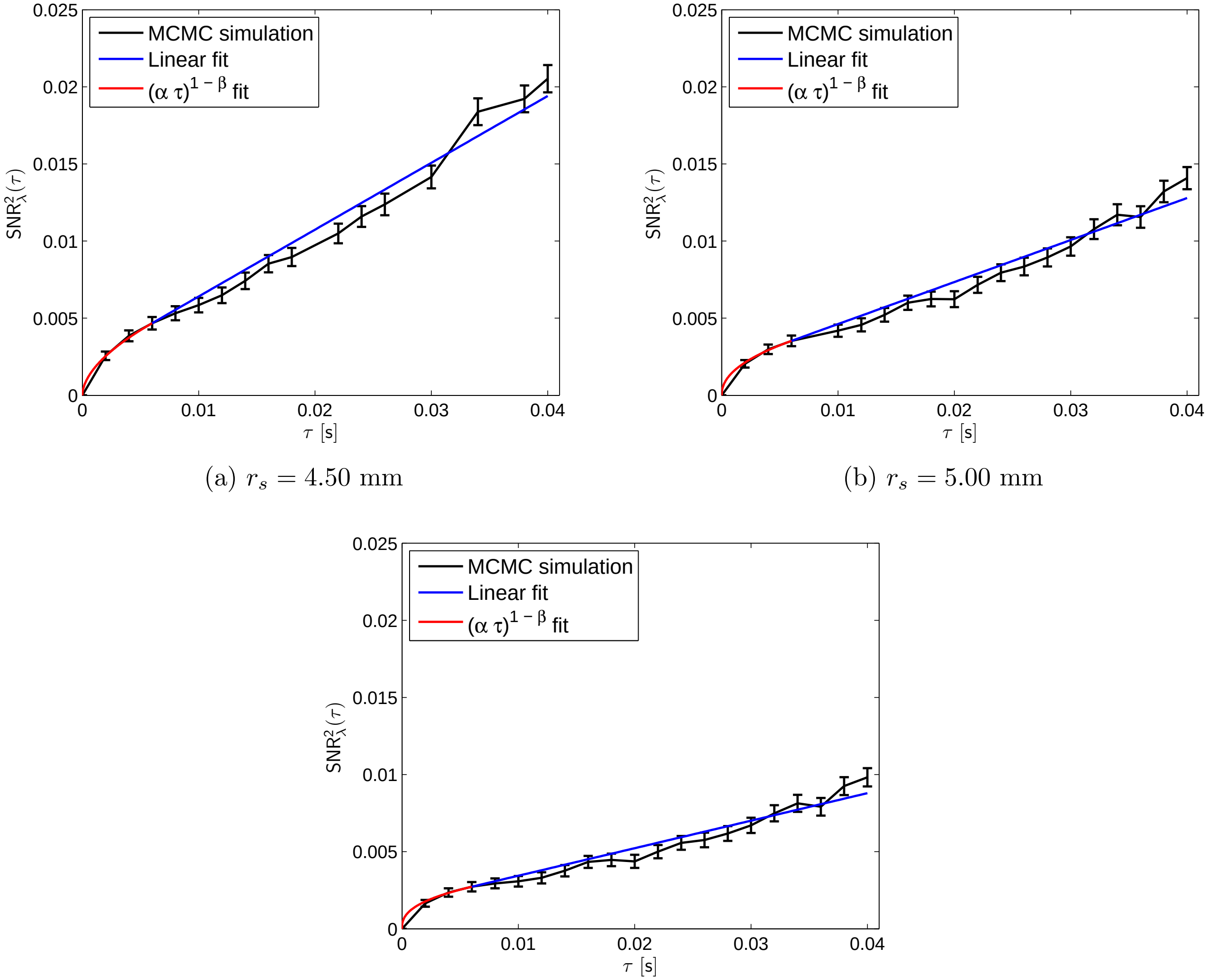
$$\begin{aligned}
\mathrm{SNR}_{\lambda}^2 &= \left\langle \overline{J}(\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}, \tau) \left\langle \left[\frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right]^2 \right\rangle_{\hat{\boldsymbol{A}}|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}|H_0} = \\
&= \tau \left\langle \left[\int_{\mathrm{FOV}} \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \right] \left\langle \left[\frac{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \Delta f(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}}{\int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) \hat{f}_{\hat{\mathscr{A}}}(\boldsymbol{r}) s(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}} \right]^2 \right\rangle_{\hat{\boldsymbol{A}}|\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}} \right\rangle_{\hat{\boldsymbol{f}}_{\hat{\mathscr{A}}}|H_0},
\end{aligned} \tag{4.62}$$

in which the last form follows from (4.60). The remarkable result we just obtained shows that so long as (4.59) is satisfied, SNR_{λ}^2 is proportional to the exposure time τ.

The numerical values of SNR_{λ}^2 reported in Table 4.7 confirm the theoretical result of (4.62). Indeed, as we can see from Figure 4.22 and with the exclusion of a few data points obtained for small values of τ, the values of SNR^2_λ are approximately aligned along straight lines. More specifically, starting from the values reported in Table 4.7, we first pruned the data by removing data points that failed the consistency checks reported in Table 4.4 through Table 4.6. Next, by visually inspecting the plots of SNR^2_λ (red line), we identified the value τ_0 such that $\mathrm{SNR}^2_\lambda \propto \tau$ for $\tau \geqslant \tau_0$, and then we performed a least square fit of SNR^2_λ with a straight line (blue line) of the data from Table 4.7. Finally, by conjecturing that for $\tau < \tau_0$, the SNR^2_λ behaves, approximatively, as $\mathrm{SNR}^2_{\mathrm{Hot}}$—see (4.35)—we considered a function of the form $(\alpha\tau)^{1-\beta}$ and we chose α and β so that, at $\tau = \tau_0$, the value of the function $(\alpha\tau)^{1-\beta}$ and its first derivative match the corresponding values obtained for the linear fit. Plots of $(\alpha\tau)^{1-\beta}$ appear in black in Figure 4.22. The overall fits we obtained match pretty well the data points obtained via our MCMC simulation code.

After examining the performance of the ideal observer in this subsection and the one of the Hotelling observer in the previous subsection, we can think about comparing the two and report them on the same plot. To this point, we need to recall first that the "weak signal" condition of (4.30) was needed to make plots of Figure 4.14 valid. We satisfied condition (4.30) by taking $b_s = 100\ \mathrm{s}^{-1}$ and $b_0 = 1000\ \mathrm{s}^{-1}$ for the Hotelling observer performance analysis. On the other hand, the MCMC results of Figure 4.22 required us to use a larger value of b_s to make the simulation more numerically stable by increasing the average number of photons emitted by the signal (hence reducing the number of cases in which no photon at all is emitted from the signal during the entire exposure time τ; this problem is especially pronounced for small values of τ). This is the reason why, as reported in Table 4.2, we set $b_s = b_0 = 1000\ \mathrm{s}^{-1}$ in our MCMC code. A direct comparison between Figure 4.14 and Figure 4.22 would, therefore, be unfair as results for the two observers were obtained under different conditions.

We can however salvage our analysis if we study how SNR^2_λ changes as the signal intensity b_s is varied. To do this, we considered once again our MCMC code and ran it for $\tau = 0.020$ s, $r_s = 5.00$ mm, and $b_s = 100, 200, 300, \ldots, 2000\ \mathrm{s}^{-1}$. The remaining



(a) $r_s = 4.50$ mm

(b) $r_s = 5.00$ mm

(c) $r_s = 5.50$ mm

FIGURE 4.22. Plots of values of SNR^2_λ (and their fits) calculated using our MCMC simulation code

parameters were the same as reported in Table 4.2. The values of SNR_λ^2 for different values of b_s have been plotted in Figure 4.23 and fit with a quadratic function of the form $\mathrm{SNR}_\lambda^2 \approx \gamma b_s^2$ for an appropriate value of γ. From Figure 4.23, we see a somewhat satisfactory fit of the values of SNR_λ^2. Further investigation is required.

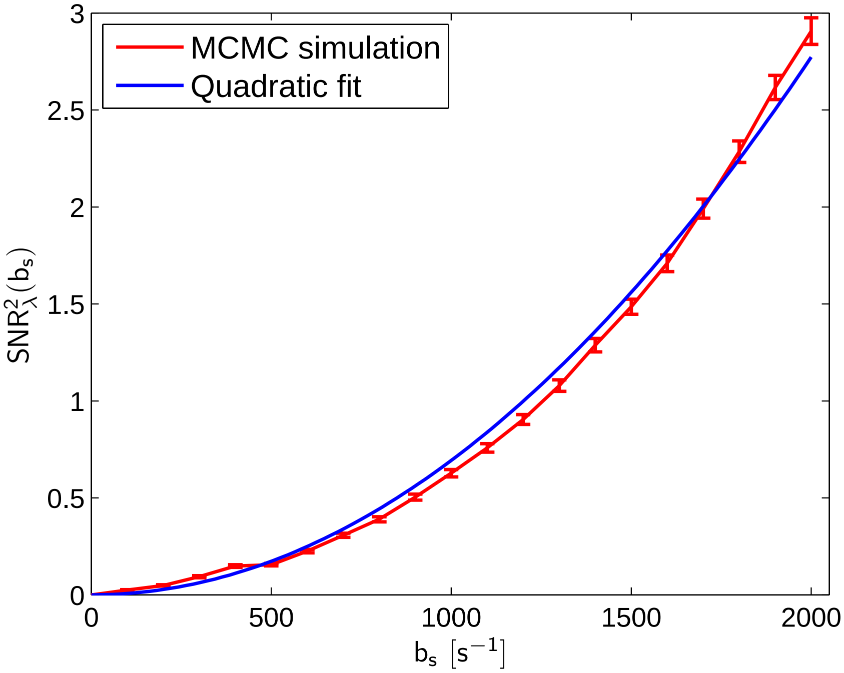


FIGURE 4.23. Plot of SNR_λ^2 for different values of b_s, along with their quadratic least-square fit

Using the result of Figure 4.23, we adjusted the plots of SNR_λ^2 in Figure 4.22 to the case $b_s = 100\ \mathrm{s}^{-1}$, which is the value we had used for the Hotelling observer task performance.

Finally, we produced plots in which the "adjusted" SNR_λ^2 is compared to $\mathrm{SNR}_{\mathrm{Hot}}^2$, as shown in Figure 4.24. As expected, the ideal observer is found to outperform the Hotelling observer and the gap in performance between the two observers increases as the exposure time τ increases (equivalently, as the mean number of collected photons increases).

We finally note that the plots in Figure 4.24 can be used as an additional test of consistency for our MCMC code for the calculation of $\mathrm{SNR}_{\lambda}^{2}$: although values of $\mathrm{SNR}_{\mathrm{Hot}}^{2}$ were calculated analytically and those for $\mathrm{SNR}_{\lambda}^{2}$ resulted from simulation code, the condition $\mathrm{SNR}_{\lambda}^{2} \geqslant \mathrm{SNR}_{\mathrm{Hot}}^{2}$ is always satisfied. Plots in Figure 4.24 also seem to suggest that for small values of τ (low mean counts), the performance of the ideal and Hotelling observes are comparable.

4.7 Estimability with List-Mode Data

In § 4.6.1 we introduced the concept of information content of a photon and we applied it to the estimation of numerical parameters from list-mode data. In this section, we provide a brief theoretical treatment of parameter estimation from list-mode data. For a more general discussion, the interested reader can consult [69]. An example relevant to the field of radiology is the estimation of the amount θ of object $\boldsymbol{f}$ within some region of interest (ROI):

$$\theta = \boldsymbol{\chi}^{\dagger}\boldsymbol{f} = \int_{\mathrm{FOV}} \chi(\boldsymbol{r}) f(\boldsymbol{r})\, \mathrm{d}^{3}\boldsymbol{r},$$

in which the vector $\boldsymbol{\chi}$ defines the region of interest. We say that the parameter θ is *estimable* if there exists an unbiased linear estimator $\hat{\theta}(\boldsymbol{u})$ of θ [69]:

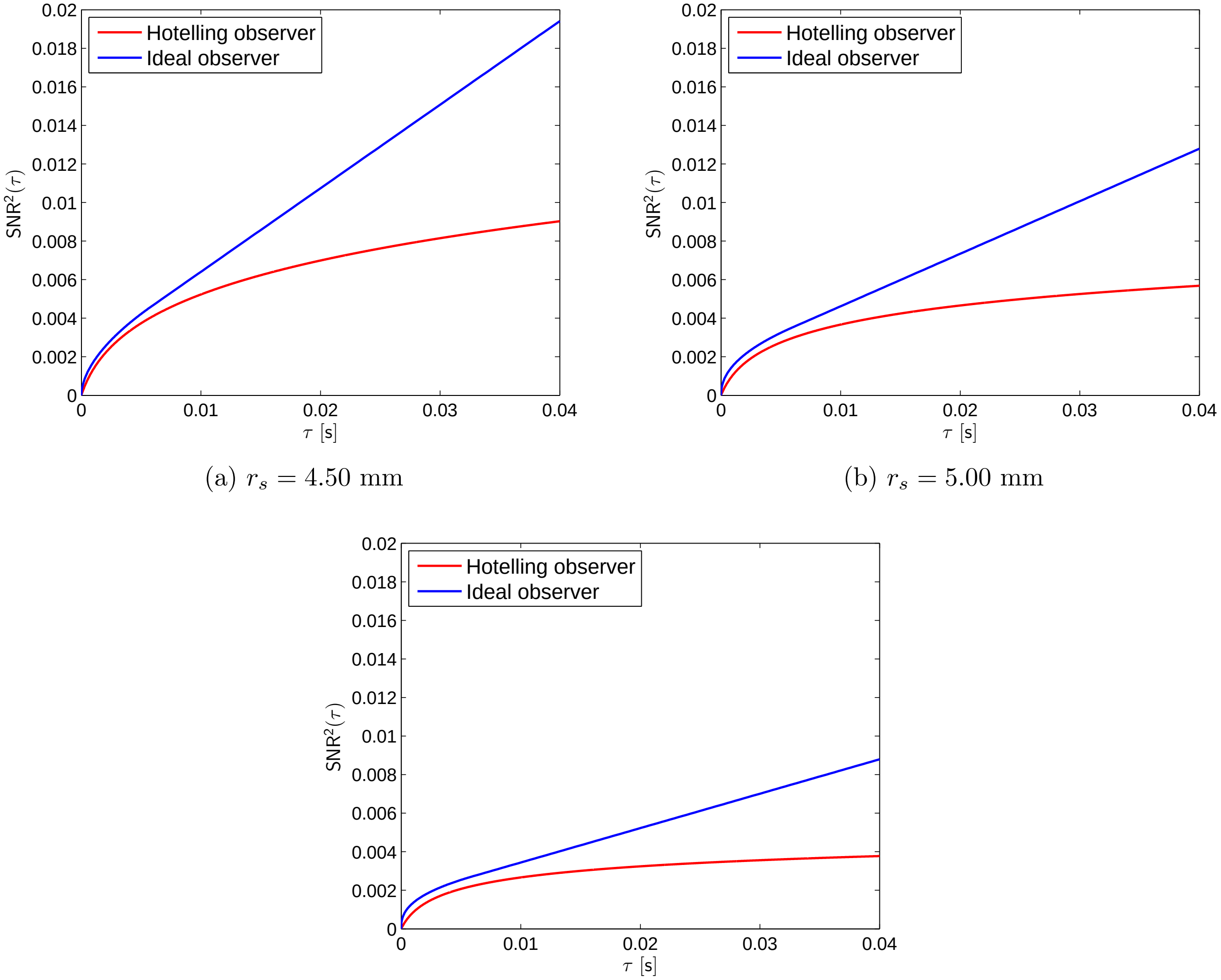
$$\theta = \left\langle \hat{\theta}(\boldsymbol{u}) \right\rangle,$$

in which $\boldsymbol{u}$ is the Poisson point process we defined in (4.19). Because the estimator $\hat{\theta}(\boldsymbol{u})$ is assumed linear, it must be of the form $\hat{\theta}(\boldsymbol{u}) = \boldsymbol{w}^{\dagger}\boldsymbol{u}$, for some vector $\boldsymbol{w}$. If we take expectation, we get

$$\left\langle \hat{\theta}(\boldsymbol{u}) \right\rangle = \boldsymbol{w}^{\dagger}\overline{\boldsymbol{u}} = \boldsymbol{w}^{\dagger}\big(\tau\boldsymbol{\mathcal{L}}\boldsymbol{f}\big) = \big(\tau\boldsymbol{\mathcal{L}}^{\dagger}\boldsymbol{w}\big)^{\dagger}\boldsymbol{f},$$

in which $\boldsymbol{\mathcal{L}}$ is the linear operator introduced in (4.23). It can be shown [316] that θ is estimable if and only if there exists a vector $\boldsymbol{w}$ that satisfies

$$\boldsymbol{\chi} = \tau\boldsymbol{\mathcal{L}}^{\dagger}\boldsymbol{w}.$$



(a) $r_s = 4.50$ mm

(b) $r_s = 5.00$ mm

(c) $r_s = 5.50$ mm

FIGURE 4.24. Comparison between $\mathrm{SNR}^2_{\mathrm{Hot}}$ and SNR^2_λ for the same signal detection problem

If such a vector $\boldsymbol{w}$ exists, then the estimator $\hat{\theta}(\boldsymbol{u})$ is automatically unbiased. Indeed

$$\Big\langle \hat{\theta}(\boldsymbol{u}) \Big\rangle = \boldsymbol{w}^\dagger \overline{\boldsymbol{u}} = \boldsymbol{w}^\dagger \big(\tau \mathcal{L} \boldsymbol{f}\big) = \big(\tau \mathcal{L}^\dagger \boldsymbol{w}\big)^\dagger \boldsymbol{f} = \boldsymbol{\chi}^\dagger \boldsymbol{f} = \theta.$$

If, on the other hand, there exists no $\boldsymbol{w}$ that satisfies $\boldsymbol{\chi} = \tau \mathcal{L}^\dagger \boldsymbol{w}$, the parameter θ is not estimable, which implies that any linear estimator of θ must be biased. Further details on estimability with list-mode data can be found in Chapter 4 of [283].

4.8 LMMLEM Reconstruction of Lumpy Backgrounds

In this section, we confront the problem of reconstructing an emission tomography image from list-mode data by performing maximum-likelihood estimation of a set of parameters that characterize the object being imaged. This way of proceeding differs from what is usually discussed in the literature, in which the object to reconstruct is often discretized and represented as a set of intensity values over a pixel or voxel grid.

To make the discussion more concrete, we still use the familiar lumpy background model of (4.25), in which the lumps are described by the list of K 2D lump centers $\boldsymbol{r}_k = (x_k, y_k)$:

$$\boldsymbol{\theta} = \big\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K\big\}.$$

We will also assume the setup of Figure 4.12 in which the imaging system and detector properties are statistically modeled as in (4.26). Our choice for the reconstruction algorithm is still the MLEM algorithm, which was derived in abstract form in § 3.4. Mathematically,

$$\hat{\boldsymbol{\theta}}^{(p+1)} = \arg\max_{\boldsymbol{\theta}} \Big\{ \langle \ln \mathrm{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) \rangle_{\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(p)}} \Big\},$$

where, using the nomenclature of § 3.1, $\boldsymbol{x}$ is the complete data, $\boldsymbol{y}$ is the incomplete data, and $\hat{\boldsymbol{\theta}}^{(p)}$ is the p^{th} estimate of the parameter $\boldsymbol{\theta}$. We will assume

$$\boldsymbol{x} = \big\{ \big(\boldsymbol{r}^{(1)}, k^{(1)}\big), \ldots, \big(\boldsymbol{r}^{(J)}, k^{(J)}\big) \big\},$$

in which J is the total number of points $\boldsymbol{r}^{(j)}$ in the list and the pair $\big(\boldsymbol{r}^{(j)}, k^{(j)}\big)$ denotes the 2D object point $\boldsymbol{r}^{(j)} = \big(x^{(j)}, y^{(j)}\big)$ at which a photon was emitted, along with the index $k^{(j)}$ of the lump that was responsible for the j^{th} photon emission. We will assume that index $k^{(j)}$ is a discrete random variable uniformly distributed over $\{1, \ldots, K\}$ and that, given the list $\boldsymbol{\theta}$ of lump locations $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K$ and index $k^{(j)}$, the point $\boldsymbol{r}^{(j)}$ is randomly drawn according to

$$\operatorname{pr}(\boldsymbol{r}^{(j)} \mid \boldsymbol{\theta}, k^{(j)}) = \frac{1}{2\pi r_b^2} \exp\left(-\frac{|\boldsymbol{r}^{(j)} - \boldsymbol{r}_{k^{(j)}}|^2}{2r_b^2}\right). \tag{4.63}$$

Notice that $\boldsymbol{\theta}$ could be estimated from $\boldsymbol{x}$ very easily: we just need to group points $\boldsymbol{r}^{(j)}$ with the same index $k^{(j)}$, and fit each of the K groups with the model $\operatorname{pr}(\boldsymbol{r}^{(j)} \mid \boldsymbol{\theta}, k^{(j)})$ above to find estimate $\hat{\boldsymbol{r}}_k$ of $\boldsymbol{r}_k$, for $k = 1, \ldots, K$. The imaging system, however, does not allow to discriminate among photons based on which lump they were emitted from, and points $\boldsymbol{r}^{(j)}$ are further perturbed according to the statistical model in (4.26). In other words, the data on which the reconstruction algorithm operates are

$$\boldsymbol{y} = \hat{\mathscr{A}} = \big\{\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}\big\},$$

in which our notation allows for the possibility that the number J' of points $\hat{\boldsymbol{R}}^{(j)}$ to differ from the number J of points $\boldsymbol{r}^{(j)}$. To find an iterative expression for our LMMLEM problem, we start by using Bayes' theorem to write

$$\operatorname{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) = \operatorname{pr}(\boldsymbol{y} \mid \boldsymbol{x}) \operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{\theta}).$$

Expanding $\operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{\theta})$ gives

$$\begin{aligned}\operatorname{pr}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \Pr(J \mid \boldsymbol{\theta}) \prod_{j=1}^{J} \operatorname{pr}(\boldsymbol{r}^{(j)}, k^{(j)} \mid \boldsymbol{\theta}) = \\ = \frac{[Kb_0\tau]^J}{J!} e^{-Kb_0\tau} \prod_{j=1}^{J} \left(\sum_{k=1}^{K} \operatorname{pr}(\boldsymbol{r}^{(j)}, k^{(j)} \mid \boldsymbol{\theta}, k) \Pr(k)\right),\end{aligned}$$

where we have assumed

$$\overline{J}(b_{\boldsymbol{\theta}}, \tau) = \tau \sum_{k=1}^{K} \int_{\text{FOV}} \ell(\boldsymbol{r} - \boldsymbol{r}_k)\, \mathrm{d}\boldsymbol{r} \approx Kb_0\tau.$$

Notice that $\mathrm{pr}(\boldsymbol{r}^{(j)}, k^{(j)} \mid \boldsymbol{\theta}, k) = \delta_{\mathrm{Kron}}(k - k^{(j)})\,\mathrm{pr}(\boldsymbol{r}^{(j)} \mid \boldsymbol{r}_{k^{(j)}})$ and recall $\Pr(k) = 1/K$ so that

$$\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \frac{e^{-Kb_0\tau}}{J!} \left[\frac{b_0\tau}{2\pi r_b^2}\right]^J \exp\left(-\frac{1}{2r_b^2}\sum_{j=1}^{J} \left|\boldsymbol{r}^{(j)} - \boldsymbol{r}_{k^{(j)}}\right|^2\right),$$

in which we also used (4.63). On the other had, by (4.26), we can write

$$\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{x}) = \delta_{\mathrm{Kron}}(J - J') \left[\frac{1}{2\pi\sigma^2}\right]^J \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{J} \left|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}^{(j)}\right|^2\right),$$

where we used the factor $\delta_{\mathrm{Kron}}(J - J')$ to model the fact that each element of $\boldsymbol{x}$ gives rise to exactly one element of $\boldsymbol{y}$. When we take the logarithm of $\mathrm{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta})$ we obtain

$$\ln \mathrm{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) = -J\ln(2\pi\sigma^2) + \ln \delta_{\mathrm{Kron}}(J - J') - \frac{1}{2\sigma^2}\sum_{j=1}^{J} \left|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}^{(j)}\right|^2 + \\ - Kb_0\tau - \ln(J!) + J\ln\left[\frac{b_0\tau}{2\pi r_b^2}\right] - \frac{1}{2r_b^2}\sum_{j=1}^{J} \left|\boldsymbol{r}^{(j)} - \boldsymbol{r}_{k^{(j)}}\right|^2,$$

which gives

$$\langle \ln \mathrm{pr}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) \rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)}} = -J\ln(2\pi\sigma^2) - Kb_0\tau - \ln(J!) + J\ln\left[\frac{b_0\tau}{2\pi r_b^2}\right] + \\ - \frac{1}{2\sigma^2}\sum_{j=1}^{J} \left\langle \left|\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}^{(j)}\right|^2 \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)}} - \frac{1}{2r_b^2}\sum_{j=1}^{J} \left\langle \left|\boldsymbol{r}^{(j)} - \boldsymbol{r}_{k^{(j)}}\right|^2 \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)}},$$

and we have used the fact that $\mathrm{pr}(\boldsymbol{x} \mid \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(p)}) = 0$ if $J \neq J'$. In the expression above, $\hat{\boldsymbol{\theta}}^{(p)}$ denotes the estimate of $\boldsymbol{\theta}$ at the p^{th} iteration of the LMMLEM algorithm. Similarly to $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}}^{(p)} = \{\hat{\boldsymbol{r}}_1^{(p)}, \ldots, \hat{\boldsymbol{r}}_{\hat{K}^{(p)}}^{(p)}\},$$

and we will start with the case in which $\hat{K}^{(p)}$ stays constant throughout the iterations of the algorithm. As we did in § 3.4 and § 4.4, we can discard from the expression above the terms that do not depend on $\boldsymbol{\theta}$ and introduce a function $Q(\boldsymbol{\theta}; \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(p)})$ defined as

$$Q(\boldsymbol{\theta}; \boldsymbol{y}, \hat{\boldsymbol{\theta}}^{(p)}) = -\frac{1}{2r_b^2}\sum_{j=1}^{J} \left\langle \left|\boldsymbol{r}^{(j)} - \boldsymbol{r}_{k^{(j)}}\right|^2 \right\rangle_{\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)}} - Kb_0\tau =$$

$$= -\frac{1}{2r_b^2}\sum_{j=1}^{J}\Bigg\{\Big\langle\big(x^{(j)}-x_{k^{(j)}}\big)^2\Big\rangle_{x^{(j)},k^{(j)}|\hat{\boldsymbol{R}}^{(j)},\hat{\boldsymbol{\theta}}^{(p)}} + \\ + \Big\langle\big(y^{(j)}-y_{k^{(j)}}\big)^2\Big\rangle_{y^{(j)},k^{(j)}|\hat{\boldsymbol{R}}^{(j)},\hat{\boldsymbol{\theta}}^{(p)}}\Bigg\} - Kb_0\tau,$$

in which—we recall—$\boldsymbol{r}^{(j)} = \big(x^{(j)}, y^{(j)}\big)$ and $\boldsymbol{r}_{k^{(j)}} = \big(x_{k^{(j)}}, y_{k^{(j)}}\big)$. To calculate the terms in the expression above, we first use the definition of expectation to write

$$\Big\langle\big(x-x_k\big)^2\Big\rangle_{x,k|\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}} = \sum_{k=1}^{\hat{K}}\int_{-\infty}^{\infty}(x-x_k)^2\,\mathrm{pr}(x,k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})\,\mathrm{d}x,$$

in which

$$\mathrm{pr}(x,k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}) = \mathrm{pr}(x \mid k,\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})\,\mathrm{pr}(k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}).$$

Using (4.24) and (4.26) and after a few pages of calculations, we obtain

$$\mathrm{pr}(x \mid k,\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}) = \int_{-\infty}^{\infty}\mathrm{pr}(x,y \mid k,\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})\,\mathrm{d}y = \\ = \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\hat{X}-x)^2}{2\sigma^2}\right)\frac{1}{\sqrt{2\pi r_b^2}}\exp\left(-\frac{(x-x_k)^2}{2r_b^2}\right)}{\frac{1}{\sqrt{2\pi\left(\sigma^2+r_b^2\right)}}\exp\left(-\frac{(\hat{X}-x_k)^2}{2\left(\sigma^2+r_b^2\right)}\right)}.$$

For $\mathrm{pr}(k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})$ we have

$$\mathrm{pr}(k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}) = \frac{\mathrm{pr}(k,\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})}{\mathrm{pr}(\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})} = \frac{\mathrm{pr}(\hat{\boldsymbol{R}} \mid k,\hat{\boldsymbol{\theta}})\,\mathrm{pr}(k,\hat{\boldsymbol{\theta}})}{\mathrm{pr}(\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}})} = \\ = \frac{\mathrm{pr}(\hat{\boldsymbol{R}} \mid \hat{\boldsymbol{r}}_k)\,\mathrm{pr}(k \mid \hat{\boldsymbol{\theta}})}{\sum_{k'=1}^{\hat{K}}\mathrm{pr}(\hat{\boldsymbol{R}} \mid \hat{\boldsymbol{r}}_{k'})\,\mathrm{pr}(k' \mid \hat{\boldsymbol{\theta}})} = \frac{\mathrm{pr}(\hat{\boldsymbol{R}} \mid \hat{\boldsymbol{r}}_k)}{\sum_{k'=1}^{\hat{K}}\mathrm{pr}(\hat{\boldsymbol{R}} \mid \hat{\boldsymbol{r}}_{k'})},$$

which, after all substitutions, gives

$$\mathrm{pr}(x,k \mid \hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}) = \sqrt{\frac{\sigma^2+r_b^2}{2\pi\sigma^2 r_b^2}}\,\frac{\exp\left(-\frac{(\hat{X}-x)^2}{2\sigma^2}\right)\exp\left(-\frac{(x-\hat{x}_k)^2}{2r_b^2}\right)\exp\left(-\frac{(\hat{Y}-\hat{y}_k)^2}{2\left(\sigma^2+r_b^2\right)}\right)}{\sum_{k'=1}^{\hat{K}}\exp\left(-\frac{|\hat{\boldsymbol{R}}-\hat{\boldsymbol{r}}_{k'}|^2}{2\left(\sigma^2+r_b^2\right)}\right)}.$$

The expectation $\langle(x-x_k)^2\rangle_{x,k|\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}}$ now becomes

$$\left\langle \left(x-x_k\right)^2\right\rangle_{x,k|\hat{\boldsymbol{R}},\hat{\boldsymbol{\theta}}} = \frac{1}{\left(\sigma^2+r_b^2\right)^2}\times$$

$$\times\frac{\sum_{k=1}^{\hat{K}}\exp\left(-\frac{(\hat{X}-\hat{x}_k)^2}{2\left(\sigma^2+r_b^2\right)}\right)\exp\left(-\frac{(\hat{Y}-\hat{y}_k)^2}{2\left(\sigma^2+r_b^2\right)}\right)\left\{\sigma^2 r_b^2(\sigma^2+r_b^2)+\left[\sigma^2(\hat{x}_k-x_k)+r_b^2(\hat{X}-x_k)\right]^2\right\}}{\sum_{k'=1}^{\hat{K}}\exp\left(-\frac{(\hat{X}-\hat{x}_{k'})^2}{2\left(\sigma^2+r_b^2\right)}\right)\exp\left(-\frac{(\hat{Y}-\hat{y}_{k'})^2}{2\left(\sigma^2+r_b^2\right)}\right)}.$$

With this result, the expression for $Q(\boldsymbol{\theta};\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)})$ turns out to be

$$Q(\boldsymbol{\theta};\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)}) =$$

$$= -\frac{1}{2r_b^2}\frac{1}{\left(\sigma^2+r_b^2\right)^2}\sum_{j=1}^{J}\left\{\frac{1}{\sum_{k'=1}^{\hat{K}^{(p)}}\exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)}-\hat{\boldsymbol{r}}_{k'}^{(p)}|^2}{2\left(\sigma^2+r_b^2\right)}\right)}\sum_{k=1}^{\hat{K}^{(p)}}\exp\left(-\frac{|\hat{\boldsymbol{R}}^{(j)}-\hat{\boldsymbol{r}}_k^{(p)}|^2}{2\left(\sigma^2+r_b^2\right)}\right)\times\right.$$

$$\left.\times\left[2\sigma^2 r_b^2(\sigma^2+r_b^2)+\left(\sigma^2(\hat{y}_k^{(p)}-y_k)+r_b^2(\hat{Y}^{(j)}-y_k)\right)^2+\left(\sigma^2(\hat{x}_k^{(p)}-x_k)+r_b^2(\hat{X}^{(j)}-x_k)\right)^2\right]\right\},$$

and if we enforce $\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{\theta};\boldsymbol{y},\hat{\boldsymbol{\theta}}^{(p)})=\boldsymbol{0}$, we get

$$\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2\hat{x}_k^{(p)}+r_b^2\hat{X}^{(j)}\right)=\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2+r_b^2\right)x_k, \tag{4.64a}$$

$$\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2\hat{y}_k^{(p)}+r_b^2\hat{Y}^{(j)}\right)=\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2+r_b^2\right)y_k, \tag{4.64b}$$

where we have set

$$\alpha_{j,k}^{(p)}=\frac{\exp\left(-\frac{\left|\hat{\boldsymbol{R}}^{(j)}-\hat{\boldsymbol{r}}_k^{(p)}\right|^2}{2\left(\sigma^2+r_b^2\right)}\right)}{\sum_{k'=1}^{\hat{K}^{(p)}}\exp\left(-\frac{\left|\hat{\boldsymbol{R}}^{(j)}-\hat{\boldsymbol{r}}_{k'}^{(p)}\right|^2}{2\left(\sigma^2+r_b^2\right)}\right)},$$

for $k=1,\ldots,\hat{K}^{(p)}$. If we solve (4.64) for x_k and y_k, we get

$$\hat{\boldsymbol{r}}_k^{(p+1)}=\frac{\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2\hat{\boldsymbol{r}}_k^{(p)}+r_b^2\hat{\boldsymbol{R}}^{(j)}\right)}{\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\left(\sigma^2+r_b^2\right)}=$$

$$=\frac{\sigma^2}{\sigma^2+r_b^2}\hat{\boldsymbol{r}}_k^{(p)}+\frac{r_b^2}{\sigma^2+r_b^2}\frac{\sum_{j=1}^{J}\alpha_{j,k}^{(p)}\hat{\boldsymbol{R}}^{(j)}}{\sum_{j=1}^{J}\alpha_{j,k}^{(p)}}. \tag{4.65}$$

Some insights on the result found above can be obtained if we assume that the object being imaged consists of only one lump: $\boldsymbol{\theta} = \{\boldsymbol{r}\}$. The maximum-likelihood problem can now be formulated as estimating the location $\boldsymbol{r}$ of the lump from the observed data $\boldsymbol{y} = \{\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}\}$. To calculate the likelihood, notice first

$$\mathrm{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{r}) = \frac{1}{2\pi\,(\sigma^2 + r_b^2)} \exp\left(-\frac{|\hat{\boldsymbol{R}} - \boldsymbol{r}|^2}{2\,(\sigma^2 + r_b^2)} \right),$$

which allows to write the likelihood as

$$\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \frac{e^{-b_0 \tau}}{J'!} \left[\frac{b_0 \tau}{2\pi\,(\sigma^2 + r_b^2)} \right]^{J'} \exp\left(-\frac{1}{2\,(\sigma^2 + r_b^2)} \sum_{j=1}^{J'} |\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}|^2 \right). \qquad (4.66)$$

Maximizing $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ is equivalent to minimizing $\sum_{j=1}^{J'} |\hat{\boldsymbol{R}}^{(j)} - \boldsymbol{r}|^2$. It is a simple exercise to show that this latter quantity is minimized by

$$\hat{\boldsymbol{r}} = \frac{1}{J'} \sum_{j=1}^{J'} \hat{\boldsymbol{R}}^{(j)},$$

which is nothing but the ensemble mean of $\hat{\boldsymbol{R}}$ calculated from $\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}$. Notice that for the case $K = 1$, $\alpha_{j,1}^{(p)} = 1$ and the reader can verify that the point $\hat{\boldsymbol{r}}$ above is indeed a fixed point of (4.65).

To handle the general case in which the number of lumps $\hat{K}^{(p+1)}$ in $\hat{\boldsymbol{\theta}}^{(p+1)}$ is allowed to change, we can see if removing one of the lumps from $\hat{\boldsymbol{\theta}}^{(p+1)}$ actually increases the likelihood. Mathematically, we can define

$$k_{\text{remove}} = \underset{k=1,\ldots,\hat{K}^{(p+1)}}{\arg\max} \left[\mathrm{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)} \setminus \{\hat{\boldsymbol{r}}_k^{(p+1)}\}\big) \right],$$

in which $\hat{\boldsymbol{\theta}}^{(p+1)} \setminus \{\hat{\boldsymbol{r}}_k^{(p+1)}\}$ denotes the set operation of removing the point $\hat{\boldsymbol{r}}_k^{(p+1)}$ from the set of points $\hat{\boldsymbol{\theta}}^{(p+1)}$. The maximization step above is easily and efficiently implemented by removing points $\hat{\boldsymbol{r}}_k^{(p+1)}$ one at a time and by evaluating the likelihood. If removing point $\hat{\boldsymbol{r}}_{k_{\text{remove}}}^{(p+1)}$ from $\hat{\boldsymbol{\theta}}^{(p+1)}$ actually increases the likelihood, that point is removed

$$\hat{\boldsymbol{\theta}}^{(p+1)} \leftarrow \begin{cases} \hat{\boldsymbol{\theta}}^{(p+1)} \setminus \{\hat{\boldsymbol{r}}_{k_{\text{remove}}}^{(p+1)}\} & \text{if } \mathrm{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)} \setminus \{\hat{\boldsymbol{r}}_{k_{\text{remove}}}^{(p+1)}\}\big) > \mathrm{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)}\big), \\ \hat{\boldsymbol{\theta}}^{(p+1)} & \text{otherwise.} \end{cases}$$

To deal with the possibility that adding one lump to $\hat{\boldsymbol{\theta}}^{(p+1)}$ might increase the likelihood, we can consider

$$\hat{\boldsymbol{r}}^{(p+1)}_{\text{add}} = \arg\max_{\hat{\boldsymbol{r}}} \left[\operatorname{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)} \cup \{\hat{\boldsymbol{r}}\}\big)\right],$$

in which $\hat{\boldsymbol{\theta}}^{(p+1)} \cup \{\hat{\boldsymbol{r}}\}$ denotes the set resulting from the union of sets $\hat{\boldsymbol{\theta}}^{(p+1)}$ and $\{\hat{\boldsymbol{r}}\}$. We implemented the maximization step above using the contracting grid algorithm of § 3.3. Once lump center $\hat{\boldsymbol{r}}^{(p+1)}_{\text{add}}$ is found, $\hat{\boldsymbol{\theta}}^{(p+1)}$ is updated according to

$$\hat{\boldsymbol{\theta}}^{(p+1)} \leftarrow \begin{cases} \hat{\boldsymbol{\theta}}^{(p+1)} \cup \{\hat{\boldsymbol{r}}^{(p+1)}_{\text{add}}\} & \text{if } \operatorname{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)} \cup \{\hat{\boldsymbol{r}}^{(p+1)}_{\text{add}}\}\big) > \operatorname{pr}\big(\boldsymbol{y} \mid \hat{\boldsymbol{\theta}}^{(p+1)}\big), \\ \hat{\boldsymbol{\theta}}^{(p+1)} & \text{otherwise.} \end{cases}$$

As we can infer from (4.65), the iterative algorithm developed here necessitates an initial guess $\hat{\boldsymbol{\theta}}^{(0)}$ to refine through the iterations. To find this initial guess, first recall that $\overline{J}(b_{\boldsymbol{\theta}}, \tau) \approx K b_0 \tau$. Hence, given the number of elements J' in $\boldsymbol{y}$ and the exposure time τ, we can take the first estimate $\hat{K}^{(0)}$ of K as

$$\hat{K}^{(0)} = \text{round}\left(\frac{J'}{b_0 \tau}\right),$$

in which the function round(z) rounds the value of z to its nearest integer. Once estimate $\hat{K}^{(0)}$ is available, we set $\hat{\boldsymbol{\theta}}^{(0)} = \big\{\hat{\boldsymbol{r}}_1^{(0)}, \ldots, \hat{\boldsymbol{r}}_{\hat{K}^{(0)}}^{(0)}\big\}$, in which points $\hat{\boldsymbol{r}}_1^{(0)}, \ldots, \hat{\boldsymbol{r}}_{\hat{K}^{(0)}}^{(0)}$ are drawn from a uniform distribution over the object's support.

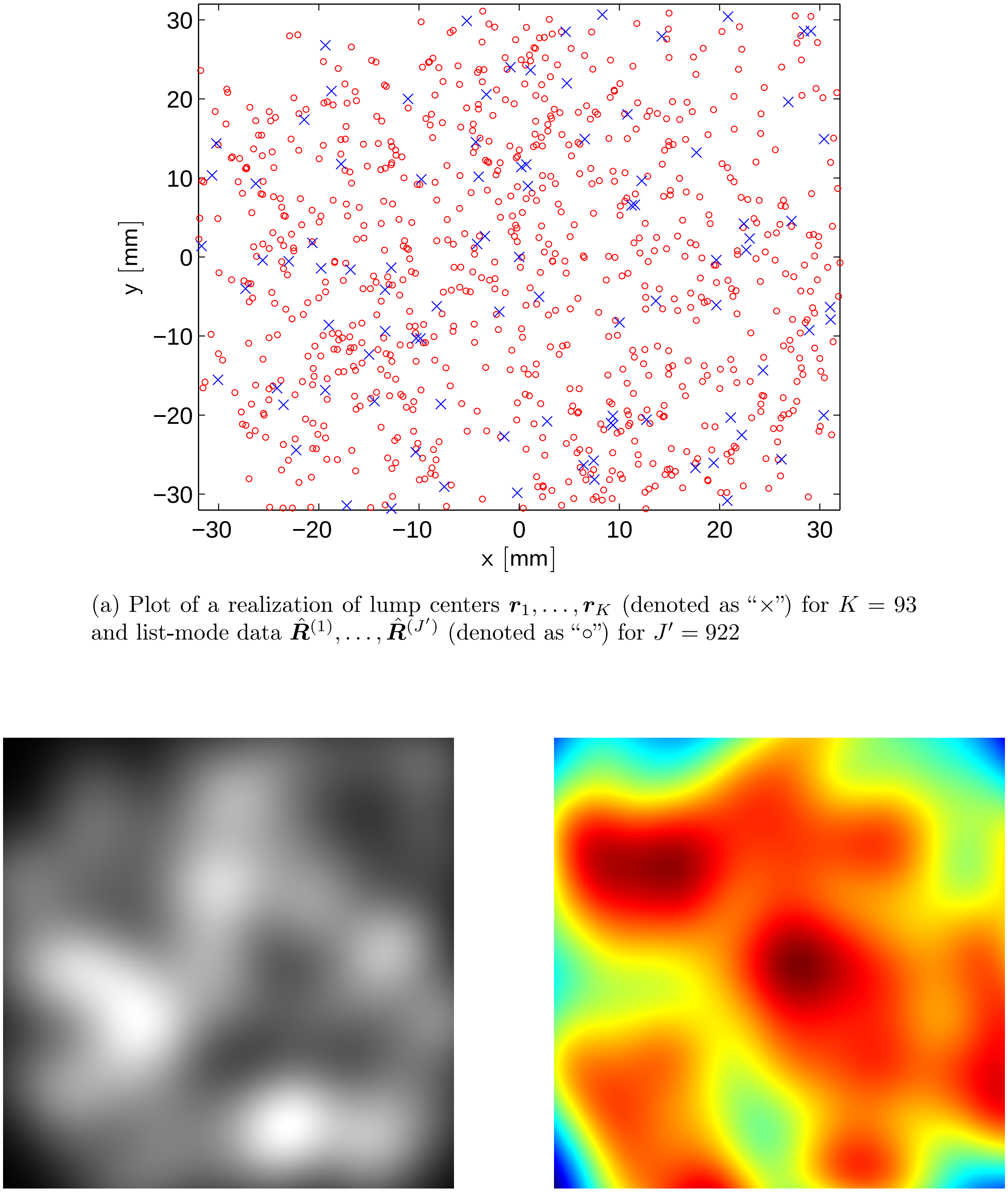
Pseudocode of the algorithm is provided below.

```
$\hat{K}^{(0)} = \text{round}\left(\frac{J'}{b_0\tau}\right)$
$\hat{\boldsymbol{\theta}}^{(0)} = \big\{\hat{\boldsymbol{r}}_1^{(0)}, \ldots, \hat{\boldsymbol{r}}_{\hat{K}^{(0)}}^{(0)}\big\}$
for $p = 1$ to $P$ do
  for $k = 1$ to $\hat{K}^{(p-1)}$ do
    $\alpha_{j,k}^{(p-1)} = \dfrac{\exp\left(-\frac{\left|\hat{\boldsymbol{R}}^{(j)} - \hat{\boldsymbol{r}}_k^{(p-1)}\right|^2}{2\left(\sigma^2 + r_b^2\right)}\right)}{\sum_{k'=1}^{\hat{K}^{(p-1)}} \exp\left(-\frac{\left|\hat{\boldsymbol{R}}^{(j)} - \hat{\boldsymbol{r}}_{k'}^{(p-1)}\right|^2}{2\left(\sigma^2 + r_b^2\right)}\right)}$
    $\hat{\boldsymbol{r}}_k^{(p)} = \frac{\sigma^2}{\sigma^2 + r_b^2} \hat{\boldsymbol{r}}_k^{(p-1)} + \frac{r_b^2}{\sigma^2 + r_b^2} \frac{\sum_{j=1}^{J} \alpha_{j,k}^{(p-1)} \hat{\boldsymbol{R}}^{(j)}}{\sum_{j=1}^{J} \alpha_{j,k}^{(p-1)}}$
  end for
```

```
    $k_{\text{remove}} = \underset{k=1,\ldots,\hat{K}^{(p-1)}}{\arg\max} \left[\text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)} \setminus \{\hat{\boldsymbol{r}}_k^{(p)}\}\big)\right]$
    if $\text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)} \setminus \{\hat{\boldsymbol{r}}_{k_{\text{remove}}}^{(p)}\}\big) > \text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)}\big)$ then
      $\hat{\boldsymbol{\theta}}^{(p)} = \hat{\boldsymbol{\theta}}^{(p)} \setminus \{\hat{\boldsymbol{r}}_{k_{\text{remove}}}^{(p)}\}$
      $\hat{K}^{(p)} = \hat{K}^{(p-1)} - 1$
    else
      $\hat{K}^{(p)} = \hat{K}^{(p-1)}$
    end if
    $\hat{\boldsymbol{r}}_{\text{add}}^{(p)} = \underset{\hat{\boldsymbol{r}}}{\arg\max} \left[\text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)} \cup \{\hat{\boldsymbol{r}}\}\big)\right]$
    if $\text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)} \cup \{\hat{\boldsymbol{r}}_{\text{add}}^{(p)}\}\big) > \text{pr}\big(\hat{\mathscr{A}} \mid \hat{\boldsymbol{\theta}}^{(p)}\big)$ then
      $\hat{\boldsymbol{\theta}}^{(p)} = \hat{\boldsymbol{\theta}}^{(p)} \cup \{\hat{\boldsymbol{r}}_{\text{add}}^{(p)}\}$
      $\hat{K}^{(p)} = \hat{K}^{(p-1)} + 1$
    else
      $\hat{K}^{(p)} = \hat{K}^{(p-1)}$
    end if
  end for
```

Code was developed to implement the reconstruction algorithm above, and simulations were set up to study its performance. We considered cases similar to those discussed in § 4.6.2 and § 4.6.3. More specifically, we assumed a field of view of size 64 mm × 64 mm, a lump amplitude $b_0 = 1000\ \text{s}^{-1}$, lump density $p = 20000\ \text{m}^{-2}$, lump width $r_b = 5.00$ mm, imaging system and detector blur uncertainty $\sigma = 1.00$ mm, and, finally, the acquisition time τ varied over $0.01, 0.02, \ldots, 0.12$ s.

As an example, Figure 4.25a reports a realization of lump centers (i.e., the vector $\boldsymbol{\theta}$), along with a realization of the list-mode data $\{\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}\}$ for the case $\tau = 0.01$ s. For the purpose of this example only, we force the location $\boldsymbol{r}_1$ of the first lump in $\boldsymbol{\theta}$ to $(0, 0)$, the center of the field of view. Figure 4.25b reports a plot of $b_{\boldsymbol{\theta}}(\boldsymbol{r})$ evaluated over a grid of point in the field of view. We want to stress that we sampled $b_{\boldsymbol{\theta}}(\boldsymbol{r})$ only to obtain the image of Figure 4.25b: as shown in (4.65), our reconstruction algorithm operates on the lump centers in $\hat{\boldsymbol{\theta}}^{(p)}$ and under no circumstances is the object being imaged represented as a set of pixel intensities. Finally, using the observed data $\boldsymbol{y} = \{\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}\}$, we can consider the logarithm of the likelihood $\text{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ in (4.66), and plot it as $\boldsymbol{\theta}$ is changed. Specifically, in Figure 4.25c



(a) Plot of a realization of lump centers $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K$ (denoted as "×") for $K = 93$ and list-mode data $\hat{\boldsymbol{R}}^{(1)}, \ldots, \hat{\boldsymbol{R}}^{(J')}$ (denoted as "∘") for $J' = 922$

(b) Background image $b_{\boldsymbol{\theta}}(\boldsymbol{r})$

(c) Plot of log-likelihood $\ln \mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ obtained by moving the first point in $\boldsymbol{\theta}$, originally located at $(0, 0)$

FIGURE 4.25. Example for LMMLEM reconstruction of lumpy background

we used the original lump locations—denotes as "×" in Figure 4.25a—except for the first one, $\boldsymbol{r}_1$, which was varied over the entire field of view to obtain an image; red areas denote large values of $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$, while small values of $\mathrm{pr}(\boldsymbol{y} \mid \boldsymbol{\theta})$ are represented in blue.

Intuitively, as the exposure time τ increases, the estimated lump locations $\hat{\boldsymbol{\theta}}$ should get "closer and closer" to the original lump locations in $\boldsymbol{\theta}$. To make this statement more precise, we first need to notice that it does not make sense to calculate the sum of the distances between points in $\hat{\boldsymbol{\theta}}$ and the corresponding points in $\boldsymbol{\theta}$. Because the order of points in $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ is irrelevant (as it leads to the same object), such sum of distances is not well defined. The number of points in $\hat{\boldsymbol{\theta}}$ could even differ from the number of points in $\boldsymbol{\theta}$. Instead, we can consider the quantity

$$d(\hat{\boldsymbol{\theta}}; \boldsymbol{\theta}) = \int_{\infty} \left[b_{\hat{\boldsymbol{\theta}}}(\boldsymbol{r}) - b_{\boldsymbol{\theta}}(\boldsymbol{r}) \right]^2 \mathrm{d}^2\boldsymbol{r}$$

as a measure of the "distance" between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$. If, however, we want to consider the "average" distance, we have to keep into account the randomness in the vector $\boldsymbol{\theta}$ and the randomness in the observed data $\boldsymbol{y}$ (conditioned on $\boldsymbol{\theta}$). In other words, there are two sources of randomness, leading to two nested expectations. Furthermore, it will not hurt normalizing this mean distance with respect to the mean background intensity $\left\langle \int_{\infty} [b_{\boldsymbol{\theta}}(\boldsymbol{r})]^2 \, \mathrm{d}^2\boldsymbol{r} \right\rangle_{\boldsymbol{\theta}}$. We call the resulting quantity the *normalized mean squared error* or NMSE

$$\mathrm{NMSE}(\tau) = \frac{\left\langle \left\langle \int_{\infty} \left[b_{\hat{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{\theta})}(\boldsymbol{r}) - b_{\boldsymbol{\theta}}(\boldsymbol{r}) \right]^2 \mathrm{d}^2\boldsymbol{r} \right\rangle_{\boldsymbol{y}|\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}}{\left\langle \int_{\infty} [b_{\boldsymbol{\theta}}(\boldsymbol{r})]^2 \, \mathrm{d}^2\boldsymbol{r} \right\rangle_{\boldsymbol{\theta}}}, \tag{4.67}$$

in which the right-hand side depends on τ via the (average) number of points $\hat{\boldsymbol{R}}^{(j)}$ in $\boldsymbol{y}$. We notice that the expectation at the denominator can be calculated analytically for the case of the lumpy background model in (4.25). Skipping all the calculations:

$$\left\langle \int_{\infty} [b_{\boldsymbol{\theta}}(\boldsymbol{r})]^2 \, \mathrm{d}^2\boldsymbol{r} \right\rangle_{\boldsymbol{\theta}} = \frac{b_0^2}{4\pi r_b^2} \left[\overline{K} + \frac{\overline{K}^2 \pi}{u^4} \left(u \, \mathrm{erf}(u) + \frac{e^{-u^2} - 1}{\sqrt{\pi}} \right)^2 \right], \qquad u = \frac{W}{2 r_b}.$$

The numerator of the quantity in (4.67) has to be estimated numerically: we generated 40 realizations of $\boldsymbol{\theta}$ and, for each of them, we generated 25 random realizations of $\boldsymbol{y}$. Each observed data vector $\boldsymbol{y}$ was fed to our LMMLEM reconstruction algorithm and 100 iterations were performed to get $\hat{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{\theta})$ to be used in (4.67). In writing $\hat{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{\theta})$ we made it clear that the observed data $\boldsymbol{y}$ depends on $\boldsymbol{\theta}$ and that, in turn, the estimate $\hat{\boldsymbol{\theta}}$ depends on $\boldsymbol{y}$. Using this process, an estimate of $\mathrm{NMSE}(\tau)$ and one for its standard deviation $\sigma_{\mathrm{NMSE}(\tau)}$ were calculated for $\tau = 0.01, 0.02, \ldots, 0.12$ s. Figure 4.26 plots in blue the estimated values of $\mathrm{NMSE}(\tau)$ along with standard deviations $\sigma_{\mathrm{NMSE}(\tau)}$ represented as error bars. These data were also fit with a shifted hyperbolic function

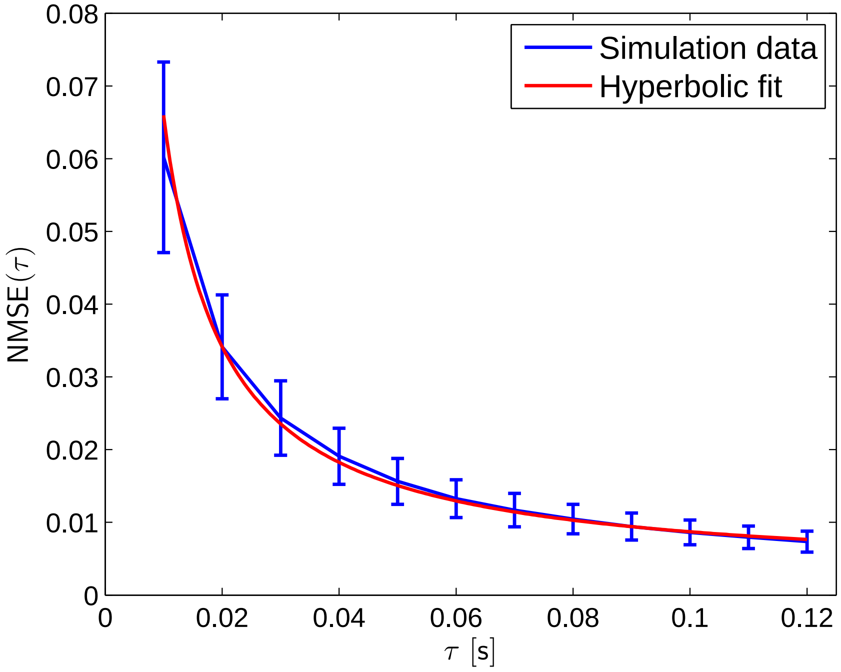


FIGURE 4.26. Plot of $\mathrm{NMSE}(\tau)$ and $\sigma_{\mathrm{NMSE}(\tau)}$ for different values of τ, along with its shifted hyperbolic fit.

of the form $\alpha/\tau + \beta$ where coefficients α and β were found by minimizing the weighted error metric

$$\varepsilon(\alpha, \beta) = \sum_{n=1}^{N} w_n \left[\mathrm{NMSE}(\tau_n) - \frac{\alpha}{\tau_n} - \beta \right]^2,$$

where $N = 12$ and $\tau_n = 0.01n$, for $n = 1, \dots, N$. Weights $w_1, \dots, w_N$ were chosen as the reciprocals of the variances $\sigma^2_{\text{NMSE}(\tau_1)}, \dots, \sigma^2_{\text{NMSE}(\tau_N)}$. Our shifted hyperbolic fit function, which was plotted in red in Figure 4.26, is in good agreement with the values of NMSE(τ) we found via our simulation code. The fact that $\beta > 0$, also suggests that NMSE(τ) does not go to 0 as more and more data are collected. This preliminary result, however, necessitates further investigation.

4.9 LMMLEM Reconstruction of PET Data

NOTE: Parts of this section have been adapted from L. Caucci, W. C. J. Hunter, L. R. Furenlid, and H. H. Barrett, *"List-mode MLEM Image Reconstruction from 3D ML Position Estimates,"* in IEEE Nuclear Science Symposium Conference Record, pages 2643–2647, Knoxville, TN, October/November 2010 [317].

Consider the PET setup shown in Figure 4.27.

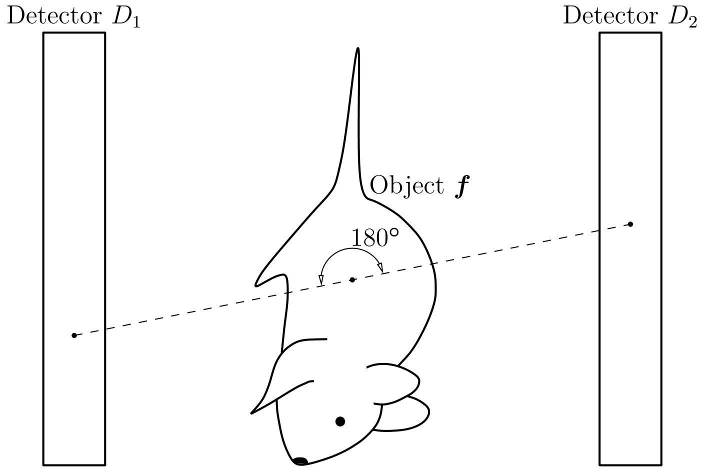


FIGURE 4.27. Setup of a simple PET system

Assume that the field of view is subdivided into N voxels, centered at locations $\boldsymbol{r}_n$, for $n = 1, \dots, N$. The goal of any image reconstruction algorithm is to calculate an estimate $\hat{\boldsymbol{f}} = \{\hat{f}_1, \dots, \hat{f}_N\}$ for the discrete set of numbers $\boldsymbol{f} = \{f_1, \dots, f_N\}$. Here,

$\boldsymbol{f}$ is a discrete approximation [69] to the unknown object $f(\boldsymbol{r})$. The object of interest has been injected with the radiotracer and pairs of photons are emitted from it. The two photons in each pair travel along the same straight line, but in opposite directions. We will refer to a coincidence event when both photons get collected by the corresponding detector, and we will assume that a mechanism for pairing photon collections is available. For example, this can be done by just recording the time of interaction and pairing events that occurred within a small time interval [6].

Each of the two detectors shown in Figure 4.27 is made of a thick scintillation crystal with PMT optical readout. From the PMT outputs for the j^{th} coincidence event collected by each detector, the 3D location of interactions $\hat{\boldsymbol{R}}_1^{(j)}$ and $\hat{\boldsymbol{R}}_2^{(j)}$ within detectors D_1 and D_2, respectively, are estimated. These two estimates of locations of interaction are paired to form the j^{th} attribute vector $\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)})$. During some acquisition time τ, a total of J coincidence events are collected, and the attribute list $\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\}$ is formed. Given the list $\hat{\mathscr{A}}$, the likelihood of $\boldsymbol{f} = \{f_1, \ldots, f_N\}$ is defined as [258, 261]

$$L(\boldsymbol{f}; \hat{\mathscr{A}}) = \operatorname{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f}) = \Pr(J \mid \boldsymbol{f}) \prod_{j=1}^{J} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f}), \tag{4.68}$$

where we assume that A_j and $A_{j'}$ are statistically independent for all $j \neq j'$. The quantity $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f})$ is the probability density function for the attribute vector $\hat{\boldsymbol{A}}^{(j)}$, conditioned on $\boldsymbol{f}$. It is useful to write [258]

$$\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{f}) = \sum_{n=1}^{N} \operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \Pr(n \mid \boldsymbol{f}), \tag{4.69}$$

in which $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ represents the probability density function of estimating $\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)})$ when a pair of photons is emitted from the n^{th} voxel in the field of view, and $\Pr(n \mid \boldsymbol{f})$ is the probability that a pair of photons is emitted from voxel n when the discretized radiotracer distribution is $\boldsymbol{f}$. In particular, we can calculate this probability as [258]

$$\Pr(n \mid \boldsymbol{f}) = \frac{s_n f_n}{\sum_{n'=1}^{N} s_{n'} f_{n'}}, \tag{4.70}$$

in which the numbers s_n, with $n = 1, \ldots, N$, are the voxel sensitivities. Each s_n is the probability that the two photons emitted from voxel n are collected by the detectors. Notice that the quantities s_n can be measured or they can also be calculated from the system geometry.

The LMMLEM reconstruction algorithm takes the form given in (4.10), which we report below for convenience:

$$\hat{f}_n^{(k+1)} = \hat{f}_n^{(k)} \left\{ \frac{1}{\tau} \sum_{j=1}^{J} \frac{\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)}} \right\}.$$

where k is the iteration index. Relevant properties of LMMLEM reconstruction are presented in [258]. For example, under broad conditions, the estimate $\hat{\boldsymbol{f}}^{(k)}$ calculated according to the formula above converges to the global maximum of the likelihood $L(\boldsymbol{f}; \hat{\mathscr{A}})$ in (4.68).

Evaluation of the LMMLEM iterative expression above requires the calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$, which is the probability density function for the attribute vector $\hat{\boldsymbol{A}}^{(j)}$ given that a photon pair is emitted from the n^{th} voxel. By Bayes' rule, we can write

$$\begin{aligned} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) &= \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)} \mid n) = \\ &= \int_{D_1} \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}, n) \, \mathrm{pr}(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)} \mid n) \, \mathrm{d}^3\boldsymbol{R}_2^{(j)} \, \mathrm{d}^3\boldsymbol{R}_1^{(j)}, \end{aligned}$$

where $\int_{D_1}$ and $\int_{D_2}$ denote integration over the 3D volume of detector 1 and detector 2, respectively. Notice that we have

$$\begin{aligned} \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}, n) &= \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}) = \\ &= \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)} \mid \boldsymbol{R}_1^{(j)}) \, \mathrm{pr}(\hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_2^{(j)}), \end{aligned}$$

because the estimation of $\hat{\boldsymbol{R}}_i^{(j)}$, for $i = 1, 2$, depends only on the true location of interaction $\boldsymbol{R}_i^{(j)}$ in detector D_i and estimation in one detector is independent on the estimation in the other detector. The expression for $\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ is now [46]

$$\operatorname{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) = \int_{D_1} \int_{D_2} \operatorname{pr}(\hat{\boldsymbol{R}}_1^{(j)} \mid \boldsymbol{R}_1^{(j)}) \operatorname{pr}(\hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_2^{(j)}) \times \\ \times \operatorname{pr}(\boldsymbol{R}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, n) \operatorname{pr}(\boldsymbol{R}_1^{(j)} \mid n) \, \mathrm{d}^3\boldsymbol{R}_2^{(j)} \, \mathrm{d}^3\boldsymbol{R}_1^{(j)}.$$

The expression above contains probability density functions of the form $\operatorname{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{R})$. These densities are the densities for the location of interaction estimates $\hat{\boldsymbol{R}}$, when the actual location of interaction is $\boldsymbol{R}$. In other words, $\operatorname{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{R})$ measures the detector performance. The concept of detector performance can be made more formal by using the Fisher information matrix $\mathbf{F}_{\boldsymbol{R}}$ at location $\boldsymbol{R}$ [69,139,318]. More details concerning $\mathbf{F}_{\boldsymbol{R}}$, including an approximate expression for $\operatorname{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{R})$ will be provided in the next chapter. For now, we will assume that an expression for $\operatorname{pr}(\hat{\boldsymbol{R}} \mid \boldsymbol{R})$ is available.

To derive an expression for $\operatorname{pr}(\boldsymbol{R}_1^{(j)} \mid n)$, consider a small cube $V_{\boldsymbol{R}_1^{(j)}}$ of side ε centered at $\boldsymbol{R}_1^{(j)}$. We will first consider the probability that an interaction occurs within the cube $V_{\boldsymbol{R}_1^{(j)}}$, given that a photon is emitted from a point $\boldsymbol{r}_n$ inside the n^{th} voxel of the field of view. We have [69]

$$\Pr\bigl(\text{interaction in } V_{\boldsymbol{R}_1^{(j)}} \mid n\bigr) \approx \varepsilon^3 \frac{\mu_{\text{pe}} e^{-\mu_{\text{tot}} \Delta_1(\boldsymbol{R}_1^{(j)}; \boldsymbol{r}_n)}}{4\pi |\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2},$$

in which μ_{pe} is the photoelectric attenuation coefficient for the detector crystal, μ_{tot} is its total attenuation coefficient, and the quantity $\Delta_1(\boldsymbol{R}_1^{(j)}; \boldsymbol{r}_n)$ measures how much the photon from $\boldsymbol{r}_n$ traveled within the detector crystal before reaching point $\boldsymbol{R}_1^{(j)}$, as shown in Figure 4.28. The approximation in the expression above comes from the fact that point $\boldsymbol{r}_n$ is allowed to vary over the entire volume of the n^{th} voxel.

By simple probability properties:

$$\operatorname{pr}(\boldsymbol{R}_1^{(j)} \mid n) = \lim_{\varepsilon \to 0} \frac{\Pr\bigl(\text{interaction in } V_{\boldsymbol{R}_1^{(j)}} \mid n\bigr)}{\varepsilon^3} \approx \frac{\mu_{\text{pe}} e^{-\mu_{\text{tot}} \Delta_1(\boldsymbol{R}_1^{(j)}; \boldsymbol{r}_n)}}{4\pi |\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2}. \tag{4.71}$$

To calculate $\operatorname{pr}(\boldsymbol{R}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, n)$, we start by recalling that, by our assumption, the two photons that interacted at $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ travel in opposite directions along the

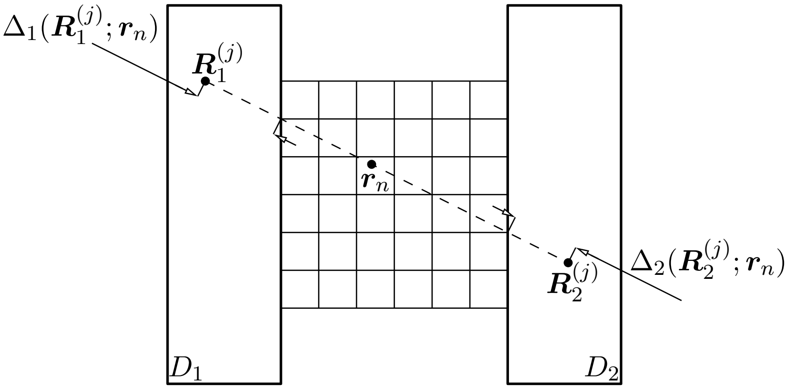

$\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)$
$\boldsymbol{R}_1^{(j)}$
$\boldsymbol{r}_n$
$\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)$
$\boldsymbol{R}_2^{(j)}$
D_1
D_2

FIGURE 4.28. Definition of $\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)$ and $\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)$

same line, and this line must contain point $\boldsymbol{r}_n$ as well. Therefore, the point $\boldsymbol{R}_2^{(j)}$ must belong to the intersection between detector D_2 and the line

$$\boldsymbol{R} = \boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell, \qquad -\infty < \ell < \infty \tag{4.72}$$

passing through $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{r}_n$. It is convenient to introduce the function

$$\psi_{D_2}(\boldsymbol{R}) = \begin{cases} 1 & \text{if } \boldsymbol{R} \in D_2, \\ 0 & \text{otherwise,} \end{cases}$$

so that

$$\begin{aligned} \mathrm{pr}(\boldsymbol{R}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, n) \approx \mu_{\mathrm{pe}} e^{-\mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)} \int_{-\infty}^{\infty} \psi_{D_2}\bigl(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\bigr) \times \\ \times \delta_{\mathrm{Dir}}\bigl(\boldsymbol{R}_2^{(j)} - \boldsymbol{R}_1^{(j)} - (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\bigr)\,\mathrm{d}\ell, \end{aligned} \tag{4.73}$$

where $\delta_{\mathrm{Dir}}(\boldsymbol{R})$ is the 3D Dirac delta function. Notice that the integrand in the expression above is nonzero only when the point $\boldsymbol{R}_2^{(j)}$ belongs to the intersection between the line defined in (4.72) with the detector D_2. Along with the units of μ_{pe}, the 3D Dirac delta function integrated over a line makes the dimensions of $\mathrm{pr}(\boldsymbol{R}_2^{(j)} \mid \boldsymbol{R}_1^{(j)}, n)$ equal to inverse volume, as they should. Putting the pieces together, we find

$$\begin{aligned} &\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \approx \\ &\quad = \frac{\mu_{\mathrm{pe}}^2}{4\pi} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1^{(j)} \mid \boldsymbol{R}_1^{(j)}) \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_2^{(j)} \mid \boldsymbol{R}_2^{(j)}) e^{-\mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)} \times \end{aligned}$$

$$\times \int_{-\infty}^{\infty} \psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) \delta_{\mathrm{Dir}}\big(\boldsymbol{R}_2^{(j)} - \boldsymbol{R}_1^{(j)} - (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) \, \mathrm{d}\ell \, \mathrm{d}^3\boldsymbol{R}_2^{(j)} \, \mathrm{d}^3\boldsymbol{R}_1^{(j)}. \tag{4.74}$$

Theoretical arguments we summarized in § 3.2 and simulation results we will discuss in the next chapter allow us to make some assumptions which, in turn, will allow fast numerical calculation of the integrals that define $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$. First of all, because the Fisher information matrix $\mathbf{F}_{\boldsymbol{R}}$ is slowly varying with respect to $\boldsymbol{R}$ and ML estimates $\hat{\boldsymbol{R}}_{\mathrm{MLE}}$ are within a short distance from $\boldsymbol{R}$, we can assume

$$\mathbf{F}_{\boldsymbol{R}} \approx \mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE}}},$$

and

$$|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2 \approx |\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)} - \boldsymbol{r}_n|^2.$$

Furthermore, if we assume $\boldsymbol{R} \approx \hat{\boldsymbol{R}}_{\mathrm{MLE}}$, then we can write

$$\Delta_i(\boldsymbol{R}_i^{(j)}; \boldsymbol{r}_n) \approx \Delta_i(\hat{\boldsymbol{R}}_{\mathrm{MLE},i}^{(j)}; \boldsymbol{r}_n), \qquad i = 1, 2.$$

With these approximations and by performing the integral over $\boldsymbol{R}_2^{(j)}$ in (4.74), we get

$$\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \approx \frac{\mu_{\mathrm{pe}}^2}{4\pi} \frac{e^{-\mu_{\mathrm{tot}}[\Delta_1(\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}; \boldsymbol{r}_n) + \Delta_2(\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)}; \boldsymbol{r}_n)]}}{|\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)} - \boldsymbol{r}_n|^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)} | \boldsymbol{R}_1^{(j)}) \times$$
$$\times \int_{-\infty}^{\infty} \psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) \, \mathrm{pr}\big(\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)} | \boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) \, \mathrm{d}\ell \, \mathrm{d}^3\boldsymbol{R}_1^{(j)}, \tag{4.75}$$

for which we will assume

$$\mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE}} | \boldsymbol{R}) \approx \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE}}}\big)}}{(2\pi)^{3/2}} \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_{\mathrm{MLE}} - \boldsymbol{R}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE}}} \big(\hat{\boldsymbol{R}}_{\mathrm{MLE}} - \boldsymbol{R}\big)\right]. \tag{4.76}$$

The expression in (4.75) is amenable to a fast implementation for the calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} | n)$. For example, because the density $\mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE}} | \boldsymbol{R})$ is sharply peaked and is essentially zero a few millimeters away from the estimated value $\hat{\boldsymbol{R}}_{\mathrm{MLE}}$, we can replace the integration over D_1 with an integration over a small detector volume centered at $\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}$. Also, the line integral that appears in (4.75) can be expressed in terms of

$\operatorname{erf}(u) = 2\pi^{-1/2}\int_0^u \exp(-v^2)\,\mathrm{d}v$. To show this, we can begin by noticing that detectors are usually shaped as a parallelepiped, hence the quantity $\psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big)$ is nonzero only when $\ell \in [\ell_1, \ell_2]$, for some values of ℓ_1 and ℓ_2 that depend on $\boldsymbol{R}_1^{(j)}$, $\boldsymbol{r}_n$, and the detector geometry, as shown in Figure 4.29.

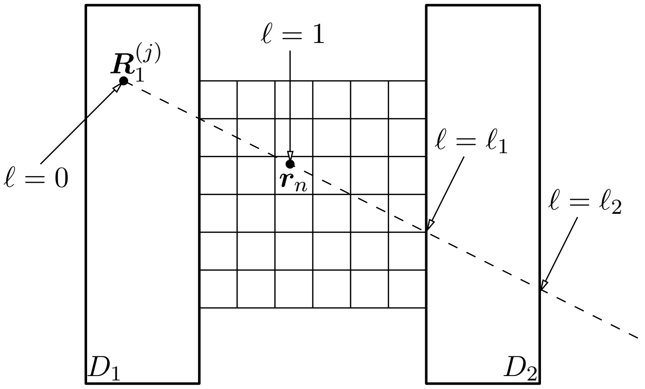


FIGURE 4.29. Definition of ℓ_1 and ℓ_2

Upon substitution of (4.76), the line integral over ℓ that appears in (4.75) becomes

$$I\big(\boldsymbol{R}_1^{(j)}\big) = \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\big)}}{(2\pi)^{3/2}} \int_{\ell_1}^{\ell_2} \exp\left[-\frac{1}{2}\big(\boldsymbol{a} - \boldsymbol{b}\ell\big)^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\big(\boldsymbol{a} - \boldsymbol{b}\ell\big)\right],$$

in which we have set

$$\boldsymbol{a} = \hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2} - \boldsymbol{R}_1^{(j)}, \qquad \boldsymbol{b} = \boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}.$$

To further simplify the notation, assume

$$A^2 = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\boldsymbol{a}, \qquad B = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\boldsymbol{b}, \qquad C^2 = \boldsymbol{b}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\boldsymbol{b},$$

so that

$$\big(\boldsymbol{a} - \boldsymbol{b}\ell\big)^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}\big(\boldsymbol{a} - \boldsymbol{b}\ell\big) = A^2 - 2B\ell + C^2\ell^2,$$

in which we used the fact that the matrix $\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}$ is symmetric. If we complete the square, we get

$$A^2 - 2B\ell + C^2\ell^2 = (\alpha - C\ell)^2 + \beta,$$

provided that we set

$$\alpha = \frac{B}{C}, \qquad \beta = A^2 - \frac{B^2}{C^2}.$$

With these substitutions, we obtain

$$\begin{aligned} I\big(\boldsymbol{R}_1^{(j)}\big) &= \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)}}\big)}}{(2\pi)^{3/2}} e^{-\frac{1}{2}\beta} \int_{\ell_1}^{\ell_2} e^{-\frac{1}{2}(C\ell-\alpha)^2}\,\mathrm{d}\ell = \\ &= \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)}}\big)}}{C(2\pi)^{3/2}} e^{-\frac{1}{2}\beta} \int_{C\ell_1-\alpha}^{C\ell_2-\alpha} e^{-\frac{1}{2}u^2}\,\mathrm{d}u = \\ &= \frac{\sqrt{\pi}}{2C} \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)}}\big)}}{(2\pi)^{3/2}} e^{-\frac{1}{2}\beta}\big[\operatorname{erf}(C\ell_2-\alpha) - \operatorname{erf}(C\ell_1-\alpha)\big]. \end{aligned}$$

Using the result above in (4.75) and (4.76) once again to expand the definition of $\mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}|\boldsymbol{R}_1^{(j)})$, we get

$$\begin{aligned} \mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \approx \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{64\pi^4} &\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}}\big)} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},2}^{(j)}}\big)} \times \\ &\times \int_{D_1} \frac{e^{-\frac{1}{2}\beta}}{C}\big[\operatorname{erf}(C\ell_2-\alpha) - \operatorname{erf}(C\ell_1-\alpha)\big] \times \\ &\times \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)} - \boldsymbol{R}_1^{(j)}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}} \big(\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)} - \boldsymbol{R}_1^{(j)}\big)\right] \mathrm{d}^3\boldsymbol{R}_1^{(j)}. \end{aligned}$$

Even though our notation does not show it, quantities ℓ_1, ℓ_2, C, α, and β do depend on the integration variable $\boldsymbol{R}_1^{(j)}$. Numerical calculation of the expression above can be performed by evaluating the integrand at a few points $\boldsymbol{R}_1^{(j)}$ around $\hat{\boldsymbol{R}}_{\mathrm{MLE},1}^{(j)}$ and by summing up all such values. Reconstruction results will be discussed in the next chapter.

CHAPTER 5
Parallel Computing and Algorithms Implementation

We begin this chapter with a brief introduction to parallel computing. We then continue by discussing in some details two popular computing platforms: the Cell Broadband Engine Architecture (CBEA) and graphics processing units (GPUs). Implementations of the contracting-grid algorithm for the 2D and 3D estimation of position of interaction within a scintillation crystal are discussed and performance results presented. GPU devices turn out to be particularly suited to performing list-mode maximum-likelihood expectation-maximization (LMMLEM) reconstruction, as we will show towards the end of this chapter. As a practical application, the LMMLEM algorithm is applied to real data acquired with the ModPET imaging system [6]. We conclude this chapter with a discussion of fast generation of random numbers on a GPU device.

5.1 Why Parallel Computing?

In this section, we will give a brief introduction to parallel computing. We will not be exhaustive; for an excellent discussion, the interested reader can consult, for example, [319] or [320].

Until a few years ago, frequency scaling was the dominant way to increase the performance of a computer. Electronic components were designed to work at higher and higher frequency so that the frequency of the clock that drove them could be increased. Assuming that the execution of an instruction took a constant number of clock cycles, as the frequency increased, the time required to carry out an instruction shortened.

However, about a decade ago, frequency scaling started to show its limitations, as shown in Figure 5.1. Circuit technology poses a limit on the speed at which a logic gate can switch from one state to the other. Furthermore, as the frequency (clock rate) increases, power consumption increases at the same rate [321]. Cooling becomes a huge problem. Another problem is that the speed at which a signal propagates inside a processing unit is finite. For a clock frequency of, say, 3 GHz, the distance a signal can travel during a clock cycle is just a few centimeters. Building small circuits might just not be possible. In any case, a small circuit is harder to cool than a larger one consuming the same amount of power.

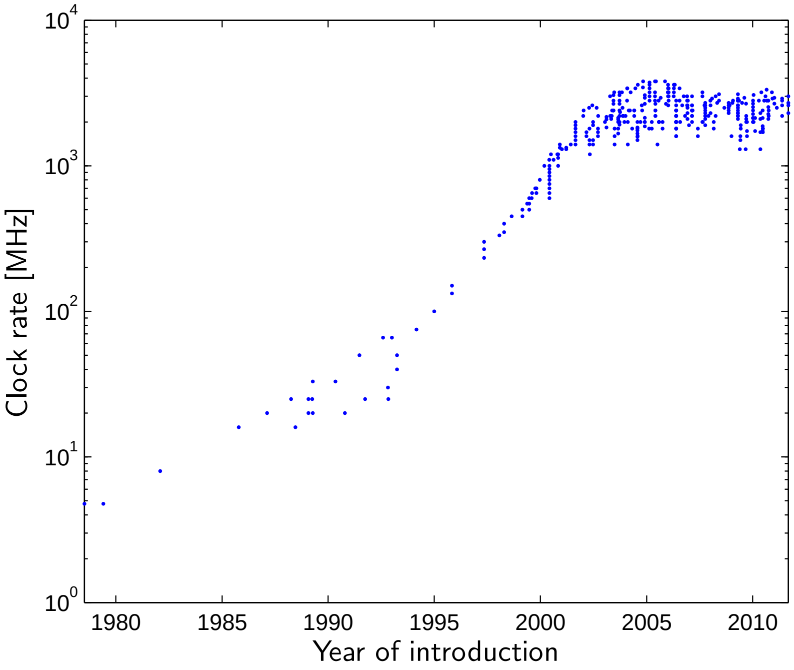


FIGURE 5.1. Log-scale plot of clock rates for some popular microprocessors introduced in the past three decades

A possible way to increase performance and solve the problems mentioned above is to abandon serial computation (in which one execution flow exists at any time and the instructions are executed sequentially, one after the other) and consider hardware and software components able to carry out parallel computing. More specifically, we refer to parallel computing as the ability to carry out many calculations simultaneously [319, 320, 322]. Parallel computing solves the problems we had with frequency scaling. In fact, rather than designing a single, powerful processing element that carries out operations fast, parallel computing allows us to break down the problem into subproblems which are solved concurrently by many slower processing elements. Collectively, the computational power of these slower processing elements surpasses the power of the single and faster processing element.

We can classify parallel hardware by the level at which parallelism is supported. We can have multi-core computer, multi-processor computer, computer clusters, or any combination of them. A multi-core computer [319, 323] is equipped with a single CPU that supports two or more execution threads. The cores in multi-core CPU execute instructions independently of each other. All the cores are implemented in the same physical package and so they might share some on-die resources, such as cache memory. The number of cores is usually rather small. Dual- or quad-core architectures are very popular. The majority of the CPUs produced today are multi-core.

In a multi-processor architecture, two or more processors are connected together by the computer's circuitry so that they share computer resources, such as the main memory and the disk [319]. A crucial part of a multi-processor architecture is the communication link between the main memory and the processors [324]. The communication can be by means of a system bus or a crossbar switch. A system bus consists of a bus that connects all the processing units and the memory. Only one communication at a time is allowed. Because of this limitation, a system bus does not scale very well with the size of the system and it is rarely used for high-performance computing. A crossbar switch is a switch able to connect multiple inputs to multiple outputs. The easiest way to conceptually visualize how a crossbar switch works is by imaging a set of horizontal input lines and a set of vertical output lines. Every time a horizontal (input) line crosses a vertical (output) line, we have a possible communication link between an input line and an output line. Thus, a crossbar switch is able to support more than one communication at a time. Crossbar switches are frequently used in high-end multi-processor systems.

Finally, in a cluster architecture, we have many computers connected together via a dedicated fast network. Each computer in the computer cluster has its local memory, disk storage, and peripherals. Usually, all the computers in the cluster have the same or similar hardware configuration. Very often, a file server is part of the cluster. Computers in the cluster might use their local disk(s) as a temporary storage media (besides for storing the operating system).

Writing programs for a parallel computer architecture is more difficult than writing sequential programs [325,326]. A parallel program consists of many subtasks running in parallel. The various subtasks might need to communicate and/or synchronize with each other to carry out the computation. Communication and synchronization requirements introduce a whole new plethora of programming difficulties and bugs. The matter is made worse by the difficulty and complexity that testing a parallel program pose. For example, the relative speed at which subtasks are completed might change the final result if synchronization is not coded appropriately.

Given a sequential application, we can usually identify portions of the code that can run in parallel and other parts that cannot be run in parallel but must be run sequentially. This gives a theoretical upper limit for the speedup S that can be obtained when a sequential program is ported to a parallel architecture and run on P processors. If α is the fraction of total running time that is spent in the sequential program by running portions of the code that can be parallelized, then the maximum theoretical speedup $S_{\max}$ is provided by Amdahl's law [327] and it is

$$S_{\max} = \frac{1}{(1-\alpha) + \frac{\alpha}{P}}.$$

In particular, we see that as the number of processors increases (i.e., $P \to \infty$), the maximum theoretical speedup is $\frac{1}{1-\alpha}$. In other words, the part of code that must run sequentially limits the speedup that can be obtained. Amdahl's law makes one important assumption, namely that the total number of communication messages between processors grows no more than linearly with the number of processors. If this is not the case, Amdahl's law cannot be applied.

In some cases, the computational overhead needed to handle P processors can be large. This would be, for example, the case in which large amounts of data are to be transmitted over a network link or system bus to all computing devices involved in the computation. To account for this, let β denote the portion of time that is spent in ancillary computation when only one processor is used. For example, β would be the fraction of total running time needed to send some amount of data over a network link. With this new parameter, we can modify Amdahl's law to account for the computational overhead needed to handle P processors:

$$S_{\max} = \frac{1}{(1-\alpha-\beta)+\frac{\alpha}{P}+P\beta}. \tag{5.1}$$

Contrary to the previous case, now when $P \to \infty$, the maximum theoretical speedup $S_{\max}$ converges to zero. Plots in Figure 5.2 summarize the behavior of the maximum theoretical speedup $S_{\max}$ as function of the number of processes P. The red curve is a plot of $S_{\max}$ according to Amdahl's law, while the blue plot shows $S_{\max}$ as calculated according to the modification to Amdahl's law we provided in (5.1). For our plots, $\alpha = 0.70$ and $\beta = 0.02$. The two curves are initially very similar. However, $S_{\max}$ for the modified version of Amdahl's law soon starts to decrease as the number of processors increases.

5.2 The Cell Broadband Engine Architecture

The Cell Broadband Engine Architecture (CBEA) or simply Cell BE, is a single-chip microprocessor architecture developed by Sony, Toshiba, and IBM over a 4-year

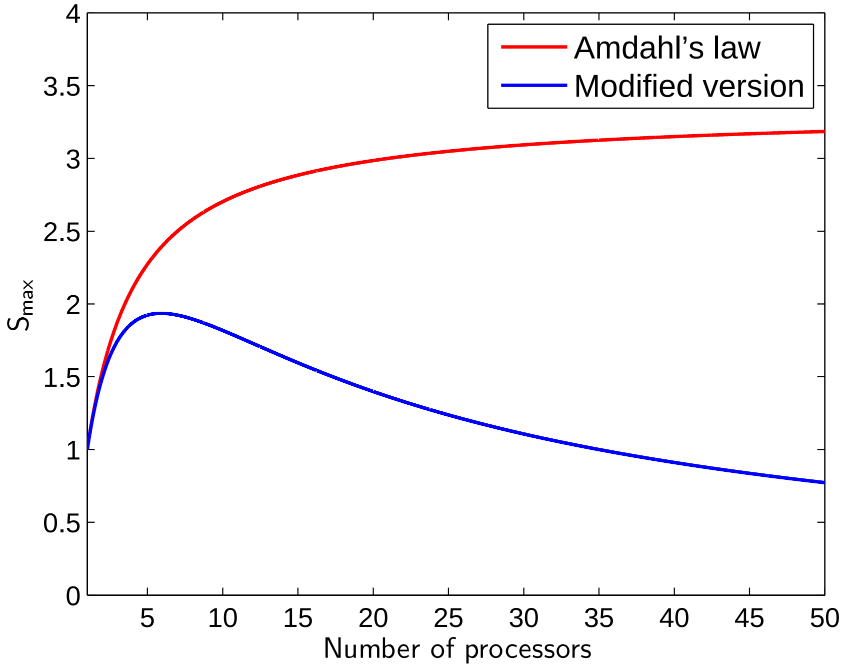


FIGURE 5.2. Maximum theoretical speedup $S_{\max}$ calculated according to Amdahl's law (red plot) and its modified version (blue plot). Parameters were as follows: $\alpha = 0.70$, $\beta = 0.02$

period starting in March 2001 [328, 329]. This microprocessor has primarily been developed to deliver high performance [330, 331] and it equips the latest generation of Sony's PlayStation game console, namely the PlayStation 3. The Cell BE has also equipped other devices as well, such as blade servers and acceleration cards. Due to its low cost and availability, the Cell BE has been widely used to carry out scientific applications, as reported in [332–336].

Schematically shown in Figure 5.3, the Cell BE architecture includes [337, 338]:

- one *Power Processor Element* (PPE);
- eight *Synergistic Processor Elements* (SPEs);
- one *Element Interconnect Bus* (EIB);
- two memory controllers;
- two input/output controllers.

The PPE the in Cell BE is a PowerPC processor capable of running two threads at the same time. Two distinct register files are available, one for each thread. Each register file consists of 32 128-bit registers. Data and instruction caches are available on the PPE as well. The PPE is capable of running 64-bit RISC code (optimized for performing simple operations fast) and vector instructions as wells [338]. Within the processor architecture, the PPE interacts with the operating system and it orchestrates other resources, the most important being the SPEs. The PPE runs at a 3.2 GHz clock frequency and it can manipulate data and perform arithmetic operations via an arithmetic-logic unit (ALU) called *Power Processing Unit* (PPU).

The SPEs are RISC processors specialized for data-intensive vector operations. Each SPE features 128 general-purpose registers, each 128 bits wide. Custom-designed vector operations are used to manipulated the content of the registers. Because each register is 128 bits (16 bytes) wide, it can store two double precision floating-point

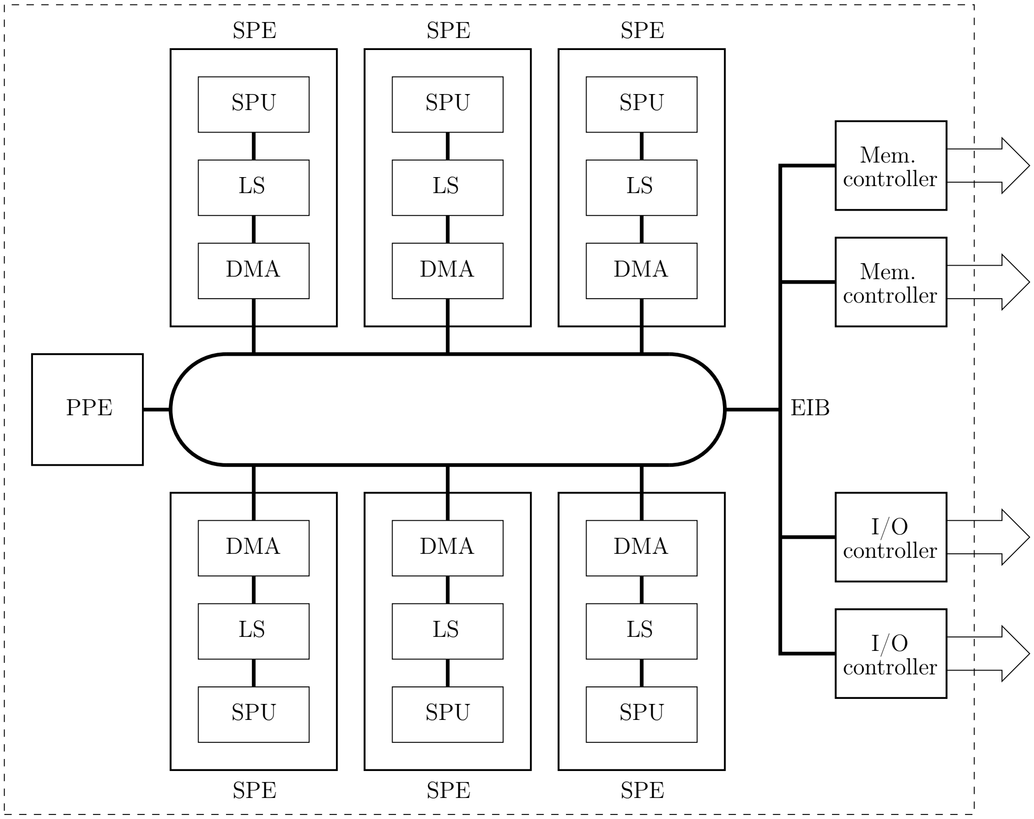


FIGURE 5.3. Diagram of the Cell BE architecture

numbers or four single-precision floating point numbers. Integer types are allowed as well [338]. In such a case, each register can store four signed or unsigned integer values, eight signed or unsigned short-integer values, or sixteen signed or unsigned character values. Vector instructions—which are executed by the SPE's *Synergistic Processing Unit* (SPU)—perform the same operation on each of the values stored in the vector registers involved in the operation. For example, we can consider two registers, each storing four floating point values, and perform a vector addition. In this case, each component of one register is added to the corresponding component of the other register. The SPE does not directly support scalar operations. Scalar operations are, however, supported by the compiler, which translates each scalar operation that appears in the code into the corresponding vector operation. Only one component of the result of the vector operation is then used; the other components are ignored [338].

Each SPE also includes a *Local Store* (LS) capable of storing up to 256 kB of data or code [338]. A *Direct Memory Access* (DMA) controller completes each SPE. The task of the DMA controller is to orchestrate data transfers from and to the SPE's LS and the main memory or another SPE's local store. A programming library provides the necessary programming interface to access the DMA controller functionalities, such as initiate a DMA transfer. It is worth noting that the DMA controller can govern data transfers while the rest of the SPE is involved in a computation [338]. This allows overlapping vector operations with data transfers. Double-buffering, which consists in transferring data to or from a portion of the LS while another portion of the LS memorizes data on which computation is being performed, is a technique commonly used to increase performance. The LS is the only memory space that an SPE can directly access: access to another SPE's LS or the main memory require a DMA data transfer.

The EIB allows communication between the processing elements (the PPE and the SPEs) with the on-chip memory and input/output controllers. The EIB consists of four 16-byte-wide data rings. Data are transferred in blocks of 128 bytes each. Because of the ring structure of the EIB, the location of the two processing units involved in the data transfer will affect latency. In other words, mean data latency increases with the number of connection "hops" the data have to go through [338]. Each ring is unidirectional, meaning that for a given ring, data always move clockwise or counterclockwise. However, to minimize mean data latency, two rings always move data in a clockwise fashion and the remaining two rings always in a counter-clockwise fashion. The EIB is capable of carrying out many data transfers concurrently, thus reaching a theoretical overall bandwidth of 204.8 GB/s [338].

The two memory controllers (shown in Figure 5.3) interface the Cell BE with the system's main memory [338]. The amount of main memory supported can vary from 64 MB to 64 GB. Again, in an effort to increase performance, the memory controllers can queue up to 64 memory reads and up to 64 memory writes. Different priority strategies and optimizations are available. For example, it is possible to make SPE reads have a higher priority with respect to all other reads. Speculative reads and "slow" mode (to reduce power consumption) are available as well [338]. The theoretical maximum bandwidth the two memory controllers provide altogether is 25.6 GB/s.

Finally, the two input/output controllers (shown in Figure 5.3) interface the Cell BE to other devices [338]. Such devices can be, for example, another memory unit, a video card, or even another Cell BE microprocessor. Each input/output controller is interfaced with twelve 8-bit-wide lines, and can work at a theoretical peak bandwidth of 76.8 GB/s.

5.2.1 Basic Programming Notions

In Cell BE programming, the PPE is treated as a conventional CPU. As we saw before, the processing unit in the PPE is capable of performing arithmetic and logic operations. In addition, the PPE has the capability to initiate DMA transfers and start code execution on the SPEs. It is this latter feature that mostly differentiate the PPE's architecture from a conventional CPU [338].

Because of design choices, some care must be paid when data structures are allocated and DMA transfers performed. For example, a single DMA data transfer cannot exceed 16 kB. If the block of data to be transferred is larger than 16 bytes, its size (in bytes) must be a multiple of 16. Shorter data transfers are allowed, provided that the size of the block of data being transferred is 1, 2, 4, or 8 bytes. Finally, the source and the destination memory buffers must be *naturally aligned*, meaning that the last four bits of the source and destination addresses must be the same [338]. As explained above, the DMA controller can operate independently from the processing unit. This implies that the processing unit can perform other operations while data are being transferred. Synchronization barriers are provided to stop code execution in the processing unit until pending DMA transfers are completed.

A Cell BE program typically makes use of threads to run parallel code on the SPEs. Each thread receives the information about the task to be performed on the SPE. A thread is typically responsible for loading the code in the SPE's LS as well as for starting code execution on the SPE. The SPE needs to have data loaded in the LS to process them. As discussed above, DMA transfers allow data transfers from the main memory to the LS, and vice versa. Once data are loaded in the LS, the SPU typically process them. Because the SPUs are optimized for data intensive vector operations, greater speedup will be attained if data are processed using vector instructions. At the end of the computation, the result can be copied to main memory using a DMA transfer and the SPU code fragment terminates executing [338].

The programming model discussed above might seem very complicated and hard to use. Fortunately, there are programming tools [339] that provide a simple, high-level data-parallel model that is easy to understand and that maps onto the capabilities of the Cell BE. Often times, these tools introduce new data types and functions to conveniently expose to the programmer the data parallelism that the hardware allows and take advantage of it.

5.2.2 An Example

In this section, we want to provide a simple reference example of a Cell BE program. The example we considered performs a componentwise addition of the elements of two vectors. The example consists of two pieces of code: one to be run on the PPU and the other on the SPUs. Because these two types of processing units have different instruction sets, the two pieces of code are stored as two different files and they have to be compiled with two different compilers, each one targeting a particular instruction set. Finally, we have a third file that contains definitions common to both pieces of code.

The PPU code is responsible for allocating and filling up the data arrays as well as filling up two particular data structures. The first data structure contains data and parameters that each SPU needs to carry out its task. We will refer to this data structure as the “control block.” Information stored in the control block would include the address of the data arrays and the number of array elements each SPU will have to process. The second data structure contains information the operating system needs to initiate code execution on the SPU. Within the Cell BE programming documentation [338], this information is collectively referred to as the “context.”

To make it possible for the PPU to start code execution on all the SPU available, six threads are created. The code was developed to be run on a Sony PlayStation 3 game console, on which only six (out of eight) SPUs can be used by the programmer. By creating six threads, we are guaranteed that each SPU available will be occupied with executing SPU code. The SPUs will run code independently from the PPU. This makes it possible to use the PPU as an additional processing unit on which computation can be carried out. In the example reported below, we decided to use the PPU to process the “leftover” array elements. For example, the total number of array elements to process might not be a multiple of six. Rather than having a different number of array elements to be processed by each SPU, we opted to have the same number of elements processed by each SPU and to leave the remaining elements for processing on the PPU. This implementation choice makes the code more understandable and easier to debug.

The PPU then waits for all the SPUs to terminate their execution and, finally, the result is displayed on the screen.

File `main_ppu.c`

```
#include <math.h>
#include <stdio.h>
#include <libspe2.h>
#include <pthread.h>
#include "my_defines.h"

////////////////////////////////////////////////////////////////////////////////////////////

extern spe_program_handle_t array_add_spu;

////////////////////////////////////////////////////////////////////////////////////////////

void *function_spe_thread(void *arg);

////////////////////////////////////////////////////////////////////////////////////////////

struct spe_thread_data_t {
  spe_context_ptr_t context;
  unsigned int entry;
  pthread_t pthread;
  void *argp;
  void *envp;
};

////////////////////////////////////////////////////////////////////////////////////////////

int main(int argc, char **argv) {
  static struct array_add_ctrl_blk_t array_add_ctrl_blks[6] __attribute__((aligned(128)));
  static float a[ARRAY_SIZE] __attribute__((aligned(128)));
  static float b[ARRAY_SIZE] __attribute__((aligned(128)));
  static float c[ARRAY_SIZE] __attribute__((aligned(128)));
  static struct spe_thread_data_t thread_infos[6];
  int elements_per_spu, i;

  // Fill in the input arrays
  for(i = 0; i < ARRAY_SIZE; ++i) {
    a[i] = cosf(1.00 + i);
    b[i] = sinf(1.00 + i);
  }
  // Calculate the number of array elements each SPU will process
  elements_per_spu = (ARRAY_SIZE / 6) & ~0x0003;
  // Loop to fill in the control blocks and start threads
  for(i = 0; i < 6; ++i) {
    // Set the number of array elements to process
    array_add_ctrl_blks[i].num_elements = elements_per_spu;
    // Set the address of arrays
    array_add_ctrl_blks[i].array_a = a + i * elements_per_spu;
    array_add_ctrl_blks[i].array_b = b + i * elements_per_spu;
    array_add_ctrl_blks[i].array_c = c + i * elements_per_spu;
    // Set the address of the control block
    thread_infos[i].argp = & array_add_ctrl_blks[i];
    thread_infos[i].envp = NULL;
    thread_infos[i].context = spe_context_create(0, NULL);
    // Load the code in the SPE's LS
    spe_program_load(thread_infos[i].context, & array_add_spu);
    thread_infos[i].entry = SPE_DEFAULT_ENTRY;
    // Start a thread to run code on an SPE
    pthread_create(& thread_infos[i].pthread, NULL, function_spe_thread, & thread_infos[i]);
  }
  // Process the ‘‘leftover’’ array elements
  for(i = (6 * elements_per_spu); i < ARRAY_SIZE; ++i) {
    c[i] = a[i] + b[i];
  }
  // Wait for the termination of the threads
  for(i = 0; i < 6; ++i) {
    pthread_join(thread_infos[i].pthread, NULL);
    spe_context_destroy(thread_infos[i].context);
  }
  // Print the result on the screen
  for(i = 0; i < ARRAY_SIZE; ++i) {
    printf("%f␣+␣%f␣=␣%f\n", a[i], b[i], c[i]);
  }
  return(0);
}

//////////////////////////////////////////////////////////////////////////////////////

void *function_spe_thread(void *arg) {
  register struct spe_thread_data_t *data = (struct spe_thread_data_t *) arg;

  // Execute SPE code and wait for it to terminate
  spe_context_run(data -> context, & (data -> entry), 0, data -> argp, data -> envp, NULL);
  // Terminate the thread
  pthread_exit(NULL);
}
```

Concerning the SPU code, whose listing is reported below, we see that the first task the SPU code needs to perform is a DMA transfer to copy the control block from main memory to the LS. The address of the control block is conveniently passed as a parameter to the main SPU program. The necessary memory buffers are then allocated (by default in the SPE's LS) and the input data is copied from main memory to the buffers in the LS. The data are processed by vector operations. In our case, vector additions are performed between the elements of the input arrays and the result is copied to a third array. Because we are performing a vector operation that involves single-precisions floating point numbers, four array elements of the input arrays are processed by each call of the vector addition operation. Thus, the body of the `for(...)` loop repeats a number of times that is given by the number of array elements divided by four. After all elements are processed, the result is copied from the LS to main memory using a DMA transfer. The SPU code then terminates.

We finally note that in an attempt to make the code clearer and to cope with some of the limitations that DMA transfers have (such as maximum block size and alignment requirements), we wrote some functions that will break up a large DMA data transfer into a sequence of smaller ones.

File `array_add_spu.c`

```
#include <libmisc.h>
#include <spu_mfcio.h>
#include "my_defines.h"

//////////////////////////////////////////////////////////////////////////////////////////

void gen_mfc_get(void *ls, uint64_t ea, uint32_t size, uint32_t tag, uint32_t tid, uint32_t rid);
void gen_mfc_put(void *ls, uint64_t ea, uint32_t size, uint32_t tag, uint32_t tid, uint32_t rid);

//////////////////////////////////////////////////////////////////////////////////////////

int main(unsigned long long speid, unsigned long long argp, unsigned long long envp) {
  static struct array_add_ctrl_blk_t array_add_ctrl_blk __attribute__((aligned(128)));
  float *a, *b, *c;
  int i, n;

  // Get the control block via a DMA transfer
  mfc_get(& array_add_ctrl_blk, argp, sizeof(array_add_ctrl_blk), 15, 0, 0);
  // Wait for the termination of the DMA transfer
  mfc_write_tag_mask(1 << 15);
  mfc_read_tag_status_all();
  // Allocate memory for the input arrays
  n = array_add_ctrl_blk.num_elements;
  a = calloc_align(n, sizeof(*a), 7);
  b = calloc_align(n, sizeof(*b), 7);
  // Get the input data via DMA transfers
  gen_mfc_get(a, (uint32_t) array_add_ctrl_blk.array_a, n * sizeof(*a), 15, 0, 0);
  gen_mfc_get(b, (uint32_t) array_add_ctrl_blk.array_b, n * sizeof(*b), 15, 0, 0);
  // Allocate memory for the output array
  c = calloc_align(n, sizeof(*c), 7);
  // Wait for the termination of the DMA transfers
  mfc_write_tag_mask(1 << 15);
  mfc_read_tag_status_all();
  // Loop to process the data
  for(i = 0; i < (n >> 2); ++i) {
    // Perform a vector addition of four data pairs at a time
    *(((vector float *) c) + i) = spu_add(*(((vector float *) a) + i), *(((vector float *) b) + i));
  }
  // Start a DMA transfer to copy the result to main memory
  gen_mfc_put(c, (uint32_t) array_add_ctrl_blk.array_c, n * sizeof(*c), 15, 0, 0);
  free_align(a);
  free_align(b);
  // Wait for the termination of the DMA transfer
  mfc_write_tag_mask(1 << 15);
  mfc_read_tag_status_all();
  free_align(c);
  return(0);
}


/////////////////////////////////////////////////////////////////////////////////////////


void gen_mfc_get(void *ls, uint64_t ea, uint32_t size, uint32_t tag, uint32_t tid, uint32_t rid) {
  unsigned register int block_size;

  while(size > (16 * 1024)) {
    mfc_get(ls, ea, 16 * 1024, tag, tid, rid);
    ls += 16 * 1024;
    ea += 16 * 1024;
    size -= 16 * 1024;
  }
  mfc_get(ls, ea, size & ~0x000f, tag, tid, rid);
  ls += size & ~0x000f;
  ea += size & ~0x000f;
  size &= 0x000f;
  block_size = 8;
  do {
    if(size >= block_size) {
      mfc_get(ls, ea, block_size, tag, tid, rid);
      ls += block_size;
      ea += block_size;
      size -= block_size;
    }
    block_size >>= 1;
  } while(block_size > 0);
  return;
}


void gen_mfc_put(void *ls, uint64_t ea, uint32_t size, uint32_t tag, uint32_t tid, uint32_t rid) {
  unsigned register int block_size;

  while(size > (16 * 1024)) {
    mfc_put(ls, ea, 16 * 1024, tag, tid, rid);
    ls += 16 * 1024;
    ea += 16 * 1024;
    size -= 16 * 1024;
  }
  mfc_put(ls, ea, size & ~0x000f, tag, tid, rid);
  ls += size & ~0x000f;
  ea += size & ~0x000f;
  size &= 0x000f;
  block_size = 8;
  do {
    if(size >= block_size) {
      mfc_put(ls, ea, block_size, tag, tid, rid);
      ls += block_size;
      ea += block_size;
      size -= block_size;
    }
    block_size >>= 1;
  } while(block_size > 0);
  return;
}
```

File `my_defines.c`

```
#define ARRAY_SIZE   1024


struct array_add_ctrl_blk_t {
  int num_elements;
  float *array_a;
  float *array_b;
  float *array_c;
} __attribute__((aligned(16)));
```

5.3 The CUDA Environment

As desktop computers' computational power increases, so does the demand for fast, real-time, 3D graphics rendering. Fueled by the game and entertainment industry, today's graphics processing units (GPUs) have become highly parallel, general-purpose computing devices [340]. Because of mass production, newer products provide higher and higher performances at very competitive prices.

In this context, NVIDIA launched in 2006 a programming environment called *Compute Unified Device Architecture* or CUDA which allows programmers to access the tremendous parallel capabilites that modern GPU devices offer. More specifically, the CUDA programming model consists of a minimal extension to the C/C++ programming language, plus a runtime library [340]. Because of its simple, yet powerful model, CUDA has become extremely popular and successful in the scientific community [341–353].

5.3.1 Basic Programming Notions

Because of their intended use in highly parallel data-intensive applications, one of the key design strategies of today's GPU devices was to optimize data manipulation and processing versus flow control. Indeed, this is the case of most gaming and entertainment applications in which the same operation is performed on many data elements, without the need of complicated flow control. As a result, the hardware implementation of a GPU greatly differs from the hardware implementation of a conventional CPU. In a GPU, many transistors are employed to optimize and speed up floating-point operations and data manipulation. Support for advanced flow-control (such as as branch prediction) is generally lacking or very limited.

In the CUDA language, the GPU unit is usually referred to as the *device*. The device acts as a coprocessor to the rest of the computer, which is usually referred to as the *host*. This notion will be used consistently in the remainder of this chapter. We can also talk about *host memory*, which is the conventional system memory, and *device memory*, which is physically installed in the same board the GPU device is installed on (see Figure 5.4). The GPU can access only the memory installed on the GPU device. Specialized library functions are provided to copy blocks of data from the host memory to the device memory, from the device memory to the host memory, or even from the device memory to the device memory [340].

In order to access the parallel capabilities of a CUDA-enabled device, the programmer writes *kernels*. A kernel in execution is usually referred to as a *thread*. To reiterate, a kernel is a piece of code, while a thread is the abstract entity represented by a piece of code that is executing. Thus, a kernel might give rise to many threads, each of them working with different inputs and, potentially, starting executing at different points in time [340]. Threads are grouped into 1D, 2D, or 3D *blocks* and blocks are grouped into 1D or 2D *grids*. In CUDA, a particular calling syntax allows the programmer to specify the size of blocks and grids. The maximum number of

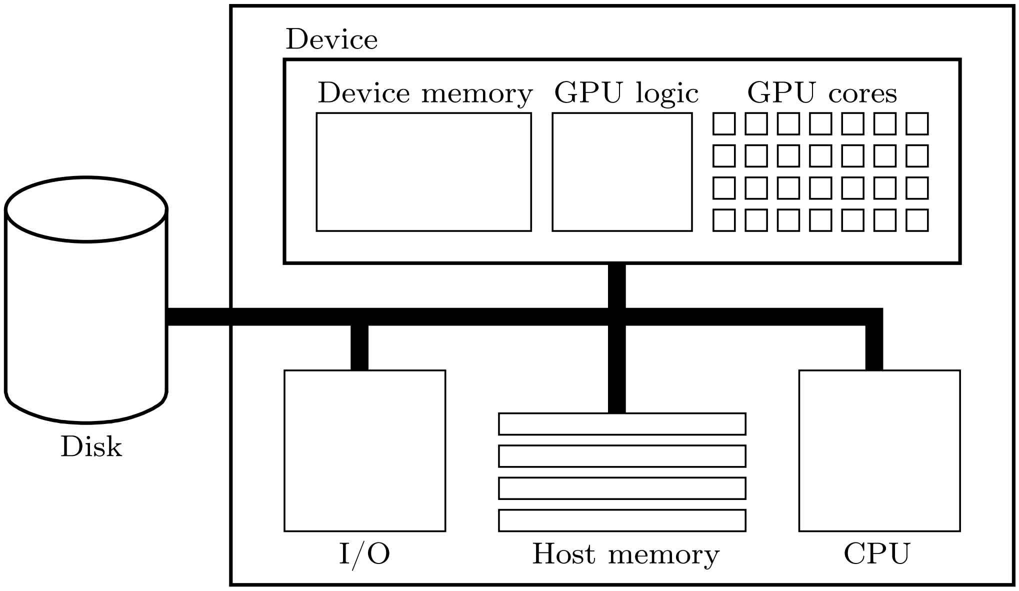


Figure 5.4. Diagram of a computer equipped with a GPU device

threads per block is a relatively small number. For current hardware implementations, such number does not exceed 1024 [340]. The number of blocks in a grid can be much higher. For example, current hardware supports grids of size up to 65535 × 65535 [340]. Figure 5.5 clarifies these concepts and presents an example of grouping threads into blocks and blocks into a grid.

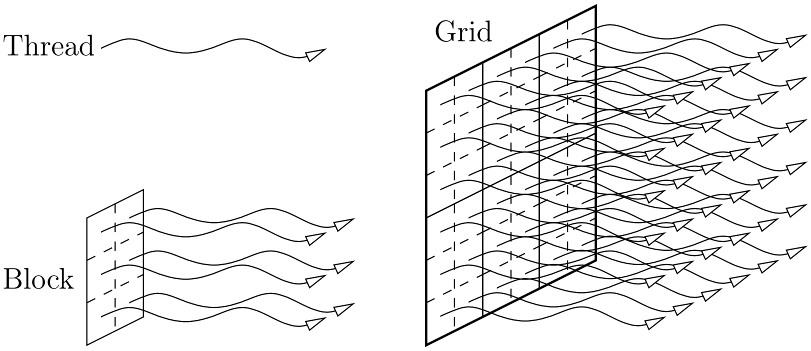


Figure 5.5. Thread and block hierarchies in CUDA

Thread blocks are automatically scheduled by the hardware, without the intervention of the operating system or the programmer. In a GPU device, threads have little context, and so their scheduling is extremely efficient: a thread ready for execution can be selected and scheduled in at a cost of just a few clock cycles [340]. Because of the large number of cores, hundreds of threads can be executed concurrently. This represents a remarkable departure from conventional multi-core CPUs, which usually have no more than sixteen cores and do not requires many threads to keep the hardware busy.

In a GPU device, the order in which threads are scheduled for execution is undefined [340]. Therefore, the programmer cannot rely on any particular order. There are cases, however, in which it is necessary to know if a thread has reached a particular point in the execution. This can be accomplished via synchronization barriers [340]. When a threads encounters a synchronization barrier, its execution is suspended until all the threads in the same block have reached the same synchronization barrier. In a GPU device, synchronization is possible only among the threads in the same block. No synchronization mechanism is provided for threads that belong to two different blocks.

A synchronization barrier is used, for example, when a thread needs to read from memory a value that another thread wrote. Without any synchronization mechanism, a thread will not be able to know if the value being read is the correct one or is the one from a previous computation. When the result of a computation depends on the order or timing of other events, we talk about *race conditions* [354]. Synchronization barriers are a common way to resolve race conditions.

A GPU device is equipped with a few different memory spaces. Memory can be *local*, *shared*, or *global*. Local memory has a scope of one thread. This means that only the thread to which a portion of this memory is associated, can access it. No other threads can access that portion of memory. The local memory has the lifetime of a thread, meaning that when the thread terminates the execution, the content of its local memory becomes undefined and not accessible from other threads. Local memory is, unfortunately, quite slow. It might take hundreds of clock cycles for retrieve a value from local memory. For this reason, the compiler tries to avoid using it and will store local variables in registers whenever possible [340].

Shared memory is particular type of on-chip memory that can be shared among all the threads in a thread block. This type of memory is very fast but, unfortunately, only 16 kB of shared memory are available for each block. Shared memory is typically used to shared data with other threads in the same block or to store temporary results. In the former case, synchronization barriers are often needed to resolve race conditions.

Global memory is visible to all threads and host, and its lifetime extends to the whole application. Global memory is off-chip and it is not cached. Similarly to local memory, accessing global memory usually requires hundreds of clock cycles. Global memory is usually used to share input and output data between the device and the host. This is accomplished by means of special function calls that the host uses to copy data from host memory to global memory and vice versa. As for the case of shared memory, the user must avoid race conditions when accessing global memory. The same synchronization barrier described for shared memory is used to solve race conditions when accessing global memory [340].

While all the types of memory spaces described above are read/write memories, a GPU device is equipped with read-only memory spaces as well. These memory spaces are referred to as *constant* and *texture* memories. To make use of these memory spaces, the host code will have to set their content first. The content of these memory spaces will be available to threads during the execution of kernel code. Contrary to the memory spaces discussed before, the constant and the texture memory spaces are cached, resulting in higher performance if the same datum is accessed multiple times during the execution of kernel code. Furthermore, because these memory spaces can be only read, no race condition between threads is possible.

The way in which texture memory is accessed is somewhat peculiar. As its name suggest, the texture memory has been designed to facilitate texture rendering and other tasks that are common in graphics rendering and computer games. This is the reason why texture memory supports features like linear interpolation and referencing via normalized floating-point coordinates. Further details about constant and texture memory spaces will not be provided here. The interested reader can consult the official documentation [340]. A summary of all the memory types is reported in Table 5.1.

Memory	**Location**	**Cached?**	**Access**	**Scope**	**Lifetime**
Local	Off-chip	No	R/W	One thread	Thread
Shared	On-chip	N/A	R/W	All threads in block	Block
Global	Off-chip	No	R/W	All threads and host	Application
Constant	Off-chip	Yes	R	All threads and host	Application
Texture	Off-chip	Yes	R	All threads and host	Application

Table 5.1. Memory types available on a GPU device

As we mentioned earlier, a GPU device performs thread scheduling in hardware with minimal overhead. Furthermore, the hardware is able to detect when a thread is waiting for data to be read from memory. This makes it possible to temporarily suspend the execution of threads that are waiting for data to be read from memory and select for execution some threads that already have data available. By scheduling out threads that are waiting for data, it is possible to hide memory access with actual computation [340]. This makes GPU programming very convenient. Contrary to Cell BE programming, the programmer does not need to worry about DMA data transfers and make sure that such transfers are completed before the data are accessed to. In a GPU device, all of this is handled transparently and automatically by the hardware [340].

Figure 5.6 summarizes the basic steps that we need to carry out in a CUDA application. First of all, the CPU instructs the GPU device to allocate some device memory for the input data. This is accomplished via the library function `cudaMalloc(...)`. Data are copied to the device memory via the library function `cudaMemcpy(...)`. The control is then passed to the GPU device by means of a kernel call. The syntax of such a call is of the form `my_kernel<<<N, M>>>(...)`, in which `my_kernel` is the name of the kernel, `N` is the grid size, and `M` is the block size. As in a C/C++ function, parameters to be passed to the kernel are enclosed in round parentheses. Once the threads have finished executing, the control returns to the CPU. The CPU then copies the result from the device memory to the host memory and the portion of device memory no longer needed is released.

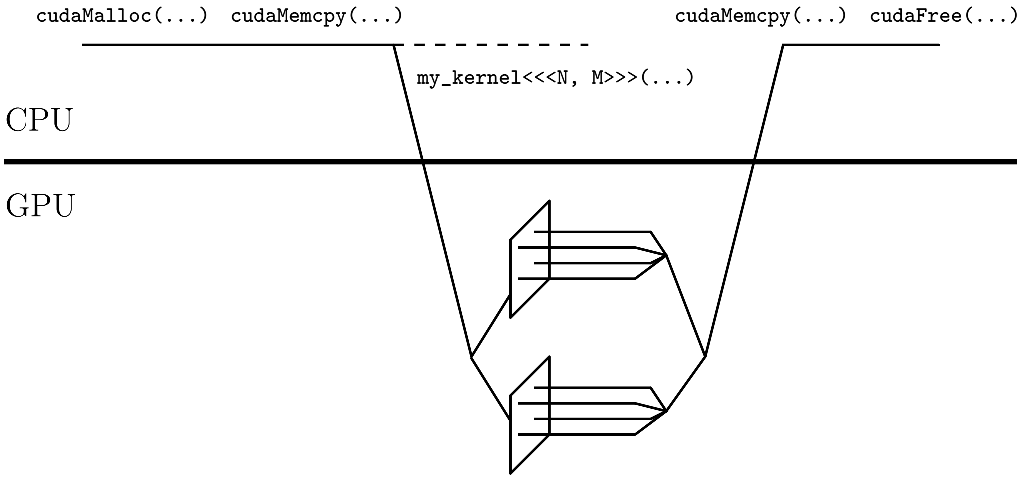


FIGURE 5.6. Workflow of a CUDA application

We conclude this introduction to GPU programming with a discussion about *scalability* [320]. With the term scalability, we refer to the ability of a system to be enlarged to accommodate growing amounts of computation or advances in technology. Recall that, in a GPU device, threads are grouped in blocks and blocks make up grids. We also noticed that only the threads in the same block can use the same bank of shared memory to share data fast. By design, the maximum number of threads in a thread block is a relatively small number–such as 1024—and it is not expected to grow as fast as technology improves. On the other hand, as technology evolves, a larger and larger number of thread blocks can be run in parallel. Recall that the programmer is not allowed to make any assumption on the order of execution of threads. This makes GPU programming scalable as the programmer is allowed only to rely on—and design his/her GPU code around—the maximum size of a thread block (which is a rather small and almost constant value across GPU technologies) and nothing else [340].

5.3.2 An Example

In this section, we will consider the same programming example we considered in § 5.2 for the Cell BE. This time, however, we will analyze CUDA code and we will run the program on a computer equipped with, at least, one GPU device. As before, the input arrays—allocated in host memory—are filled up with data. Recall that the GPU device can access only device memory. Therefore, to make it possible for the GPU device perform computation with the input data, two memory blocks are allocated in the device memory space. The host code then calls a particular library function to copy the input data to the blocks of device memory previously allocated.

The next step is to decide what each thread has to do and set up the thread blocks and block grid. In our implementation, we decided that the n^{th} thread will add together the n^{th} elements of the input arrays and put the result in the destination array. Therefore, the total number of threads has to be equal to or greater than the number of elements in each array. In our implementation, we decided to use 1D thread blocks of 256 threads each. We also chose a 1D thread grid and the number of blocks in the thread grid has to be calculated so that all the elements in the array are processed.

In the next step, the kernel function is called. The particular CUDA <<< . . . >>> notation instructs the hardware to generate a block grid, populate its blocks with threads, and start executing the threads. Threads are scheduled in hardware with no intervention from the programmer. Once all the threads have completed their execution, the result is copied from the device memory to the host memory. At this point, the computation on the device has terminated and the content of the device memory is no longer needed. Memory previously allocated on the device is released and the result is printed on the screen.

Let us consider now the portion of code that is executed on the GPU device. Recall that in CUDA, this piece of code is usually referred to as the kernel. In the programming example reported below, the kernel is defined by the `__global__` keyword. Its implementation is extremely simple. From the block index, the block size, and the thread index, a 1D index n is calculated. The code then fetches the n^{th} components of the two input arrays, add them together, and stores the result as the n-th component of the output array. Some care must be paid to make sure the code does not access memory locations beyond the arrays. This is accomplished by the `if(...)` statement that appear in the kernel implementation.

File `main.cu`

```
#include <cuda.h>
#include <math.h>
#include <stdio.h>

///////////////////////////////////////////////////////////////////////////////////////////

#define ARRAY_SIZE   1024

///////////////////////////////////////////////////////////////////////////////////////////

__global__ void array_add_kernel(float *c_dev, float *a_dev, float *b_dev, int array_size);

///////////////////////////////////////////////////////////////////////////////////////////

int main(int argc, char **argv) {
  float *a_dev, *b_dev, *c_dev;
  static float a[ARRAY_SIZE];
  static float b[ARRAY_SIZE];
  static float c[ARRAY_SIZE];
  int threads_per_block;
  int num_blocks;
  int i;

  // Fill in the input arrays
  for(i = 0; i < ARRAY_SIZE; ++i) {
    a[i] = cosf(1.00 + i);
    b[i] = sinf(1.00 + i);
  }
  // Allocate memory buffers on the device
  cudaMalloc(((void **) (& a_dev)), ARRAY_SIZE * sizeof(*a_dev));
  cudaMalloc(((void **) (& b_dev)), ARRAY_SIZE * sizeof(*b_dev));
  cudaMalloc(((void **) (& c_dev)), ARRAY_SIZE * sizeof(*c_dev));
  // Copy the input data from the host memory to the device memory
  cudaMemcpy(a_dev, a, ARRAY_SIZE * sizeof(*a), cudaMemcpyHostToDevice));
  cudaMemcpy(b_dev, b, ARRAY_SIZE * sizeof(*b), cudaMemcpyHostToDevice));
  // Set the number of threads per block
  threads_per_block = 256;
  // Calculate the number of blocks needed to process all the elements of the arrays
  num_blocks = ARRAY_SIZE / threads_per_block + ((ARRAY_SIZE % threads_per_block) > 0);
  // Call the kernel
  array_add_kernel<<<num_blocks, threads_per_block>>>(c_dev, a_dev, b_dev, ARRAY_SIZE);
  // Copy the result from the host memory to the device memory
  cudaMemcpy(c, c_dev, ARRAY_SIZE * sizeof(*c), cudaMemcpyDeviceToHost);
  // Free the memory that was allocated on the device
  cudaFree(a_dev);
  cudaFree(b_dev);
  cudaFree(c_dev);
  // Print the result on the screen
  for(i = 0; i < ARRAY_SIZE; ++i) {
    printf("%f␣+␣%f␣=␣%f\n", a[i], b[i], c[i]);
  }
  return(0);
}

///////////////////////////////////////////////////////////////////////////////////////////

__global__ void array_add_kernel(float *c_dev, float *a_dev, float *b_dev, int array_size) {
  const int n = blockIdx.x * blockDim.x + threadIdx.x;

  // Check if the array index points within the array
  if(n < array_size) {
    // Add the elements and store the result
    c_dev[n] = a_dev[n] + b_dev[n];
  }
  return;
}
```

5.4 The Future: Cell BE or GPU?

If we compare the CUDA code above with the same piece of code that we developed for the Cell BE, we will immediately notice that the CUDA code is simpler and easy to understand. We mentioned that one of the strategies often adopted by Cell BE programmers to increase performance is to use double buffering to overlap data transfer with computation. In the case of a GPU device, this step is automatically performed by the hardware. By pausing threads that are waiting for data from the device memory and by allowing execution of threads that already have data available, a GPU device can effectively attain high performance without any effort from the programmer. Another important difference between the Cell BE and a GPU device is that new and more powerful GPU devices are being regularly launched year after year and this trend is expected to continue. Because GPU devices are scalable, programs written for today's GPU devices will continue to work properly when run on tomorrow's GPU devices. On the other hand, the Cell BE architecture makes only a few provisions for future improvements and extensions. One of them regards the size of the local store. Although rumors that IBM is working on another processor in the Cell BE "family" are circulating, no release of an improved version of the Cell BE is expected for the next year or so.

While Cell BE programming for scientific applications was a hot topic around the year 2007, more recent developments in GPU technology have been able to provide scalable solutions, lower prices, and hardware that is easy to program. Although greatly increasing performance, the Cell's SPUs have been notoriously difficult to develop code for. However, and as we will see in the next section, the Cell BE still remains a viable solution in some applications. For example, in some algorithms it might be necessary to make access to a relatively large amount of data fast. The Cell BE's LS might be a possible candidate to store such data. A GPU device might simply not have enough shared memory to store all the data, resulting in the need to access global or constant memory every time a datum is needed. The same algorithm might, therefore, run slower on a GPU device than on the Cell BE.

5.5 2D ML Position Estimation on the Cell BE

In § 4.3 we described a simple algorithm to perform maximum-likelihood estimation of the 2D or 3D position of interaction within a scintillation crystal. The algorithm we described, namely the contracting-grid algorithm [45, 173, 253], uses the PMT data $\boldsymbol{g}$

as well as calibration data—in the form of the MDRF $\overline{\boldsymbol{g}}(\boldsymbol{R})$—to calculate an estimate $\hat{\boldsymbol{R}}_{\mathrm{MLE}}$ of $\boldsymbol{R}$ according to

$$\hat{\boldsymbol{R}}_{\mathrm{MLE}} = \underset{\boldsymbol{R}_0 \in D}{\arg\max} \left[\sum_{k=1}^{K} \left\{ g_k \ln \left[\overline{g}_k(\boldsymbol{R}_0) \right] - \overline{g}_k(\boldsymbol{R}_0) \right\} \right]. \tag{5.2}$$

In this section, we want to present some details of the implementation—in the C programming language—of the 2D maximum-likelihood estimation algorithm on the Cell BE architecture, which we introduced in § 5.2.

When an emitter distribution is imaged, a list of events is collected by gamma-ray cameras. For each event in the list, a PMT data vector is given. The goal of the 2D maximum-likelihood estimation algorithm we are describing in this section is to use the PMT data to estimate the location of interaction for each event. As we discussed in some detail in § 5.2, the Cell BE architecture includes one Power Processor Element (PPE) and eight Synergistic Processor Elements (SPEs). The main idea of the parallel algorithm we are going to propose is to break down the list of events into many sublists, and assign each sublist to one of the SPEs or PPE for processing. The SPEs and the PPE will, independently, perform maximum-likelihood estimation for each entry of the sublist they were assigned to. When the SPEs and the PPE have completed their calculations, the partial results are combined.

As we noted in § 4.3, in a practical case, the calibration data $\overline{\boldsymbol{g}}(\boldsymbol{R})$ is available only for a discrete set of 2D points $\boldsymbol{R}_n$, for $n = 1, \ldots, N$. For the problem at hand, the scintillation crystal was divided into an array of 161 × 161 pixels and the crystal itself was coupled via a light guide to a 3×3 array of PMTs. As it was remarked in § 4.3, it is usually convenient to zero-pad the 161 × 161 calibration data $\overline{\boldsymbol{g}}(\boldsymbol{R}_n)$ to a power of 2 in each dimension to make the algorithm run faster. With zero padding, the calibration data $\overline{\boldsymbol{g}}(\boldsymbol{R}_n)$ consist of an array of size 256 × 256 (thus, $N = 256^2 = 65536$) and the estimation algorithm will require just seven iterations to converge. This number of iterations differs from the one reported in [45] because, in our case, the scintillation crystal was of size 161 × 161 pixels, while in [45] its size was just 81 × 81. Apart from the number of iterations, our algorithm is very similar to the one described in [45].

Conceptually, the zero padding is equivalent to stating that, if a hypothetical scintillation event were to occur outside the crystal, then no signal would be produced by the PMTs. With this assumption and noticing that the number of components in the vectors $\boldsymbol{g}$ and $\overline{\boldsymbol{g}}(\boldsymbol{R}_n)$ is $K = 9$, we can rewrite (5.2) as

$$\hat{n}_{\mathrm{MLE}} = \operatorname*{arg\,max}_{n_0=1,\ldots,65536} \left[\sum_{k=1}^{9} \left\{ g_k \ln \left[\overline{g}_k(\boldsymbol{R}_{n_0}) \right] - \overline{g}_k(\boldsymbol{R}_{n_0}) \right\} \right].$$

The total size of calibration data is thus $NK = 65536 \cdot 9 = 589824$ numerical values. Recall that these data represent the means of Poisson random variables, thus they are, in general, non-integer numbers. In our Cell BE implementation, we opted to store these data as floating-point numbers in single precision. Each of such values requires 4 bytes of storage, bringing the total storage requirements for the calibration data to 2304 kB. Because this number exceeds the size of the local store (which is 256 kB, see § 5.2), an *ad hoc* data arrangement is needed.

An important observation is that, during the first few iterations, the contracting-grid algorithm accesses only a subset of the whole calibration data $\overline{\boldsymbol{g}}(\boldsymbol{R}_n)$. Recall that, at each iteration, the algorithm calculates the likelihood for points $\boldsymbol{R}_n$ in a 4×4 grid. Thus, at the first iteration, only 16 calibration data vectors will be needed. To carry out the second iteration, all that is needed are 16 out of 256 calibration vectors. Which ones in this set of 256 vectors will actually be needed depends on the calculations that were carried out during the first iteration. Similarly, for the third iteration, only 16 out of 4096 calibration data vectors will actually be needed. Again, which vectors will actually be needed depends on calculations carried out in the previous two iterations. A direct consequence of this simple observation is that, to carry out the first three iterations of the algorithm, only a limited amount of data is actually needed. For example, for the first iteration $16 \cdot 9 \cdot 4 = 576$ bytes of data are needed. The second iteration would require $16^2 \cdot 9 \cdot 4 = 9216$ bytes of data to be available. For the third iteration, $16^3 \cdot 9 \cdot 4 = 147456$ bytes of data will be needed. To summarize, the algorithm needs no more than $576+9216+147456 = 157248 \approx 154$ kB of data to complete the first three iterations of the algorithm. Notice that these data would fit in the local store available on each SPE. Therefore, we opted to store the data for the first three iterations of the algorithm in each SPE's local store, thus avoiding the need to fetch calibration data via time-consuming DMA transfers during the first three iterations of the algorithm.

Calibration data for the remaining four iterations of the algorithm were arranged in such a way that estimations for scintillation events occurring at or near the center of the scintillation crystal face would be performed faster than estimation for scintillation events occurring near the edge of the crystal. The fundamental reason for this implementation choice comes from the fact that, usually, the object being imaged is placed at or near the center of the field of view. Thus, it seems reasonable to expect that the central area of the scintillation crystal would receive a larger number of gamma-ray photons than the areas near the edge of the crystal. Calibration data for the 56×56 central area of the crystal was partitioned into four regions, labeled A, B, C, and D, as shown in Figure 5.7. Storing one of the 32×16 blocks—marked as A and B in Figure 5.7—of calibration data takes $32 \cdot 16 \cdot 9 \cdot 4 = 18432$ bytes of storage, while the two semi-frames (denoted as C and D) require $(56^2 - 32^2)/2 \cdot 9 \cdot 4 = 21600$ bytes. Thus, all these data blocks are small enough that any of them can be stored in the SPEs' local stores, along with the calibration data needed during the first three iterations. In our C implementation of the algorithm, parameters such as the size of calibration data regions A, B, C, and D were defined via the pre-processor directive `#define`. This increased flexibility and allowed us to easily change the size of such data regions when fine-tuning our implementation to obtain the best performance within the limitations imposed by the hardware. The algorithm was specifically written to be run on a Sony PlayStation 3, for which only six out of the eight SPEs are accessible to the programmer. Thus each of the 32×16 blocks (A and B) shown in Figure 5.7 was stored twice, in two different local stores. The two remaining areas (C

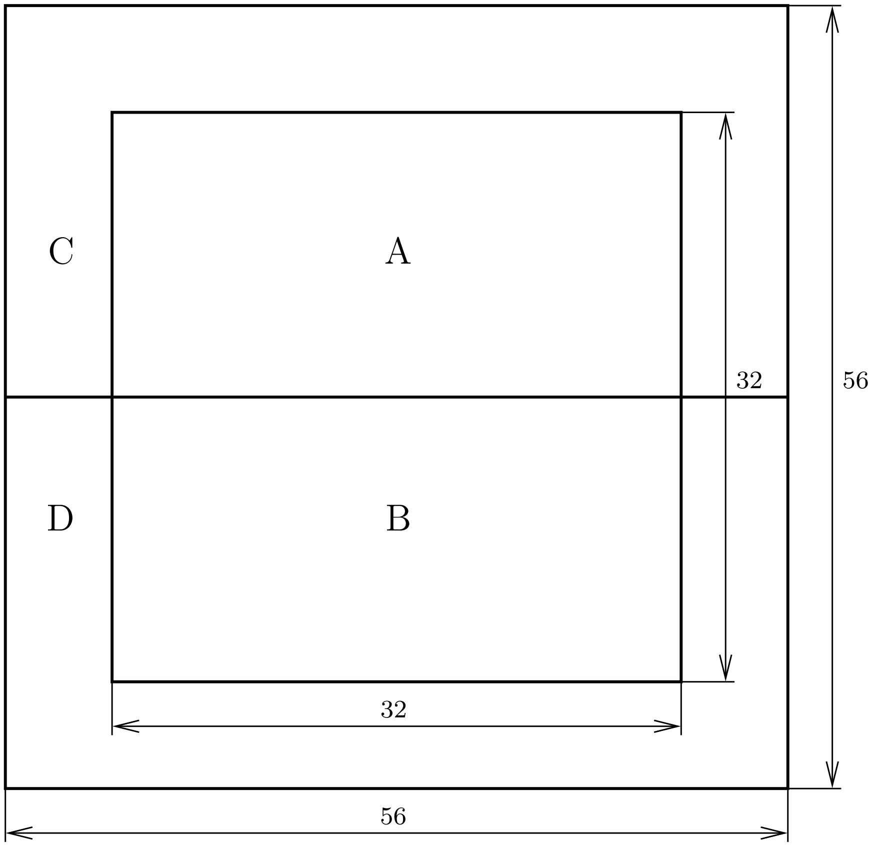


Figure 5.7. Partitioning of the 56 × 56 calibration data for the central area of the crystal for our PlayStation 3 implementation

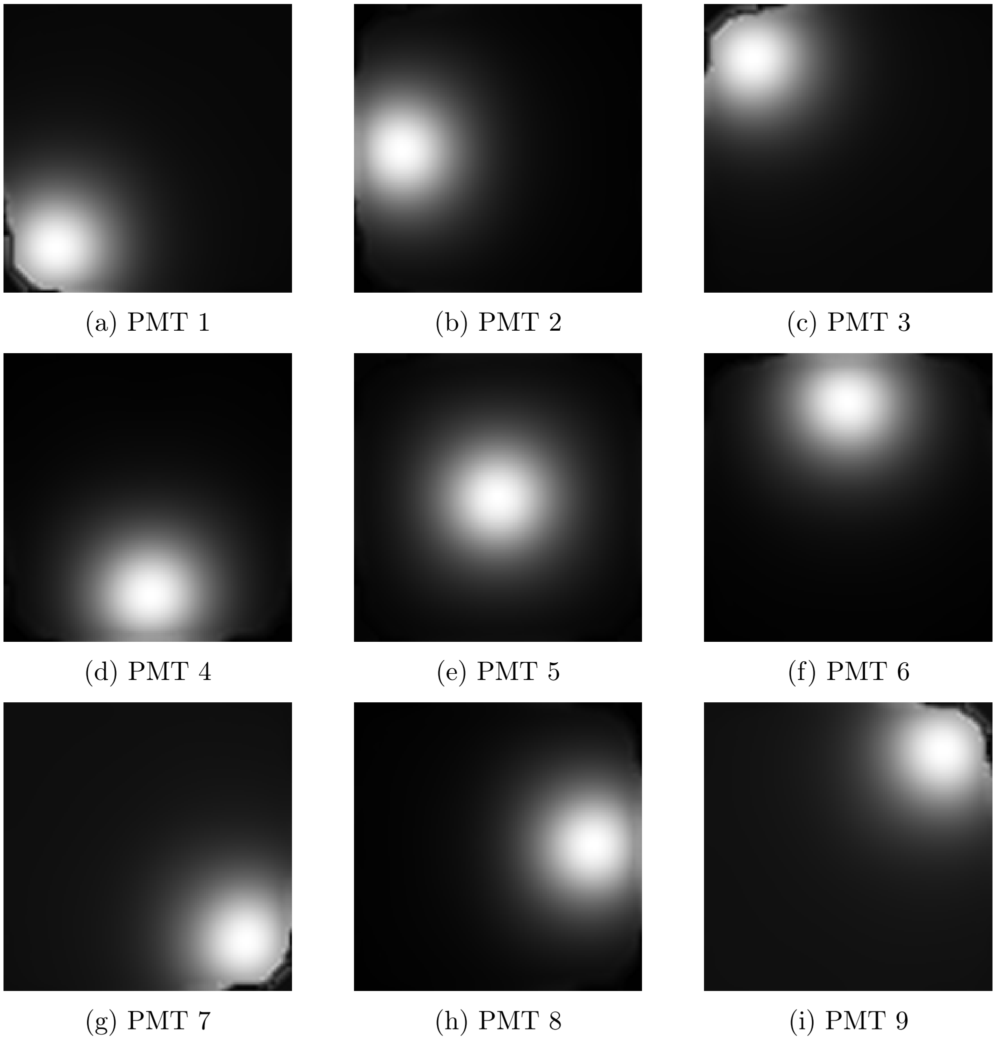
and D) depicted in Figure 5.7 were stored only once.

When, after the first three iterations of the algorithm, calibration data for the 56×56 central area of the crystal are needed, the index of the SPE that stores such data is determined, and an appropriate SPE-to-SPE DMA transfer is initiate to fetch the data. Again, the reason for which the 32×16 blocks (A and B) were stored twice is that it would seem reasonable to expect that such data would be accessed more often than the data in the two semi-frame regions (C and D). Having two copies of each of the 32×16 blocks of data allowed us to obtain load balancing when accessing data: each SPE storing either data block A or B locally would serve data requests from two other SPEs as well as data requests from itself.

Finally, when during the computation, calibration data outside the central 56×56 region of the detector's crystal are needed, an appropriate DMA transfer from main memory is performed. This type of DMA transfer is much slower than an SPE-to-SPE DMA transfer. However, owning to the assumption that most of the gamma-ray interactions will take place in the central area of the crystal, the algorithm still performs well.

As a final note, we want to remark that our implementation heavily relies on the parallel capabilities that the Cell BE offers. Besides using all the SPEs available for computation, the algorithm resorts to vector instructions to operate on vector operands and to efficiently calculate the values of the log-likelihood. Because our algorithm stores intermediate values as single-precision floating-point numbers, each vector instruction we used operated on vectors of four elements. Thus, just by using vector instructions, each steps in our algorithm would replace four steps in an equivalent non-vectorized implementation of the same algorithm. To further speed up the calculation and to avoid time-consuming `for(...)` and `if(...)` statements, our implementation took advantage of the Cell BE's "bits and mask" intrinsics [338]. For example, when a vector comparison was needed to calculate the maxima between corresponding entries in two different vectors, a bit pattern was computed using a specific Cell BE intrinsic. The calculated bit pattern and another intrinsic were then used to select the entries at which the maxima occurred. This way of manipulating the data allowed us to calculate at which point of the 4 × 4 grid the log-likelihood attained the largest value without actually making use of any `if(...)` statement. Iterating over the 4 × 4 grid was replaced with manual `for(...)`-loop unrolling and Cell BE vector instructions.

The Cell BE code we developed was compiled with IMB's XL C/C++ compiler version 9.0, and run on a PlayStation 3 hardware platform. The Linux operating system and the Cell BE Software Development Kit (SDK) version 3.0 had been previously installed on the PlayStation 3. The ML position estimation algorithm was run on real data consisting of a list of 867685 events. The calibration data we used are depicted in Figure 5.8. For each PMT, the mean response was measured by moving a gamma-ray point source over the camera face [6]. Thus, the whole calibration data form a set of nine 161 × 161 images, one for each PMT. The algorithm was also implemented for a conventional computer architecture, compiled with the GCC compiler version 4.4.1, and run on different hardware platforms. Running time was recorded and the average number of events processed per second was calculated by dividing the number of events in the list by the average running time (estimated over at least 100 iterations of the algorithm). Performance results are reported in Table 5.2. The estimated locations of interaction (one for each event in the list) were further processed to generate an image by simply counting the number of interactions that fell within each pixel of the detector. The image we obtained is depicted as Figure 5.9. The object being imaged was a mouse that had previously been injected with a radiotracer. Bright regions in the image correspond to organs and tissues with high activity. Structures that can easily be recognized are the bladder, the tail, arms, and legs.



(a) PMT 1

(b) PMT 2

(c) PMT 3

(d) PMT 4

(e) PMT 5

(f) PMT 6

(g) PMT 7

(h) PMT 8

(i) PMT 9

Figure 5.8. Images of the PMT responses. Each pixel in the images represents the PMT response for a gamma-ray point source centered over that location on the camera face

Hardware Platform	**Events/s**
PlayStation 3 (Cell Broadband Engine)	866374.69
Intel® Xeon® CPU E5540, 2.53 GHz	391270.03
Intel® Xeon® CPU E5506, 1.6 GHz	286228.78
AMD Phenom™ II X4 965 Processor, 3.4 GHz	147307.02
AMD Phenom™ 9950 Quad-Core Processor, 2.6 GHz	99412.38
AMD Phenom™ 9850 Quad-Core Processor, 2.5 GHz	84158.03
Dual Core AMD Opteron™ Processor 270, 2 GHz	54676.95
AMD Athlon™ Processor MP 2400+, 2 GHz	11827.62

TABLE 5.2. Performance results for the 2D ML estimation algorithm

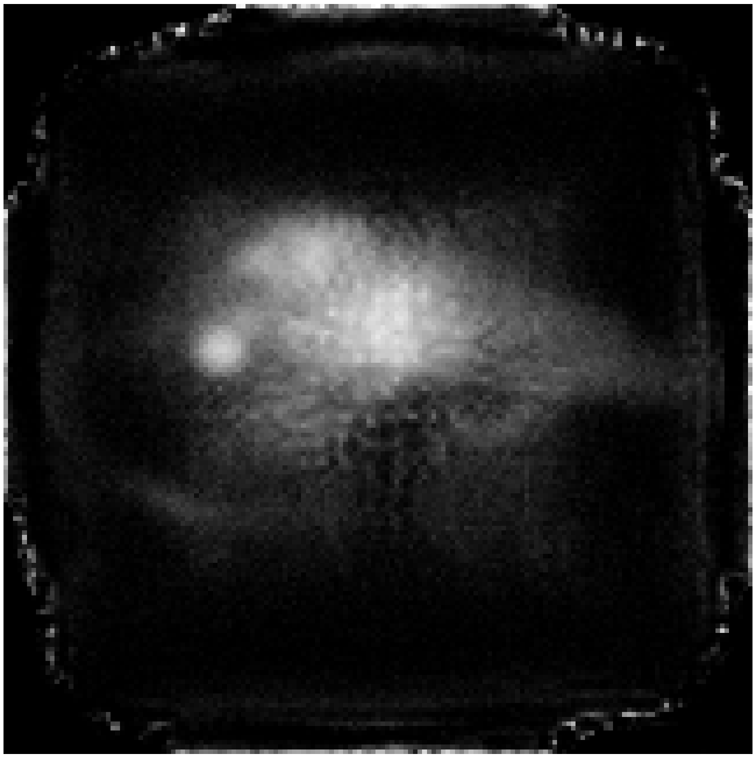


FIGURE 5.9. Result of the 2D ML position estimation from real data shown as a log-scale image

5.6 3D ML Position Estimation on a GPU Device

Consider now the case of a gamma-ray camera with a thick scintillation crystal. As we saw in § 1.4, thick crystals have higher sensitivity than thin crystals. We also remarked that, unless the depth of interaction Z within a thick crystal is estimated along with the (X, Y) location, parallax errors will occur. In this section, we discuss an implementation of a GPU algorithm for the maximum-likelihood (ML) estimation of the (X, Y, Z) location of interaction of a gamma-ray photon within a thick scintillation crystal. Our GPU implementation of the ML algorithm for the estimation of the 3D location of interaction uses the same ideas that were used to develop Cell BE code for the 2D ML estimation of the location of interaction. The two architectures—the Cell BE and NVIDIA's GPUs—are, however, substantially different. Therefore, one of the goals of this section is to show how the same problem and the same mathematical approach to solving it are implemented in two completely different ways. Another goal of this section is to verify through simulation one of the properties of ML estimation we briefly summarized in § 3.2, namely that, as the number of photoelectrons per event increases, ML estimates approach Gaussian statistics with covariance matrix given by the inverse of the Fisher information matrix.

The ML estimation of the 3D location of interaction of a gamma-ray photon with the crystal is mathematically formalized as

$$\hat{\boldsymbol{R}}_{\mathrm{MLE}} = \operatorname*{arg\,max}_{\boldsymbol{R}_0 \in D} \left[\sum_{k=1}^{K} \left\{ g_k \ln \left[\overline{g}_k(\boldsymbol{R}_0) \right] - \overline{g}_k(\boldsymbol{R}_0) \right\} \right],$$

in which $\hat{\boldsymbol{R}}_{\mathrm{MLE}}$ is the estimate for the unknown location of interaction $\boldsymbol{R}$, g_k are noisy PMT outputs for $k = 1, \ldots, K$ and, finally, the $\overline{g}_k(\boldsymbol{R}_0)$ are the calibration data. As in the case of 2D ML estimation, the calibration data are provided for a set of discrete points $\boldsymbol{R}_0$ within the crystal. For our 3D estimation, however, $\overline{g}_k(\boldsymbol{R}_0)$ is sampled on a 3D lattice of points, instead of on a grid. The contracting-grid algorithm of [45,173] is still well suited to solve this estimation problem. For convenience, we will still use the word “grid” to refer to the 3D lattice of points at which the log-likelihood is calculated at each iteration of the algorithm.

To describe the implementation of the algorithm on a GPU device, it is convenient to recall the thread hierarchy depicted in Figure 5.5 and the different memory spaces available on a GPU device (see Table 5.1), along with their main features. The calibration data $\overline{g}_k(\boldsymbol{R}_0)$ for discrete points $\boldsymbol{R}_0$ do not change throughout the iterations of the algorithm. Thus, in our implementation, we chose to store these data in constant memory, as opposed to global memory. The main advantage of this implementation decision is that constant memory is cached, while global memory is not (see Table 5.1).

As in the case of 2D ML estimation, the input to the algorithm is a list of J PMT data vectors $\{\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(J)}\}$. Once this list is copied from host memory to global memory, the appropriate thread hierarchy is generated. Because threads in the same block have the ability of share data via shared memory, we decided to associate a thread block to each of the PMT data vectors $\boldsymbol{g}^{(j)}$. Upon invocation of the kernel, each thread in the block first loads one of the components of $\boldsymbol{g}^{(j)}$ from global memory to shared memory. These loads are executed concurrently and are localized in space (meaning that most of the loads that are invoked during a short time interval are loads from the same memory bank), thus the hardware is able to carry out a few of them at the same time.

An iteration of the contracting-grid algorithm consists on calculating the log-likelihood for points $\boldsymbol{R}_0$ on a $4 \times 4 \times 4$ grid. Recall that, in CUDA, a thread block can be 1D, 2D, or 3D and the total number of threads in the block cannot exceed 1024 [340]. This particular computational model turned out to be very convenient for implementation of the 3D contracting-grid algorithm. Indeed, 3D thread blocks of size $4 \times 4 \times 4$ were generated, with each thread in the block calculating the log-likelihood for one of the points in the grid. Because the thread block is 3D, the thread index associated to each thread is actually a vector index of three components, which were used to quickly index the point $\boldsymbol{R}_0$ within the $4 \times 4 \times 4$ grid. Once the values of the log-likelihood are calculated, they are stored in the block's global memory. One of the threads of the block is in charge of finding the maximum of the log-likelihood values, calculate the center of the grid for the next iteration, and update the grid step sizes. The process described in this paragraph is then repeated until all the step sizes are reduces to one, at which point the algorithm terminates.

To test the effectiveness of the algorithm, we considered a NaI(Tl) thick crystal of size 52.30 mm × 52.30 mm × 25 mm divided into $69 \times 69 \times 25$ voxels. The crystal was coupled to an 8×8 array of PMTs [60]. The same calibration data of [60] were used for our tests. The GPU algorithm was run with simulated PMT data from [355]. This data set consisted of a list of $J = 2 \cdot 10^5$ vectors of simulated PMT responses for a 45°, 511 keV gamma-ray incident on the camera's crystal. A cross-section (for $y = 0$ mm) of the 3D ML-estimated distribution of the location of interaction is reported as Figure 5.10, while the first 5,000 estimated locations of interaction are shown in the scatter plot of Figure 5.11.

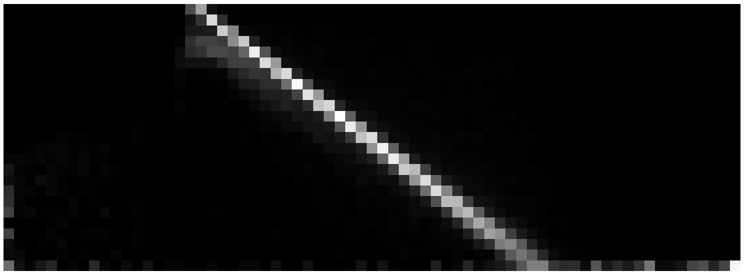


FIGURE 5.10. Cross-section of the 3D ML-estimated distribution of the location of interaction for a 45°, 511 keV beam of gamma-ray photons incident on the camera's crystal

As for the case of 2D ML estimation, C code was developed for a conventional hardware platform as well. To evaluate algorithm performance we used the same data we used before (i.e., a list of $J = 2 \cdot 10^5$ vectors of simulated PMT responses) and we ran the algorithm 100 times on different hardware platforms, including GPU devices. Results, measured in terms of number of processed events per seconds, are

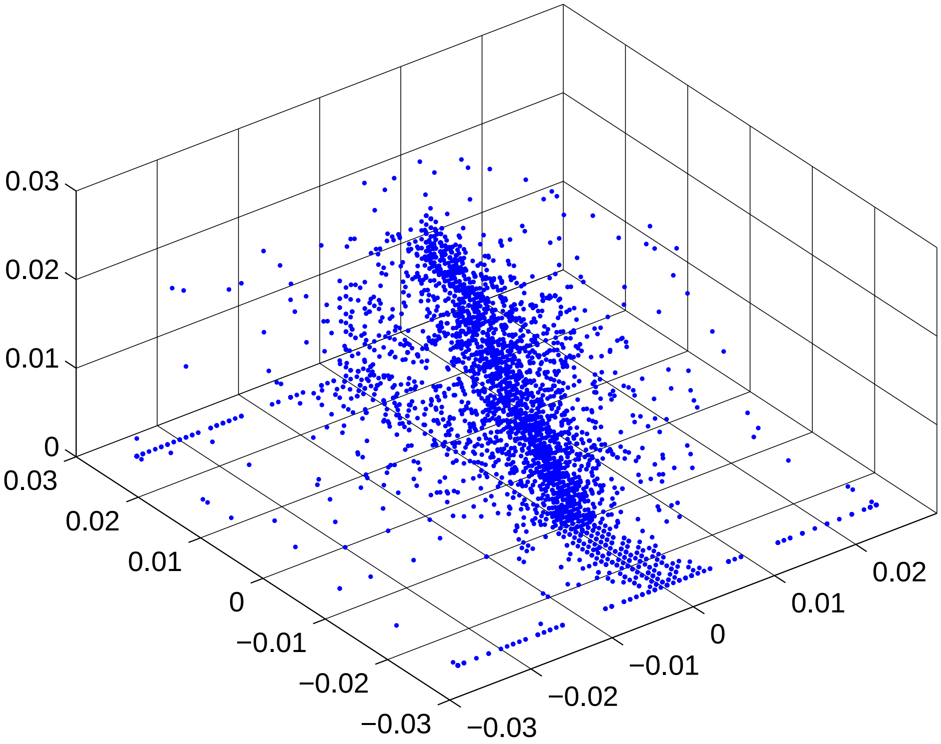


Figure 5.11. Scatter plot of the first 5,000 estimated locations of interaction for a 45°, 511 keV gamma-ray incident on the camera's crystal (the detector's crystal measured 52.30 mm × 52.30 mm × 25.00 mm)

reported in Table 5.3. When more than one GPU device was available, the list of events was divided into sublists of approximatively the same length and each sublist was assigned to a GPU device.

Our implementation takes advantage of the lifetime of memory spaces for a GPU device. As Table 5.1 shows, global, constant, and texture memories have a lifetime of the whole application. This means that so long as the application does not terminate or the memory is not deallocated or overwritten, the content of such memory spaces is left unchanged. In a typical application, we can decide to load in the device memory the calibration only once, at the beginning of our code. Every time PMT data will have to be processed to estimate the 3D locations of interaction, the calibration data will already be in the device's constant memory and ready to be used, without incurring in any time penalty due to host-to-device calibration data transfers.

Hardware Platform	**Events/s**	**Speedup**
Intel® Xeon® CPU E5540, 2.53 GHz*	3279.92	—
NVIDIA GeForce GTX 295, 1 device	52561.00	16.03
NVIDIA GeForce GTX 295, 2 devices	104335.32	31.81
NVIDIA GeForce 9800 GX2, 1 device	40654.62	12.40
NVIDIA GeForce 9800 GX2, 2 devices	80607.25	24.58
NVIDIA GeForce 9800 GX2, 3 devices	120516.67	36.74
NVIDIA GeForce 9800 GX2, 4 devices	160645.01	48.98
NVIDIA GeForce 9800 GX2, 5 devices	197454.39	60.20
NVIDIA GeForce 9800 GX2, 6 devices	232461.37	70.87
NVIDIA Tesla C1060, 1 device	80036.25	24.40
NVIDIA Tesla C1060, 2 devices	158945.45	48.46
NVIDIA Tesla C1060, 3 devices	237228.06	72.33
NVIDIA Tesla C1060, 4 devices	313789.08	95.67
NVIDIA Tesla C2050, 1 device	241359.90	73.59
NVIDIA Tesla C2050, 2 devices	471183.70	143.66
NVIDIA Tesla C2050, 3 devices	686089.59	209.18
NVIDIA Tesla C2050, 4 devices	883516.32	269.37

*Using only one core.

Table 5.3. Performance results for the 3D ML estimation algorithm

We have said numerous times that, because of measurement noise, the data $\boldsymbol{g}$ that are used to perform ML estimation are random, and so is any non-constant quantity calculated from $\boldsymbol{g}$. We also have discussed in § 4.6.1 how the Fisher information matrix can be used to evaluate the performance of any unbiased estimator. For clarity, we will repeat here some of the material that was presented in § 4.6.1, but we will use the notation relevant to the problem of estimating the 3D location of interaction from noisy data $\boldsymbol{g}$. We will first consider the case of estimation by means of any estimator $\hat{\boldsymbol{R}}$, and then we will move on by examining the case in which estimates are calculated via ML methods.

The bias $\boldsymbol{b}(\boldsymbol{R})$ at $\boldsymbol{R}$ for the random variable $\hat{\boldsymbol{R}}$ is written as

$$\boldsymbol{b}(\boldsymbol{R}) = \left\langle \hat{\boldsymbol{R}} - \boldsymbol{R} \right\rangle_{\hat{\boldsymbol{R}}|\boldsymbol{R}} = \left\langle \hat{\boldsymbol{R}} \right\rangle_{\hat{\boldsymbol{R}}|\boldsymbol{R}} - \boldsymbol{R},$$

which, for our 3D estimation problem above, turns out to be a 3 × 1 vector. The covariance matrix of the estimates $\hat{\boldsymbol{R}}$ at $\boldsymbol{R}$ is given by

$$\mathbf{K}_{\hat{\boldsymbol{R}}|\boldsymbol{R}} = \left\langle \left[\hat{\boldsymbol{R}} - \langle \hat{\boldsymbol{R}} \rangle_{\hat{\boldsymbol{R}}|\boldsymbol{R}} \right] \left[\hat{\boldsymbol{R}} - \langle \hat{\boldsymbol{R}} \rangle_{\hat{\boldsymbol{R}}|\boldsymbol{R}} \right]^{\mathsf{T}} \right\rangle_{\hat{\boldsymbol{R}}|\boldsymbol{R}},$$

while an expression for the $(m,n)^{\text{th}}$ component of the Fisher information matrix $\mathbf{F}_{\boldsymbol{R}}$ turns out to be [69]

$$[\mathbf{F}_{\boldsymbol{R}}]_{m,n} = \left\langle \left[\frac{\partial}{\partial R_m} \ln \mathrm{pr}\big(\boldsymbol{g} \mid \boldsymbol{R}\big) \right] \left[\frac{\partial}{\partial R_n} \ln \mathrm{pr}\big(\boldsymbol{g} \mid \boldsymbol{R}\big) \right] \right\rangle_{\boldsymbol{g}|\boldsymbol{R}},$$

for $m = 1, 2, 3$, $n = 1, 2, 3$, and we denoted the components of the vector $\boldsymbol{R}$ as R_1, R_2, and R_3. Finally, the Cramér-Rao lower bound (CRLB) assumes the form [69]:

$$\left[\mathbf{K}_{\hat{\boldsymbol{R}}|\boldsymbol{R}}\right]_{m,m} = \sigma^2_{\hat{R}_m} \geqslant \left[\mathbf{F}^{-1}_{\boldsymbol{R}}\right]_{m,m}. \tag{5.3}$$

We now turn our attention to the special case in which $\hat{\boldsymbol{R}} = \hat{\boldsymbol{R}}_{\text{MLE}}$. As we briefly alluded to in § 3.2, maximum-likelihood estimators are asymptotically efficient (meaning that, as more and more photoelectrons per event are measured by the PMTs, $\mathbf{K}_{\hat{\boldsymbol{R}}|\boldsymbol{R}}$ approaches $\mathbf{F}^{-1}_{\boldsymbol{R}}$ and the inequality in (5.3) becomes an equality). Although the actual probability density function $\mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE}} \mid \boldsymbol{R})$ of the ML estimates $\hat{\boldsymbol{R}}_{\mathrm{MLE}}$ for an interaction at $\boldsymbol{R}$ can, in theory, be calculate with very good accuracy [356], the method of [356] requires conditioning on an auxiliary parameters, which can be difficult to define and calculate in a practical case. For this reason, in the remainder of this section, we will adopt the method presented in [357]. Even though [357] assumes a Gaussian model for the noise, many practical cases allow us to invoke the central limit theorem [127, 128] and apply the theory developed in [357] to non-Gaussian noise models as well. For the case of 3D estimation from PMT data, the conditions of the central limit theorem are satisfied when we have a sufficiently large number of low-noise PMTs. For the problem at hand, results reported in [357] and [268] show:

$$\mathrm{pr}(\hat{\boldsymbol{R}}_{\mathrm{MLE}} \mid \boldsymbol{R}) \approx \frac{\sqrt{\det(\mathbf{F}_{\boldsymbol{R}})}}{(2\pi)^{3/2}} \exp\left[-\frac{1}{2}\left(\hat{\boldsymbol{R}}_{\mathrm{MLE}} - \boldsymbol{R}\right)^{\mathsf{T}} \mathbf{F}_{\boldsymbol{R}} \left(\hat{\boldsymbol{R}}_{\mathrm{MLE}} - \boldsymbol{R}\right)\right]. \tag{5.4}$$

The expression above is the probability density function of a multivariate normally-distributed random variable with mean $\boldsymbol{R}$ and covariance matrix $\mathbf{F}_{\boldsymbol{R}}^{-1}$.

To derive an expression for $\mathbf{F}_{\boldsymbol{R}}$, we first need to recall that in § 4.1.6 we denoted by n_k the number of photoelectrons produced when a gamma-ray photon entered the k^{th} PMT and struck the photocathode (see Figure 4.7). In § 4.1.6, we also argued that the random variable n_k follows Poisson statistics with mean $\overline{n}_k(\boldsymbol{R}, E_{\mathrm{int}})$. Through a noisy channel, the voltage at the output of the k^{th} PMT reaches an analog-to-digital (A/D) converter, which converts the output voltage into an integer number g_k. The noisy channel and A/D conversion introduce a small error, which we can model as normally distributed with standard deviation σ of a few A/D units [358]. Hence [279],

$$\begin{aligned}\mathrm{pr}(\boldsymbol{g} \mid \boldsymbol{R}) = \prod_{k=1}^{K} \mathrm{pr}(g_k \mid \boldsymbol{R}) = \prod_{k=1}^{K} \sum_{n_k=0}^{\infty} \mathrm{pr}(g_k \mid n_k) \, \mathrm{Pr}(n_k \mid \boldsymbol{R}) = \\ = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{n_k=0}^{\infty} \exp\left[-\frac{1}{2}\frac{(g_k - r n_k)^2}{\sigma^2}\right] \frac{\left[\overline{n}_k(\boldsymbol{R}, E_{\mathrm{int}})\right]^{n_k}}{n_k!} e^{-\overline{n}_k(\boldsymbol{R}, E_{\mathrm{int}})},\end{aligned}$$

in which the number r is the A/D conversion factor. A useful approximation is [279]

$$\begin{aligned}
[\mathbf{F}_{\boldsymbol{R}}]_{m,n} &\approx \sum_{k=1}^{K} \frac{r^2}{\sigma^2 + r^2 \overline{n}_k(\boldsymbol{R}, E_{\text{int}})} \frac{\partial \overline{n}_k(\boldsymbol{R}, E_{\text{int}})}{\partial R_m} \frac{\partial \overline{n}_k(\boldsymbol{R}, E_{\text{int}})}{\partial R_n} = \\
&= \sum_{k=1}^{K} \frac{1}{\sigma^2 + r \overline{g}_k(\boldsymbol{R})} \frac{\partial \overline{g}_k(\boldsymbol{R})}{\partial R_m} \frac{\partial \overline{g}_k(\boldsymbol{R})}{\partial R_n},
\end{aligned} \tag{5.5}$$

where the second form follows from $\overline{g}_k(\boldsymbol{R}) = r\overline{n}_k(\boldsymbol{R}, E_{\text{int}})$.

One necessity that arises in the calculation of the Fisher information matrix $\mathbf{F}_{\boldsymbol{R}}$ above is that partial derivatives of the calibration data $\overline{\boldsymbol{g}}(\boldsymbol{R})$ are needed. In our case, however, calibration data were available only for a discrete set of points $\boldsymbol{R}$. We therefore turned our attention to data smoothing with spines to obtain an analytic expression for the calibration data from which partial derivatives can be calculated. Alternatives to smoothing with splines have been proposed as well [24, 60]. Spline functions are introduced in Appendix A, along with their main properties. For our data smoothing procedure, we used 3D spline approximation for each of the K PMT outputs. With respect to the notation developed in Appendix A, we used $m_x = m_y = m_z = 3$, $M_x = M_y = 10$, and $M_z = 5$. Plots of the calibration data and their fit are shown in Figure 5.12 through Figure 5.17. Each figure reports plots of the $K = 64$ PMT outputs when only one variable at a time is changed and the others are kept fixed. Figure 5.12 through Figure 5.14 correspond to an interaction location $\boldsymbol{R}$ close to crystal's entrance face: PMT outputs are numerically low and most of the PMTs output a non-zero signal. On the contrary, when the interaction occurs deeper into the crystal and at a point $\boldsymbol{R}$ closer to the PMTs, only a few PMTs produce a large signal, as shown in Figure 5.15 through Figure 5.17.

Fitting the calibration data also allowed us to re-sample them and obtain calibration data sampled on a much finer grid than was originally provided. With this finer sampling, more iterations of the contracting-grid algorithm could be calculated. Although ML estimation on this finely re-sampled calibration data was considerably slower than ML estimation on the original calibration data, the re-sampled calibration

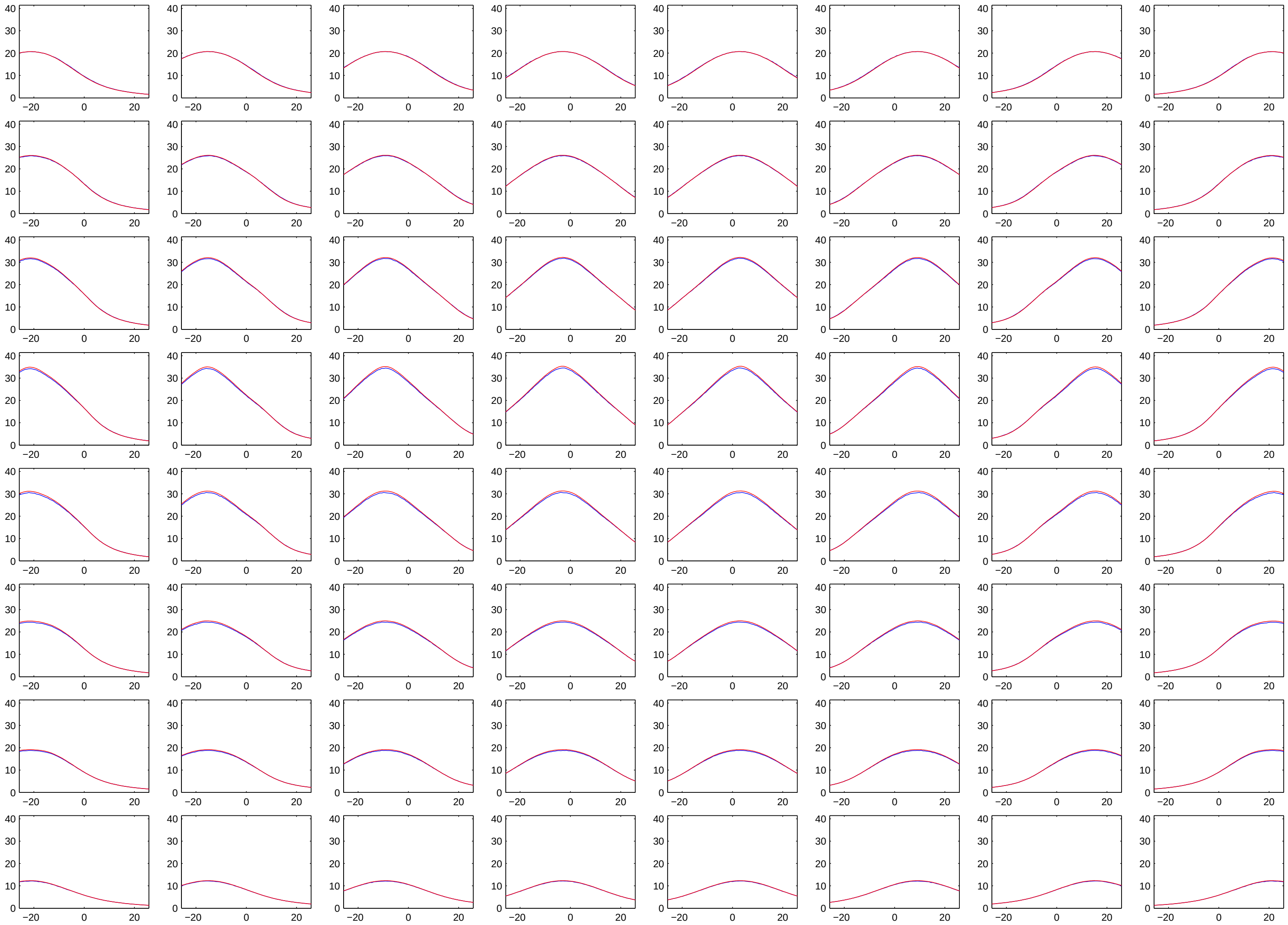


FIGURE 5.12. Plots as functions of X of PMT response functions $\overline{\boldsymbol{g}}(X, Y_0, Z_0)$ for $X \in [-26.15, 26.15]$ mm, $Y_0 = -3.789855$ mm and $Z_0 = 0.50$ mm (blue lines) along with their spline approximation (red lines)

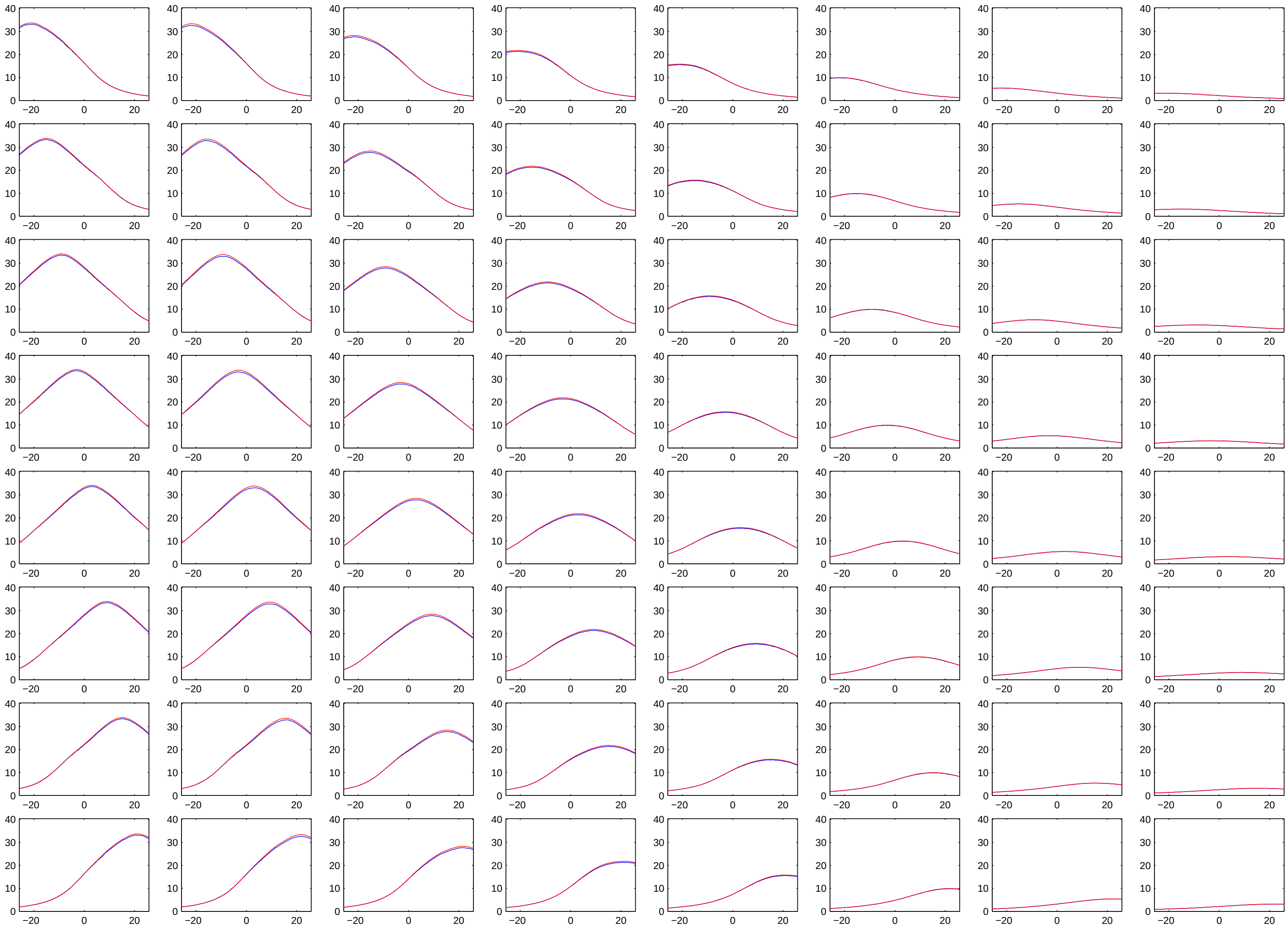


FIGURE 5.13. Plots as functions of Y of PMT response functions $\overline{\boldsymbol{g}}(X_0, Y, Z_0)$ for $X_0 = -18.949275$ mm, $Y \in [-26.15, 26.15]$ mm and $Z_0 = 0.50$ mm (blue lines) along with their spline approximation (red lines)

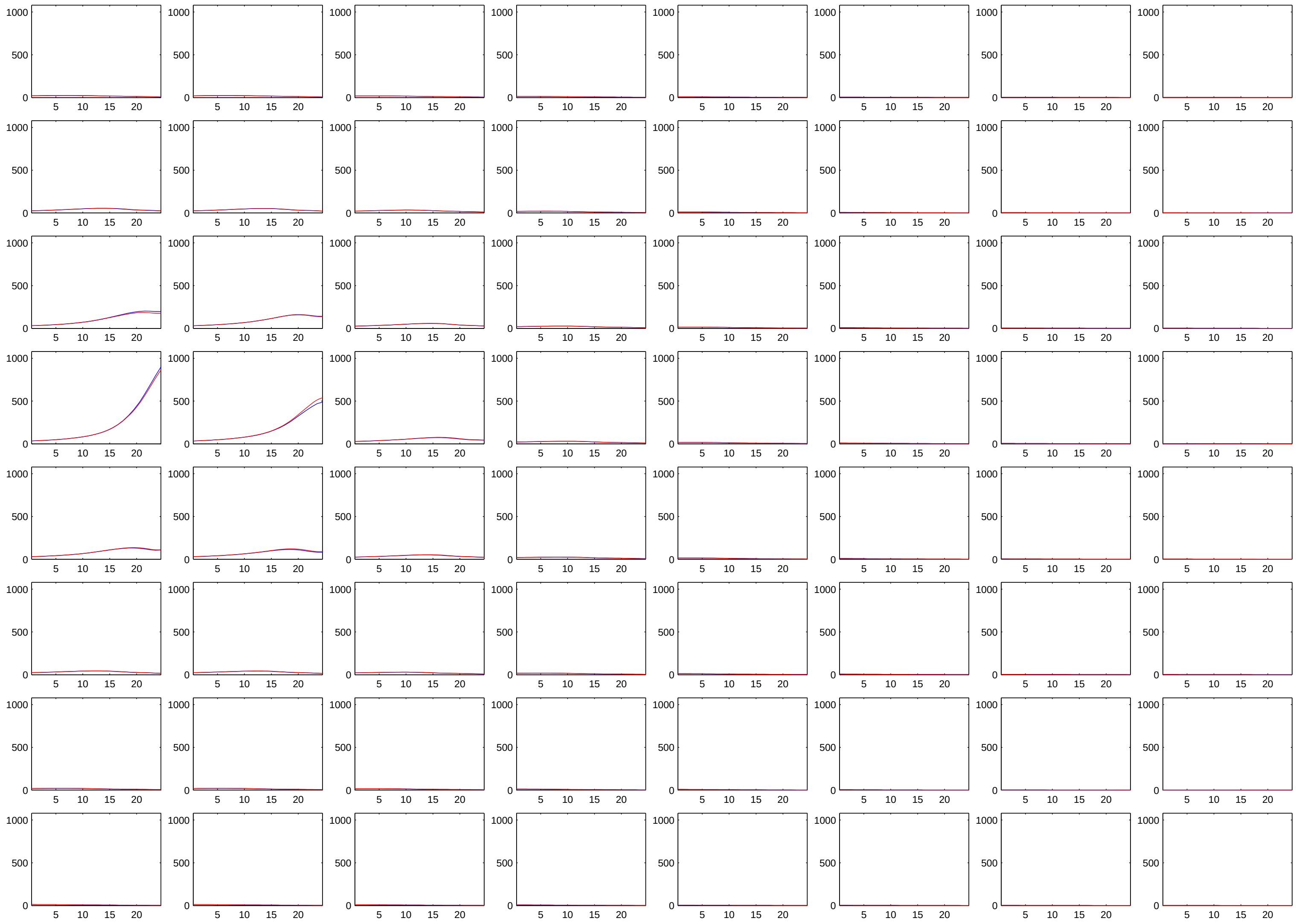


FIGURE 5.14. Plots as functions of Z of PMT response functions $\overline{\boldsymbol{g}}(X_0, Y_0, Z)$ for $X_0 = -18.949275$ mm, $Y_0 = -3.789855$ mm and $Z_0 \in [0.00, 25.00]$ mm (blue lines) along with their spline approximation (red lines)

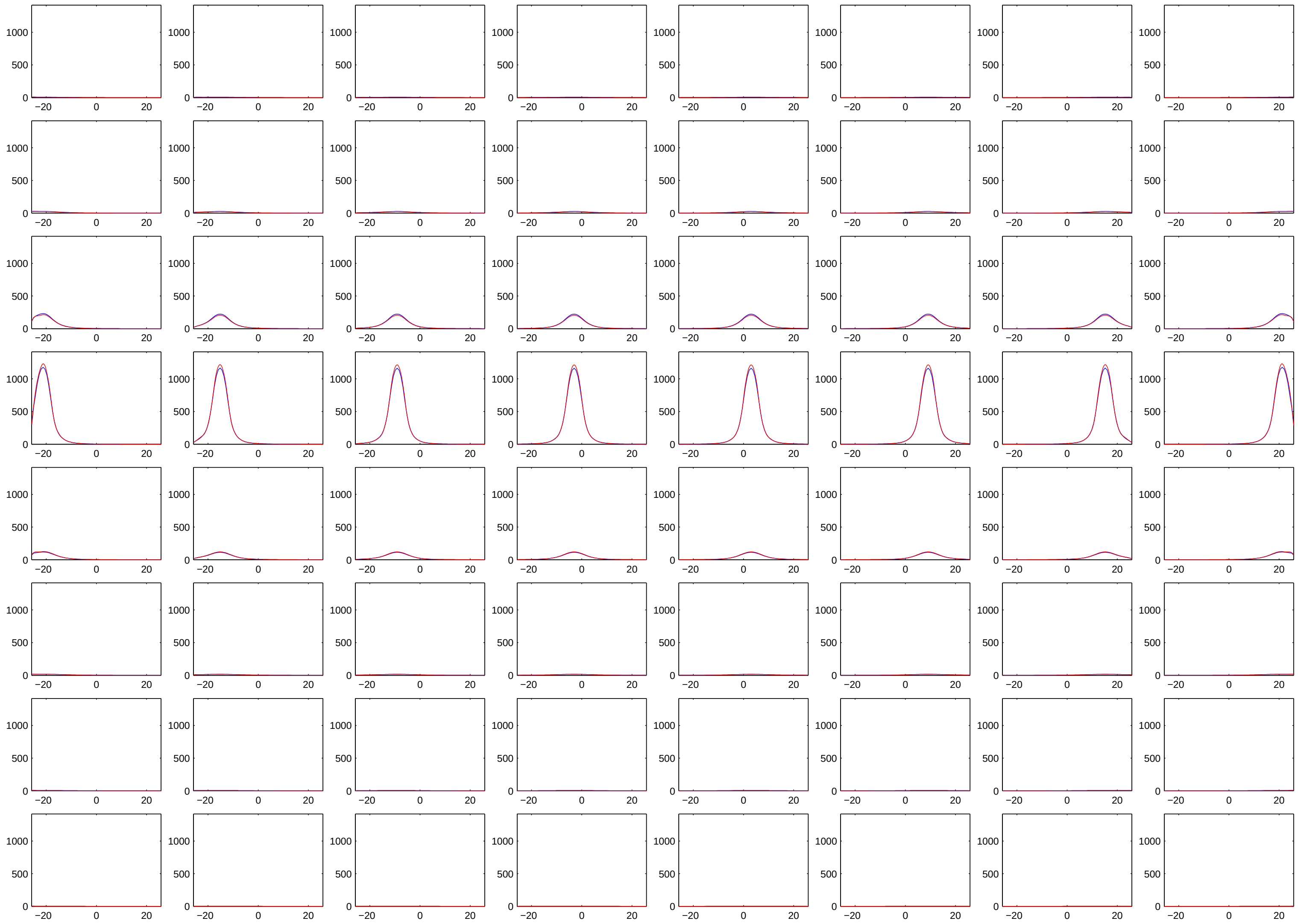


FIGURE 5.15. Plots as functions of X of PMT response functions $\overline{\boldsymbol{g}}(X, Y_0, Z_0)$ for $X \in [-26.15, 26.15]$ mm, $Y_0 = -3.789855$ mm and $Z_0 = 24.50$ mm (blue lines) along with their spline approximation (red lines)

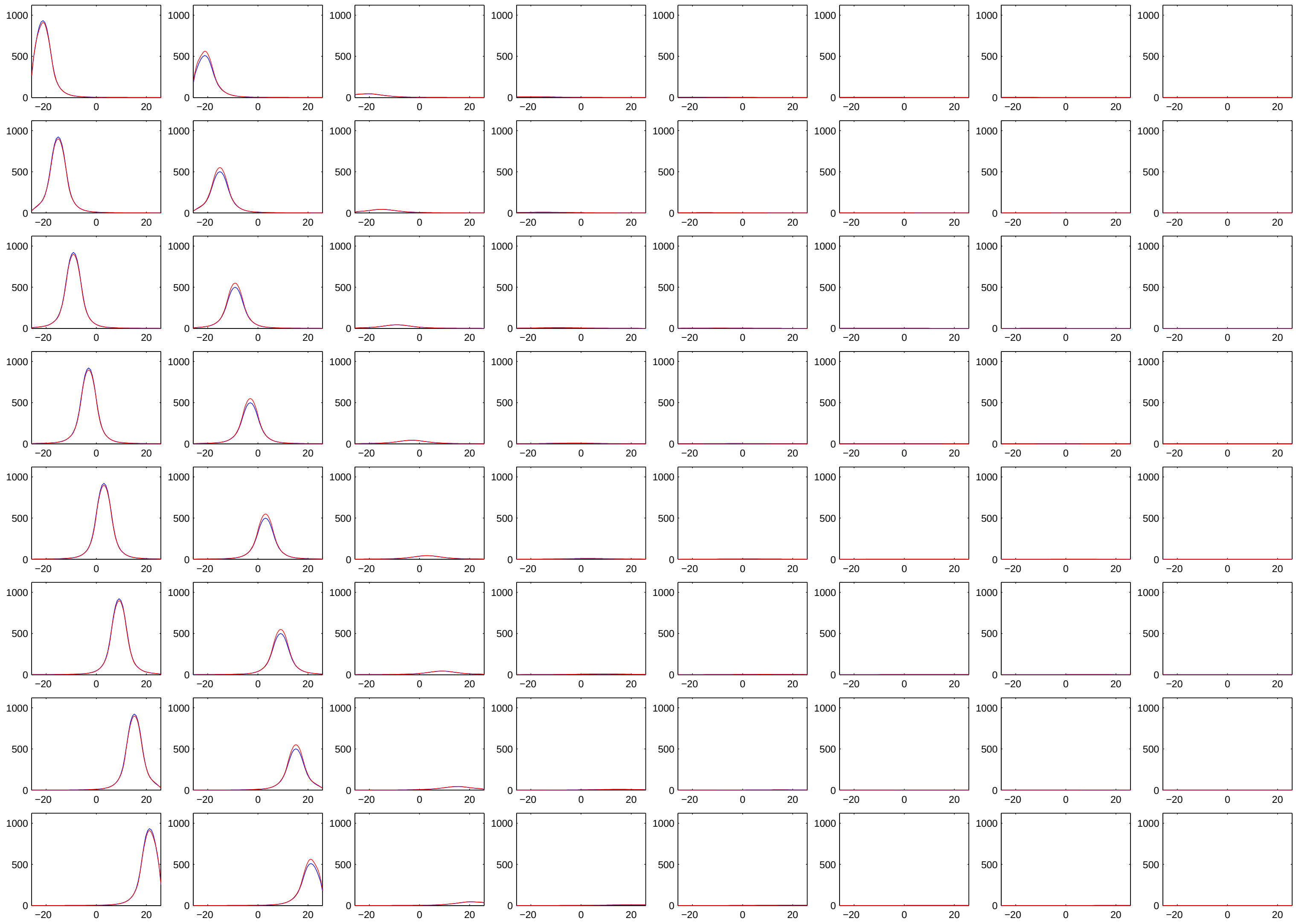


FIGURE 5.16. Plots as functions of Y of PMT response functions $\overline{\boldsymbol{g}}(X_0, Y, Z_0)$ for $X_0 = -18.949275$ mm, $Y \in [-26.15, 26.15]$ mm and $Z_0 = 24.50$ mm (blue lines) along with their spline approximation (red lines)

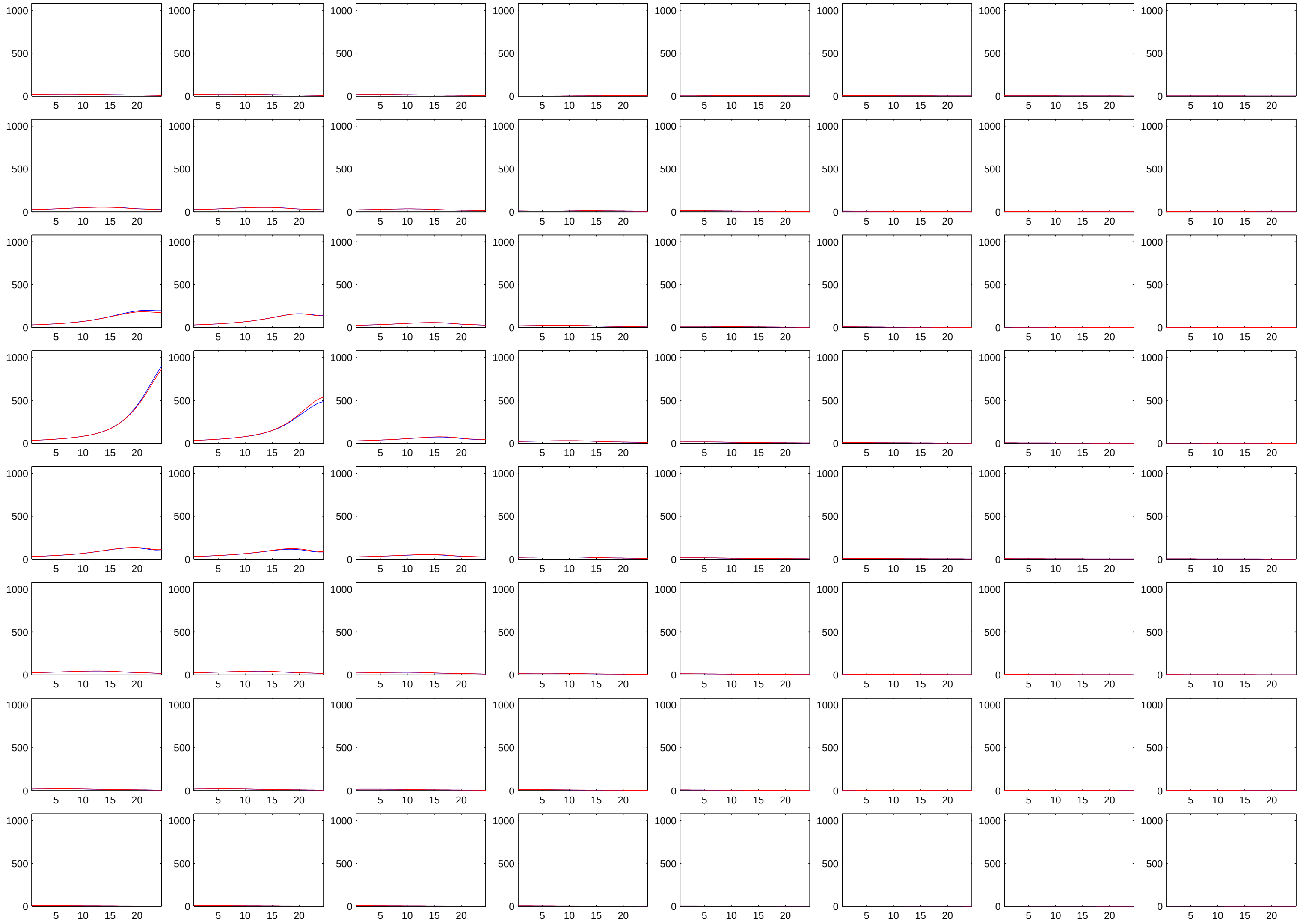


FIGURE 5.17. Plots as functions of Z of PMT response functions $\overline{\boldsymbol{g}}(X_0, Y_0, Z)$ for $X_0 = -18.949275$ mm, $Y_0 = -3.789855$ mm and $Z_0 \in [0.00, 25.00]$ mm (blue lines) along with their spline approximation (red lines)

data yielded estimates whose accuracy was limited by noise in the data $\boldsymbol{g}$ only.

As a test, we considered the same 52.30 mm×52.30 mm×25.00 mm NaI(Tl) crystal considered before and we assumed an interaction at $\boldsymbol{R} = (26.30, 26.30, 1.50)$ mm in the crystal. In other words, the point $\boldsymbol{R}$ was 1.50 mm underneath the central point of the crystal's entrance face. We generated 10^7 realizations of noisy PMT outputs $\boldsymbol{g} = \{g_1, \ldots, g_K\}$ for $K = 64$ and, for each of them, we ran 15 iterations of the contracting-grid algorithm above (appropriately modified to handle the re-sampled calibration data). By binning the 10^7 estimates $\hat{\boldsymbol{R}}_{\text{MLE}}$, a discrete representation of $\text{pr}(\hat{\boldsymbol{R}}_{\text{MLE}} \mid \boldsymbol{R})$ as a function of $\hat{\boldsymbol{R}}_{\text{MLE}}$ could be calculated. To compare $\text{pr}(\hat{\boldsymbol{R}}_{\text{MLE}} \mid \boldsymbol{R})$ estimated from the simulation data with the theoretical model predicted in (5.4), we considered three sets of plots of probability density functions, as shown in Figure 5.18. In each set, only one component of $\hat{\boldsymbol{R}}_{\text{MLE}}$ at a time was being changed while the others were held fixed and near the true location. The agreement between the plots in each set is striking, and it confirms that the noise model summarized by (5.4) is appropriate for our case.

Plots of the Cramér-Rao lower bounds are reported in Figure 5.19 through Figure 5.21. More specifically, Figure 5.19, Figure 5.20, and Figure 5.21 report the lower limits on the standard deviations for, respectively, the X, Y, and Z components of any estimate $\hat{\boldsymbol{R}}$ of the true location of interaction $\boldsymbol{R}$. The Cramér-Rao lower bound results are reported for different values of the Z coordinate inside the crystal. As expected, as the depth of interaction increases, the uncertainty on the estimate of the location of interaction reduces.

5.7 LMMLEM Image Reconstruction on a GPU Device

In this section, we provide some details about the simulation code we developed to test list-mode reconstruction on a GPU device. Recall that the derivation we presented in § 4.9 was for the PET setup of Figure 5.22, which sandwiched the 3D

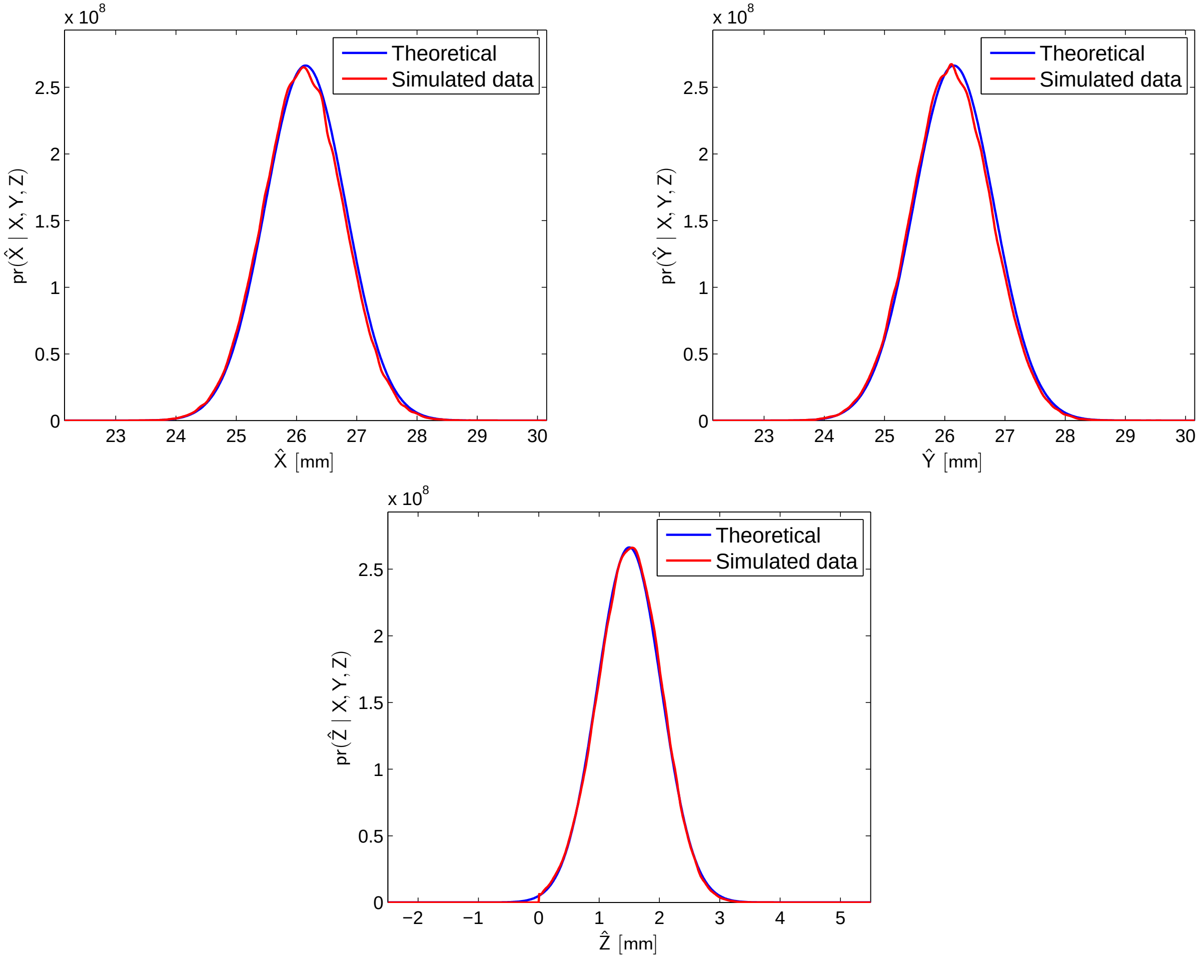


FIGURE 5.18. Comparison of the probability density functions of 3D maximum-likelihood estimates of location of interaction calculated through simulated data with the theoretical probability density functions calculated using the Fisher information matrix. For all plots, the true location of interaction was at $X = 26.30$ mm, $Y = 26.30$ mm, and $Z = 1.50$ mm

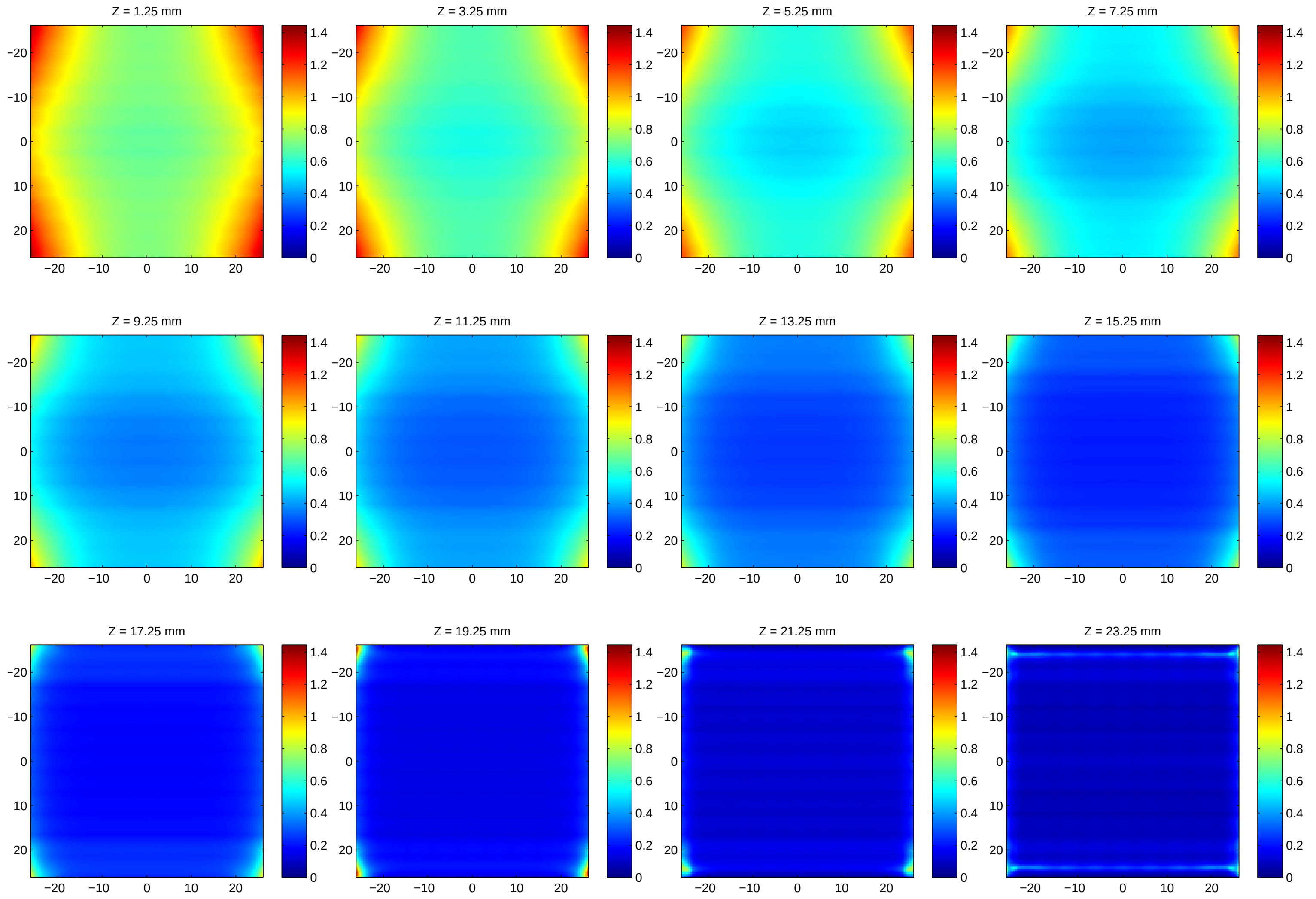
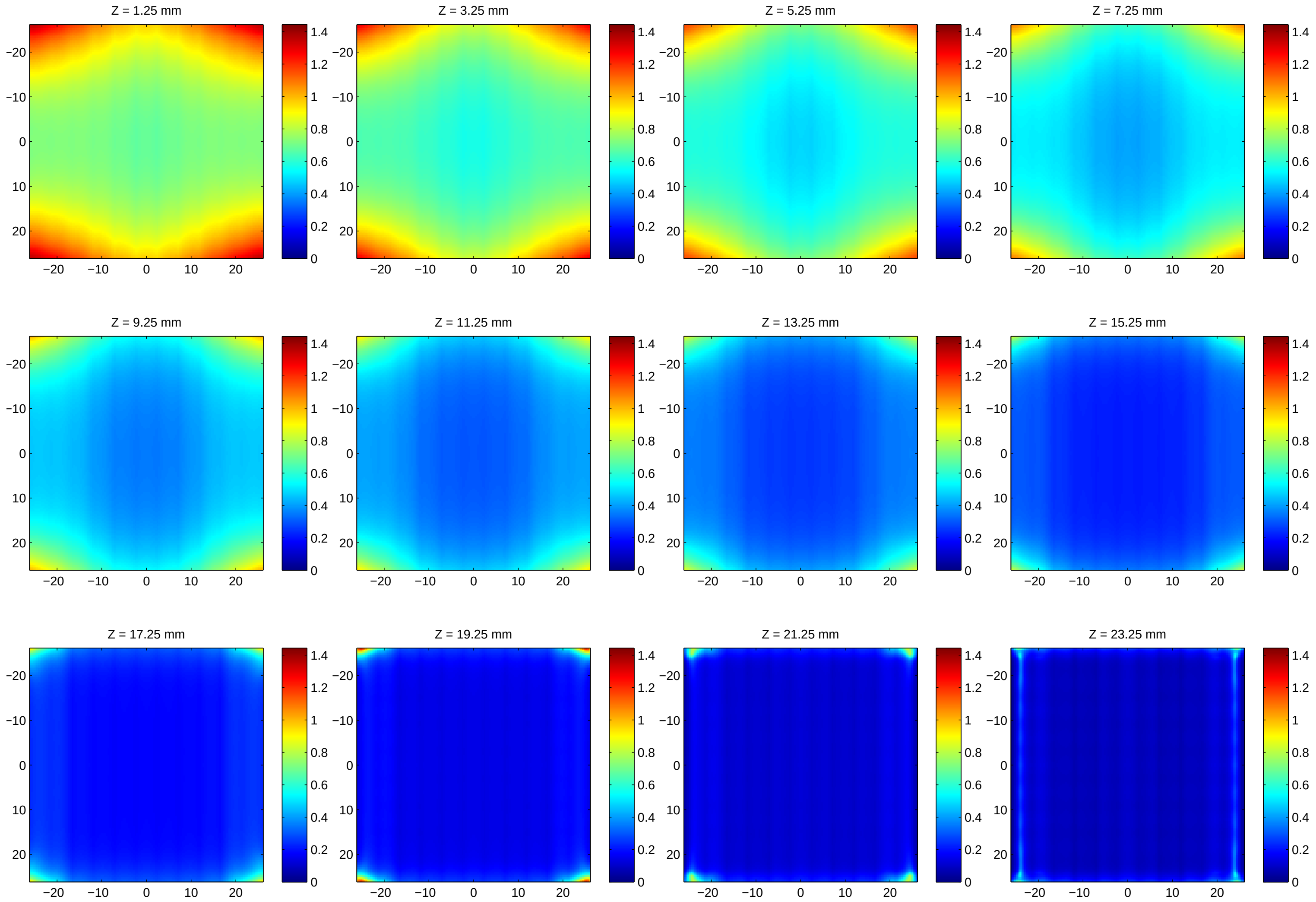


FIGURE 5.19. Plots for different values of depth of interaction of the lower limit on the standard deviation of the X component of any 3D estimate $\hat{\boldsymbol{R}}$ of the location of interaction


Z = 1.25 mm

Z = 3.25 mm

Z = 5.25 mm

Z = 7.25 mm

Z = 9.25 mm

Z = 11.25 mm

Z = 13.25 mm

Z = 15.25 mm

Z = 17.25 mm

Z = 19.25 mm

Z = 21.25 mm

Z = 23.25 mm

FIGURE 5.20. Plots for different values of depth of interaction of the lower limit on the standard deviation of the Y component of any 3D estimate $\hat{\boldsymbol{R}}$ of the location of interaction

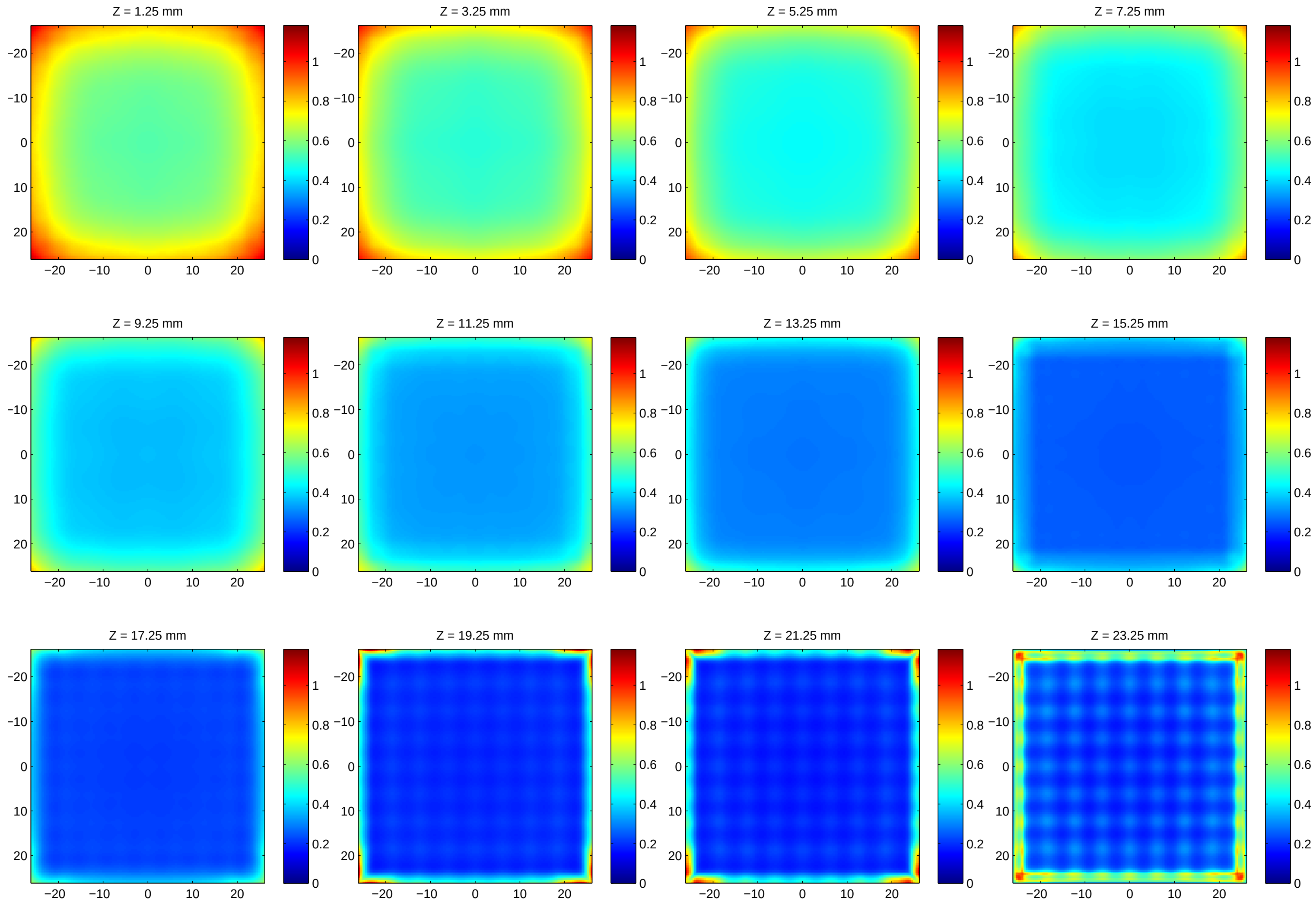


FIGURE 5.21. Plots for different values of depth of interaction of the lower limit on the standard deviation of the Z component of any 3D estimate $\hat{\boldsymbol{R}}$ of the location of interaction

radiotracer distribution $f(\boldsymbol{r})$ between two gamma-ray cameras facing each other. For our simulation, we assumed that the field of view (FOV) measured 20 mm $\times$ 20 mm $\times$ 20 mm and that it was subdivided into a 3D grid of $40 \times 40 \times 40$ square voxels. Each voxel thus measured 0.50 mm along each side.

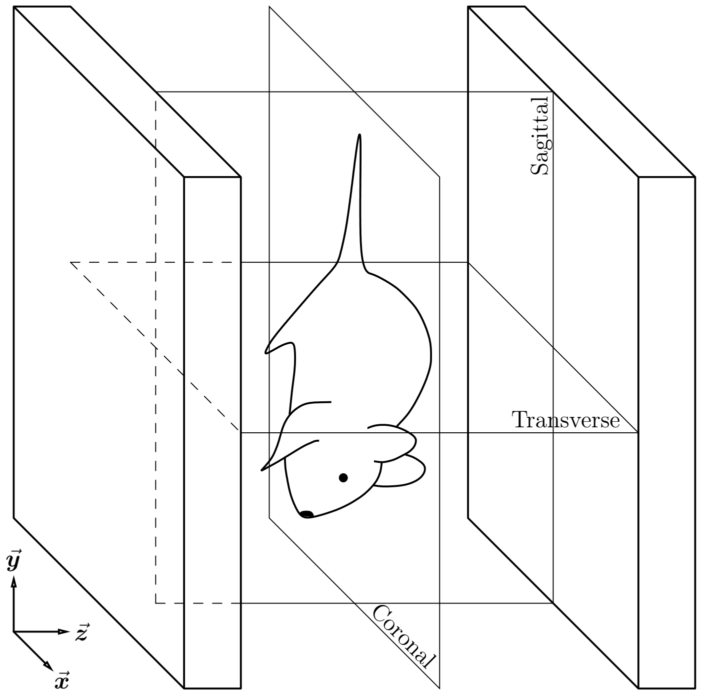


FIGURE 5.22. Setup of the simulated 3D PET system

One of the ingredients that enters in the expression of the LMMLEM algorithm in (4.10) is the vector $\boldsymbol{s}$ of sensitivities $s_1, \ldots, s_N$, where N denotes the total number of voxels in the FOV (for our case, $N = 40^3 = 64000$). We recall that each number s_n is the probability that a pair of gamma-ray photons emitted from the n^{th} voxel gets detected by both detectors and gives rise to an attribute vector $\hat{\boldsymbol{A}}^{(j)}$ of the list $\hat{\mathscr{A}}$. Voxel sensitivities, which are solely determined by the system's geometry and the linear absorption coefficient of the camera's crystal, could, in principle, be calculated analytically. In practice, however, this might require the calculation of multidimensional integrals over complicated integration domains. For this reason, we decided to calculate each of the s_n by simulating a radiotracer distribution inside the

n^{th} voxel, isotropically emitting pairs of gamma-ray photons. The probability that both photons are detected by the corresponding detectors is then averaged over all possible propagation directions

$$s_n \approx \frac{2\pi}{N_\vartheta N_\varphi} \sum_{p=0}^{N_\vartheta-1} \sum_{q=0}^{N_\varphi-1} \Pr\left(\ell \leqslant \ell_1\left(\boldsymbol{r}_n, \arccos\left(\tfrac{p+1/2}{N_\vartheta}\right), 2\pi\left(\tfrac{q+1/2}{N_\varphi}\right)\right)\right) \times \\ \times \Pr\left(\ell \leqslant \ell_2\left(\boldsymbol{r}_n, \arccos\left(\tfrac{p+1/2}{N_\vartheta}\right), 2\pi\left(\tfrac{q+1/2}{N_\varphi}\right)\right)\right),$$

in which $\ell_i(\boldsymbol{r}, \vartheta, \varphi)$ denotes the length of the segment defined by the intersection between detector D_i and the line passing through the point $\boldsymbol{r}$ and parallel to the versor $\vec{\boldsymbol{p}} = \big(\sin(\vartheta)\cos(\varphi), \sin(\vartheta)\sin(\varphi), \cos(\vartheta)\big)$. In our code, we set $N_\vartheta = 4096$ and $N_\varphi = 1024$. It is important to notice that by setting $\vartheta = \arccos\left(\frac{p+1/2}{N_\vartheta}\right)$ in the expression above we take care of the differential element that appears in the calculation of s_n over a solid angle

$$s_n = \int_{2\pi} \Pr\big(\ell \leqslant \ell_1(\boldsymbol{r}_n, \vartheta, \varphi)\big) \Pr\big(\ell \leqslant \ell_2(\boldsymbol{r}_n, \vartheta, \varphi)\big)\, \mathrm{d}\Omega,$$

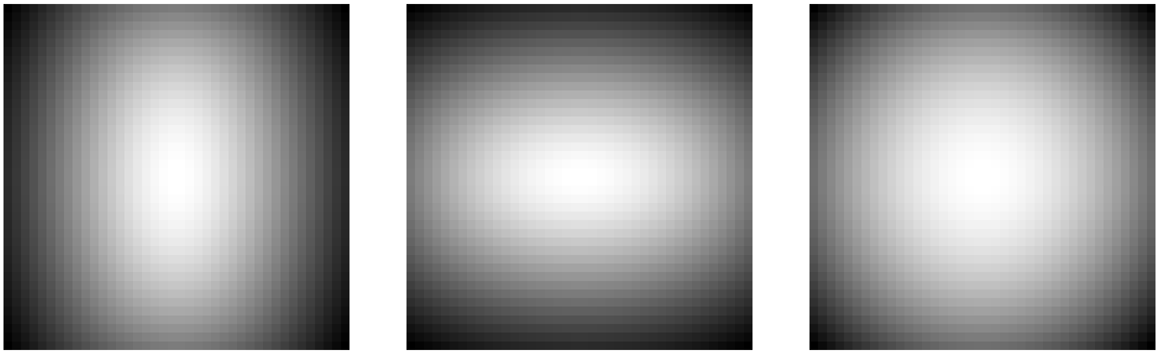
in which $\Pr\big(\ell \leqslant \ell_i(\boldsymbol{r}_n, \vartheta, \varphi)\big)$ denotes the probability of a photoelectric interaction before the gamma-ray proton has traveled the distance $\ell_i(\boldsymbol{r}_n, \vartheta, \varphi)$ within the crystal

$$\begin{aligned} \Pr\big(\ell \leqslant \ell_i(\boldsymbol{r}_n, \vartheta, \varphi)\big) &= \frac{\mu_{\text{pe}}}{\mu_{\text{tot}}} \int_0^{\ell_i(\boldsymbol{r}_n, \vartheta, \varphi)} \operatorname{pr}(\ell)\, \mathrm{d}\ell = \\ &= \frac{\mu_{\text{pe}}}{\mu_{\text{tot}}} \left[1 - e^{-\mu_{\text{tot}} \ell_i(\boldsymbol{r}_n, \vartheta, \varphi)}\right], \end{aligned}$$

where

$$\operatorname{pr}(\ell) = \begin{cases} \mu_{\text{tot}} e^{-\mu_{\text{tot}} \ell} & \text{if } \ell \geqslant 0, \\ 0 & \text{otherwise,} \end{cases}$$

and we denoted with μ_{pe} the photoelectric attenuation coefficient of the crystal material and with μ_{tot} the sum of the photoelectric and scattering (μ_{sc}) attenuation coefficients. For NaI(Tl) and photon energy 511 keV, values of μ_{pe} and μ_{tot} are 5.8537 m^{-1} and 34.2228 m^{-1}, respectively [359]. Figure 5.23 shows planar images obtained by restricting the sensitivity function along three planes normal to one another and passing through the center of the field of view. Please refer to Figure 5.22 for a pictorial representation of the location of these planes. As expected, the sensitivity attains its maximum value at the center of the field of view and decreases as we get closer and closer to one of the corners of either detector's entrance face. We also want to remark that, because sensitivities $s_1, \ldots, s_N$ do not depend on the object being imaged, they can be calculated once for all and saved to a file for later use. This, however, will not be true if we allow our system to be adaptive [7, 205], but we can turn to GPU computing for the efficient recalculation of sensitivities $s_1, \ldots, s_N$ every time the system is reconfigured.



(a) Sagittal view

(b) Transverse view

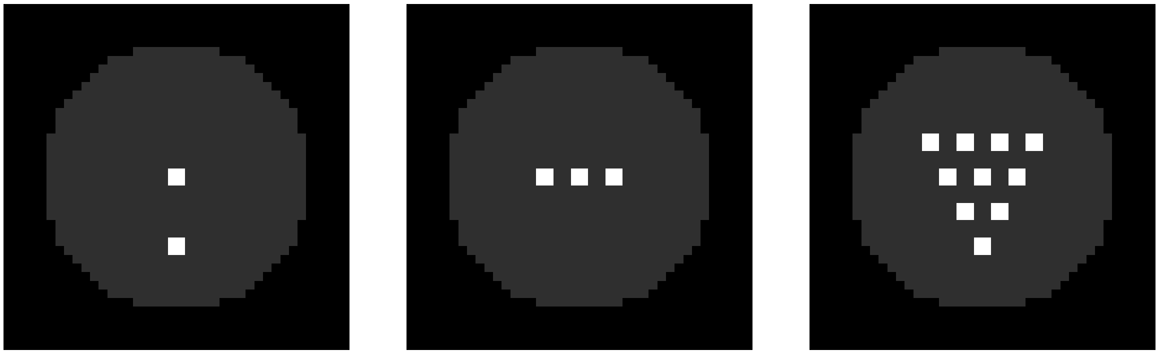
(c) Coronal view

FIGURE 5.23. Sensitivities $\boldsymbol{s}$ along three planes passing through the center of the field of view

Each gamma-ray camera used the same crystal we first presented in § 5.6, and we used the same PMT calibration data [355]. Recall that the PMT calibration data consist of a set of vectors of the form $\overline{\boldsymbol{g}}(\boldsymbol{R}_n) = \{\overline{g}_1(\boldsymbol{R}_n), \ldots, \overline{g}_K(\boldsymbol{R}_n)\}$, one for each point $\boldsymbol{R}_n$ of the $69 \times 69 \times 25$ grid into which the crystal's volume was divided. The index k denotes the PMT index, with $K = 64$ the total number of PMTs.

The original PMT data were first resampled on a finer grid. This step was carried out by first fitting the PMT calibration data with spline functions $\overline{g}_k(X, Y, Z)$ for $k = 1, \ldots, K$ and then by evaluating such function on a $138 \times 138 \times 50$ grid of points. Furthermore and in virtue of (5.5), the same spline functions also allow us to calculate the Fisher information matrix (FIM) at each point of such grid. In our reconstruction algorithm, both the resampled PMT calibration data and the FIMs were calculated once and store to binary files to be used in subsequent runs of the same code. Our implementation also takes advantage of the fact that each FIM is a 3×3 symmetric matrix; hence only six of its elements need to be stored.

For our simulation, we assumed that the real object $\boldsymbol{f}$ being imaged was a sphere of 15.50 mm of radius surrounding ten high-activity cubes of size $1\ \text{mm} \times 1\ \text{mm} \times 1\ \text{mm}$ and arranged as shown in Figure 5.24. The total activity from the ten cubes was equal to $25 \cdot 10^4$ collected counts per second and so was the total activity coming from the sphere surrounding the cubes, for a grand total of $50 \cdot 10^4$ collected counts per second. Each voxel in the field of view measured 500 µm × 500 µm × 500 µm.



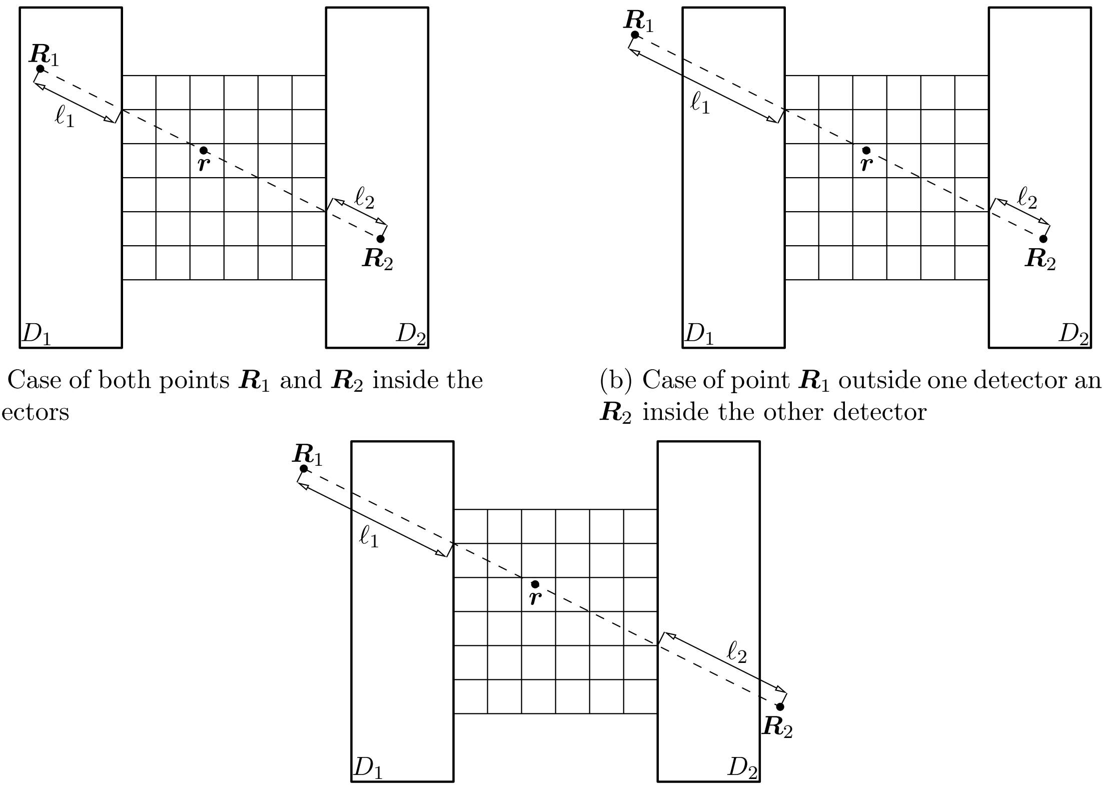
(a) Sagittal view

(b) Transverse view

(c) Coronal view

FIGURE 5.24. Log-scale images of the simulated object $\boldsymbol{f}$ along three planes passing through the center of the field of view

The generation of the noise-free list-mode data $\mathscr{A}$ was carried out as follows. A voxel in the field of view was randomly selected according to the probability model of (4.5) and a point $\boldsymbol{r}$ inside the selected voxel was chosen according to a uniform probability distribution over the voxel volume. Next, a propagation direction $\vec{\boldsymbol{p}}$ was isotropically and randomly generated, and two exponentially distributed random variables ℓ_1 and ℓ_2 were generated to account for how much each photon traveled inside the camera's crystal before getting absorbed. If both ℓ_1 and ℓ_2 took on values smaller that the length of the intersection between the line passing through $\boldsymbol{r}$ and propagating along the direction $\vec{\boldsymbol{p}}$, and the corresponding detector's crystals (see Figure 5.25a), then we assumed that the gamma-ray photons were absorbed by the cameras and that they produced measurable PMT outputs. The actual location of interactions $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ were calculated and paired to form the list-mode datum $\boldsymbol{A} = (\boldsymbol{R}_1, \boldsymbol{R}_2)$ to be added to the list $\mathscr{A}$. If either $\boldsymbol{R}_1$ or $\boldsymbol{R}_2$—as shown in Figure 5.25b—or both—see Figure 5.25c—fell outside the detector, the process just delineated started over with the selection of a new voxel in the field of view.



(a) Case of both points $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ inside the detectors

(b) Case of point $\boldsymbol{R}_1$ outside one detector and $\boldsymbol{R}_2$ inside the other detector

(c) Case of both points $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ outside the detectors

FIGURE 5.25. Generation of noise-free list-mode data $\mathscr{A}$

The generation of list-mode data $\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\}$ from noise-free list-mode data $\mathscr{A} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(J)}\}$ is depicted in Figure 5.26. In our notation, $\boldsymbol{A}^{(j)}$ is the pair $(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$ of the actual locations of interaction for the j^{th} line of response, while $\hat{\boldsymbol{A}}^{(j)}$ denotes the pair $(\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}, \hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2})$ of maximum-likelihood estimates of $(\boldsymbol{R}^{(j)}_1, \boldsymbol{R}^{(j)}_2)$, calculate from PMT data using the contracting-grid algorithm of § 5.6.

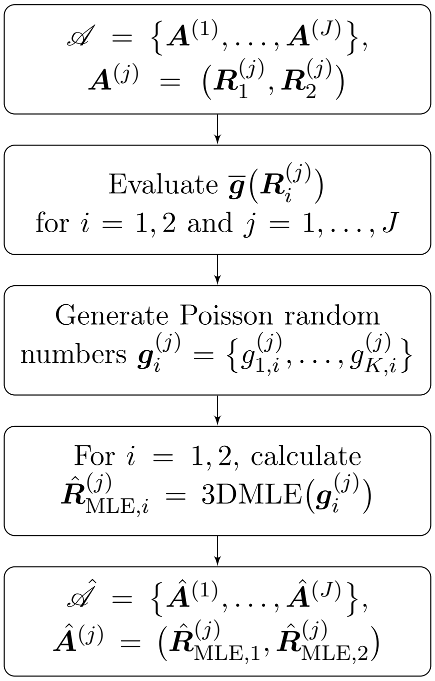

$\mathscr{A} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(J)}\},$
$\boldsymbol{A}^{(j)} = (\boldsymbol{R}^{(j)}_1, \boldsymbol{R}^{(j)}_2)$
Evaluate $\overline{\boldsymbol{g}}(\boldsymbol{R}^{(j)}_i)$
for $i = 1, 2$ and $j = 1, \ldots, J$
Generate Poisson random
numbers $\boldsymbol{g}^{(j)}_i = \{g^{(j)}_{1,i}, \ldots, g^{(j)}_{K,i}\}$
For $i = 1, 2$, calculate
$\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},i} = \text{3DMLE}(\boldsymbol{g}^{(j)}_i)$
$\hat{\mathscr{A}} = \{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\},$
$\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}, \hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2})$

FIGURE 5.26. Steps involved in the generation of $\hat{\mathscr{A}}$ from $\mathscr{A}$

We now discuss in more details the procedure summarized by Figure 5.26. First, thanks to the spline functions $\overline{g}_1(\boldsymbol{R}), \ldots, \overline{g}_K(\boldsymbol{R})$ we discussed above and in § 5.6, the MDRF at points $\boldsymbol{R}^{(j)}_1$ and $\boldsymbol{R}^{(j)}_2$ for $j = 1, \ldots, J$ was evaluated. By the procedure we delineated in the previous paragraph, points $\boldsymbol{R}^{(j)}_i$ are guaranteed to lie inside the corresponding detector volumes. It is important to remember that, under the Poisson noise model for the PMT outputs, $\overline{g}_k(\boldsymbol{R}^{(j)}_i)$ gives the mean of the Poisson random variable g_k for the k^{th} PMT output and interaction location $\boldsymbol{R}^{(j)}_i$. Hence, for each interaction location $\boldsymbol{R}^{(j)}_i$, K random numbers were drawn from K Poisson random variables with means $\overline{g}_1(\boldsymbol{R}^{(j)}_i), \ldots, \overline{g}_K(\boldsymbol{R}^{(j)}_i)$, respectively. These random numbers were then used in the 3D ML position of interaction estimation algorithm of § 5.6 to obtain the estimate $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},i}$. The collection of the $\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}, \hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2})$, in which each $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},i}$ was calculated as we just discussed, makes up the noisy list-mode data

$\hat{\mathscr{A}}$. This whole process is graphically summarized in Figure 5.26.

We are now ready to apply the iterative expression (4.10) to the list-mode data $\hat{\mathscr{A}}$. To do so, we need to evaluate quantities of the form $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$, in which $\hat{\boldsymbol{A}}^{(j)}$ is one element of $\hat{\mathscr{A}}$ and n is a voxel index in the field of view. Recall that, at the end of § 4.9, we derived an approximate expression for $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$, which we report below for convenience

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n) \approx \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{64\pi^4}\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}}\big)}\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}}\big)} \times \\ \times \int_{D_1} \frac{e^{-\frac{1}{2}\beta}}{C}\big[\operatorname{erf}(C\ell_2-\alpha)-\operatorname{erf}(C\ell_1-\alpha)\big] \times \\ \times \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}-\boldsymbol{R}^{(j)}_1\big)^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}}\big(\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}-\boldsymbol{R}^{(j)}_1\big)\right] \mathrm{d}^3\boldsymbol{R}^{(j)}_1. \quad (5.6)\end{aligned}$$

It is important to remember that, even though our notation does not show it, quantities ℓ_1, ℓ_2, C, α, and β also depend on the integration variable $\boldsymbol{R}^{(j)}_1$. Indeed,

$$\alpha = \frac{B}{C}, \qquad \beta = A^2 - \frac{B^2}{C^2},$$

where

$$A^2 - 2B\ell + C^2\ell^2 = (\alpha - C\ell)^2 + \beta,$$

$$A^2 = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}}\boldsymbol{a}, \qquad B = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}}\boldsymbol{b}, \qquad C^2 = \boldsymbol{b}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}}\boldsymbol{b},$$

and

$$\boldsymbol{a} = \hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2} - \boldsymbol{R}^{(j)}_1, \qquad \boldsymbol{b} = \boldsymbol{r}_n - \boldsymbol{R}^{(j)}_1.$$

To speed up the reconstruction procedure in (4.10), we can start by noticing that for voxels that are a short distance (say, 2 mm) away from the line passing through points $\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}$ and $\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}$, the value of $\mathrm{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ is very small. One way to see this, for example, is by analyzing the integral over ℓ that appears in (4.76); if voxel n (and, hence, its center $\boldsymbol{r}_n$) is far away from the straight line connecting $\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},1}$ with $\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}$, then for no value of ℓ the point $\boldsymbol{R}^{(j)}_2 = \boldsymbol{R}^{(j)}_1 + (\boldsymbol{r}_n - \boldsymbol{R}^{(j)}_1)\ell$ will be close to $\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2}$ enough so that $\mathrm{pr}(\hat{\boldsymbol{R}}^{(j)}_{\mathrm{MLE},2} \mid \boldsymbol{R}^{(j)}_2) \not\approx 0$. For this reason, our implementation of the reconstruction algorithm finds, for each $\hat{\boldsymbol{A}}^{(j)} = (\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}, \hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2})$ in $\hat{\mathscr{A}}$, the voxel indices n that are "near" the straight line defined by $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}$ and $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}$. Such values of n form a list, which we will denote as $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$. The quantity $\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ is then evaluated only for values of n in $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$.

Our CUDA code for the calculation of $\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ is based on the approximate expression in (5.6). More specifically, a 2D block grid is defined, and each thread block in the grid is tasked with calculating $\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ for one element $\hat{\boldsymbol{A}}^{(j)}$ of $\hat{\mathscr{A}}$ and all the values of n in $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$. The reason why we used a 2D block grid as opposed to just a 1D block grid is that, as we discussed in § 5.3, the GPU hardware limits the size of the grid in each dimension to 65535, whereas the number J of elements in $\hat{\mathscr{A}}$ can easily exceed such value. By using a 2D block grid, the $J = 65535$ limit of 1D block grid can easily be overcome. The volume integration that appears in (5.6) is well suited for a GPU architecture. For this reason, we took our thread blocks to be 3D, and each thread in the block evaluated the integrand for a particular value of the integration variable $\boldsymbol{R}^{(j)}_1$. Once again, we took advantage of the fact that the exponential factor that appears in the integrand makes the integrand itself negligible a few millimeters away from the point $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}$. In other words, the integral over the whole detector space D_1 is replaced with the sum of a few hundred terms obtained by varying $\boldsymbol{R}^{(j)}_1$ near $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}$. In our implementation, we used thread blocks of size $7 \times 7 \times 7$ which gives $7^3 = 343$ threads per block, below the limit we mentioned in § 5.3. Once all the threads in the block have completed calculating the value of the integrand, their sum is calculated and stored to global memory. This process is then repeated for the next value of n in $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$ and so on, until the whole list $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$ has been processed.

Besides taking advantage of the parallel processing capabilities of today's GPU devices, we further increased the computation speed by judicious usage of the different memory spaces that GPU devices feature. For example, the elements of the

Fisher information matrices $\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}}$ and $\mathbf{F}_{\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}}$ depend only on $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},1}$ and $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},2}$, respectively. Hence they can be loaded to shared memory and used repeatedly during the evaluation of $\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ for all the values of n in $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$. Similarly, values of the integrand in (5.6) were saved to shared memory to speed up the computation and let one of the threads sum all the values up.

The final step of the reconstruction algorithm involves one or more iterations of (4.10), which, using sets $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$, takes the form

$$\hat{f}_n^{(k+1)} = \hat{f}_n^{(k)} \left\{ \frac{1}{\tau} \sum_{j=1}^{J} \frac{\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\displaystyle\sum_{n' \in \mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}} \text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(k)}} \right\}, \tag{5.7}$$

in which τ denotes the total exposure time and the vector $\hat{\boldsymbol{f}}^{(k)}$ is the estimate of $\boldsymbol{f}$ at the k^{th} iteration of the reconstruction algorithm. For our example, we assumed $\tau = 2$ s and that the list $\hat{\mathscr{A}}$ contained $J = 10^5$ elements. The choice of $\hat{\boldsymbol{f}}^{(0)}$ is arbitrary (as long as $\hat{f}_n^{(0)} > 0$ for $n = 1, \ldots, N$); the algorithm will ensure convergence to a maximum-likelihood $\hat{\boldsymbol{f}}_{\text{MLE}}$ of $\text{pr}(\hat{\mathscr{A}} \mid \boldsymbol{f})$ no matter what $\hat{\boldsymbol{f}}^{(0)}$ is. Convergence speed might, however, depend on the choice of $\hat{\boldsymbol{f}}^{(0)}$. In our implementation we opted for $\hat{\boldsymbol{f}}^{(0)} = \boldsymbol{s}$, where $\boldsymbol{s}$ is the vector of the voxel sensitivities. Another possible choice is $\hat{f}_n^{(0)} = f_0$ for some constant positive value f_0 and $n = 1, \ldots, N$.

Our implementation of the reconstruction algorithm has been broken down into three main steps (or tasks). They are:

1. 3D maximum-likelihood estimation of $\hat{\boldsymbol{R}}^{(j)}_{\text{MLE},i}$ to get $\hat{\mathscr{A}}$;
2. evaluation of $\text{pr}(\hat{\boldsymbol{A}}^{(j)} \mid n)$ for $\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}$ and $n \in \mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$;
3. application of the iterative formula (5.7) above.

The first two steps are those that require substantial computational power, and they also parallelize well. For this reason, GPU code was developed to carry out those two tasks, as we discussed in § 5.6 and above in this section. For all the steps listed above, CPU code was developed as well. This has a few advantages. First of all, developing CPU code is easier and less prone to programming bugs than developing GPU code: once the CPU code has been developed, the numerical results can be used to test the correctness of the GPU implementation of the same numerical calculation. Second, by developing CPU code first, we can identify the portions of the code in which most of the computation time is spent, so that we can develop faster code only for those portions. Finally, the CPU code can guide and be used as starting point for the GPU code that performs the same task. For example, a `for(...)` loop in CPU code can be implemented on a GPU device by creating the appropriate thread hierarchy in which each thread implements the body of the `for(...)` with minor modifications.

Table 5.4 reports performance results for our CPU and GPU implementations of the LMMLEM reconstruction algorithm when run on the example of this section (see Figure 5.24). The expression in (5.7) was iterated ten times. In Table 5.4 we also indicated in parenthesis the GPU speedup with respect to the CPU implementation, and for a different number of GPU devices used. The speedups of Table 5.4 are graphically reported in Figure 5.27. It is interesting to note that the performance of the GPU 3D maximum-likelihood estimation of position of interaction actually deteriorates as more GPU devices are used. This can be explained using the modified Amdahl's law model of (5.1): in our example, the list $\hat{\mathscr{A}}$ contained a relatively small number of events and most of the time was spent to initialize the GPU devices and set up the GPU computation. These steps included establishing a communication channel with each GPU device, allocating the necessary memory buffers on the GPU devices, and transferring GPU code from the host memory to each GPU device. This overhead grows linearly with the number of GPU devices used. For the case of a relatively small number of events in the list $\hat{\mathscr{A}}$, the initialization overhead will constitute a substantial fraction of the total computation time. In other words, $0 < \alpha \ll \beta < 1$ in (5.1).

Finally, Figure 5.28 shows the results of the LMMLEM reconstruction algorithm

		Task	
Computing Platform	**3D MLE**	$\mathbf{pr}(\hat{\boldsymbol{A}}^{(j)} \mid \boldsymbol{n})$	**LMMLEM**
Intel® Xeon® L5506 2.13 GHz	104.39 s 1915.84 events/s	9347.17 s 10.70 events/s	2.17 s 0.22 s/iter
NVIDIA Tesla C2050, 1 device	1.11 s (94.06×) 180401.22 events/s	60.09 s (155.55×) 1664.06 events/s	*n/a*
NVIDIA Tesla C2050, 2 devices	1.16 s (89.99×) 171799.61 events/s	30.30 s (308.49×) 3311.45 events/s	*n/a*
NVIDIA Tesla C2050, 3 devices	1.47 s (71.01×) 135632.30 events/s	20.29 s (460.68×) 4929.06 events/s	*n/a*
NVIDIA Tesla C2050, 4 devices	1.80 s (57.99×) 111151.81 events/s	15.52 s (602.27×) 6442.47 events/s	*n/a*
NVIDIA Tesla C2050, 5 devices	2.86 s (36.50×) 69942.32 events/s	12.66 s (738.32×) 7897.76 events/s	*n/a*
NVIDIA Tesla C2050, 6 devices	3.95 s (26.43×) 50670.69 events/s	11.22 s (833.08×) 8911.21 events/s	*n/a*
NVIDIA Tesla C2050, 7 devices	4.78 s (21.84×) 41839.17 events/s	10.09 s (926.38×) 9911.99 events/s	*n/a*
NVIDIA Tesla C2050, 8 devices	6.01 s (17.37×) 33261.77 events/s	9.74 s (959.67×) 10264.06 events/s	*n/a*

TABLE 5.4. CPU and GPU running times and speedups for the LMMLEM reconstruction algorithm

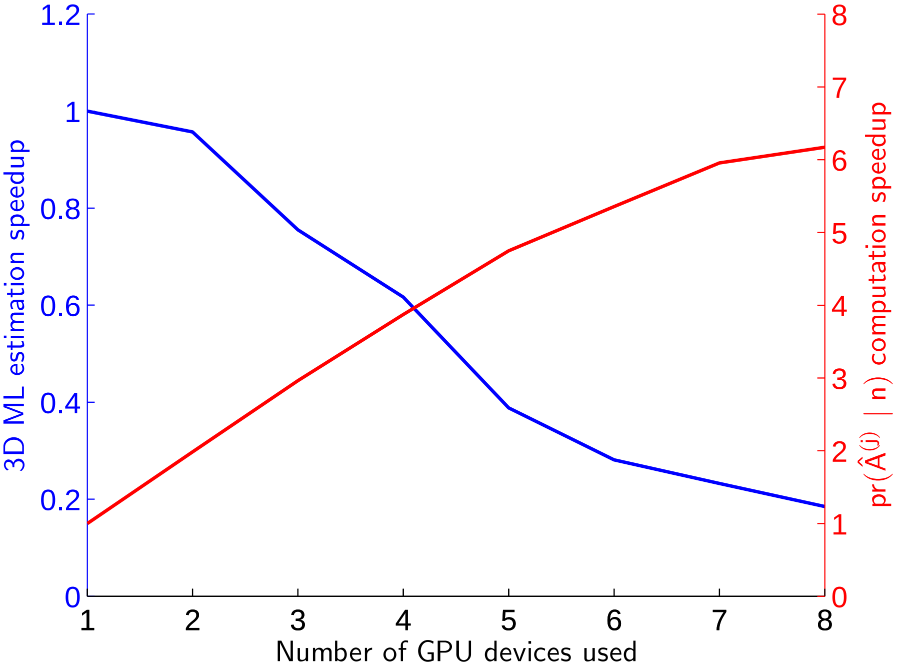
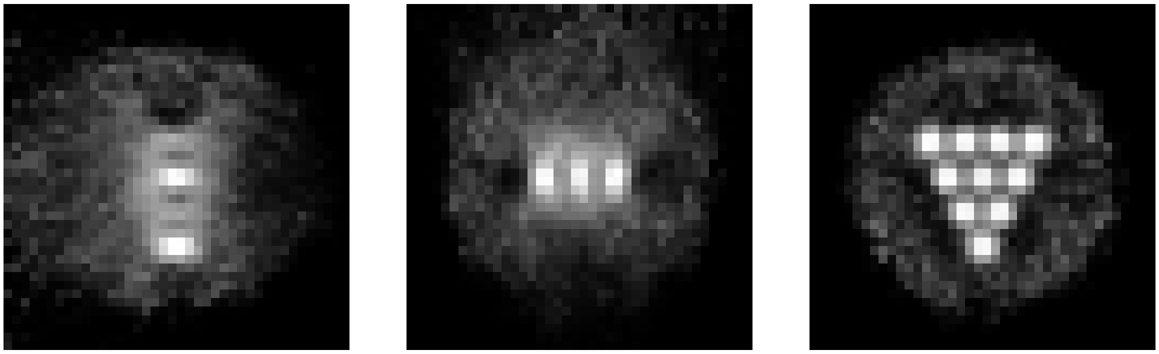


FIGURE 5.27. Speedups (with respect to conventional CPU code) obtained by running a portion of the LMMLEM reconstruction algorithm on a different number of GPU devices

$\hat{\boldsymbol{f}}^{(10)}$ after 10 iterations of (5.7). Comparison with Figure 5.24 shows the effectiveness of the LMMLEM algorithm in producing good estimates of the unknown radiotracer distribution $\boldsymbol{f}$.



(a) Sagittal view

(b) Transverse view

(c) Coronal view

FIGURE 5.28. Log-scale images of the reconstructed object $\hat{\boldsymbol{f}}^{(10)}$ along three planes passing through the center of the field of view

5.8 LMMLEM Reconstruction with ModPET

In this section, we apply the LMMLEM algorithm to an existing system, namely ModPET, a diagram of which is reported in Figure 5.29.

Each gamma-ray camera shown in the setup of Figure 5.29 is equipped with a thin (5 mm) NaI(Tl) crystal, and the camera has been experimentally calibrated [6] to perform 2D estimation of location of interaction of gamma-ray photons within the camera's crystal. In other words, no information about the depth of interaction will be available to the LMMLEM reconstruction algorithm.

Even though the actual depth of interaction Z cannot be estimated from PMT data, we can still characterize it statistically. We will denote with Δ the continuous random variable that indicates how much a gamma-ray photon has traveled within the detector before being absorbed and producing measurable PMT signals, and we will assume that Δ takes on values in $[0, \Delta_{\max}]$. Notice that $\Delta_{\max}$ depends on the

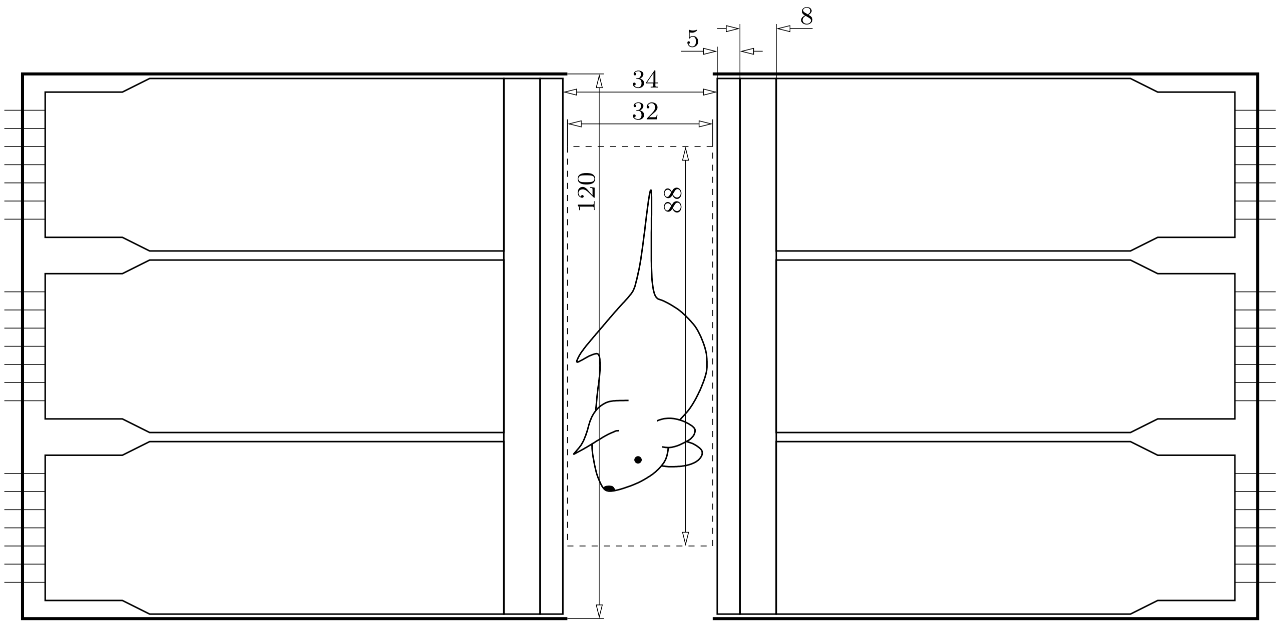


FIGURE 5.29. Schematic diagram of ModPET (dimensions in millimeters)

direction of propagation $\vec{\boldsymbol{p}}$ of the gamma-ray photon: Δ_{max} attains its minimum of $Z_{\text{max}} = 5$ mm (the crystal's thickness) at normal incidence and varies as $\Delta_{\text{max}} = Z_{\text{max}}/\cos\theta$, where θ is the angle between $\vec{\boldsymbol{p}}$ and the versor $\vec{\boldsymbol{n}}$ normal to the crystal's entrance face.

In this framework, we can consider the probability density function of Δ conditioned on the fact that the gamma-ray photon has been absorbed:

$$\text{pr}(\Delta \mid \Delta \leqslant \Delta_{\text{max}}) = \frac{\text{pr}(\Delta, \Delta \leqslant \Delta_{\text{max}})}{\text{Pr}(\Delta \leqslant \Delta_{\text{max}})} = \begin{cases} \dfrac{\text{pr}(\Delta)}{\text{Pr}(\Delta \leqslant \Delta_{\text{max}})} & \text{if } \Delta \leqslant \Delta_{\text{max}}, \\ \\ 0 & \text{otherwise,} \end{cases}$$

in which Δ (without conditioning) follows an exponential distribution

$$\text{pr}(\Delta) = \begin{cases} \mu_{\text{tot}} e^{-\mu_{\text{tot}}\Delta} & \text{if } \Delta \geqslant 0, \\ 0 & \text{otherwise.} \end{cases}$$

Integrating $\text{pr}(\Delta)$ over $[0, \Delta_{\text{max}}]$ gives

$$\text{Pr}(\Delta \leqslant \Delta_{\text{max}}) = 1 - e^{-\mu_{\text{tot}}\Delta_{\text{max}}},$$

so that

$$\text{pr}(\Delta \mid \Delta \leqslant \Delta_{\text{max}}) = \begin{cases} \dfrac{\mu_{\text{tot}} e^{-\mu_{\text{tot}}\Delta}}{1 - e^{-\mu_{\text{tot}}\Delta_{\text{max}}}} & \text{if } \Delta \in [0, \Delta_{\text{max}}], \\ \\ 0 & \text{otherwise.} \end{cases}$$

The mean of Δ (conditioned on the event $\Delta \leqslant \Delta_{\text{max}}$) is calculated as

$$\begin{aligned} \langle \Delta \rangle_{\Delta \mid \Delta \leqslant \Delta_{\text{max}}} &= \frac{\mu_{\text{tot}}}{1 - e^{-\mu_{\text{tot}}\Delta_{\text{max}}}} \int_0^{\Delta_{\text{max}}} \Delta e^{-\mu_{\text{tot}}\Delta} \, \text{d}\Delta = \\ &= \frac{1}{\mu_{\text{tot}}} \left[\frac{1 - e^{-\mu_{\text{tot}}\Delta_{\text{max}}} \big(\mu_{\text{tot}}\Delta_{\text{max}} + 1\big)}{1 - e^{-\mu_{\text{tot}}\Delta_{\text{max}}}} \right]. \end{aligned}$$

For NaI(Tl) and gamma-ray energy of 511 keV, $\mu_{\text{tot}} \approx 34.2228\ \text{m}^{-1}$ [359], which, for $\Delta_{\text{max}} = 5$ mm (normal incidence), gives $\langle \Delta \rangle_{\Delta \mid \Delta \leqslant \Delta_{\text{max}}} = 2.4287\ \text{mm} \approx \Delta_{\text{max}}/2$.

By a change of variables, we can also consider the probability density function of the depth of interaction Z for different values of the angle of incidence θ

$$\mathrm{pr}(Z \mid Z \leqslant Z_{\mathrm{max}}) = \frac{1}{\cos\theta} \begin{cases} \dfrac{\mu_{\mathrm{tot}} e^{-\mu_{\mathrm{tot}} Z / \cos\theta}}{1 - e^{-\mu_{\mathrm{tot}} Z_{\mathrm{max}} / \cos\theta}} & \text{if } \Delta \in [0, Z_{\mathrm{max}}], \\ 0 & \text{otherwise.} \end{cases}$$

Some plots of $\mathrm{pr}(Z \mid Z \leqslant Z_{\mathrm{max}})$ for a few values of θ are shown in Figure 5.30.

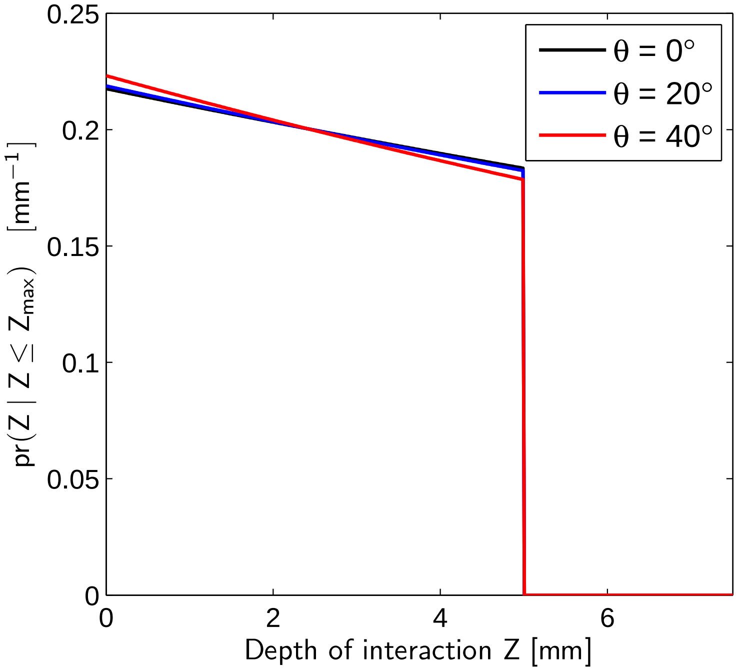


FIGURE 5.30. Plots of the probability density function $\mathrm{pr}(Z \mid Z \leqslant Z_{\mathrm{max}})$ for different values of the angle of incidence θ

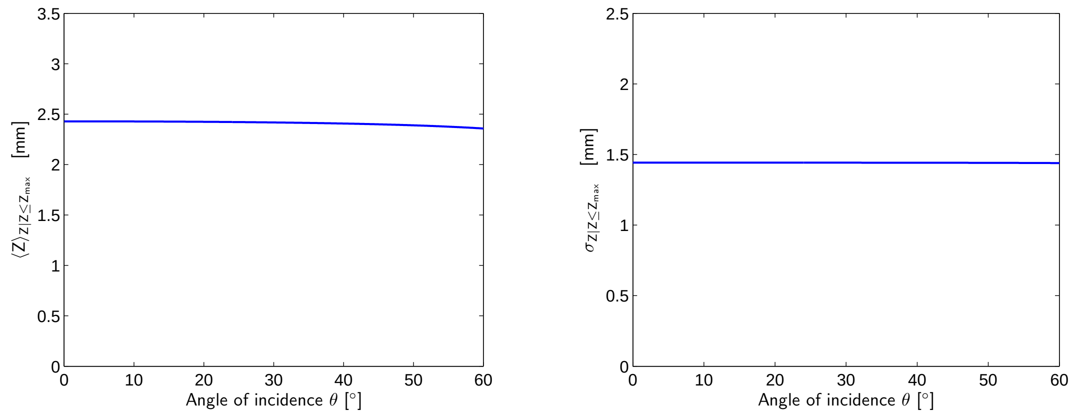
We can also use the expression of $\mathrm{pr}(Z \mid Z \leqslant Z_{\mathrm{max}})$ to calculate the mean and variance of Z:

$$\langle Z \rangle_{Z \mid Z \leqslant Z_{\mathrm{max}}} = \frac{\cos\theta}{\mu_{\mathrm{tot}}} \left[\frac{1 - e^{-\mu_{\mathrm{tot}} Z_{\mathrm{max}} / \cos\theta} (\mu_{\mathrm{tot}} Z_{\mathrm{max}} / \cos\theta + 1)}{1 - e^{-\mu_{\mathrm{tot}} Z_{\mathrm{max}} / \cos\theta}} \right],$$

and

$$\sigma^2_{Z|Z\leqslant Z_{\max}} = \langle Z^2\rangle_{Z|Z\leqslant Z_{\max}} - \left[\langle Z\rangle_{Z|Z\leqslant Z_{\max}}\right]^2 = \frac{\cos^2\theta}{\mu^2_{\text{tot}}\left(1 - e^{-\mu_{\text{tot}} Z_{\max}/\cos\theta}\right)^2} \times$$
$$\times \left[1 - e^{-\mu_{\text{tot}} Z_{\max}/\cos\theta}\left(\frac{\mu^2_{\text{tot}} Z^2_{\max}}{\cos^2\theta} + 2\right) + e^{-2\mu_{\text{tot}} Z_{\max}/\cos\theta}\right],$$

which are plotted in Figure 5.31a and Figure 5.31b, respectively.



(a) Mean

(b) Standard deviation

FIGURE 5.31. Plots of mean $\langle Z\rangle_{Z|Z\leqslant Z_{\max}}$ and standard deviation $\sigma_{Z|Z\leqslant Z_{\max}}$ as function of the angle of incidence θ

The results of Figure 5.31 show that, because of the relatively small attenuation coefficient of NaI(Tl), the mean depth of interaction for our $Z_{\max} = 5$ mm crystal is approximatively centered around the middle of the detector, with little variability as the angle of incidence θ is varied. This also implies relatively large values for the standard deviation of the depth of interaction, as shown in Figure 5.31b. Notice that the value of $\sigma_{Z|Z\leqslant Z_{\max}}$ plotted in Figure 5.31b is just below the standard deviation $\sigma_U = \sqrt{5^2/12}$ mm ≈ 1.44 mm for a uniform random variable U over $[0, Z_{\max}]$, where $Z_{\max} = 5$ mm. For this reason and even though ModPET features thin crystals, the integrals over the detector volumes that were used in § 4.9 cannot be just replaced with area integrals over the detector entrance faces: our treatment of ModPET must take into account the almost-uniformly-distributed depth of interaction Z.

Much of the theory we developed in § 4.9 still applies to the case of ModPET, with the notable different given by the fact that ModPET employs two thin ($Z_{\max} = 5$ mm) crystals of NaI(Tl) is each gamma-ray camera. For this reason, the theory we developed will be modified as needed. One important clarification on the notation is in order: as we did in § 4.9, we will use the vector $\boldsymbol{R}$ to denote a 3D point inside some crystal volume, and we will use the notation $\boldsymbol{R}'$ to denote the first two components of $\boldsymbol{R}$. In other words, if $\boldsymbol{R} = (X, Y, Z)$, then $\boldsymbol{R}' = (X, Y)$ and we can also write $\boldsymbol{R} = (\boldsymbol{R}', Z)$. This notation will often be used inside integrals, so, for example, given functions $f(\boldsymbol{R})$ and $g(\boldsymbol{R}')$, we can write $\int_D f(\boldsymbol{R}) g(\boldsymbol{R}') \,\mathrm{d}\boldsymbol{R}$, with the understanding that $g(\boldsymbol{R}')$ depends only on the first two components of the integration variable $\boldsymbol{R}$. With this in mind, we will use $\hat{\boldsymbol{A}}'^{(j)} = (\hat{\boldsymbol{R}}_1'^{(j)}, \hat{\boldsymbol{R}}_2'^{(j)})$ to denote the j^{th} element of the list mode data $\hat{\mathscr{A}}'$, in which $\hat{\boldsymbol{R}}_1'^{(j)}$ and $\hat{\boldsymbol{R}}_2'^{(j)}$ are 2D estimates of location of interaction within detector D_1 and detector D_2, respectively. In a similar fashion, $\boldsymbol{r}'$ will denote the first two components of the vector $\boldsymbol{r}$ in the field of view.

Our reconstruction algorithm is still mainly based on (4.10), which makes use of the probability density function $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ of $\hat{\boldsymbol{A}}'^{(j)} = (\hat{\boldsymbol{R}}_1'^{(j)}, \hat{\boldsymbol{R}}_2'^{(j)})$ conditioned on a pair of photons being emitted from the n^{th} voxel in the field of view. One way to calculate such probability density functions is by means of Monte Carlo sampling, as it has been discussed and done in [6]. In that work, $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ is calculated by drawing 3D samples $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ normally distributed about $(\hat{\boldsymbol{R}}_1'^{(j)}, Z_{\max}/2)$ and $(\hat{\boldsymbol{R}}_2^{(j)}, Z_{\max}/2)$, respectively. The line of response connecting samples $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ is then supplied to a line-projector algorithm [360] to calculated the length of the segment resulting from the intersection between the line of response itself and each voxel in the field of view. Repeated multiple times, this builds up a finite sum approximating $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ up to a constant factor. Some of the limitations of this Monte Carlo-based method for the calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ have already been noted in [6], one of them being that the detector blur is assumed constant over the whole detector space. Moreover, the Z components of $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ are sampled from an univariate and normally distributed random variable, in contrast with the (almost-uniform) truncated exponential distribution of Figure 5.30. For these reasons, we depart from [6] and calculate $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ in another way.

The starting point for our method for calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ is the exact expression in (4.74), which for our 2D estimates $\hat{\boldsymbol{R}}_1'^{(j)}$ and $\hat{\boldsymbol{R}}_2'^{(j)}$ has to be rewritten as

$$\begin{aligned}
&\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \\
&= \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}} \Delta_1(\boldsymbol{R}_1^{(j)}; \boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_2'^{(j)}) e^{-\mu_{\mathrm{tot}} \Delta_2(\boldsymbol{R}_2^{(j)}; \boldsymbol{r}_n)} \times \\
&\times \int_{-\infty}^{\infty} \psi_{D_2}\bigl(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\bigr) \delta_{\mathrm{Dir}}\bigl(\boldsymbol{R}_2^{(j)} - \boldsymbol{R}_1^{(j)} - (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\bigr) \, \mathrm{d}\ell \, \mathrm{d}^3\boldsymbol{R}_2^{(j)} \, \mathrm{d}^3\boldsymbol{R}_1^{(j)},
\end{aligned}$$

where we have used the superscript "(D_1)" in $\mathrm{pr}^{(D_1)}(\ldots)$ to mean that the outermost integral is over detector D_1. It is important to remark that in the expression above we have assumed depths of interaction $Z_1^{(j)}$ and $Z_2^{(j)}$ uniformly distributed over $[0, Z_{\max}]$, no matter what the angle of incidence θ is. This is why a ${}^1\!/\!_{Z_{\max}^2}$ factor appears in front of the integral over D_1. If we repeat the same process but we change the order of integration, we get

$$\begin{aligned}
&\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \\
&= \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_2'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}} \Delta_2(\boldsymbol{R}_2^{(j)}; \boldsymbol{r}_n)}}{|\boldsymbol{R}_2^{(j)} - \boldsymbol{r}_n|^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) e^{-\mu_{\mathrm{tot}} \Delta_1(\boldsymbol{R}_1^{(j)}; \boldsymbol{r}_n)} \times \\
&\times \int_{-\infty}^{\infty} \psi_{D_1}\bigl(\boldsymbol{R}_2^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_2^{(j)})\ell\bigr) \delta_{\mathrm{Dir}}\bigl(\boldsymbol{R}_1^{(j)} - \boldsymbol{R}_2^{(j)} - (\boldsymbol{r}_n - \boldsymbol{R}_2^{(j)})\ell\bigr) \, \mathrm{d}\ell \, \mathrm{d}^3\boldsymbol{R}_1^{(j)} \, \mathrm{d}^3\boldsymbol{R}_2^{(j)}.
\end{aligned}$$

The expressions for $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ and $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ above represent the same quantity, but calculated in two different ways. Because neither $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ nor $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ can be calculated analytically, we will introduce some approximations and we will evaluate some of the 3D integrals numerically. More specifically, our code for the evaluation of $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ performs numerically the integration over D_1, while in the evaluation of $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ it is the integration over D_2 that gets performed numerically. This introduces an asymmetry in our code, which we fix by taking

$$\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{1}{2}\Big[\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) + \mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)\Big] \tag{5.8}$$

as an approximate expression for $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$. As in § 4.9, the presence of the Dirac delta function $\delta_{\mathrm{Dir}}(\ldots)$ makes it trivial to perform one of the 3D integrals and get

$$\begin{aligned} &\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \\ &= \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \int_{-\infty}^{\infty} \psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) \times \\ &\qquad \times \mathrm{pr}\big(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_1'^{(j)} + (\boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)})\ell\big) e^{-\mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_1^{(j)}+(\boldsymbol{r}_n-\boldsymbol{R}_1^{(j)})\ell;\boldsymbol{r}_n)}\,\mathrm{d}\ell\,\mathrm{d}^3\boldsymbol{R}_1^{(j)}. \end{aligned}$$

With the help of Figure 4.29, we can find two numbers ℓ_1 and ℓ_2 such that $\psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) = 1$ for $\ell \in [\ell_1, \ell_2]$ and $\psi_{D_2}\big(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell\big) = 0$ otherwise. It is important to notice that the straight line passing through $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{r}_n$ might not intersect detector D_2: in such a case, numbers ℓ_1 and ℓ_2 do not exist and $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = 0$. If numbers ℓ_1 and ℓ_2 do exist, we have

$$\Delta_2(\boldsymbol{R}_1^{(j)} + (\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\ell; \boldsymbol{r}_n) = \big|(\ell - \ell_1)(\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)})\big| = (\ell - \ell_1)\big|\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}\big|,$$

provided that $\ell \in [\ell_1, \ell_2]$. With this observation, the expression for $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ has now become

$$\begin{aligned} \mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) &= \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \times \\ &\times \int_{\ell_1}^{\ell_2} \mathrm{pr}\big(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_1'^{(j)} + (\boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)})\ell\big) e^{-\mu_{\mathrm{tot}}(\ell-\ell_1)|\boldsymbol{r}_n-\boldsymbol{R}_1^{(j)}|}\,\mathrm{d}\ell\,\mathrm{d}^3\boldsymbol{R}_1^{(j)}. \end{aligned} \tag{5.9}$$

Figure 5.33 reports the Cramér-Rao lower bounds (CRLBs) for the estimation of the 2D location of interaction from PMT data. These CRLBs were calculated using (5.3) and (5.5), which we report below for convenience

$$[\mathbf{F}_{\boldsymbol{R}}]_{m,n} \approx \sum_{k=1}^{K} \frac{1}{\sigma^2 + r\overline{g}_k(\boldsymbol{R})} \frac{\partial \overline{g}_k(\boldsymbol{R})}{\partial R_m} \frac{\partial \overline{g}_k(\boldsymbol{R})}{\partial R_n}.$$

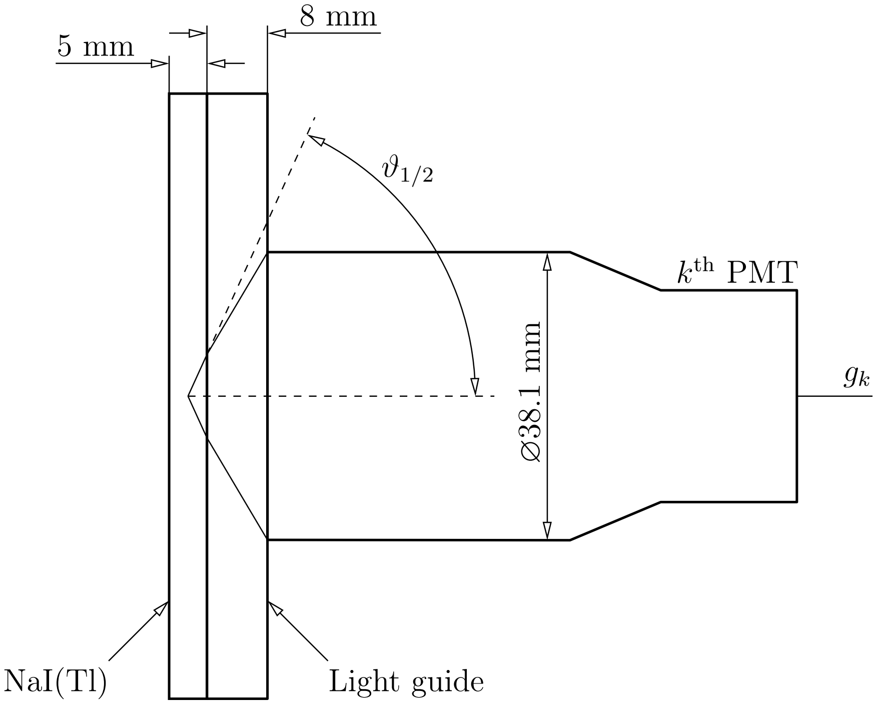


FIGURE 5.32. Diagram for the calculation of ModPET PMT gain

The value of σ^2 was set to 2 [358], while r was calculated with the help of Figure 5.32.

The angle $\vartheta_{1/2}$ measuring the spread of the bundle of optical photons that make it to the entrance face of the k^{th} PMT can be calculated as

$$\vartheta_{1/2} \approx \tan^{-1}\left[\frac{(38.1\ \mathrm{mm})/2}{(2.50 + 8.00)\ \mathrm{mm}}\right] = 1.0670\ \mathrm{rad}.$$

Hence, the solid angle Ω subtended by the same bundle of photos is

$$\Omega = 2\pi\bigl(1 - \cos\vartheta_{1/2}\bigr) = 3.2502\ \mathrm{ster}.$$

The PMT output g_k is related to r according to

$$\frac{2\Omega}{4\pi} r\mathcal{E}_\gamma \eta_{\mathrm{NaI(Tl)}} \eta = g_k,$$

where $\mathcal{E}_\gamma$ is the energy of a gamma-ray photon in units of MeV, $\eta_{\mathrm{NaI(Tl)}}$ is the number of optical photons per MeV for NaI(Tl), and η is the PMT's quantum efficiency.

Experimentally, $g_k \approx 1575$, so that:

$$r = \frac{g_k}{\dfrac{2\Omega}{4\pi}\mathcal{E}_\gamma \eta_{\mathrm{NaI(Tl)}} \eta} \approx 0.7902,$$

in which we have taken $\mathcal{E}_\gamma = 0.511$ MeV, $\eta_{\mathrm{NaI(Tl)}} = 37700$ photons/MeV [235], and we have assumed $\eta = 0.20$ [358].

Recall that the two cameras have been experimentally calibrated [6], so any manufacturing difference in the two cameras has to be taken into account by having a set of calibration data for each camera, as shown in Figure 5.34 and Figure 5.35. Plots in Figure 5.33 suggest that if we exclude pathological cases near the edges of the detector (as emphasized in Figure 5.33 by red squares superimposed on the CRLB maps), maximum-likelihood estimates $\hat{\boldsymbol{R}}'$ are very accurate, as the standard deviations of the components of $\hat{\boldsymbol{R}}'$ are typically less than 1.00 mm. Our calculations show that the average standard deviation for the $\hat{X}$ component of $\hat{\boldsymbol{R}}'$ is $\sigma_{\hat{X}} = 0.5350$ mm and that the average standard deviation for the $\hat{Y}$ component of $\hat{\boldsymbol{R}}'$ is $\sigma_{\hat{Y}} = 0.5365$ mm. These standard deviations can be related to the full width at half maximum (FWHM) of the distribution of the $\hat{X}$ and $\hat{Y}$ estimates using the relationship $\mathrm{FWHM} = 2\sqrt{2 \ln 2}\sigma \approx 2.35\sigma$, giving $\mathrm{FWHM}_{\hat{X}} = 1.2598$ mm and $\mathrm{FWHM}_{\hat{Y}} = 1.2633$ mm. These values are comparable to the detector's X and Y intrinsic spatial resolutions experimentally found in [6].

As in § 4.9, the density $\mathrm{pr}(\hat{\boldsymbol{R}}' \mid \boldsymbol{R}')$ is well approximated with a Gaussian:

$$\mathrm{pr}(\hat{\boldsymbol{R}}' \mid \boldsymbol{R}') \approx \frac{\sqrt{\det\left(\mathbf{F}_{\hat{\boldsymbol{R}}'}\right)}}{2\pi} \exp\left[-\frac{1}{2}\left(\hat{\boldsymbol{R}}' - \boldsymbol{R}'\right)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}'} \left(\hat{\boldsymbol{R}}' - \boldsymbol{R}'\right)\right], \tag{5.10}$$

in which $\mathbf{F}_{\hat{\boldsymbol{R}}'}$ is the 2×2 Fisher information matrix at $\hat{\boldsymbol{R}}'$. By examining the elements of $\mathbf{F}_{\hat{\boldsymbol{R}}'}$ (or $[\mathbf{F}_{\hat{\boldsymbol{R}}'}]^{-1}$) for different values of $\hat{\boldsymbol{R}}'$, we notice that there is little correlation between $\hat{X}$ and $\hat{Y}$ estimates. Indeed, if we assume $\mathbf{K}_{\hat{\boldsymbol{R}}'} \approx [\mathbf{F}_{\hat{\boldsymbol{R}}'}]^{-1}$, then Figure 5.36 suggests $-0.15\ \mathrm{mm}^2 \lesssim \sigma_{\hat{X}\hat{Y}} \lesssim 0.15\ \mathrm{mm}^2$ for most of the detector face. On the other hand, values of $\sigma^2_{\hat{X}}$ and $\sigma^2_{\hat{Y}}$ range from $0.30\ \mathrm{mm}^2$ to $1.20\ \mathrm{mm}^2$ (see Figure 5.33).

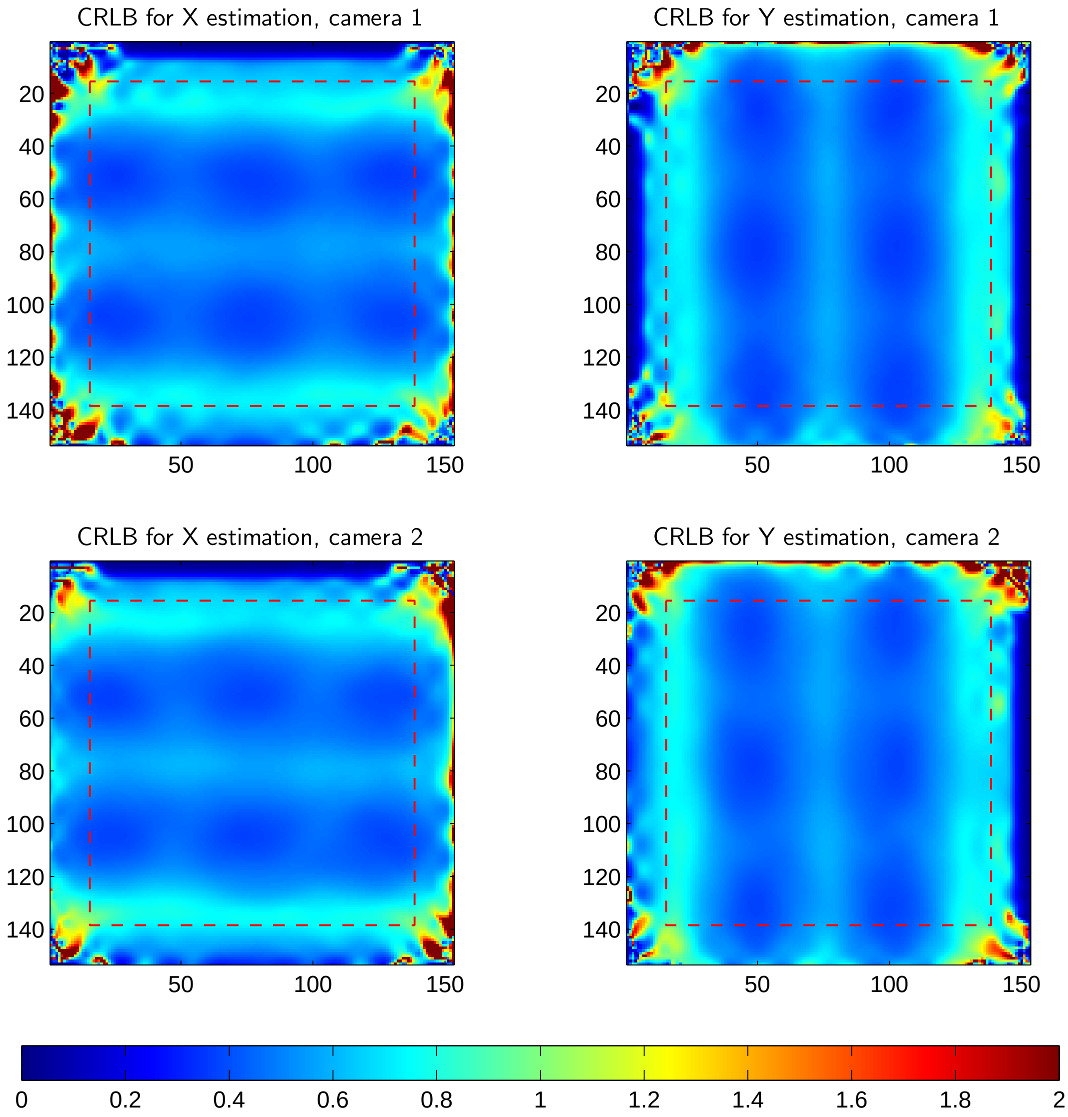
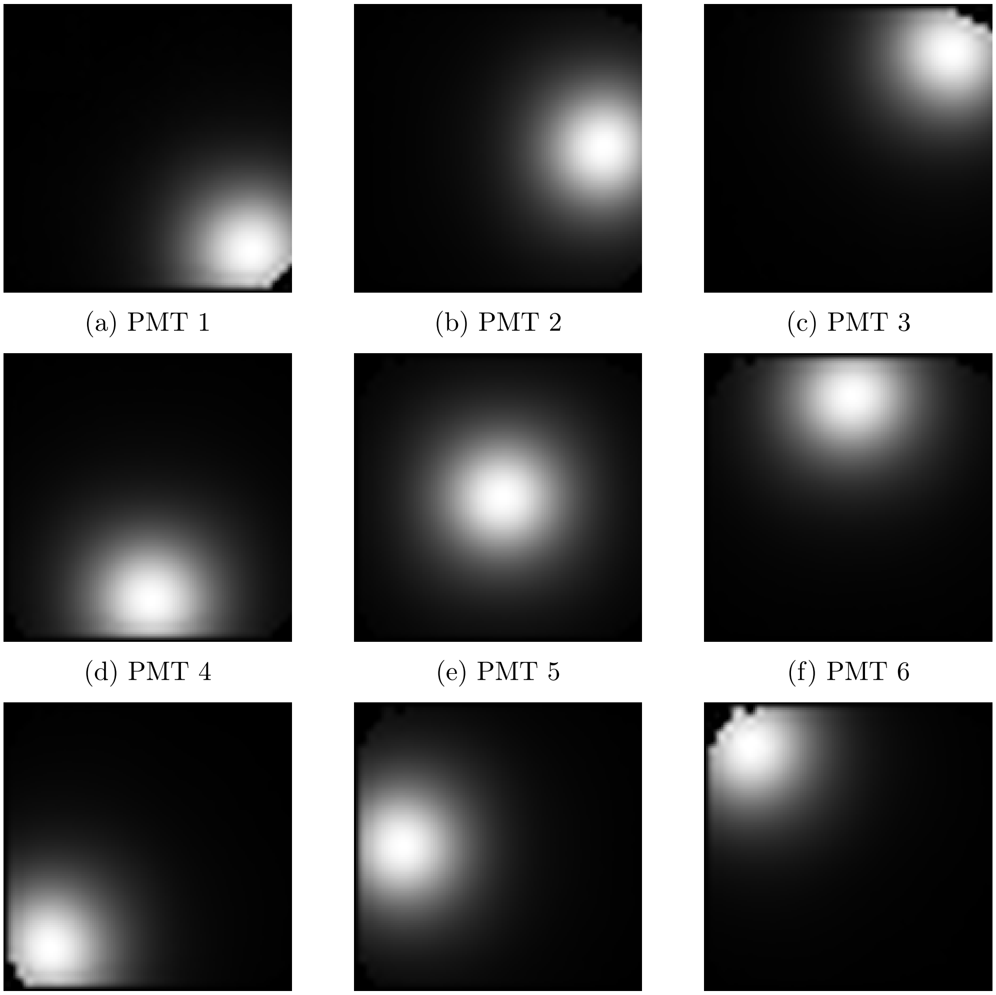


FIGURE 5.33. Plots of Cramér-Rao lower bounds (in millimeters) for the 2D estimation of location of interaction for ModPET gamma-ray cameras. The regions outside the red squares indicate detector pixels for which estimates $\hat{\boldsymbol{R}}'$ are inaccurate and/or their calculation and processing is numerically unstable



(a) PMT 1

(b) PMT 2

(c) PMT 3

(d) PMT 4

(e) PMT 5

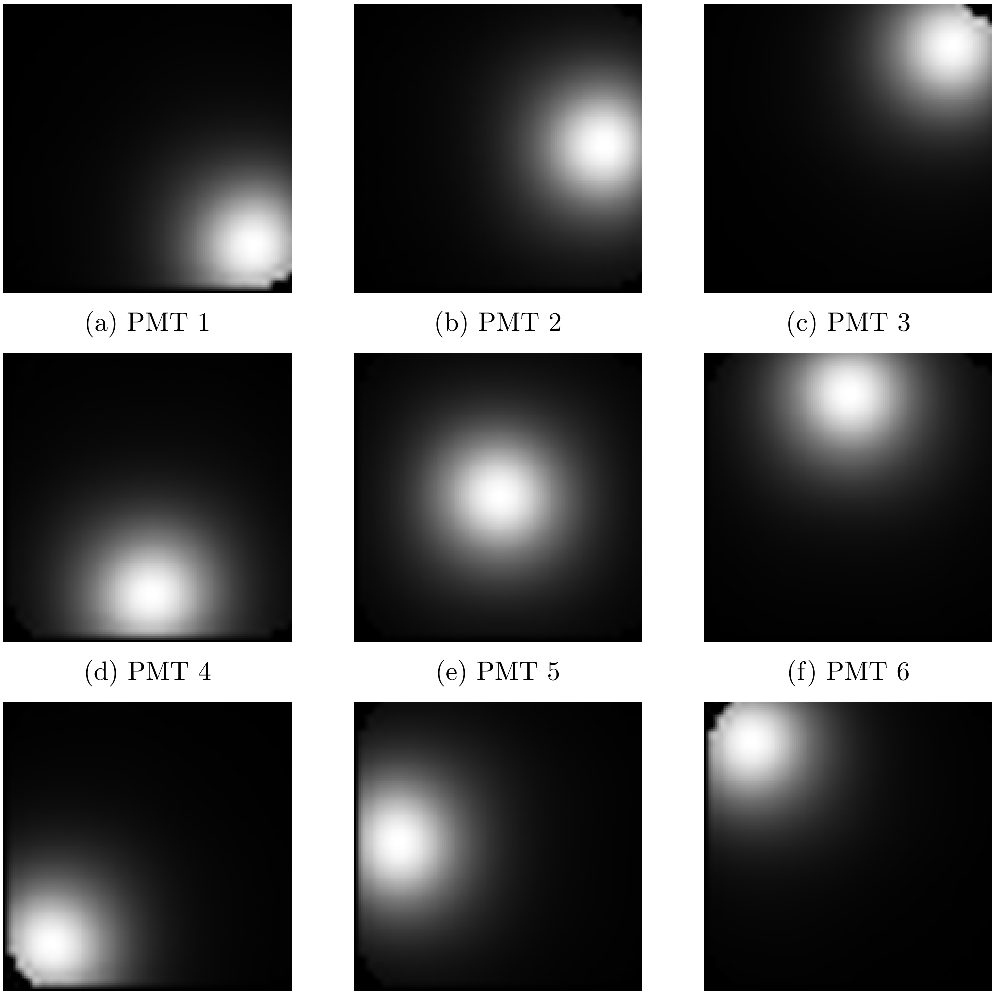
(f) PMT 6

(g) PMT 7

(h) PMT 8

(i) PMT 9

FIGURE 5.34. MDRF calibration data for ModPET camera 1



(a) PMT 1

(b) PMT 2

(c) PMT 3

(d) PMT 4

(e) PMT 5

(f) PMT 6

(g) PMT 7

(h) PMT 8

(i) PMT 9

FIGURE 5.35. MDRF calibration data for ModPET camera 2

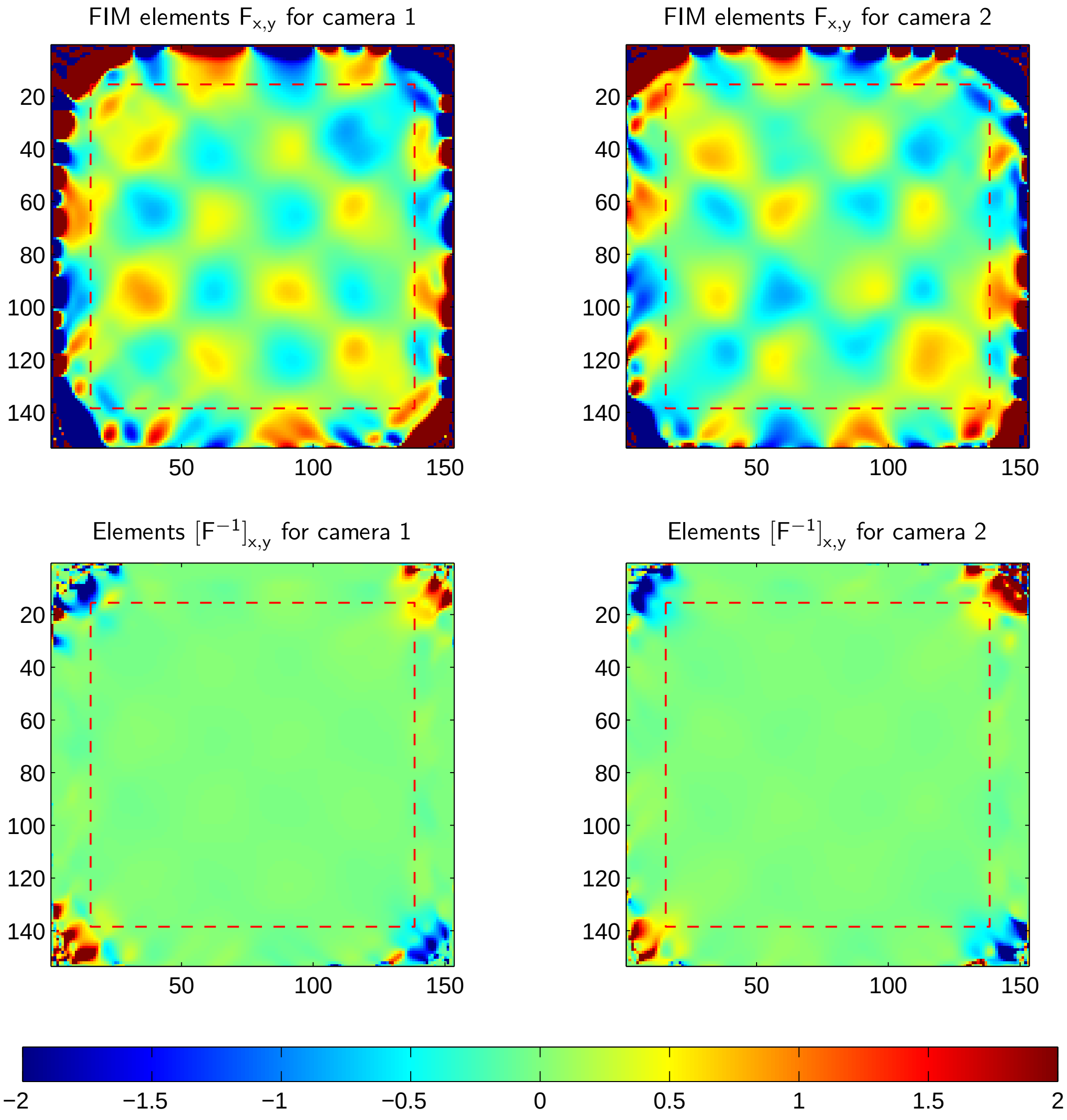


FIGURE 5.36. In the top row, plots of the off-diagonal elements of the Fisher information matrix $\mathbf{F}_{\hat{\boldsymbol{R}}'}$ (in inverse square millimeters) for the 2D estimation of location of interaction for ModPET gamma-ray cameras. In the bottom row, plots of the off-diagonal elements of $[\mathbf{F}_{\hat{\boldsymbol{R}}'}]^{-1}$ (in square millimeters). The regions outside the red squares indicate detector pixels for which estimates $\hat{\boldsymbol{R}}'$ are inaccurate and/or their calculation and processing is numerically unstable

Hence, depending on the values of estimates $\hat{X}$ and $\hat{Y}$, the probability density function $\mathrm{pr}(\hat{\boldsymbol{R}}' \mid \boldsymbol{R}')$ could alternatively be modeled as the probability density function of two independent and normally distributed random variables.

With the assumption in (5.10), the line integral over ℓ in (5.9) becomes

$$\int_{\ell_1}^{\ell_2} \mathrm{pr}\big(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_1'^{(j)} + (\boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)})\ell\big) e^{-\mu_{\mathrm{tot}}(\ell-\ell_1)|\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}|}\,\mathrm{d}\ell =$$

$$= \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)}}{2\pi}\int_{\ell_1}^{\ell_2}\exp\left[-\frac{1}{2}\big(\boldsymbol{a}-\boldsymbol{b}\ell\big)^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big(\boldsymbol{a}-\boldsymbol{b}\ell\big)\right] e^{-c(\ell-\ell_1)},$$

where we have set

$$\boldsymbol{a} = \hat{\boldsymbol{R}}_2'^{(j)} - \boldsymbol{R}_1'^{(j)}, \qquad \boldsymbol{b} = \boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)}, \qquad c = \mu_{\mathrm{tot}}\big|\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}\big|.$$

If we further set

$$A^2 = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\boldsymbol{a}, \qquad B = \boldsymbol{a}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\boldsymbol{b}, \qquad C^2 = \boldsymbol{b}^{\mathsf{T}}\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\boldsymbol{b},$$

and complete the square, we find

$$\int_{\ell_1}^{\ell_2} \mathrm{pr}\big(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_1'^{(j)} + (\boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)})\ell\big) e^{-\mu_{\mathrm{tot}}(\ell-\ell_1)|\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}|}\,\mathrm{d}\ell =$$

$$= \frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)}}{2\pi} e^{c\ell_1 - \beta/2}\int_{\ell_1}^{\ell_2}\exp\left[-\frac{1}{2}(C\ell-\alpha)^2\right]\mathrm{d}\ell,$$

where

$$\alpha = \frac{B-c}{C}, \qquad \beta = A^2 - \left(\frac{B-c}{C}\right)^2.$$

Finally, we can use the $\mathrm{erf}(\ldots)$ function to express the integral above, yielding

$$\int_{\ell_1}^{\ell_2} \mathrm{pr}\big(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_1'^{(j)} + (\boldsymbol{r}_n' - \boldsymbol{R}_1'^{(j)})\ell\big) e^{-\mu_{\mathrm{tot}}(\ell-\ell_1)|\boldsymbol{r}_n - \boldsymbol{R}_1^{(j)}|}\,\mathrm{d}\ell =$$

$$= \frac{\sqrt{\pi}}{2C}\frac{\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)}}{2\pi} e^{c\ell_1 - \frac{1}{2}\beta}\big[\mathrm{erf}(C\ell_2-\alpha) - \mathrm{erf}(C\ell_1-\alpha)\big].$$

With this result, the expression for $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ assumes the form,

$$\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{32\pi^3 Z_{\mathrm{max}}^2}\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}}\big)}\sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)} \times$$

$$\times \int_{D_1} \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \big[\operatorname{erf}(C\ell_2 - \alpha) - \operatorname{erf}(C\ell_1 - \alpha)\big] \times$$
$$\times \frac{e^{c\ell_1 - \frac{1}{2}\beta}}{C} \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}} \big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)\right] \mathrm{d}^3\boldsymbol{R}_1^{(j)}.$$

Next, we observe that some of the quantities that appear in the expression above only depend on $\boldsymbol{R}_1^{(j)}$, so that we can write

$$\begin{aligned}\operatorname{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{32\pi^3 Z_{\max}^2} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}}\big)} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)} \times \\ \times \int_{D_1'} \frac{1}{C} \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}} \big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)\right] \times \\ \times \left\{ \int_0^{Z_{\max}} \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} e^{c\ell_1 - \frac{1}{2}\beta} \big[\operatorname{erf}(C\ell_2 - \alpha) - \operatorname{erf}(C\ell_1 - \alpha)\big] \, \mathrm{d}Z_1^{(j)} \right\} \mathrm{d}^2\boldsymbol{R}_1'^{(j)}.\end{aligned}$$

Furthermore, we can use $e^{-\varepsilon} \approx 1 - \varepsilon$ for $|\varepsilon| \ll 1$ to write

$$\begin{aligned}\operatorname{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{32\pi^3 Z_{\max}^2} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}}\big)} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)} \times \\ \times \int_{D_1'} \frac{1}{C} \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}} \big(\hat{\boldsymbol{R}}_1'^{(j)} - \boldsymbol{R}_1'^{(j)}\big)\right] \times \\ \times \left\{ \int_0^{Z_{\max}} \frac{1 - \mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} e^{c\ell_1 - \frac{1}{2}\beta} \big[\operatorname{erf}(C\ell_2 - \alpha) - \operatorname{erf}(C\ell_1 - \alpha)\big] \, \mathrm{d}Z_1^{(j)} \right\} \mathrm{d}^2\boldsymbol{R}_1'^{(j)}.\end{aligned} \tag{5.11a}$$

If we repeat the same process but for $\operatorname{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, we get

$$\begin{aligned}\operatorname{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\sqrt{\pi}\mu_{\mathrm{pe}}^2}{32\pi^3 Z_{\max}^2} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}}\big)} \sqrt{\det\big(\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}\big)} \times \\ \times \int_{D_2'} \frac{1}{C} \exp\left[-\frac{1}{2}\big(\hat{\boldsymbol{R}}_2'^{(j)} - \boldsymbol{R}_2'^{(j)}\big)^{\mathsf{T}} \mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}} \big(\hat{\boldsymbol{R}}_2'^{(j)} - \boldsymbol{R}_2'^{(j)}\big)\right] \times \\ \times \left\{ \int_0^{Z_{\max}} \frac{1 - \mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)}{|\boldsymbol{R}_2^{(j)} - \boldsymbol{r}_n|^2} e^{c\ell_1 - \frac{1}{2}\beta} \big[\operatorname{erf}(C\ell_2 - \alpha) - \operatorname{erf}(C\ell_1 - \alpha)\big] \, \mathrm{d}Z_2^{(j)} \right\} \mathrm{d}^2\boldsymbol{R}_2'^{(j)}.\end{aligned} \tag{5.11b}$$

As an example, Figure 5.37 reports a grayscale plot of $\operatorname{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as calculate using the procedure discussed above. In this figure, large values of $\operatorname{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ are denoted as dark regions. The uncertainty in the depth of interaction Z for points $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ is the main reason why the non-zero values $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ extend a few millimeters in the X and Y directions. Also notice that the large values of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ near either crystal's entrance face are due to the large numerical value for either solid angle factors $\frac{1}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2}$ or $\frac{1}{|\boldsymbol{R}_2^{(j)} - \boldsymbol{r}_n|^2}$ when point $\boldsymbol{r}_n$ is near one of the crystals.

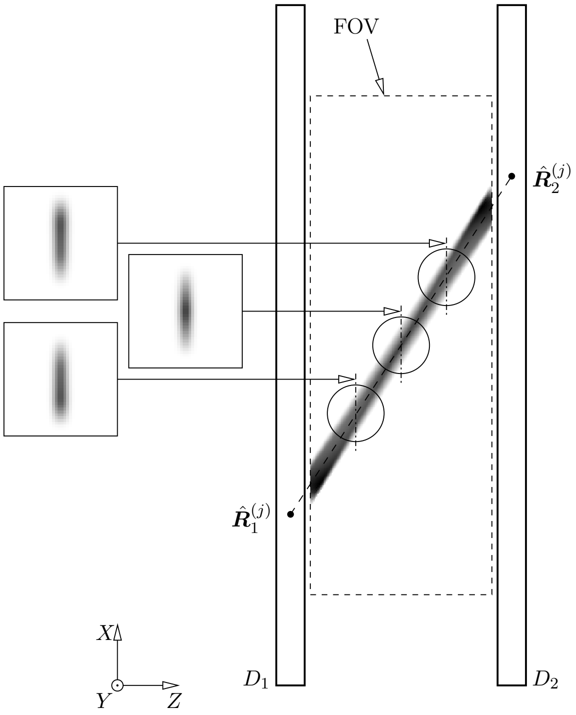

FOV
$\hat{\boldsymbol{R}}_2^{(j)}$
$\hat{\boldsymbol{R}}_1^{(j)}$
X
Y
Z
D_1
D_2

FIGURE 5.37. Grayscale plot of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$

One of the drawbacks of the expressions we found in (5.11) is that when voxel n is close to the detector over which we perform the integration numerically, the other detector space is poorly sampled. Indeed, if voxel n is close to, say, detector D_1 and we numerically evaluate $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, then a small change in the value of the integration variable $(\boldsymbol{R}_1'^{(j)}, Z_1^{(j)}) = \boldsymbol{R}_1^{(j)}$ will result in a large change in the value of the integrand, with the risk of poorly estimate $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$. This problem is most severe when the object support runs right up to the sensitive area of the camera. The same problem occurs when voxel n is close to detector D_2 and we evaluate $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ numerically. In this work, this problem has been partially solved by defining $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as in (5.8). A more satisfactory solution would be to retain the integrals over D_1 and D_2 and consider the following expression

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2} \times \\ \times \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_2'^{(j)}) e^{-\mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)} \psi_{B_n}(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}) \, \mathrm{d}^3\boldsymbol{R}_2^{(j)} \, \mathrm{d}^3\boldsymbol{R}_1^{(j)},\end{aligned}$$

in which $\psi_{B_n}(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}) = 1$ if the line of response defined by points $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ intersects voxel B_n; otherwise, $\psi_{B_n}(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}) = 0$. Equivalently, we can consider

$$\begin{aligned}\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n) = \frac{\mu_{\mathrm{pe}}^2}{4\pi Z_{\max}^2} \int_{D_2} \mathrm{pr}(\hat{\boldsymbol{R}}_2'^{(j)} \mid \boldsymbol{R}_2'^{(j)}) \frac{e^{-\mu_{\mathrm{tot}}\Delta_2(\boldsymbol{R}_2^{(j)};\boldsymbol{r}_n)}}{|\boldsymbol{R}_2^{(j)} - \boldsymbol{r}_n|^2} \times \\ \times \int_{D_1} \mathrm{pr}(\hat{\boldsymbol{R}}_1'^{(j)} \mid \boldsymbol{R}_1'^{(j)}) e^{-\mu_{\mathrm{tot}}\Delta_1(\boldsymbol{R}_1^{(j)};\boldsymbol{r}_n)} \psi_{B_n}(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}) \, \mathrm{d}^3\boldsymbol{R}_1^{(j)} \, \mathrm{d}^3\boldsymbol{R}_2^{(j)}.\end{aligned}$$

Numerical evaluation of the two expressions above is, however, computationally expensive, as two nested 3D numerical integrations are required.

Rather than using the same algorithm we used in § 5.7 to reconstruct the data, here we opted for a more general algorithm, still based on the MLEM methods. This algorithm is known in the literature as the ordered subset (OS) MLEM algorithm, or OSMLEM. First proposed in [204] for the case of binned data, the OSMLEM algorithm groups projection data into an ordered sequence of P subsets, where the number P is also called the OS *level* [204]. One iteration of the OSMLEM algorithm is now defined as a single pass through to all the P projections (or subsets) [204]. The equivalent of (3.10) but for the case of the OSMLEM algorithm takes the form:

$$\hat{f}_n^{(kP+p+1)} = \hat{f}_n^{(kP+p)} \left\{ \frac{1}{\sum_{m=1}^{M} h_{m,n}^{(p)}} \sum_{m=1}^{M} \frac{g_m^{(p)} h_{m,n}^{(p)}}{\sum_{n'=1}^{N} h_{m,n'}^{(p)} \hat{f}_{n'}^{(kP+p)}} \right\}, \tag{5.12}$$

in which $\boldsymbol{g}^{(p)}$ is the projection data associated to the p^{th} subset, for $p = 0, \ldots, P-1$, and, as usual, k denotes the iteration number. Hence, kP represents the effective number of image updates after k iterations of the algorithm. The recursion formula provided above shows that the standard MLEM algorithm is sequentially applied to each subset, with the resulting reconstruction becoming the starting value for use with the next subset [204].

There are different choices for the subsets [204]. For example, in the case of non-overlapping subsets, a natural choice for subset p is simply the data from the p^{th} projection only. On the other hand, if we employ cumulative subsets, we define subset p as the concatenation of the data from the first $p+1$ projections, for $p = 0, \dots, P-1$. Finally, we notice that if we use only one subset and define it as the union of the all P projections, the OSMLEM simply reduces to the MLEM algorithm defined in (3.10).

In the simulation examples of [204], the use of ordered subset has been found to speed up reconstruction by an order of magnitude over the classical, non-OS MLEM algorithm for the same restoration "quality." Some convergence results of the OSMLEM algorithm for noise-free data are discussed in [204] as well. For the case of noisy data, it has been empirically determined that, for a large number of iterations, the OSMLEM algorithm follows a cycle of different limiting images. One way to break this cycle and obtain a unique solution is to monotonically reduce the OSMLEM level down to 1 as the algorithm proceeds throughout the iterations [204].

The adaptation of the OSMLEM algorithm to the case of list-mode data is trivial: by comparing (5.12) with (4.10) we can conclude [259, 260, 264, 342, 361]

$$\hat{f}_n^{(kP+p+1)} = \hat{f}_n^{(kP+p)} \left\{ \frac{1}{\tau^{(p)}} \sum_{\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}^{(p)}} \frac{\mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(kP+p)}} \right\}, \tag{5.13}$$

in which $\hat{\mathscr{A}}^{(0)}, \dots, \hat{\mathscr{A}}^{(P-1)}$ are the P data subsets (or projections), along with their exposure times $\tau_0, \dots, \tau_{P-1}$, and we have explicitly included in $\mathrm{pr}^{(p)}(\hat{\boldsymbol{A}} \mid n)$ the dependence on the subset index p (this would be useful, for example, if the imaging system allows us to collect data along different projections, if the imaging system is reconfigured as the different data subsets are acquired, or even if we want to introduce motion compensation into our model [362–366]). We will refer to the iterative algorithm mathematically formalized above as the ordered subset (OS) LMMLEM or OSLMMLEM algorithm.

The application of the OSLMMLEM algorithm to emission tomography data has become very popular in the last decade or so, mainly for its reconstruction speed when compared to the standard MLEM algorithm operating on binned data. Indeed, it has been noticed that often just one or two iterations [259] of the OSLMMLEM reconstruction algorithm already produce an image that would not change significantly if additional iterations were performed. In particular, if just one iteration of the OSLMMLEM algorithm is used, the algorithm passes through the data $\hat{\mathscr{A}}$ only once [259, 361], and subset $\hat{\mathscr{A}}^{(p)}$ can be discarded as soon as it has been used in (5.13). This way, a hypothetical one-pass OSLMMLEM imaging system would be able to display estimate $\hat{\boldsymbol{f}}^{(p)}$ in "real-time" [259] and while the next projection data $\hat{\mathscr{A}}^{(p)}$ is being acquired and processed to calculate $\hat{\boldsymbol{f}}^{(p+1)}$ according to (5.13). As noted in [361], however, convergence of the OSLMMLEM algorithm is no longer guaranteed (unless, of course, for the case in which the OSLMMLEM algorithm reduces to the LMMLEM algorithm), but a regularized version of the OSLMMLEM algorithm has been proposed and its convergence properties proven [260].

Application to ModPET: Reconstruction Results

The OSLMMLEM algorithm discussed above was applied to list-mode data acquired with ModPET, a schematic diagram of which is reported in Figure 5.29. The input data $\hat{\mathscr{A}}$ consisted of a list of 11,450,814 lines or response (LORs) for a ^{18}F-NaF bone scan for a normal mouse. Figure 5.38, which was originally published in [6], shows a maximum intensity projection of reconstructed volume, superimposed on a grayscale optical image of the mouse. The data were reconstructed using the LMMLEM reconstruction algorithm of [6].

The elements $\hat{\boldsymbol{A}}^{(j)}$ of $\hat{\mathscr{A}}$ are of the form $\hat{\boldsymbol{A}}^{(j)} = \left(\hat{\boldsymbol{R}}_1^{(j)}, \ell_1^{(j)}, \hat{\boldsymbol{R}}_2^{(j)}, \ell_2^{(j)}\right)$, where the vector $\hat{\boldsymbol{R}}_i^{(j)}$ denotes a 3D point within crystal D_i, and the scalar value $\ell_i^{(j)}$ is the

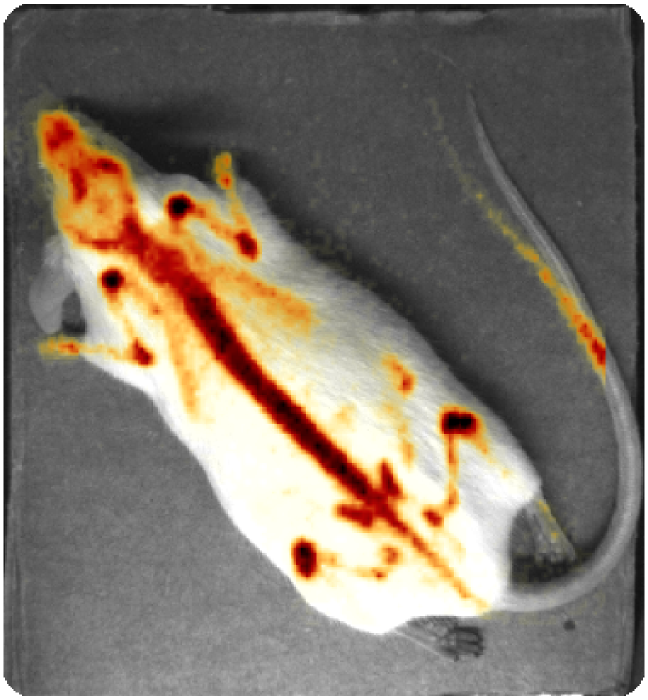


Figure 5.38. Maximum intensity projection of reconstructed volume from a ^{18}F-NaF mouse bone scan superimposed on a grayscale optical image (courtesy of Stephen K. Moore [6])

value of the log-likelihood defined in (3.4) that was previously calculated as part of the estimation of point $\hat{\boldsymbol{R}}_i^{(j)}$ from PMT data. The $\hat{Z}_i^{(j)}$ component (i.e., the depth of interaction) of each point $\hat{\boldsymbol{R}}_i^{(j)}$ in the data $\hat{\mathscr{A}}$ was either -2.50 mm or $+2.50$ mm, depending which crystal space the point $\hat{\boldsymbol{R}}_i^{(j)}$ belongs to. This makes sense if we recall that ModPET features a thin (5 mm) NaI(Tl) crystal in each of the two 2D gamma-ray cameras and, as we argued above, a depth of interaction of 2.50 mm is approximatively the expected value of the random variable representing the actual depth of interaction. Contrary to [6], our implementation does not use the estimates $\hat{Z}_i^{(j)}$ of the depths of interaction. Indeed and as we can see from (5.11), only the 2D points $\hat{\boldsymbol{R}}_i'^{(j)}$ enter in our expressions for $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ and $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, and the distribution of the actual depth of interaction $Z_i^{(j)}$ is assumed uniform over the crystal thickness. The remaining components $\big(\hat{X}_i^{(j)}, \hat{Y}_i^{(j)}\big) = \hat{\boldsymbol{R}}_i'^{(j)}$ of the vector $\hat{\boldsymbol{R}}_i^{(j)}$ took on values over a 153 × 153 regular grid of points spaced by 750 μm along each dimension.

The elements $\hat{\boldsymbol{A}}^{(j)}$ of the list $\hat{\mathscr{A}}$ were initially filtered using the values of the log-likelihood $\ell_i^{(j)}$: given a lower bound for the log-likelihood on the form of a threshold ℓ_{min}, attribute vector $\hat{\boldsymbol{A}}^{(j)}$ of $\hat{\mathscr{A}}$ is retained if and only if $\ell_1^{(j)} > \ell_{\mathrm{min}}$ and $\ell_2^{(j)} > \ell_{\mathrm{min}}$. This process allows us to discard lines of response that, overall, have a low likelihood given the PMT data acquired by the hardware. In our implementation, we set $\ell_{\mathrm{min}} = -30.00$ [6], which resulted in the rejection of about 65.13% of the lines of response originally contained in $\hat{\mathscr{A}}$. Another 14.95% of the original lines of response in $\hat{\mathscr{A}}$ that passed the likelihood threshold test were rejected because at least one of the points $\hat{\boldsymbol{R}}_1'^{(j)}$ or $\hat{\boldsymbol{R}}_2'^{(j)}$ that define the j^{th} line of response fell outside the red squares drawn in Figure 5.33. The reason we decided to exclude these lines of response is because the Fisher information matrices $\mathbf{F}_{\hat{\boldsymbol{R}}}$ are poorly estimated for points $\hat{\boldsymbol{R}}$ that lie outside the red squares, resulting in numerical instability during the calculation of $\mathrm{pr}^{(D_1)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ and $\mathrm{pr}^{(D_2)}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ according to (5.11). After these two selection steps, only 2,280,715 lines of response were retained, corresponding to 19.92% of the original lines of response in $\hat{\mathscr{A}}$. We will denote the list of the retained lines of response as $\hat{\mathscr{A}}^{(\checkmark)}$:

$$\hat{\mathscr{A}}^{(\checkmark)} = \big\{\hat{\boldsymbol{A}}^{(1)}, \ldots, \hat{\boldsymbol{A}}^{(J)}\big\}, \tag{5.14}$$

where $J = 2280715$.

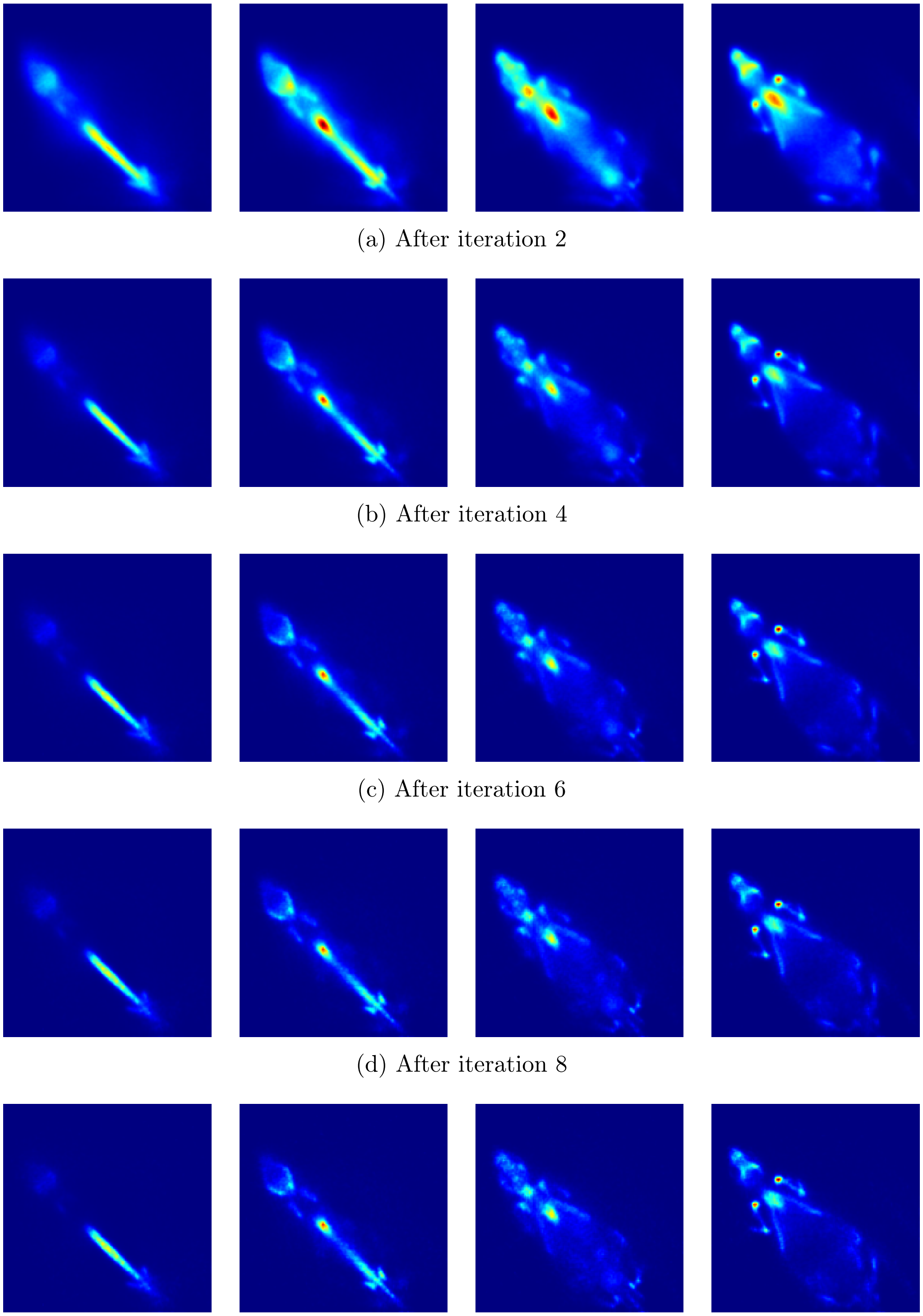
The calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ for $\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}^{(\checkmark)}$ according to (5.8) and (5.11) was performed using computer code developed for a GPU device. For easy handling, the list $\hat{\mathscr{A}}^{(\checkmark)}$ was partitioned into 512 sublists, which we called "fragments." Each fragment contained about 4,455 lines of response. Specialized CUDA code was developed to evaluate $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ for each line of response in each fragment, resulting in 512 data files. To speed up the computation, we also decided to restrict the values of the voxel index n to voxels "near" the line of response. This is justified by the fact that $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n) \approx 0$ for voxels n that were more than 2 mm away from the line of response, as shown in Figure 5.37. Voxel indices n were saved to disk along with the values of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$. We likewise sped up the evaluation of the integrals over $\boldsymbol{R}_i'^{(j)}$ in (5.11) by only considering an $N_X \times N_Y = 16 \times 16$ grid of points $\boldsymbol{R}_i'^{(j)}$ inside a $\Delta_X \times \Delta_Y = 4\ \mathrm{mm} \times 4\ \mathrm{mm}$ region centered around $\hat{\boldsymbol{R}}_i'^{(j)}$. On the other hand, the integral over $Z_i^{(j)}$ used $N_Z = 16$ samples uniformly spaced over the whole $\Delta_Z = 5$ mm detector thickness. Despite all these attempts to speed the calculation up, the evaluation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ for $\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}^{(\checkmark)}$ took approximatively 1.63 hours when a total of eight NVIDIA Tesla C2050 GPU cards were used. This figure translates to about 2.58 ms per line of response. As shown in Figure 5.29, the field of view measured 88 mm × 88 mm × 32 mm and it was divided into a 3D grid of 160 × 160 × 64 voxels. Thus, each voxel measured $\delta_x \times \delta_y \times \delta_z = 550\ \mu\mathrm{m} \times 550\ \mu\mathrm{m} \times 500\ \mu\mathrm{m}$.

To test the effectiveness of the OSLMMLEM on reconstructing the list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$, we first ran the LMMLEM algorithm discussed in § 4.4 on the data $\hat{\mathscr{A}}^{(\checkmark)}$. This can effectively be done by setting $P = 1$ in (5.13). We ran 10 iterations of (5.13) with $P = 1$ and reported some reconstruction results in Figure 5.39. Because the reconstructed object is 3D, we limited ourselves to displaying some significant slices through the object. Going from left to right, Figure 5.39 shows sections of the reconstructed object along x-y planes in the field of view for $z = -6.25$ mm, $z = -3.25$ mm, $z = -0.25$ mm, and $z = 2.25$ mm, where the point $\boldsymbol{r} = (x, y, z) = (0, 0, 0)$ is located at the center of the field of view. The same planes will be used in the next figures for ease of comparison. Performing 10 iterations of the LMMLEM algorithm took approximatively 3.10 hours, of which about 2.65 hours were spent in loading from the disk the data $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$.

To test the effectiveness of the OSLMMLEM algorithm in speeding up the reconstruction when compared to the LMMLEM algorithm, we applied (5.13) with $P > 1$ to the data $\hat{\mathscr{A}}^{(\checkmark)}$. More specifically, data subsets $\hat{\mathscr{A}}^{(0)}, \ldots, \hat{\mathscr{A}}^{(P-1)}$ were defined in terms of the 512 fragments into which the list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ was subdivided. As an example, for the case $P = 2$, $\hat{\mathscr{A}}^{(0)}$ was made up of the first 256 fragments and $\hat{\mathscr{A}}^{(1)}$ was made up of the remaining 256 fragments. In this way, roughly the first half of $\hat{\mathscr{A}}^{(\checkmark)}$ defined $\hat{\mathscr{A}}^{(0)}$ and the remaining elements of $\hat{\mathscr{A}}^{(\checkmark)}$ were used to form $\hat{\mathscr{A}}^{(1)}$. A similar scheme was used for the cases $P = 4$ and $P = 16$. Figure 5.40 shows OSLMMLEM reconstructions for $P = 2$; Figure 5.41 for $P = 4$ and, finally, Figure 5.42 for $P = 16$. In all these reconstructions, we took the values of the sensitivities as the initial guess: $\hat{\boldsymbol{f}}^{(0)} = \boldsymbol{s}$.

We do not claim that visual examination of just one reconstructed object suffices to define image quality, but we note that the LMMLEM reconstruction after 10 iterations (shown in Figure 5.39) is comparable to the $P = 2$ OSLMMLEM reconstruction after 4 iterations (see Figure 5.40), which, in turn, is comparable to the $P = 4$ OSLMMLEM reconstruction after 2 iterations (see Figure 5.41). This comparison is summarized in Figure 5.43.

The reconstructed images reported in Figure 5.42 show one of the shortcomings of the multiplicative nature of any MLEM scheme (hence, the OSLMMLEM algorithm as well). To understand this point, let us consider the extreme case in which each ordered



(a) After iteration 2

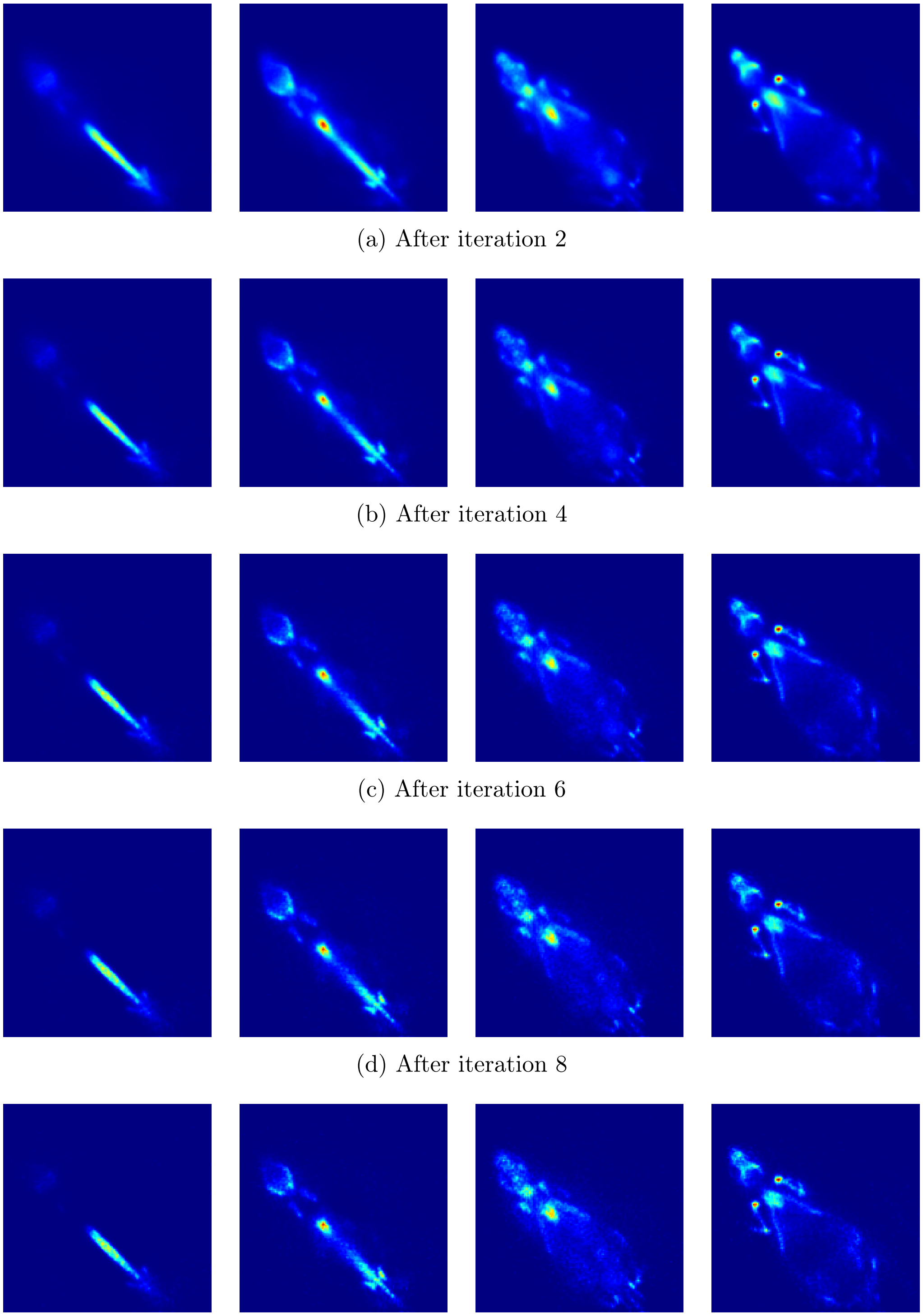
(b) After iteration 4

(c) After iteration 6

(d) After iteration 8

(e) After iteration 10

FIGURE 5.39. LMMLEM reconstructions of list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$



(a) After iteration 2

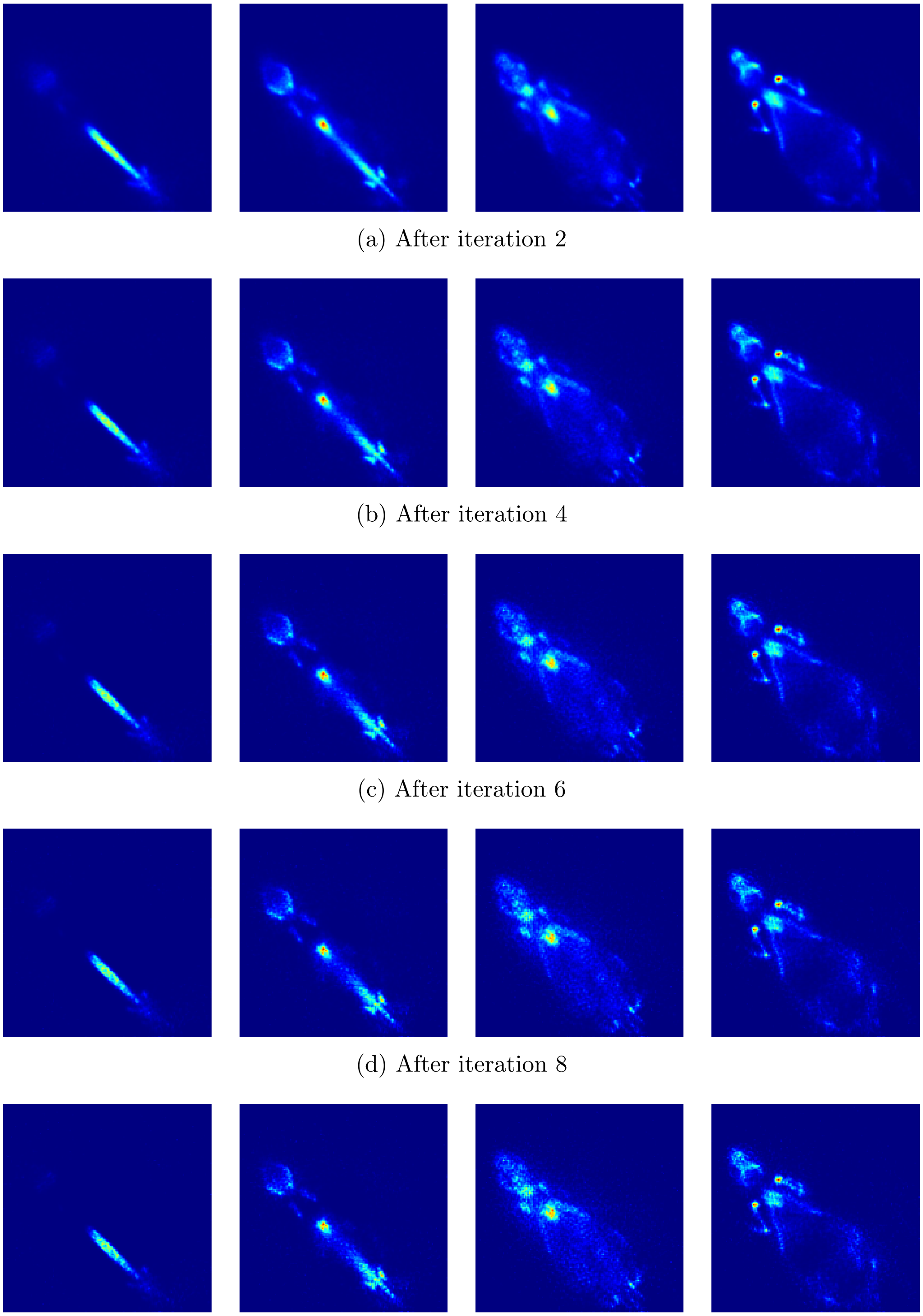
(b) After iteration 4

(c) After iteration 6

(d) After iteration 8

(e) After iteration 10

FIGURE 5.40. OSLMMLEM reconstructions of list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ for $P = 2$



(a) After iteration 2

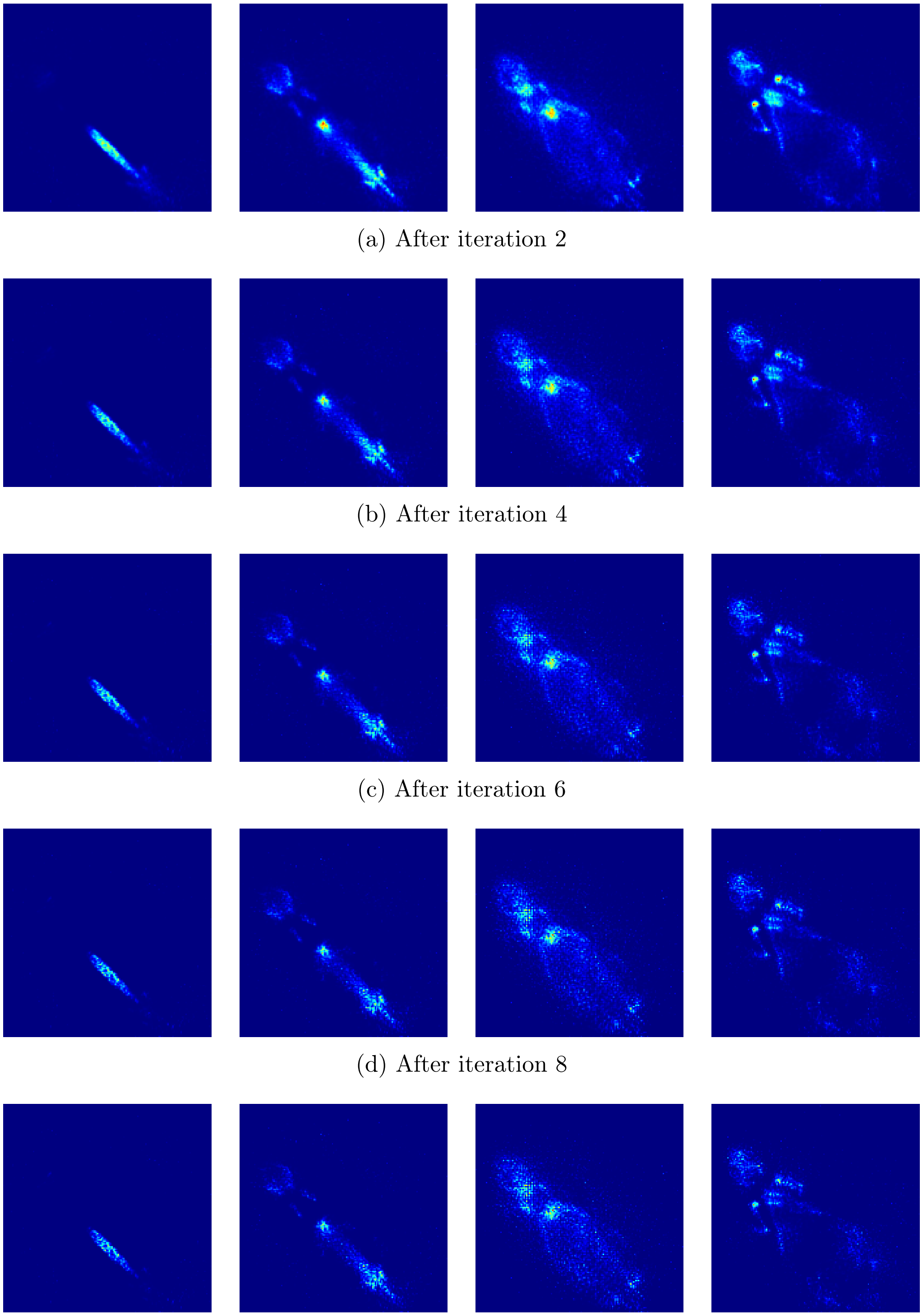
(b) After iteration 4

(c) After iteration 6

(d) After iteration 8

(e) After iteration 10

FIGURE 5.41. OSLMMLEM reconstructions of list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ for $P = 4$



(a) After iteration 2

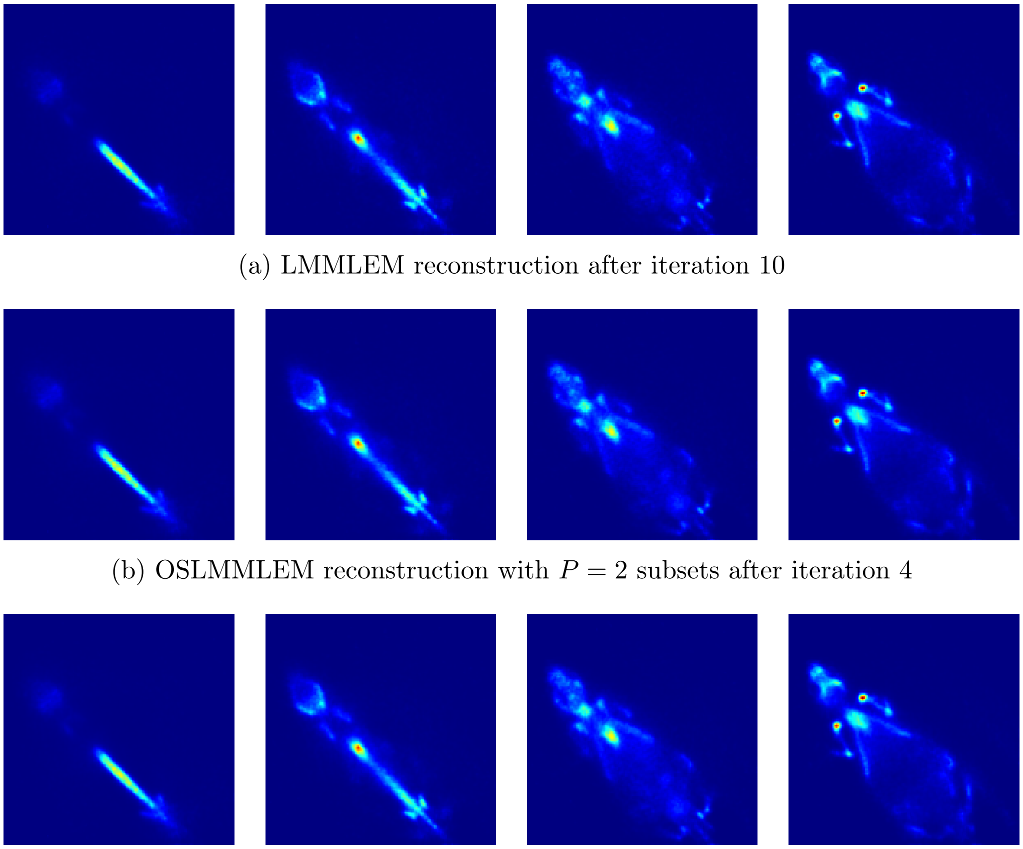
(b) After iteration 4

(c) After iteration 6

(d) After iteration 8

(e) After iteration 10

FIGURE 5.42. OSLMMLEM reconstructions of list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ for $P = 16$



(a) LMMLEM reconstruction after iteration 10

(b) OSLMMLEM reconstruction with $P = 2$ subsets after iteration 4

(c) OSLMMLEM reconstruction with $P = 4$ subsets after iteration 2

FIGURE 5.43. Comparison of LMMLEM reconstructions with OSLMMLEM reconstructions

subset $\hat{\mathscr{A}}^{(p)}$ contains just one of the elements of $\hat{\mathscr{A}}^{(\checkmark)}$ and let us assume, for simplicity, that $\hat{f}_n^{(0)} = 1$ for $n = 1, \ldots, N$. When the first ordered subset $\hat{\mathscr{A}}^{(0)} = \{\hat{\boldsymbol{A}}^{(1)}\}$ is processed according to (5.13), only a subset $\mathscr{N}_{\hat{\boldsymbol{A}}^{(1)}}$ of the voxel indices $\{1, \ldots, N\}$ will satisfy $\mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(1)} \mid n) \not\approx 0$ for $n \in \mathscr{N}_{\hat{\boldsymbol{A}}^{(1)}}$, as suggested by Figure 5.37. Hence, $\hat{f}_n^{(1)} \approx 0$ unless $n \in \mathscr{N}_{\hat{\boldsymbol{A}}^{(1)}}$. When the second ordered subset $\hat{\mathscr{A}}^{(1)} = \{\hat{\boldsymbol{A}}^{(2)}\}$ is processed, additional voxels whose index does not belong to $\mathscr{N}_{\hat{\boldsymbol{A}}^{(2)}}$ will be set to zero (or to a very small value). This process repeats until all the ordered subsets are processed. Unless *all* the $\mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}}$ have non-empty intersection,

$$\bigcap_{j=1}^{J} \mathscr{N}_{\hat{\boldsymbol{A}}^{(j)}} \neq \varnothing,$$

the estimated image $\hat{\boldsymbol{f}}^{(P)}$ after the first iteration of (5.13) will satisfy $\hat{f}_n^{(P)} = 0$ for all $n = 1, \ldots, N$. A similar result, albeit to a lesser extent, will occur if the number of elements in each ordered subset is "too small." Indeed, as shown in Figure 5.42, only a relatively small number of pixels of the reconstructed object will assume a non-zero value and the reconstructed object itself will not look smooth.

To solve this problem, we start by rewriting the OSLMMLEM algorithm of (5.13) in additive form [367]:

$$\hat{f}_n^{(kP+p+1)} = \hat{f}_n^{(kP+p)} + \Delta \hat{f}_n^{(kP+p)}, \tag{5.15}$$

where

$$\Delta \hat{f}_n^{(kP+p)} = \hat{f}_n^{(kP+p)} \left\{ \frac{1}{\tau^{(p)}} \sum_{\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}^{(p)}} \frac{\mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(kP+p)}} - 1 \right\}.$$

The scheme above, which is mathematically equivalent to (5.13), is then modified by introducing the relaxation parameter $\lambda^{(kP+p)}$ so that [264, 367]

$$\hat{f}_n^{(kP+p+1)} = \hat{f}_n^{(kP+p)} + \lambda^{(kP+p)} \Delta \hat{f}_n^{(kP+p)}, \tag{5.16}$$

or also

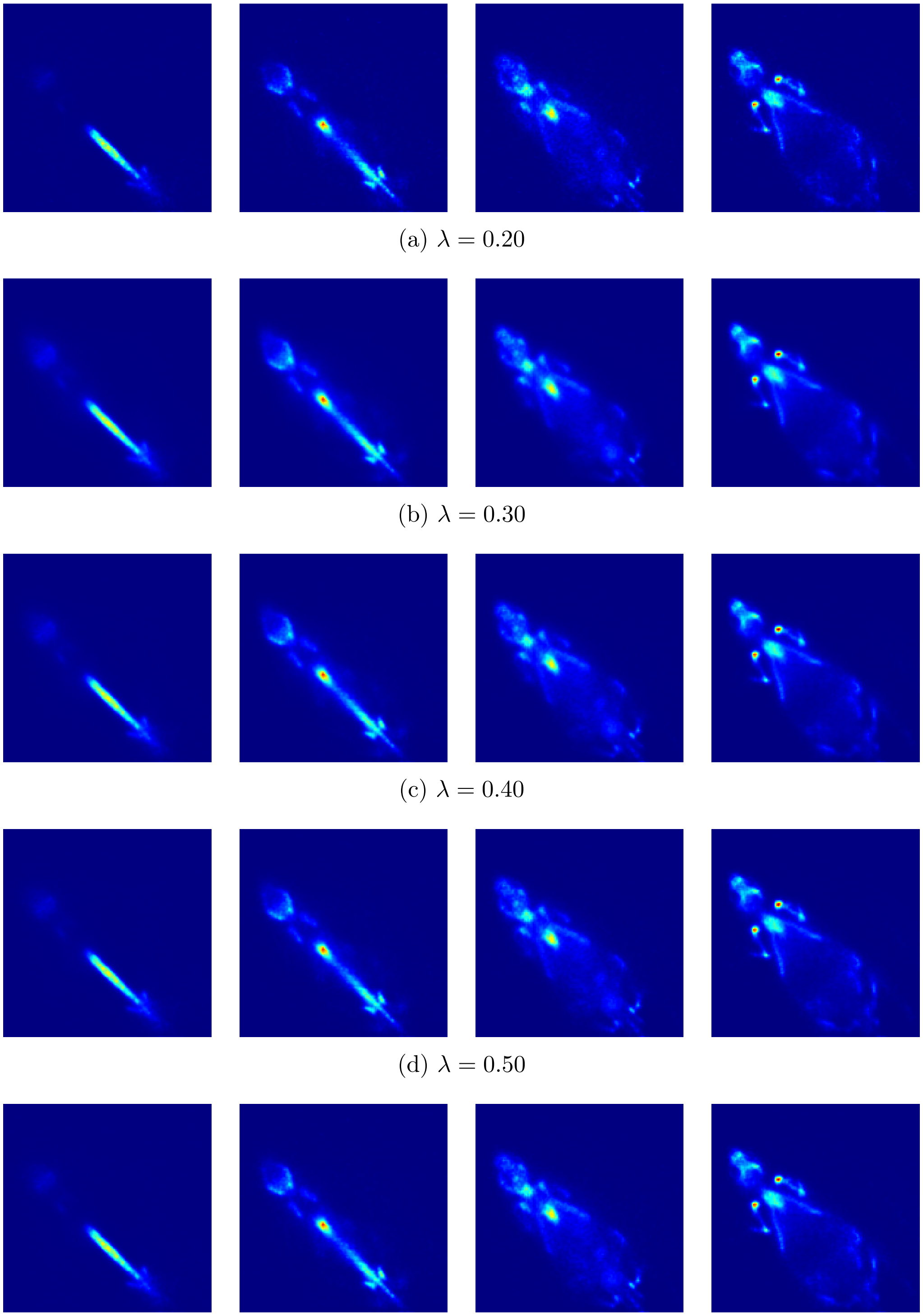
$$\hat{f}_n^{(kP+p+1)} = \left(1 - \lambda^{(kP+p)}\right) \hat{f}_n^{(kP+p)} +$$

$$+ \lambda^{(kP+p)} \hat{f}_n^{(kP+p)} \left\{ \frac{1}{\tau^{(p)}} \sum_{\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}^{(p)}} \frac{\mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n)}{\sum_{n'=1}^{N} \mathrm{pr}^{(p)}(\hat{\boldsymbol{A}}^{(j)} \mid n') s_{n'} \hat{f}_{n'}^{(kP+p)}} \right\}.$$

The form above shows that, so long as $0 \leqslant \lambda^{(kP+p)} \leqslant 1$ and $\hat{f}_n^{(kP+p)} \geqslant 0$, then $\hat{f}_n^{(kP+p+1)} \geqslant 0$. More important, however, is the fact that by introducing the relaxation parameter $\lambda^{(kP+p)}$ we have lost the original multiplicative nature of the OSLMMLEM algorithm so that the case $\hat{\boldsymbol{f}}^{(kP+p)} = \boldsymbol{0}$ we discussed above no longer occurs: choosing $\lambda^{(kP+p)}$ so that $0 \leqslant \lambda^{(kP+p)} < 1$ has the advantage that $\hat{f}_n^{(kP+p)} > 0$ also enters in additive form in the expression for $\hat{f}_n^{(kP+p+1)}$, preventing $\hat{f}_n^{(kP+p+1)} \to 0$ throughout the iterations. This, however, comes at a price: by introducing the relaxation parameter $\lambda^{(kP+p)}$, the algorithm in (5.15) might no longer converge [367] or, if it does, it might converge to a solution that is no longer a maximum-likelihood solution. In practice, just one or two passes through the data are usually enough to obtain reconstruction comparable to those obtained with many iterations of the LMMLEM algorithm. Convergence of the relaxed OSLMMLEM algorithm is thus reduced to a question with only theoretical relevance.

To test the relaxed version of the OSLMMLEM algorithm, we ran it on the same data $\hat{\mathscr{A}}^{(\checkmark)}$ we used before. In our reconstruction, we set the number P of ordered subsets to 16 and we tested the algorithm for different values of $\lambda^{(kP+p)}$. For simplicity, we set $\lambda^{(kP+p)} = \lambda$ for all k and p, and for some value of λ we picked up before running the reconstructions. We also wanted to test the claims in [264,361] and decided to run just one pass throughout the data $\hat{\mathscr{A}}^{(\checkmark)}$. Some reconstructions obtained after just one iteration and for different values of λ are reported in Figure 5.44. By visually comparing these reconstructions with those reported in Figure 5.43, we conclude that the $\lambda = 0.50$ "one-pass" OSLMMLEM algorithm with $P = 16$ ordered subsets is capable of producing reconstructed data comparable to those of the LMMLEM algorithm but after 10 iterations (see Figure 5.39).

So far, we have not discussed how the values $\lambda^{(kP+p)}$ can be chosen; in Figure 5.44,



(a) $\lambda = 0.20$

(b) $\lambda = 0.30$

(c) $\lambda = 0.40$

(d) $\lambda = 0.50$

(e) $\lambda = 0.60$

FIGURE 5.44. Relaxed OSLMMLEM reconstructions of list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ with $P = 16$ subsets and after iteration 1

we tried different values and we noticed that some particular choices gave "better" reconstructions than others. This idea can be made more formal by carrying out an objective assessment of image quality [5, 65, 69, 72, 124, 318] or OAIQ. From the statistics of the objects being imaged, the noise properties of the imaging system, the task we want to perform (such as detection of a signal at a known location), and how we perform it, an appropriate figure of merit for task performance—such as the SNR^2—can be introduced and maximized. In this context, the "optimal" values of $\lambda^{(kP+p)}$ are defined as those that maximize the task performance figure of merit.

Originally introduced in [367], the parameters $\lambda^{(kP+p)}$ were allowed to be larger than 1 in the hope that would speed up the reconstruction. As noticed in [367], if $\lambda^{(kP+p)} > 1$, non-negativity of estimates $\hat{\boldsymbol{f}}^{(kP+p)}$ is, in general, no longer guaranteed. Indeed, if we multiply both sides of (5.15) by s_n and sum over n, then, by (4.11) we get:

$$\sum_{n=1}^{N} s_n \Delta \hat{f}_n^{(kP+p)} = 0.$$

Unless $\Delta \hat{f}_n^{(kP+p)} = 0$ for all n, the result above shows that some of the $\hat{f}_n^{(kP+p+1)}$ will be negative if $\lambda^{(kP+p)}$ is chosen too large in (5.16). To circumvent this problem, a pseudorelaxation parameter $\mu_n^{(kP+p)}$ is introduced [367]

$$\mu_n^{(kP+p)} = \begin{cases} \infty & \text{if } \Delta \hat{f}_n^{(kP+p)} \geqslant 0, \\ -\dfrac{\hat{f}_n^{(kP+p)}}{\Delta \hat{f}_n^{(kP+p)}} & \text{if } \Delta \hat{f}_n^{(kP+p)} < 0. \end{cases}$$

In [367], the relaxation parameter $\lambda^{(kP+p)}$ is then heuristically calculated as

$$\lambda^{(kP+p)} = \min\left\{ \frac{\mu_{\min}^{(kP+p)} + 1}{2}, 4 \right\},$$

where we have set

$$\mu_{\min}^{(kP+p)} = \min\left\{ \mu_n^{(kP+p)}, \text{ for } n = 1, \ldots N \right\}.$$

A more interesting alternative, explored in [182], is to chose $\lambda^{(kP+p)}$ as the value λ that maximizes the likelihood $\mathrm{pr}\big(\hat{\mathscr{A}}^{(p)} \mid \hat{f}_n^{(kP+p)} + \lambda \Delta \hat{f}_n^{(kP+p)}\big)$:

$$\lambda^{(kP+p)} = \operatorname*{arg\,max}_{0 \leqslant \lambda \leqslant \mu_{\min}^{(kP+p)}} \Big[\mathrm{pr}\big(\hat{\mathscr{A}}^{(p)} \mid \hat{f}_n^{(kP+p)} + \lambda \Delta \hat{f}_n^{(kP+p)}\big)\Big].$$

The line-search over λ is easily carried out via a 1D implementation of the contracting-grid algorithm we discussed in § 3.3. Although this scheme does not immediately fall within the OAIQ scenario [5,65,69,72,124,318] we alluded to above, we can interpret the line-search step as the maximization of the iteration-by-iteration figure of merit given by the likelihood $\mathrm{pr}\big(\hat{\mathscr{A}}^{(p)} \mid \hat{f}_n^{(kP+p)} + \lambda \Delta \hat{f}_n^{(kP+p)}\big)$.

An interesting question is whether or not the complex calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as mathematically formalized in (5.8) and (5.11) is actually providing an improvement in terms of image quality with respect to simpler ways of calculating $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$. For example, we could decide to obtain $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ by calculating the intersection length between lines of response and voxels in the field of view. To pursue this approach, we first rewrite $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as [6]

$$\mathrm{pr}\big(\hat{\boldsymbol{A}}'^{(j)} \mid n\big) = \int_{\infty} \mathrm{pr}\big(\hat{\boldsymbol{A}}'^{(j)} \mid \boldsymbol{A}^{(j)}\big) \mathrm{pr}\big(\boldsymbol{A}^{(j)} \mid n\big)\, \mathrm{d}\boldsymbol{A}^{(j)},$$

where $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid \boldsymbol{A}^{(j)})$ is the probability density function for estimating the pair $\hat{\boldsymbol{A}}'^{(j)} = (\hat{\boldsymbol{R}}_1'^{(j)}, \hat{\boldsymbol{R}}_2'^{(j)})$, when the true coincidence pair endpoints are $\boldsymbol{A}^{(j)} = (\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$.

The term $\mathrm{pr}(\boldsymbol{A}^{(j)} \mid n)$ represents the geometric radiometry of the imaging system. For fixed $\boldsymbol{A}^{(j)} = (\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$, the quantity $\mathrm{pr}(\boldsymbol{A}^{(j)} \mid n)$ has to be understood as the likelihood of a pair of gamma-ray photons being produced within voxel n, when the actual line of response $\ell(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$ is defined by points $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ [6]. Hence, $\mathrm{pr}(\boldsymbol{A}^{(j)} \mid n)$ is proportional to the length $L(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}; n)$ of the segment resulting from the intersection between the line of response $\ell(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$ and the volume space defined by the n^{th} voxel

$$\mathrm{pr}\big(\boldsymbol{A}^{(j)} \mid n\big) = \mathrm{pr}\big(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)} \mid n\big) = \frac{L\big(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}; n\big)}{\sum_{n'=1}^{N} L\big(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}; n'\big)}.$$

If $\ell(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$ does not intersect voxel n, we will set $\mathrm{pr}(\boldsymbol{A}^{(j)} \mid n) = 0$. Lengths $L(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)}; n)$ are efficiently calculated via the Siddon algorithm [360, 368].

For the case of ModPET, points $\hat{\boldsymbol{R}}_1'^{(j)}$ and $\hat{\boldsymbol{R}}_2'^{(j)}$ making up $\hat{\boldsymbol{A}}'^{(j)}$ are 2D random estimates corresponding to actual 3D locations of interaction within the corresponding camera's crystals. For this reason, Monte Carlo (MC) integration [369, 370] is used in [6] to estimate $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ according to

$$\mathrm{pr}^{(\mathrm{MC})}\big(\hat{\boldsymbol{A}}'^{(j)} \mid n\big) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{pr}\big(\boldsymbol{R}_1^{(j,m)}, \boldsymbol{R}_2^{(j,m)} \mid n\big), \tag{5.17}$$

where we used the superscript "(MC)" to differentiate the probability density function above from the one introduced in (5.11). In the calculation of $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, points $\boldsymbol{R}_i^{(j,m)} = (X_i^{(j,m)}, Y_i^{(j,m)}, Z_i^{(j,m)})$ are randomly generated according to

$$\mathrm{pr}\big(\hat{X}_i^{(j)} \mid X_i^{(j,m)}\big) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left(-\frac{\big(\hat{X}_i^{(j)} - X_i^{(j,m)}\big)^2}{2\sigma_X^2}\right),$$

$$\mathrm{pr}\big(\hat{Y}_i^{(j)} \mid Y_i^{(j,m)}\big) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{\big(\hat{Y}_i^{(j)} - Y_i^{(j,m)}\big)^2}{2\sigma_Y^2}\right),$$

$$\mathrm{pr}\big(Z_i^{(j,m)}\big) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} \exp\left(-\frac{\big(Z_i^{(j,m)} - Z_i^{(\mathrm{middle})}\big)^2}{2\sigma_Z^2}\right),$$

in which $Z_i^{(\mathrm{middle})}$ denotes the depth of a point midway inside crystal i, for $i = 1, 2$, and 2D points $\hat{\boldsymbol{R}}_1'^{(j)} = (\hat{X}_1^{(j)}, \hat{Y}_1^{(j)})$ and $\hat{\boldsymbol{R}}_2'^{(j)} = (\hat{X}_2^{(j)}, \hat{Y}_2^{(j)})$ define the list-mode datum $\hat{\boldsymbol{A}}'^{(j)} = (\hat{\boldsymbol{R}}_1'^{(j)}, \hat{\boldsymbol{R}}_2'^{(j)})$. From the expressions above, we see that the components of $\boldsymbol{R}_i^{(j,m)}$ are independent of each other and their standard deviations are σ_X, σ_Y, and σ_Z, respectively.

There are a few substantial differences between $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as calculated according to (5.8) and $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ defined in (5.17):

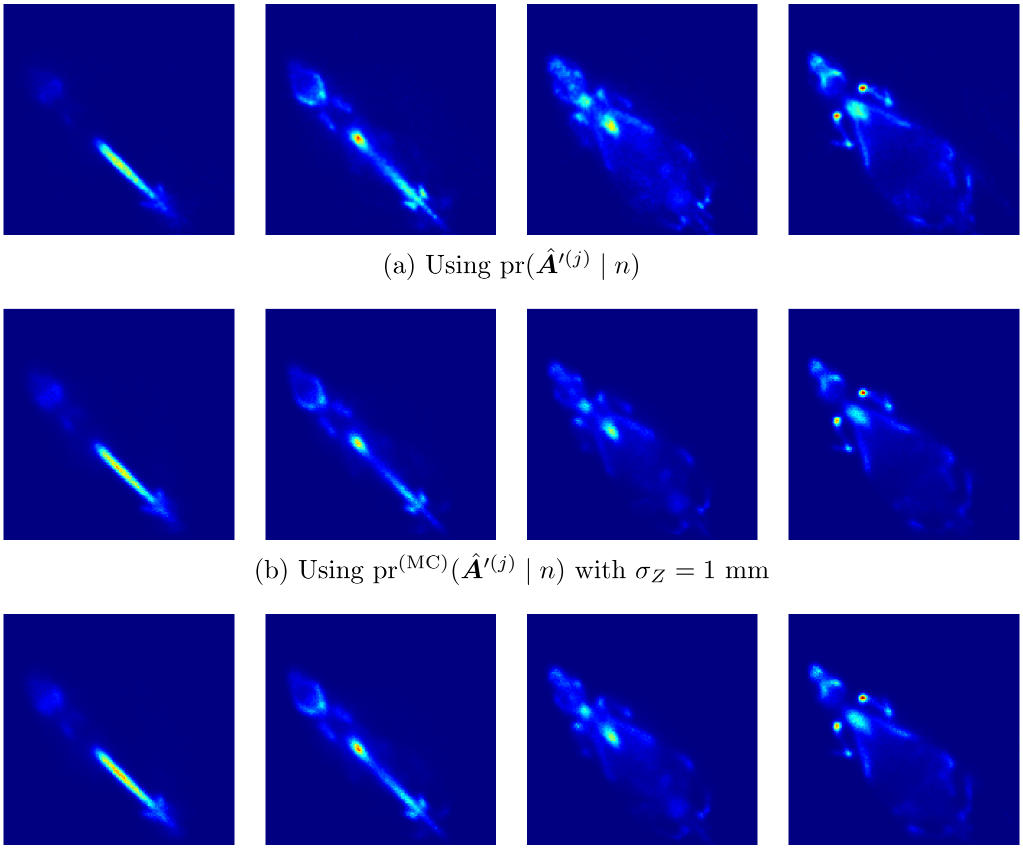
- The expression for $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ assumes uniform performance over the entrance face of the gamma-ray cameras. Furthermore, X estimates are assumed independent from Y estimates. On the other hand, the expression in (5.8) uses the Fisher information matrices $\mathbf{F}_{\hat{\boldsymbol{R}}_1'^{(j)}}$ and $\mathbf{F}_{\hat{\boldsymbol{R}}_2'^{(j)}}$ to model the position-dependent performance of each camera.

- In the calculation of $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, the depth of interaction is assumed normally distributed with standard deviation σ_Z and mean $Z_i^{(\mathrm{middle})}$. In Figure 5.30, we argued that the distribution of the depth of interaction actually follows a truncated exponential distribution which, for our 5 mm NaI(Tl) crystals, is well approximated by a uniform distribution over the crystal thickness. For this reason, the expressions in (5.11) assume a uniform distribution of the depths of interaction $Z_1^{(j)}$ and $Z_2^{(j)}$.
- Factors $\frac{1}{|\boldsymbol{R}_1^{(j)} - \boldsymbol{r}_n|^2}$ and $\frac{1}{|\boldsymbol{R}_2^{(j)} - \boldsymbol{r}_n|^2}$ in (5.11) model the solid angles at $\boldsymbol{R}_1^{(j)}$ and $\boldsymbol{R}_2^{(j)}$ subtended by the projection of voxel n along the line of response $\ell(\boldsymbol{R}_1^{(j)}, \boldsymbol{R}_2^{(j)})$. No solid-angle calculation enters in the derivation of $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$.

Lastly, we considered reconstructions obtained using the LMMLEM algorithm in (5.13) with $P = 1$ and we applied it to the list-mode data $\hat{\mathscr{A}}^{(\checkmark)}$ that were defined in (5.14). For our reconstructions, we first used $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ as defined by (5.8) and (5.11). Then we ran the same reconstruction algorithm on the same data but we used $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ instead. This latter case actually included two subcases, one in which we assumed $\sigma_Z = 1$ mm, and the other in which we assumed $\sigma_Z = 2$ mm. Other parameters used in (5.17) that were found in [6] to give a good compromise between reconstruction speed and resolution, were as follows: $\sigma_X = \sigma_Y = 1$ mm and $M = 15$.

We will not carry out a formal image quality study here to compare reconstructions obtained using different expressions for the calculation of $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ or $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$. Instead, we will limit ourselves to a simple visual inspection of the ^{18}F-NaF bone scan reconstructions reported in Figure 5.45. From a close look at Figure 5.45, we conclude that, by using $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$, more detailed reconstructions are obtained.

For example, the pelvic bone, the coccyx (tailbone), and some vertebrae are easily recognized in the top image of the second column of the arrangement of Figure 5.45.



(a) Using $\mathrm{pr}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$

(b) Using $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ with $\sigma_Z = 1$ mm

(c) Using $\mathrm{pr}^{(\mathrm{MC})}(\hat{\boldsymbol{A}}'^{(j)} \mid n)$ with $\sigma_Z = 2$ mm

Figure 5.45. LMMLEM reconstructions after iteration 10

5.9 Random-Number Generation on a GPU Device

In § 4.6.3 we discussed our Markov-chain Monte Carlo (MCMC) algorithm for the investigation of the ideal observer's task performance for the detection of a known signal buried in random lumpy background. The algorithm was developed in CUDA and run on a GPU device. As our algorithm makes use of random numbers distributed according to a few probability distributions, GPU code for the generation of random numbers was developed. In this section, we provide an introduction to the topic of random-number generation, and we discuss the rationale and details that went into the development of our GPU random-number generation routines.

With the term *random-number generator*, we refer to any hardware device or computer algorithm able to generate a sequence of numbers that "appear random." Random-number generation finds applications in gambling, statistical analysis (such as Markov chain Monte Carlo), cryptography, computer simulations, and randomized algorithms.

An example of a device able to generate random numbers could be a photon-counting device (such as a photodiode) counting the number of photons emitted by a constant source during a predetermined time interval [371]: the laws of quantum mechanics predict that the number of photons counted follows a particular distribution. Other physical phenomena often used for the generation of random numbers include nuclear decay, thermal noise, avalanche noise, and atmospheric noise. All these process are, in theory, completely unpredictable and so well suited for the generation of random numbers. Hardware random-number generators are available on the market [372, 373], usually in the form of computer cards that can be plugged inside a computer. Their major application is in the field of high-security data encryption.

Methods for the generation of random numbers via a computer algorithm [374,375] uses the value of a "state" variable—called *seed*—and two functions: one to calculate a number from the current state and the other to update the state by generating a new one. Mathematically, we can denote the initial state as S_1 and we can introduce the function $r(S_n)$ to calculate a "random" number, say x_n, from S_n, and the function $u(S_n)$ to calculate the next state S_{n+1}. In some random-number generator algorithms, the state S_n and the random number x_n coincide, making generation of random numbers simpler and faster.

Contrary to the case of hardware devices for the generation of random numbers, the scheme just presented is completely deterministic: given S_n, the values of x_n and

S_{n+1} are deterministically computed via the functions $r(S_n)$ and $u(S_n)$, respectively. However, by carefully constructing the functions $r(S_n)$ and $u(S_n)$ it is possible to obtain sequences $x_1, x_2, \ldots$ that pass many rigorous statistical tests and lack any obvious pattern. In other words, the sequence $x_1, x_2, \ldots$ "looks random." Often, the development and analysis of a computer algorithm for the generation of random numbers require advance knowledge of abstract algebra and number theory. Because of their deterministic nature, the random-number generators just described are more appropriately termed as *pseudorandom-number generators*. In the reminder of this section we will focus on this type of random-number generators.

Historically, most of the pseudorandom-number generators that have been used belong to the class of linear congruential generators [376,377]:

$$x_{n+1} = S_{n+1} = (aS_n + c) \mod m,$$

where m $(0 < m)$ is an integer number called *modulus*, a $(0 < a < m)$ is called *multiplier*, c $(0 \leqslant c < m)$ is called *increment*, and, finally, with the "$(aS_n + c)$ mod m" notation we mean the remainder of the integer division between $aS_n + c$ and m. It is important to note that the sequence $x_1, x_2, \ldots$ is necessary periodic: there exists an integer number $k > 0$, called *period*, such that $x_{n+k} = x_n$ for all positive values of n. This is an immediate consequence of the fact that the state S_n can only take on values in $\{0, \ldots, m-1\}$ and, as we keep generating random numbers, a previous value of the state must sooner or later reappear.

In addition to the mathematical expression shown above, the definition of the pseudorandom-number generator includes the values of a, c, and m. The choice of these parameters is critical in the design of the pseudorandom-number generator: poorly selected values will results in embarrassedly non-random numbers and/or very short period. In [376], some theoretical results and algorithms are presented. In particular, the authors advocate the following choices for a, c, and m:

$$a = 7^5 = 16807, \qquad c = 0, \qquad m = 2^{31} - 1 = 2147483647.$$

It is shown in [376] that, because $a = 7^5$ is a primitive root of m, the period k of the pseudorandom-number generator above is $m = 2^{31} - 1$. Besides the values reported above, many other values have been used or proposed [378].

Despite their popularity and extensive theoretical analysis, linear congruential pseudorandom-number generators have recently started to show their shortcomings. The relative short period they allow has become problematic when a long sequence of good-quality random numbers is needed. Even more problematic is the high correlation between values in the sequence $x_1, x_2, \ldots$. For example, if a linear congruential generator is used to generate points uniformly spaced in an n-dimensional space, then such points will lie on, at most, $m^{1/n}$ parallel hyperplanes [379]. An example of this phenomenon is depicted in Figure 5.46 for the case $n = 3$.

For this reason linear congruential pseudorandom-number generators have fallen into disuse in favor of pseudorandom-number generator that are implementations of higher-performance methods. In parallel with algorithms, validations tools have been developed [380, 381]. Just to give the reader an idea of what these tools look like, we cite short descriptions of some of the randomness tests developed in [381]:

"Overlapping permutations." *Analyze sequences of five consecutive random numbers. The 120 possible orderings should occur with statistically equal probability.*

"Parking lot test." *Randomly place unit circles in a 100 × 100 square. If the circle overlaps an existing one, try again. After 12000 tries, the number of successfully "parked" circles should follow a certain normal distribution.*

"Minimum distance test." *Randomly place 8000 points in a 10000 × 10000 square, then find the minimum distance between the pairs. The square of this distance should be exponentially distributed with a certain mean.*

Popular algorithms in use today for the generation of pseudorandom numbers include "Mersenne twister" [382], "multiply-with-carry" and their lagged variants [383, 384],

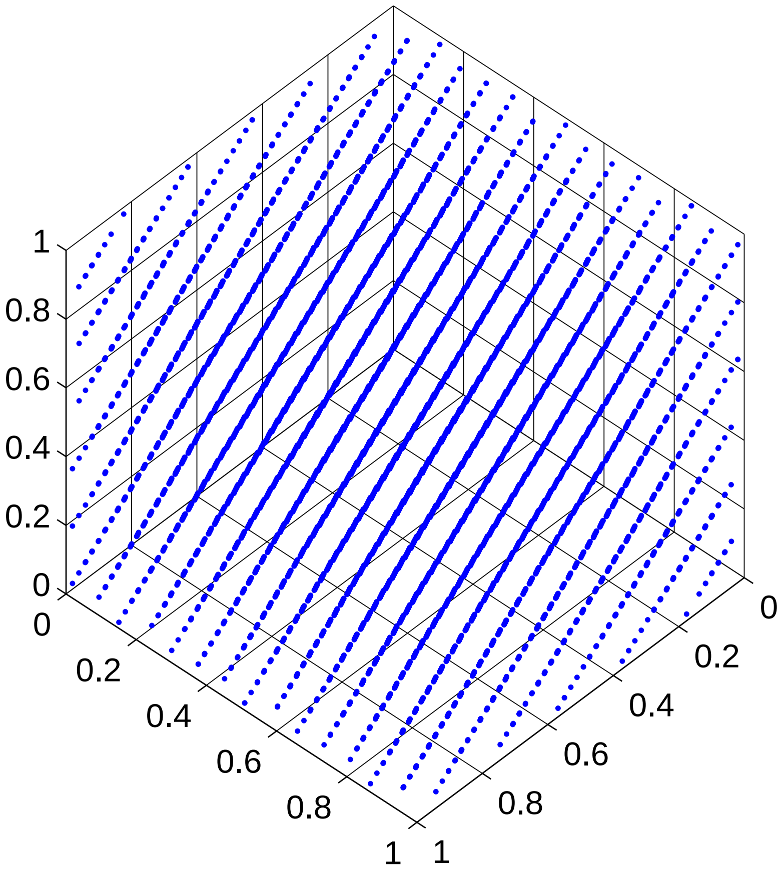


FIGURE 5.46. Scatter plot showing the clustering along hyperplanes for points generated using a linear congruential pseudorandom-number generator with $a = 884$, $c = 0$, and $m = 8191$

and “Xorshift” [385, 386].

Recall that in a typical GPU application, the computation is arranged in a hierarchy of threads, each one executing a fragment of code usually referred to as a kernel. The GPU device is able to access different memory spaces, such as the global memory and the shared memory. A GPU code that necessitate the generation of pseudorandom numbers might have to do so inside a kernel, hence the need for a GPU pseudorandom-number generator library.

GPU implementation of some of the pseudorandom-number generator algorithms described above is often problematic or—from a theoretical point of view—impossible. Ideally, we would like to ensure two properties:

1. the algorithm must be fast and use a limited amount of resources on the device;
2. there has to be no correlation between the random numbers generated by different threads.

The first requirement is easy to understand. Its implications are that the state of the pseudorandom-number generator must be small enough that a few GPU registers or bytes of shared memory are enough to memorize it. Furthermore, the pseudorandom-number generator algorithm should avoid lengthy calculations that require many clock cycles to execute. The second requirement implies that the stream of random numbers calculated by one threads should not be a “lagged” copy of the stream of random numbers calculated by another thread. This requirement will not be satisfied if, for example, the same algorithm but with different initial states is used by different threads running in parallel. If this were the case, one stream of random numbers calculated by one thread will be a lagged copy of the stream of random numbers calculated by another thread. The lag might be very large, but it is nonetheless finite.

GPU implementations of some pseudorandom-number generator algorithms have appeared in the literature. Among them, [387] proposes a fast implementation of the linear congruential pseudorandom-number generator of [376]. GPU implementation of other algorithms have appeared in [388–393], and a general discussion on portability issues is presented in [394]. Although one of these algorithms—namely the Mersenne Twister pseudorandom-number generator [382,391]—is often the algorithm of choice in many applications, its implementation on a GPU device is problematic: the algorithm is rather complex and storing the state of the generator takes up 624 32-bit integer number, for a total of 2496 bytes. Storing this large state in the limited shared memory is usually a bad idea, whereas accessing and manipulating 2496 bytes of global memory each time a random number is needed will greatly reduce performances. For these reasons, we decided not to take advantage of the Mersenne Twister algorithm implementation of [391], instead, we coded our algorithm.

A comparison of different implementations on a GPU device of pseudorandom-number generator algorithms has appeared in [395]. Based on these results, we chose the lagged multiply-with-carry (MWC) algorithm introduced in [383,384] as the basic algorithm for our GPU pseudorandom-number generator. We will provide now the basic details of the algorithm. A lag-ℓ multiply-with-carry pseudorandom-number generator requires a *multiplier* a, a *base* b and a state S_n of $\ell + 1$ values: $S_n = \{x_n, \dots, x_{n-\ell+1}, c_n\}$. The definitions of $r(S_n)$ and $u(S_n)$ dictate how a random number is generated and how the state is updated. More specifically,

$$x_{n+1} = r(S_n) = (ax_{n-r+1} + c_n) \mod b,$$

and

$$S_{n+1} = u(S_n) = \{x_{n+1}, \dots, x_{n-\ell+2}, c_{n+1}\},$$

where

$$c_{n+1} = \left\lfloor \frac{ax_{n-\ell+1} + c_n}{b} \right\rfloor,$$

and $\lfloor \dots \rfloor$ denotes rounding towards 0. The initialization of seed requires picking up a value for c_1 such that $0 \leqslant c_1 < a$ and values for $x_1, \dots, x_{-\ell}$ satisfying $0 \leqslant$

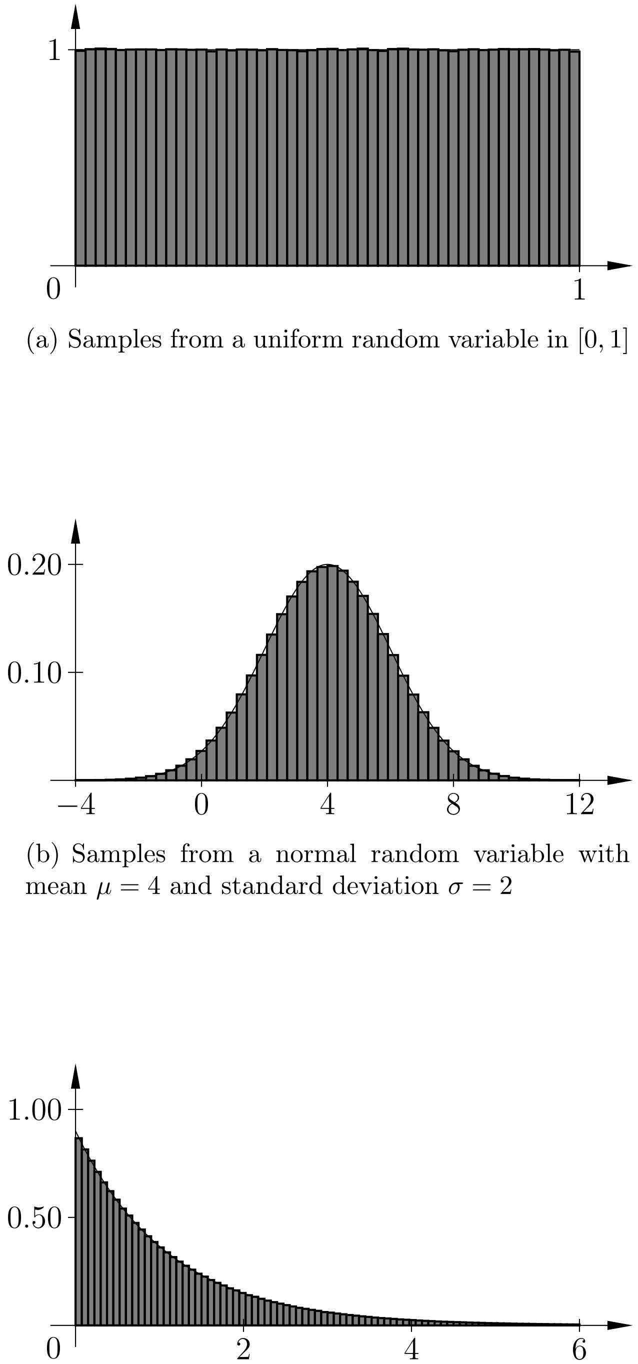
$x_1, \ldots, x_{-\ell} < b$. Calculation of the period of the pseudorandom-number generator requires some knowledge of group theory. Without going too much into details, it suffices to say that the period of the random-number generator is given by the smallest positive integer k such that

$$b^k = 1 \mod ab^\ell - 1.$$

It can be shown that the largest period is obtain when $ab^\ell - 1$ is a *safe prime*.[1] When $ab^\ell - 1$ is a safe prime, the period of the random-number generator is $ab^\ell/2 - 1$. As an example, assume $\ell = 2$. Then, if the base is $b = 2^{16} = 65536$, the largest 16-bit multiplier such that $ab^\ell - 1$ is a safe prime is $a = 65274$. The period of the random-number generator will therefore be $ab^\ell/2 - 1 = 140174847639551 \approx 1.40 \cdot 10^{14}$, which is usually adequate in many applications.

The small size of the state $S_n = \{x_n, \ldots, x_{n-\ell+1}, c_n\}$ of a multiply-with-carry pseudorandom-number generator satisfies our need for a simple and fast GPU algorithm. To obtain the other requirement—namely, absence of correlations between sequences of random numbers—different threads can use different values for the multiplier a. Effectively, this makes the pseudorandom-number generators running on different threads different. Furthermore and as suggested in [396], we have combined two independent random-number generators to improve randomness and increase the period of the overall random-number generator. This results in an efficient algorithm that produces 32-bit integer numbers uniformly distributed in the interval $[0, 2^{32} - 1]$. From this, other distributions (including the Poisson distribution [397]) can be easily obtained. For example, floating-point random numbers uniformly distributed in the interval $[0, 1]$ are obtained by normalization with the constant $2^{31} - 1$, as shown in Figure 5.47a. Normally distributed random numbers (see Figure 5.47b) can be

[1] A prime number of the form $2p + 1$ is said to be a *safe prime* if p is also a prime number. Examples of safe primes are 5, 7, 11, 23, 47, 59, 83, 107, and so on.



(a) Samples from a uniform random variable in $[0, 1]$

(b) Samples from a normal random variable with mean $\mu = 4$ and standard deviation $\sigma = 2$

(c) Samples from an exponential random variable with parameter $\lambda = 0.90$

FIGURE 5.47. Example of different distributions obtained from the multiply-with-carry pseudorandom-number generator algorithm

generated using [398]:

$$X_1 = (-2\ln U_1)^{1/2}\cos(2\pi U_2),$$
$$X_2 = (-2\ln U_1)^{1/2}\sin(2\pi U_2),$$

where U_1 and U_2 are two realizations of a random variable with uniform distribution over $(0,1)$. Similarly, if U is a random variable with uniform distribution over $(0,1)$ and $\lambda > 0$ is a real number, then:

$$Y = -\frac{\ln U}{\lambda}$$

follows an exponential distribution with parameter λ, as shown in Figure 5.47c. Methods for the generation of samples drawn from other common probability distributions are available as well [399].

We have not performed extensive statistical analysis of the random numbers generated using the methods we briefly mentioned above. We have simply limited ourselves to comparing values of sample mean, sample variance, sample skewness, and sample kurtosis to the corresponding theoretical values. Given a random variable X, skewness and kurtosis are defined as [128]

$$\gamma_1 = \frac{\big\langle (X - \langle X\rangle)^3 \big\rangle}{\big[\big\langle (X - \langle X\rangle)^2 \big\rangle\big]^{3/2}}, \qquad \gamma_2 = \frac{\big\langle (X - \langle X\rangle)^4 \big\rangle}{\big[\big\langle (X - \langle X\rangle)^2 \big\rangle\big]^{2}},$$

respectively.

Table 5.5 reports our results, which were obtained using $5 \cdot 10^6$ samples. Our results reveal good agreement with theoretical values.

		Uniform	**Normal**	**Exponential**
		in $[0,1]$	$(\boldsymbol{\mu = 4,\ \sigma = 2})$	$(\boldsymbol{\lambda = 0.90})$
Mean	Theory	0.5000	4.0000	1.1111
	Sample	0.4999	4.0003	1.1081
Variance	Theory	0.0833	4.0000	1.2346
	Sample	0.0830	3.9776	1.2260
Skewness	Theory	0.0000	0.0000	2.0000
	Sample	−0.0001	−0.0010	2.0327
Kurtosis	Theory	−1.2000	0.0000	6.0000
	Sample	−1.1960	0.0117	6.0445

Table 5.5. Comparison between theoretical values and values estimated from samples for some continuous probability distributions

CHAPTER 6
Conclusions

This chapter begins by providing a concise summary of the main results obtained and work carried out in this dissertation. The chapter continues with a detailed possible application of list-mode data to X-ray digital mammography. We will argue that, a few years from now, CMOS cameras and computing hardware will have become fast enough to make list-mode X-ray digital mammography feasible. The system we propose can be understood as a proof-of-concept for list-mode X-ray imaging, with the long-term intent to apply the same ideas and techniques to other X-ray imaging systems, mainly computed tomography (CT). The chapter concludes by providing a list of topics strictly related to those developed in this dissertation. Each item in this list has to be understood as the main topic of a journal paper.

6.1 Summary

In this dissertation, we have investigated the use of list-mode data for detection, estimation, and image reconstruction problems for small-animal emission tomography imaging.

Our treatment of list-mode data was preceded by a discussion of maximum-likelihood (ML) estimation, and we presented two classes of algorithms that can be used to solve maximum-likelihood estimation problems efficiently. These two classes were contracting-grid-type algorithms (which work extremely well when just a few numerical parameters characterize the statistics of the observed noisy data) and maximum-likelihood expectation-maximization (MLEM) algorithms (which are iterative and might require many iterations to converge, but are preferred when more than a few parameters are to be estimated). Performance of estimation algorithms is analyzed via the Fisher information matrix.

After introducing list-mode data, we discussed their advantages over binned data and we showed how list-mode data fit in the contest of parameter estimation. Two topics of interest were considered: estimation of the 2D or 3D position of interaction from photomultiplier (PMT) data via a contracting-grid algorithm, and list-mode MLEM image reconstruction. Our investigation of the benefits of list-mode data continued with the study of detection of a known signal buried in random lumpy background. We first applied the optimal linear observer (known in the literature as the Hotelling observer) directly to list-mode data. By theorizing a simple imaging system, we were able to calculate analytically a figure of merit for our detection task. Using this figure of merit for detection performance, a surprising and unexpected result was obtained: if the location of the signal is known, detecting a signal whose width is smaller than the width of the background lumps is easier than detecting a signal whose width is larger than the width of the background lumps. The same identical study was then repeated for the case of the ideal observer. As expected, the ideal observer outperformed the Hotelling observer. The interesting result, however, was that the performance gap between the Hotelling observer and the ideal observer kept increasing as the mean number of detected counts increased. This behavior is in sharp contrast to what one would find for the case of binned data. Indeed, if the mean number of detected counts is large enough, the statistics of binned data are approximatively normally distributed and the ideal observer reduces to the Hotelling observer. Another interesting result was that the performance of the ideal observer was found to be asymptotically proportional to the mean number of detected counts.

Besides presenting theoretical results for list-mode data, we considered practical aspects of their efficient parallel processing. In this dissertation, we considered two lost-cost parallel computing platforms: Sony's PlayStation 3 (equipped with the Cell Broadband Engine chip) and NVIDIA's graphics processing unit (GPU) architecture. Due to their scalability, programming convenience, and constant performance improvement, GPU devices are revolutionizing the world of scientific computing. It is our expectation that, thanks to constant computing hardware improvement, the application of list-mode data to medical imaging will find a growing number of applications. For this reason, this dissertation discussed the implementation of accurate models for list-mode data processing on a GPU device. In concrete terms, this led us to develop maximum-likelihood GPU code for volumetric reconstruction from raw PMT data. The reconstruction algorithm was applied to ^{18}F-NaF bone scan list-mode data of a normal mouse acquired with the ModPET imaging system. Using these data, we compared different list-mode algorithms, including an ordered-subset list-mode variant of the MLEM algorithm that uses a relaxation parameter to improve reconstruction speed.

6.2 Possible Application

In this dissertation, we provided a general theoretical framework for the analysis and processing of list-mode data and we applied it to the case of emission tomography imaging. Based on the promising—and, in some cases, surprising—results we obtained, we propose to use the same formalism for a completely different imaging modality, namely X-ray imaging. This idea has been formalized in response to the 2012 NIH Director's Early Independence Award (EIA), colloquially known as "skip your postdoc." In this section, we report part of the application that was submitted at the end of January 2012. If funded, this 5-year project will start in September/October 2012. The author would like to thank Dr. Harrison H. Barrett, Dr. Lars R. Furenlid, Dr. Matthew A. Kupinski, Dr. Diego R. Martin, and Dr. Thomas L. Koch for the support and help throughout the application process.

It is reasonable to assume that the results we briefly summarized in the previous section are inherent to list-mode data, and the project we propose represents the first time anyone has attempted to build a system that embodies them. We will use the proposed imaging system to introduce and demonstrate the concept of "ideal dose utilizer" (IDU) as the extension of the ideal observer to list-mode data, optimally calculated via maximum-likelihood methods.

In this research project, we propose to apply the theoretical framework developed in this dissertation, to X-ray digital mammography and breast tomosynthesis. By building a novel imaging system and by designing data-processing methods that preserve and take full advantage of all the information stored as the list of parameters estimated for each X-ray photon, we expect to increase the accuracy of digital mammography and breast tomosynthesis and, at the same time, reduce the radiation dose delivered to the patient. Second only to lung cancer as the cause of death from a malignant disease, breast cancer is the most common and feared cancer among women in the United States [400]. As the incidence of breast cancer is increasing, accurate low-dose methods and imaging systems for early detection and diagnosis are expected to have a substantial impact on the treatment of such disease.

In Chapter 4, we started from the characterization of the object as a continuous function—namely, the probability density function of the location of production of gamma-ray photons—and we analyzed the statistical properties of the imaging system and gamma-ray cameras to derive an exact expression that characterizes the list-mode data collected by the hardware. Within the same framework, we derived expressions for figures of merit for two different observers: the ideal and the optimal-linear observers. The novelty of this general approach is represented by the fact that these observers operated directly on list-mode data, as opposed to the more common binned data. Our analytical calculations, which were later confirmed by simulation results, have provided valuable insights on the tremendous benefits—numerically quantified in terms of observer's figure of merit—that list-mode data allow.

Our theoretical framework applies equally well to image reconstruction, signal detection, and parameter quantitation. As such, the research we propose includes the development of computer-aided detection (CADe) and computer-aided diagnosis (CADx) techniques for digital mammography. Conventional CADe/CADx methods typically use *ad hoc* procedures for the extraction and classification of features from an image, without fully taking into account the statistical properties of the image itself. On the other hand, our formalism allows us to apply the methods of statistical decision theory to list-mode data for X-ray digital mammography, hence letting us to analyze and prove the optimality of our methods. By designing a new imaging system and by developing optimal algorithms for signal detection and parameter quantitation from list-mode data, we will build a proof-of-concept CADe/CADx system that will provide clinicians additional information in the form of a “second opinion.”

Recent advantages in CMOS technology are starting to provide cameras fast enough that X-ray list-mode data collection is becoming feasible. Currently high-end scientific cameras can read 10^9 or even 10^{10} pixels per second. By using a fluorescent screen, X-ray photons that make through the object produce localized flashes of visible light, which are seen as clusters of bright pixels on the camera’s detector. At high enough frame rates (50,000 frames per second or more), a reasonably low number of clusters of bright pixels will show up in each frame, allowing for discrimination among clusters and calculation of attributes (such as location and deposited energy within the fluorescent screen) about each X-ray photon that gets absorbed.

6.2.1 Approach

The system we propose is meant to demonstrate a new concept, which we have called the “ideal dose utilizer” (IDU). The IDU uses maximum-likelihood methods to estimate list-mode photon attributes, which are then fed to the ideal observer to perform the task of interest. In other words, not only we define optimal methods that fully utilize all the information in the data; we also establish what type of data we have to use (and how we calculate them) so that task performance is optimized and all we can learn by imaging the object is fully used.

A schematic diagram of the system we propose to build is illustrated in Figure 6.1. The setup includes an X-ray tube, a safety shutter, a fluorescent screen, a fiber-optic taper, an image intensifier, and a camera. The setup also includes an aluminum frame, stages, and appropriate lead shielding. Table 6.1 lists expected costs.

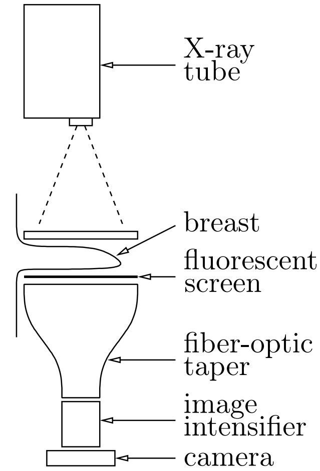


FIGURE 6.1. Setup of the proposed X-ray mammography system

Component	**Cost ($)**
X-ray tube	10,000
Safety shutter	1,775
Thick fluorescent screen	2,500
Fiber-optic taper	22,000
Image intensifier	20,000
Fast CMOS camera	100,000
Miscellaneous items	5,000
TOTAL:	161,275

TABLE 6.1. Summary of the components and their estimated cost

Processing X-ray list-mode data requires the ability to handle large amounts of data very fast. In this dissertation we have taken advantage of the CUDA computational model running on GPU hardware. Thanks to mass production and constant hardware improvements, GPU technology offers an impressive amount of computational power at affordable prices. At the time of writing (January 2012), GPU technology offers the capability of putting together a computing system capable of 100 TFLOPS (i.e., 10^{14} floating-point operations per second) at an estimated cost of less than $40,000.

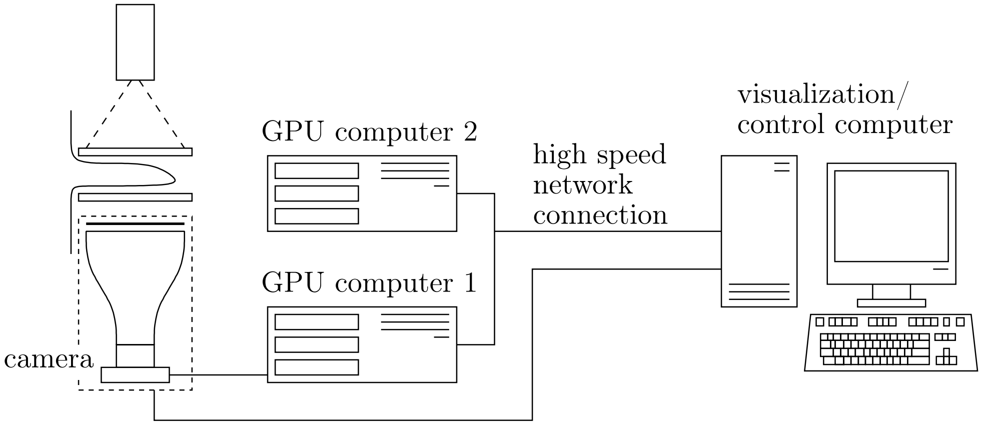


FIGURE 6.2. Diagram of the proposed computing setup

For this research, we are planning to use off-the-shelf components to build two rack-mount GPU supercomputers for the acquisition and processing of the data, as shown in Figure 6.2. Because these machines will be dedicated to processing large amounts of data, no peripherals, with the exception of network connectivity and image acquisition circuitry, will be part of these machines. The setup also includes a third machine, which will provide the input/output interface to the user. No particular GPU capabilities are planned for this machine. Appropriate high-speed network connectivity completes the setup.

As some of the GPU hardware we plan on using has not appeared on the market yet, costs are necessarily approximate. Based on current prices and trends, we estimate a cost of around $9,000 for the computing unit labeled as “GPU computer 1” and a cost of around $30,000 for the unit labeled as “GPU computer 2.” The difference in cost between these two machines reflects the different tasks to which they will be dedicated, which, in turn, influences computing requirements. More specifically, GPU computer 1 will be used to acquire data from the camera and process the frames to extract photon-counting information, while GPU computer 2 will run reconstruction/detection/quantitation codes. Processing camera frames will not necessitates large amounts of GPU memory. Therefore, inexpensive "gaming" GPU hardware can be used to build GPU computer 1. On the other hand, we expect that reconstruction algorithms will benefit from the usage of GPU hardware specifically designed for scientific applications, which usually, comes with much larger memory spaces (currently, up to 6 GB on high-end GPU cards). These types of device are considerably more expensive than gaming GPU hardware.

Flashes of light in the visible range are detected at the output face of the image intensifier by a fast CMOS camera. To reduce the probability of overlapping of flashes of lights corresponding to two or more X-ray photons [401], the camera has to operate at very high frame rates, such as 50,000 frames per second or higher. Suitable cameras able to meet this and other requirements are manufactured by Photron. As an example, Photron's FASTCAM SA1.1 camera is able to operate at 67,500 frames per second at a frame size of 256-by-256 pixels. Unless dedicated circuitry is designed, real-time data transfer of this huge amount of data (of the other of 10^9 pixel intensities per second) to the computer is not possible. For this reason, the camera is equipped with a configurable amount of memory (such as 8 GB or 16 GB or 32 GB). As an example, at 67,5000 frames per second and 256-by-256 pixel array, the camera mentioned above is capable of storing about 350,000 frames during a total collection time of about 5 seconds. It is important to emphasize that, by using maximum-likelihood methods [45, 174, 268], we will attain subpixel resolution for the estimation of location of light flashes, which will depend in a complicated way on the actual number of pixels in the frame. We also remark that a thick fluorescent screen increases the probability that an X-ray photon gets absorbed and produces a measurable flash of light on the camera's detector. However, as the thickness of the fluorescent screen increases, the spatial size of light flashes on the detector increases as well. Once again, simulation studies can help to find the screen thickness that maximizes performance.

6.2.2 Mathematical Model

The mathematical model we need for X-ray imaging is quite different than the one we assumed for emission tomography. Recall that in emission tomography, the object being imaged is injected with a radiotracer, which forms an unknown emission distribution under the patient's metabolism. On the other hand, X-ray imaging is a transmission-based technique in which X-ray photons produced by an external source enter the object at one side, pass through it, and get detected by a detector placed at the opposite side of the object [402], as shown in Figure 6.3.

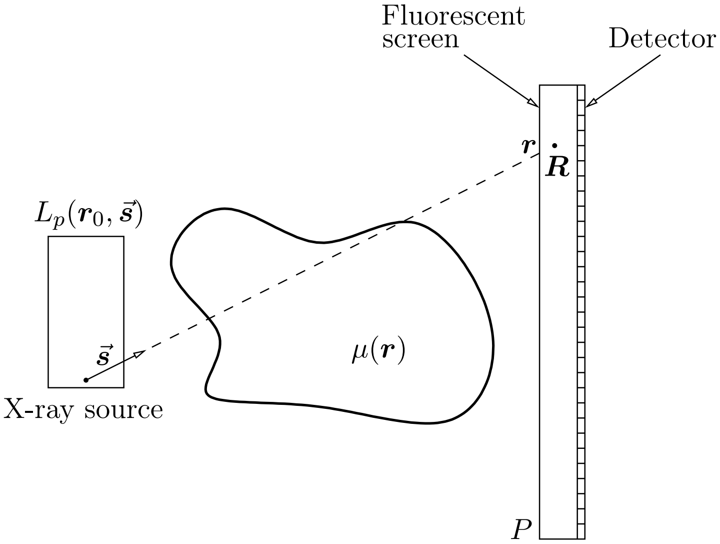


FIGURE 6.3. Mathematical model for X-ray imaging

The image-formation mechanism is hence based on the object's linear attenuation coefficient μ, so that contrast between different tissues and structures in the object arises from different attenuation of X-ray radiation. The object is thus modeled via a spatially-dependent linear attenuation coefficient $\mu(\boldsymbol{r})$, where $\boldsymbol{r}$ denotes a 3D point within the system's field of view. The linear attenuation coefficient $\mu(\boldsymbol{r})$ also depends on the X-ray photon energy, but we will ignore such dependency for simplicity. Similarly to the background-and-signal decomposition $f(\boldsymbol{r}) = b_{\boldsymbol{\theta}}(\boldsymbol{r}) + \Delta f(\boldsymbol{r})$ we postulated in § 4.6.2 and § 4.6.3, we will make the assumption that, when the signal is present, $\mu(\boldsymbol{r}) = \mu_{\boldsymbol{\theta}}(\boldsymbol{r}) + \Delta\mu(\boldsymbol{r})$. As our notation suggests, we will make the assumption that $\Delta\mu(\boldsymbol{r}) \ll \mu_{\boldsymbol{\theta}}(\boldsymbol{r})$. We also assumed that the background part $\mu_{\boldsymbol{\theta}}(\boldsymbol{r})$ of the object is parameterized by a set of parameters, collectively denoted as the vector $\boldsymbol{\theta}$. For the case of X-ray mammography, object models $\mu(\boldsymbol{r})$ that have appeared in the literature include [403–407].

To characterize the image-formation mechanism for the case of the setup in Figure 6.3, we start by characterizing the X-ray source via the photon radiance function [69] $L_p(\boldsymbol{r}_0, \vec{\boldsymbol{s}})$, in which $\boldsymbol{r}_0$ is a 3D point in space and the versor $\vec{\boldsymbol{s}}$ defines a direction of propagation. If $\boldsymbol{r}_0$ is outside the source, then $L_p(\boldsymbol{r}_0, \vec{\boldsymbol{s}}) = 0$ for all $\vec{\boldsymbol{s}}$. The function $L_p(\boldsymbol{r}_0, \vec{\boldsymbol{s}})$ models the number of photons emitted per second per unit area and unit solid angle at point $\boldsymbol{r}_0$ and along direction $\vec{\boldsymbol{s}}$.

Photons that make it through the object interact with a fluorescent screen, which we model via a response function $d^{\mathrm{FS}}(\boldsymbol{r}, \vec{\boldsymbol{s}}; \boldsymbol{R})$, for any point $\boldsymbol{R}$ within the volume defined by the fluorescent screen. By introducing $d^{\mathrm{FS}}(\boldsymbol{r}, \vec{\boldsymbol{s}}; \boldsymbol{R})$, we model how the fluorescent screen's response at point $\boldsymbol{R}$ to X-ray photons that at, at point $\boldsymbol{r}$, are traveling in direction $\vec{\boldsymbol{s}}$. With this framework, the light output $g^{\mathrm{FS}}(\boldsymbol{R})$ inside the fluorescent screen is [69]

$$\begin{aligned} g^{\mathrm{FS}}(\boldsymbol{R}) = C \int_P \int_{2\pi} \int_0^\infty & d^{\mathrm{FS}}(\boldsymbol{r}, \vec{\boldsymbol{s}}; \boldsymbol{R}) L_p(\boldsymbol{r} - \vec{\boldsymbol{s}}\ell, \vec{\boldsymbol{s}}) \times \\ & \times \exp\left[- \int_0^\ell \mu(\boldsymbol{r} - \vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell' \right] \mathrm{d}\ell\,\mathrm{d}\Omega\,\mathrm{d}^3\boldsymbol{r}, \end{aligned} \tag{6.1}$$

in which P is the set of 3D points on the entrance face of the fluorescent screen and C is a constant. From $g^{\mathrm{FS}}(\boldsymbol{R})$, we can calculate the mean pixel intensity $\overline{g}_m^{\mathrm{det}}$ as

$$\overline{g}_m^{\mathrm{det}} = \int_{\mathrm{FS}} d_m^{\mathrm{det}}(\boldsymbol{R}) g^{\mathrm{FS}}(\boldsymbol{R})\,\mathrm{d}^3\boldsymbol{R},$$

in which $d_m^{\mathrm{det}}(\boldsymbol{R})$ is the response function for the m^{th} pixel of the detector and point $\boldsymbol{R}$ in the fluorescent screen. It is interesting to note that by virtue of $\mu(\boldsymbol{r}) = \mu_{\boldsymbol{\theta}}(\boldsymbol{r}) +$

$\Delta\mu(\boldsymbol{r})$ and $\Delta\mu(\boldsymbol{r}) \ll \mu_{\boldsymbol{\theta}}(\boldsymbol{r})$, we can write

$$\exp\left[-\int_0^\ell \Delta\mu(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \approx 1 - \int_0^\ell \Delta\mu(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell',$$

so that (6.1) can be rewritten as

$$\begin{aligned} g^{\mathrm{FS}}(\boldsymbol{R}) \approx C \int_P \int_{2\pi} \int_0^\infty & d^{\mathrm{FS}}(\boldsymbol{r},\vec{\boldsymbol{s}};\boldsymbol{R}) L_p(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell,\vec{\boldsymbol{s}}) \times \\ & \times \exp\left[-\int_0^\ell \mu_{\boldsymbol{\theta}}(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \mathrm{d}\ell\,\mathrm{d}\Omega\,\mathrm{d}^3\boldsymbol{r} + \\ - C \int_P \int_{2\pi} \int_0^\infty & d^{\mathrm{FS}}(\boldsymbol{r},\vec{\boldsymbol{s}};\boldsymbol{R}) L_p(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell,\vec{\boldsymbol{s}}) \times \\ & \times \exp\left[-\int_0^\ell \mu_{\boldsymbol{\theta}}(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \left[\int_0^\ell \Delta\mu(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \mathrm{d}\ell\,\mathrm{d}\Omega\,\mathrm{d}^3\boldsymbol{r}, \end{aligned}$$

which gives

$$\begin{aligned} \overline{g}_m^{\mathrm{det}} \approx C \int_{\mathrm{FS}} d_m^{\mathrm{det}}(\boldsymbol{R}) \int_P \int_{2\pi} \int_0^\infty & d^{\mathrm{FS}}(\boldsymbol{r},\vec{\boldsymbol{s}};\boldsymbol{R}) L_p(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell,\vec{\boldsymbol{s}}) \times \\ & \times \exp\left[-\int_0^\ell \mu_{\boldsymbol{\theta}}(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \mathrm{d}\ell\,\mathrm{d}\Omega\,\mathrm{d}^3\boldsymbol{r}\,\mathrm{d}^3\boldsymbol{R} + \\ - C \int_{\mathrm{FS}} d_m^{\mathrm{det}}(\boldsymbol{R}) \int_P \int_{2\pi} \int_0^\infty & d^{\mathrm{FS}}(\boldsymbol{r},\vec{\boldsymbol{s}};\boldsymbol{R}) L_p(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell,\vec{\boldsymbol{s}}) \times \\ & \times \exp\left[-\int_0^\ell \mu_{\boldsymbol{\theta}}(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \left[\int_0^\ell \Delta\mu(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \mathrm{d}\ell\,\mathrm{d}\Omega\,\mathrm{d}^3\boldsymbol{r}\,\mathrm{d}^3\boldsymbol{R}. \end{aligned}$$

If we define the function:

$$\Delta\tilde{\mu}(\boldsymbol{r},\vec{\boldsymbol{s}},\ell) = \int_0^\ell \Delta\mu(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'$$

and the operator $\mathcal{X}_{m,\boldsymbol{\theta}}$ with integral kernel:

$$\begin{aligned} [\mathcal{X}_{m,\boldsymbol{\theta}}](\boldsymbol{r},\vec{\boldsymbol{s}},\ell) = -C \int_{\mathrm{FS}} & d_m^{\mathrm{det}}(\boldsymbol{R}) d^{\mathrm{FS}}(\boldsymbol{r},\vec{\boldsymbol{s}};\boldsymbol{R}) L_p(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell,\vec{\boldsymbol{s}}) \times \\ & \times \exp\left[-\int_0^\ell \mu_{\boldsymbol{\theta}}(\boldsymbol{r}-\vec{\boldsymbol{s}}\ell')\,\mathrm{d}\ell'\right] \mathrm{d}^3\boldsymbol{R}, \end{aligned}$$

then we can write

$$\overline{g}_m^{\mathrm{det}} \approx \mathcal{X}_{m,\boldsymbol{\theta}} \Delta\tilde{\boldsymbol{\mu}} - \mathcal{X}_{m,\boldsymbol{\theta}} \mathbf{1}_{P,2\pi,(0,\infty)},$$

in which the function $\mathbf{1}_{P,2\pi,(0,\infty)}$ is the function that identically evaluates to 1. The expression above allows us to conclude that, with good approximation, $\overline{g}_m^{\text{det}}$ can be written as the sum of two terms: the first one resulting by applying to $\Delta\tilde{\boldsymbol{\mu}}$ the linear "background" operator $\mathcal{X}_{m,\boldsymbol{\theta}}$ (which, in turn, is related in a non-linear way to $\mu_{\boldsymbol{\theta}}(\boldsymbol{r})$ and m) and the second term given by $\mathcal{X}_{m,\boldsymbol{\theta}}$ acting on the identically constant function $\mathbf{1}_{P,2\pi,(0,\infty)}$. Hence, apart from the "background" term $\mathcal{X}_{m,\boldsymbol{\theta}}\mathbf{1}_{P,2\pi,(0,\infty)}$, the expression for $\overline{g}_m^{\text{det}}$ resembles the expression given in (4.23).

6.2.3 Innovation

- **Information Content of a Photon.** We propose a new theoretical framework for the rigorous analysis of the information content of a photon by introducing appropriate figures of merit for detection/quantitation tasks for X-ray digital mammography. By building an actual imaging system featuring list-mode data capabilities, we will show the effectiveness of such approach and we will pave the way for its adaptation to other related imaging problems, most notably, X-ray computed tomography.

- **Hardware for List-Mode X-Ray Imaging.** Collecting and processing X-ray list-mode data for complete information extraction requires tremendously fast data acquisition hardware and massively parallel computing capabilities that are just becoming available. We will design an unprecedented imaging system for the acquisition of list-mode data for X-ray digital mammography and we will show how reasonably-priced mass-produced graphics processing unit (GPU) hardware can provide sufficient computational power to keep up with data acquisition and processing requirements.

- **Accurate Methods for Detection/Quantitation.** Starting from a rigorous characterization of the object, the imaging system, and the list-mode data, we will introduce a number of novel methods for the detection of signals of interest and quantitation of clinically relevant parameters for the screening and monitoring of breast tumors. Contrary to popular *ad hoc* algorithms routinely used in digital mammography, our methods are optimal with respect to statistically significant metrics, such as the probability of correct detection of a tumor for a fixed value of the probability of a false positive (often referred to as "call-back rate").

- **The Ideal Dose Utilizer.** We will introduce the concept of the ideal dose utilizer (IDU) by extending the ideal observer to list-mode data calculated via maximum-likelihood estimation. The IDU paradigm provides an optimal way to use the dose delivered to the patient by careful extraction and use of all the information that can be learned about the object.

- **Real-Time Maximum-Likelihood Tomosynthesis.** List-mode data allow great flexibility in the way in which the data are used for 3D reconstruction. This would include ordered subset reconstruction algorithms, in which portions (or "subsets") of the data are processed for reconstruction while the next few subsets are being collected. Moreover, the imaging system we propose allows for maximum-likelihood (ML) methods be used in cascade for both calculation of X-ray attributes and list-mode reconstruction, resulting in what we call "double ML" processing. Within this new concept, desirable statistical properties of maximum-likelihood estimates are preserved and they positively contribute to task performance.

- **Low-Dose CADe/CADx Schemes.** The computerized methods for detection/diagnosis we have discussed in the proposal do come with an enormous benefit for the patient: since they are able to make an optimal use of the data, they attain the same level of performance of existing methods but for much lower radiation dose delivered to the patient. Equivalently, our novel methods are more accurate than existing ones for the same radiation dose.

- **List-Mode Data vs. Binned Data.** By software integration of frames acquired by the camera, our system can also collect digitized mammographic images, thus mimicking conventional systems. Hence, the imaging apparatus we propose offers the unique opportunity to run comparative studies between list-mode methods and conventional methods based on projection data, further shedding some light on the advantages of list-mode techniques in the field of medical imaging.

- **Adaptive List-Mode X-Ray Imaging.** Even though not specifically included in the research proposal, the system we will design can be conveniently and effortlessly augmented with adaptive imaging capabilities. For example, by translating the X-ray source and/or by placing a tungsten beam shaper right after the X-ray tube, different parts of the object can be imaged at different magnifications. In conjunction with fast algorithms for volume reconstruction and accurate methods for CAD diagnosis, an adaptive imaging system will enhance diagnostic capabilities by reducing screening time and call-back rate and making an optimal use of the dose delivered to the patient.

6.3 Future Investigations

In this section, we itemize some possible future work for which the concepts and theory developed in this dissertation provide a starting point. Each item listed below has to be understood as a concise summary for a potential journal paper.

- **Parameter Estimation from List-Mode Data.** In this dissertation, we mainly considered detection problems. Far more interesting are, however, estimation problems [124, 408, 409], in which we want to extract numerical information from the data. This dissertation discussed how PMT data can be used to perform estimation of the 2D or 3D location of interaction of a gamma-ray photon within a scintillation crystal. We can, of course, consider the estimation of any parameter $\boldsymbol{\theta}$ (one example relevant to the field of radiology would be the estimation of volume and location of a tumoral mass [410]) and we can use list-mode data $\hat{\mathscr{A}}$ to perform the estimation. Task performance will now be measured via the Fisher information matrix associated to the parameter $\boldsymbol{\theta}$ being estimated. For example, we could use the same idealized imaging system of Figure 4.12 and use "scanning" versions [107, 410] of the Hotelling and ideal observers for the estimation of the location of a signal. Some work on similar problems has appeared in [411, 412]. It would be interesting to consider plots similar to those of Figure 4.24, but when the location of the signal is unknown and it needs to be estimated if the observer concludes that the signal is present. It seems reasonable to expect that, if the signal is of the same shape and size as the background lumps, any observer will perform poorly, as the signal would be indistinguishable from any of the background lumps.

- **Comparison with Binned Data.** One of the advantages of list-mode data is that they allow a more accurate data representation if compared to binned data. Intuitively, this translates into higher performance for the task of interest. For example, we could consider again the same setup of Figure 4.12, along with the same SKE/BKS detection problem. This time, however, the observer would operate on binned data as opposed to list-mode data, so that the binned data observer performance can be compared to list-mode data observer performance for the same "class" of observers (for example, Hotelling observer in both cases). Depending on the mathematical form of the observers, simulation might be needed to calculate performance.

- **List-Mode Channelized Observers.** In § 2.5 we introduced the channelized Hotelling observer (CHO) and we mentioned that one way to reduce the size of the data covariance matrix is by passing the data $\boldsymbol{g}$ through a bank of K channels $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, as shown in Figure 6.4 for the case of binned data $\boldsymbol{g}$. The same concept is applicable to list-mode data, in which, given list-mode data $\hat{\mathscr{A}}$, the channels are implemented as

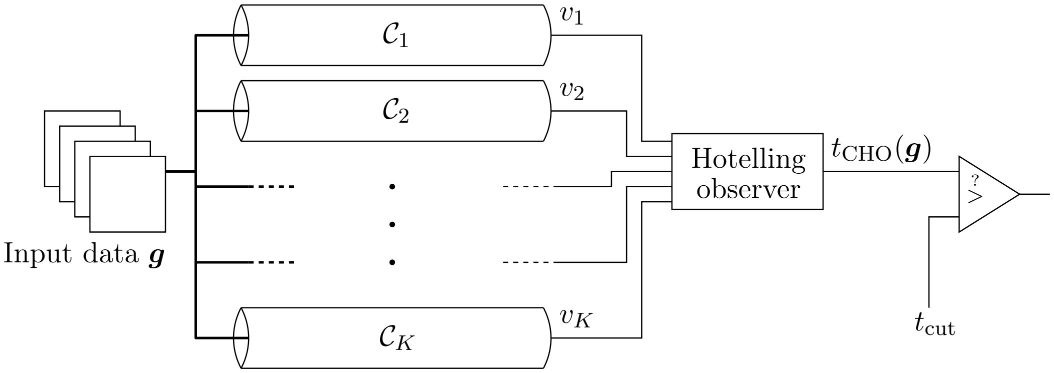


FIGURE 6.4. Diagram of channelized Hotelling observer (adapted from [7])

$$\mathcal{C}_k(\hat{\mathscr{A}}) = \int_{\infty} \boldsymbol{c}_k(\hat{\boldsymbol{A}}) u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) \,\mathrm{d}\hat{\boldsymbol{A}},$$

for some template channel function $\boldsymbol{c}_k(\hat{\boldsymbol{A}})$ and $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ defined as in (4.19). One topic of interest could be comparing the binned-data channelized Hotelling observer with the list-mode channelized Hotelling observer. Similarly, we can compare the list-mode channelized Hotelling observer with its non-channelized counterpart.

- **List-Mode Data vs. Reconstructed Data: Task Performance.** In this work, we used list-mode data directly to perform detection. We also used list-mode data to perform reconstruction via the LMMLEM (or the OSLMMLEM) algorithm and calculate an estimate $\hat{\boldsymbol{f}}$ of the real object $\boldsymbol{f}$. The reconstructed object $\hat{\boldsymbol{f}}$ is usually presented to an observer to make a decision. In both cases, performance are assessed via an appropriate figure of merit (such as the AUC for a detection problem). Hence, we could use the same figure of merit to assess task performance for two different scenarios: in one case, the observer operates directly on the list-mode data $\hat{\mathscr{A}}$; in the other case, list-mode data $\hat{\mathscr{A}}$ are first used to calculate estimate $\hat{\boldsymbol{f}}$ via the LMMLEM algorithm (or any other reconstruction algorithm) and the actual observer is fed $\hat{\boldsymbol{f}}$ (as opposed to $\hat{\mathscr{A}}$). It is reasonable to assume that the observer that operates directly on $\hat{\mathscr{A}}$ will not be outperformed by the one that operates on $\hat{\boldsymbol{f}}$, as processing can never increase the "information" present in the data [307].

- **Statistical Properties of LMMLEM-Reconstructed Data.** Because many reconstruction algorithms calculate deterministic (non random) function of the input noisy data, the resulting reconstructed data can be interpreted as realizations of random vectors in an appropriate Hilbert space. The distribution of MLEM-reconstructed data has received some attention in the literature, and it has been found [190, 191] that pixel (or voxel) intensities in the reconstructed data approximatively follow log-normal statistics. With [190] as a starting point and by using the code developed in this dissertation, we propose to carry out the same analysis but for the case of list-mode data and the LMMLEM reconstruction algorithm.

- **OSLMMLEM Reconstruction from Multiple Projections.** This dissertation has just touched the tip of the iceberg for the application of list-mode data in medical imaging. One concrete case we have considered was the application of the OSLMMLEM reconstruction algorithm to ModPET data [6]. The OSLMMLEM algorithm would be particularly suited for an imaging system similar to ModPET but featuring more than two cameras (so that more that one projection through the object would be defined) or also a tomographic version of ModPET, in which cameras are mounted on a rotating gantry and rotated around the object as the data are acquired [361].

- **List-Mode Data: Bounds on Null Functions.** In § 3.4 we briefly mentioned null functions in the context of image reconstruction via the MLEM algorithm. a null function $\boldsymbol{f}_{\text{null}}$ for an object $\boldsymbol{f}$ was defined as any function that when added to $\boldsymbol{f}$, produces the same mean data as $\boldsymbol{f}$ does. Null functions have been studied in the context of binned data [192,193] and relevant results (in the form of bounds on the L^1, L^2, or L^∞ norm of $\boldsymbol{f}_{\text{null}}$) were found for some relevant cases. The same analysis can be carried out for the case of list-mode data. We would expect the bounds for the list-mode data case to be—at least—as tight as the corresponding bounds found for the binned data case, as binned data can be obtained by deterministically processing list-mode data in way that will never increase the “information” contained in the data themselves.

- **Characteristic Functionals.** Recall that in § 4.6.2 we expressed the relationship between the object $f(\boldsymbol{r})$ being imaged and the mean $\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau)$ of the Poisson point process $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ as

  $$\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau) = (\tau \boldsymbol{\mathcal{L}} \boldsymbol{f})(\hat{\boldsymbol{A}}),$$

  in which the linear operator $\boldsymbol{\mathcal{L}}$ played the same role as the operator $\boldsymbol{\mathcal{H}}$ in the usual relationship $\overline{\boldsymbol{g}} = \boldsymbol{\mathcal{H}} \boldsymbol{f}$ we had for the binned data case [69]. Because the object $\boldsymbol{f}$ is a stochastic process, we can calculate its characteristic functional

  $$\Psi_{\boldsymbol{f}}(\boldsymbol{\xi}) = \left\langle e^{-2\pi i \boldsymbol{\xi}^\dagger \boldsymbol{f}} \right\rangle_{\boldsymbol{f}},$$

  in which $\boldsymbol{\xi}$ represents the Fourier conjugate of the function $\boldsymbol{f}$. Using $\Psi_{\boldsymbol{f}}(\boldsymbol{\xi})$, we can study how the imaging system maps this characteristic functional to the characteristic functional of the output noisy data. This calculation has successfully been carried out for $\Psi_{\boldsymbol{g}}(\boldsymbol{\rho})$, which corresponds to the case of binned data $\boldsymbol{g}$ [413]. The same analysis but for the case of list-mode data (i.e., finding $\Psi_{\hat{\mathscr{A}}}(\boldsymbol{\aleph})$ or $\Psi_{u(\,\cdot\,\mid \hat{\mathscr{A}})}(\boldsymbol{\aleph})$ for $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ defined as in (4.19)) would represent a huge step forward in the statistical characterization of list-mode data.

- **List-Mode Backus-Gilbert Reconstruction.** In this dissertation we mainly used the LMMLEM algorithm (or variants of it) to perform reconstruction of list-mode data. A different reconstruction scheme, which was originally introduced to calculate Earth models from a finite set of noisy measurements [414], is the Backus-Gilbert method [415, 416]. We will present below a description of the Backus-Gilbert method that closely resembles the one provided in [417], but we will rewrite it using the notation and language of stochastic processes and linear operators we developed in Chapter 4. Other formulations are possible [69]. We will start by assuming that the estimate $\hat{\boldsymbol{f}}$ and the object $\boldsymbol{f}$ being imaged satisfies the linear relationship:

$$\hat{f}(\boldsymbol{r}) = \int_{\mathrm{FOV}} d(\boldsymbol{r}, \boldsymbol{r}') f(\boldsymbol{r}') \, \mathrm{d}^3\boldsymbol{r}', \tag{6.2}$$

or, more compactly, $\hat{\boldsymbol{f}} = \mathcal{D}\boldsymbol{f}$, where $\mathcal{D}$ is the linear operator with integral kernel $[\mathcal{D}](\boldsymbol{r}, \boldsymbol{r}') = d(\boldsymbol{r}, \boldsymbol{r}')$, for some function $d(\boldsymbol{r}, \boldsymbol{r}')$. Intuitively, "good" estimates $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ will be obtained when $d(\boldsymbol{r}, \boldsymbol{r}') \approx \delta_{\mathrm{Dir}}(\boldsymbol{r} - \boldsymbol{r}')$. Contrary to most of the reconstruction algorithms presented in this work, the estimate $\hat{f}(\boldsymbol{r})$ above is a function of the continuous variable $\boldsymbol{r}$, not a discrete vector of pixel/voxel values over a 2D or 3D grid. As in Chapter 4, we introduce

$$u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) = \sum_{\hat{\boldsymbol{A}}^{(j)} \in \hat{\mathscr{A}}} \delta_{\mathrm{Dir}}\big(\hat{\boldsymbol{A}} - \hat{\boldsymbol{A}}^{(j)}\big),$$

in which $\hat{\mathscr{A}}$ is the noisy list-mode data. By introducing the integral kernel $q(\boldsymbol{r}, \hat{\boldsymbol{A}})$, the Backus-Gilbert method postulates a linear relationship between the stochastic process $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ and the estimate $\hat{\boldsymbol{f}}$:

$$\hat{f}(\boldsymbol{r}) = \int_{\infty} q(\boldsymbol{r}, \hat{\boldsymbol{A}}) u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) \, \mathrm{d}\hat{\boldsymbol{A}}, \tag{6.3}$$

which, in more abstract notation, is rewritten as $\hat{\boldsymbol{f}} = \mathcal{Q}\boldsymbol{u}$. Hence, we can interpret the operator $\mathcal{Q}$ as a "reconstruction operator" because when we apply it to the process $\boldsymbol{u}$ (which is defined in terms of $\hat{\mathscr{A}}$) we get the estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$. The expression above also shows that all we need to calculate $\hat{\boldsymbol{f}}$ from $\boldsymbol{u}$ is the linear (and so far unknown) operator $\mathcal{Q}$. As in Chapter 4, we make use of a linear operator $\mathcal{L}$ to calculate the mean of $u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}})$ with respect to the list-mode data $\hat{\mathscr{A}}$, and for fixed object $\boldsymbol{f}$ and exposure time τ:

$$\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau) = \big(\tau \mathcal{L} \boldsymbol{f}\big)(\hat{\boldsymbol{A}}) = \tau \int_{\mathrm{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}) s(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r}.$$

If we insert the expression for $\overline{u}(\hat{\boldsymbol{A}} \mid \boldsymbol{f}, \tau)$ above into (6.3) and compare it with (6.2), we see that

$$d(\boldsymbol{r}, \boldsymbol{r}') = \tau \int_{\infty} q(\boldsymbol{r}, \hat{\boldsymbol{A}}) \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}') s(\boldsymbol{r}') \, \mathrm{d}\hat{\boldsymbol{A}},$$

which, if we assume $d(\boldsymbol{r}, \boldsymbol{r}') \approx \delta_{\mathrm{Dir}}(\boldsymbol{r} - \boldsymbol{r}')$, can be interpreted as imposing that $\mathcal{QL}$ approximates the identity operator over the set of all object functions. The "goodness" of the reconstruction operator $\mathcal{Q}$ is mathematically quantified as

$$\begin{aligned} A(\boldsymbol{r}) &= \int_{\mathrm{FOV}} |\boldsymbol{r} - \boldsymbol{r}'|^2 \left[d(\boldsymbol{r}, \boldsymbol{r}')\right]^2 \, \mathrm{d}^3\boldsymbol{r}' = \\ &= \iint_{\infty} q(\boldsymbol{r}, \hat{\boldsymbol{A}}) W(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'; \boldsymbol{r}) q(\boldsymbol{r}, \hat{\boldsymbol{A}}') \, \mathrm{d}\hat{\boldsymbol{A}} \, \mathrm{d}\hat{\boldsymbol{A}}', \end{aligned}$$

where

$$W(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'; \boldsymbol{r}) = \tau^2 \int_{\mathrm{FOV}} |\boldsymbol{r} - \boldsymbol{r}'|^2 \, \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}') s(\boldsymbol{r}') \mathrm{pr}(\hat{\boldsymbol{A}}' \mid \boldsymbol{r}') s(\boldsymbol{r}') \, \mathrm{d}^3\boldsymbol{r}'.$$

Because the vector $\boldsymbol{u}$ is a realization of a stochastic process, the estimate $\hat{\boldsymbol{f}} = \mathcal{Q}\boldsymbol{u}$ is a random function. Hence, another condition we can impose on $\mathcal{Q}$ regards the variance of the random variable $\hat{f}(\boldsymbol{r})$ for random $\hat{\mathscr{A}}$ and $\boldsymbol{f}$:

$$B(\boldsymbol{r}) = \iint_{\infty} q(\boldsymbol{r}, \hat{\boldsymbol{A}}) K_{\boldsymbol{u}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') q(\boldsymbol{r}, \hat{\boldsymbol{A}}') \, \mathrm{d}\hat{\boldsymbol{A}} \, \mathrm{d}\hat{\boldsymbol{A}}',$$

where

$$K_{\boldsymbol{u}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}') = \left\langle \left[u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) - \overline{\overline{\boldsymbol{u}}}(\hat{\boldsymbol{A}} \mid \tau)\right] \left[u(\hat{\boldsymbol{A}}' \mid \hat{\mathscr{A}}) - \overline{\overline{\boldsymbol{u}}}(\hat{\boldsymbol{A}}' \mid \tau)\right] \right\rangle_{\hat{\mathscr{A}}, \boldsymbol{f}},$$

and

$$\overline{\overline{\boldsymbol{u}}}(\hat{\boldsymbol{A}} \mid \tau) = \left\langle \left\langle u(\hat{\boldsymbol{A}} \mid \hat{\mathscr{A}}) \right\rangle_{\hat{\mathscr{A}}|\boldsymbol{f}} \right\rangle_{\boldsymbol{f}}.$$

Finally notice

$$\begin{aligned}\int_{\text{FOV}} d(\boldsymbol{r}, \boldsymbol{r}')\, \mathrm{d}^3\boldsymbol{r}' = \int_{\infty} q(\boldsymbol{r}, \hat{\boldsymbol{A}}) \left[\tau \int_{\text{FOV}} \mathrm{pr}(\hat{\boldsymbol{A}} \mid \boldsymbol{r}') s(\boldsymbol{r}')\, \mathrm{d}^3\boldsymbol{r}'\right] \mathrm{d}\hat{\boldsymbol{A}} =\\ = \big(\mathcal{Q}(\tau\mathcal{L}\mathbf{1})\big)(\boldsymbol{r}),\end{aligned}$$

in which $\mathbf{1}$ is the vector that corresponds to the unit function defined on $\boldsymbol{r}$: $1(\boldsymbol{r}) = 1$. In words, $\tau\mathcal{L}\mathbf{1}$ is the vector $\overline{\boldsymbol{u}}$ that corresponds to the mean function $\overline{u}(\hat{\boldsymbol{A}} \mid \mathbf{1}, \tau)$, and the reconstruction operator $\mathcal{Q}$ is then applied to $\overline{\boldsymbol{u}}$, giving estimate $\hat{\mathbf{1}}$. The vector $\hat{\mathbf{1}}$ correspond to the function $\hat{1}(\boldsymbol{r})$ and, for a "good" $\mathcal{Q}$, $\hat{1}(\boldsymbol{r}) = 1$ for all $\boldsymbol{r}$. Functions $A(\boldsymbol{r})$ and $B(\boldsymbol{r})$ measure two different properties of the estimate $\hat{\boldsymbol{f}}$; $A(\boldsymbol{r})$ quantifies how close $\hat{\boldsymbol{f}}$ is to $\boldsymbol{f}$, while $B(\boldsymbol{r})$ quantifies how much $\hat{\boldsymbol{f}}$ is—on average—susceptible to noise in the input data $\hat{\mathscr{A}}$. Notice that quantities $A(\boldsymbol{r})$ and $B(\boldsymbol{r})$ work one against the other. Indeed, if we are looking for an estimate $\hat{f}(\boldsymbol{r})$ that exhibits little susceptibility to noise in $\hat{\mathscr{A}}$ (i.e., small $B(\boldsymbol{r})$), the operator $\mathcal{Q}$ might produce a smooth estimate $\hat{\boldsymbol{f}}$, which, on average, might not be a good approximation to $\boldsymbol{f}$ (i.e., large $A(\boldsymbol{r})$). Vice versa, if we demand a good agreement between $\hat{\boldsymbol{f}}$ and the noisy data $\hat{\mathscr{A}}$ (i.e., small $A(\boldsymbol{r})$), the algorithm will be oversensitive to the noise in $\hat{\mathscr{A}}$, leading to large values for the variance $\sigma^2_{\hat{f}(\boldsymbol{r})}$ of $\hat{f}(\boldsymbol{r})$ at any given point. In seeking a compromise between $A(\boldsymbol{r})$ and $B(\boldsymbol{r})$, we can introduce a parameter λ and define the kernel $q(\boldsymbol{r}, \hat{\boldsymbol{A}})$ of the operator $\mathcal{Q}$ as the function that minimizes $A(\boldsymbol{r})+\lambda B(\boldsymbol{r})$ under the condition $\mathcal{Q}(\tau\mathcal{L}\mathbf{1}) \equiv \hat{\mathbf{1}} = \mathbf{1}$:

$$\begin{cases}\text{minimize: } A(\boldsymbol{r}) + \lambda B(\boldsymbol{r}),\\ \text{subject to: } \big(\mathcal{Q}(\tau\mathcal{L}\mathbf{1})\big)(\boldsymbol{r}) = 1.\end{cases} \tag{6.4}$$

If $\mathcal{S}_{\boldsymbol{r},\lambda}$ is the linear operator with integral kernel

$$[\mathcal{S}_{\boldsymbol{r},\lambda}](\boldsymbol{A}, \boldsymbol{A}') = W(\boldsymbol{A}, \boldsymbol{A}'; \boldsymbol{r}) + \lambda K_{\boldsymbol{u}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'),$$

then the solution [418] to (6.4) is

$$q(\boldsymbol{r}, \hat{\boldsymbol{A}}) = \frac{\displaystyle\int_{\infty} \big[\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\big](\boldsymbol{A}, \boldsymbol{A}')p(\boldsymbol{A}')\,\mathrm{d}\boldsymbol{A}'}{\displaystyle\iint_{\infty} p(\boldsymbol{A})\big[\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\big](\boldsymbol{A}, \boldsymbol{A}')p(\boldsymbol{A}')\,\mathrm{d}\boldsymbol{A}\,\mathrm{d}\boldsymbol{A}'},$$

where $\boldsymbol{p} = \tau\boldsymbol{\mathcal{L}}\boldsymbol{1}$. If we insert the expression above into (6.3), we get

$$\hat{f}(\boldsymbol{r}) = \frac{\displaystyle\iint_{\infty} u(\boldsymbol{A} \mid \hat{\mathscr{A}})\big[\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\big](\boldsymbol{A}, \boldsymbol{A}')p(\boldsymbol{A}')\,\mathrm{d}\boldsymbol{A}\,\mathrm{d}\boldsymbol{A}'}{\displaystyle\iint_{\infty} p(\boldsymbol{A})\big[\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\big](\boldsymbol{A}, \boldsymbol{A}')p(\boldsymbol{A}')\,\mathrm{d}\boldsymbol{A}\,\mathrm{d}\boldsymbol{A}'},$$

or, more compactly

$$\hat{f}(\boldsymbol{r}) = \frac{\boldsymbol{u}^{\dagger}\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\boldsymbol{p}}{\boldsymbol{p}^{\dagger}\mathcal{S}_{\boldsymbol{r},\lambda}^{-1}\boldsymbol{p}}.$$

APPENDIX A
Spline Functions

A.1 Introduction

In the simplest case, a spline function $s(x)$ is a piecewise polynomial function defined on a bounded and closed subinterval $[a, b]$ of the real line. Spline functions are particularly useful in numerical interpolation and approximation [419–422] due to their ability to "bend" around points while, maintaining a low degree for the polynomials $s(x)$ is made up with. Splines have become ubiquitous in computer graphics because of the simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design [419, 423–426]. Implementations on specialized hardware of algorithms on splines have been proposed as well [352, 353, 427].

An example of a spline function defined on the interval $[2, 9]$ is shown below:

$$s(x) = \begin{cases} {}^{5}\!/_{2} - (x-3) & \text{if } x \in [2, {}^{9}\!/_{2}), \\ 1 - (x - {}^{9}\!/_{2}) + {}^{16}\!/_{25}(x - {}^{9}\!/_{2})^2 & \text{if } x \in [{}^{9}\!/_{2}, 7), \\ {}^{5}\!/_{2} + {}^{11}\!/_{5}(x-7) - {}^{8}\!/_{5}(x-7)^2 & \text{if } x \in [7, 9]. \end{cases}$$

A plot of $s(x)$ is shown in Figure A.1. As we can see from Figure A.1, the function $s(x)$ is continuous over the interval $[2, 9]$, even though higher continuity might break down at $x = {}^{9}\!/_{2}$ and $x = 7$ [424].

A.2 Definition of Spline Space

Consider a bounded and closed interval $[a, b]$ and k points $t_1, \ldots, t_k$ such that $a < t_1 < t_2 < \cdots < t_{k-1} < t_k < b$. Let m be a positive integer—called *order*—and consider a vector $\boldsymbol{m} = \{m_1, \ldots, m_k\}$ of positive integers such that $1 \leqslant m_i \leqslant m$ for

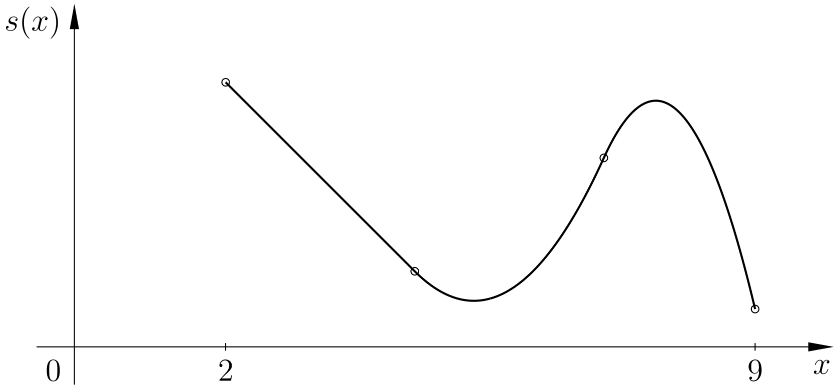


Figure A.1. Plot of the spline function $s(x)$ for $x \in [2, 9]$

$i = 1, \ldots, k$. The integers $m_1, \ldots, m_k$ are called *multiplicities*. Let $M = \sum_{i=1}^{k} m_i$ and define the *knot vector* as the set $\Delta = \{w_1, \ldots, w_{2m+M}\}$ of $2m + M$ numbers (called *knots*) such that

- $w_1 \leqslant w_2 \leqslant \ldots \leqslant w_{2m+M-1} \leqslant w_{2m+M}$;
- $w_m = a$ and $w_{m+M+1} = b$;
- $\{w_{m+1}, \ldots, w_{m+M}\} = \{\underbrace{t_1, \ldots, t_1}_{m_1 \text{ times}}, \ldots, \underbrace{t_k, \ldots, t_k}_{m_k \text{ times}}\}$.

We define the set of B-spline basis functions

$$\mathcal{B} = \big\{N_{i,m}(x), \text{ for } i = 1, \ldots, m + M\big\},$$

according to the recurring formula [421, 423, 424]

$$N_{i,n}(x) = \frac{x - w_i}{w_{i+n-1} - w_i} N_{i,n-1}(x) + \frac{w_{i+n} - x}{w_{i+n} - w_{i+1}} N_{i+1,n-1}(x), \tag{A.1a}$$

for $n = 2, \ldots, m$, where

$$N_{i,1}(x) = \begin{cases} 1 & \text{if } w_i \leqslant x < w_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{A.1b}$$

Notice that (A.1a) can yield the quotient 0/0; we define this ratio to be 0. A plot of the B-spline basis functions in $\mathcal{B}$ for a simple case is shown in Figure A.2.

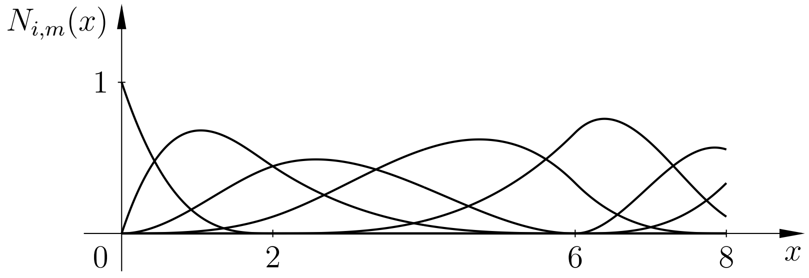


FIGURE A.2. Plot of B-spline functions $N_{1,m}(x), \ldots, N_{7,m}(x)$ for $x \in [0, 8]$, with $m = 4$, $k = 2$, $\boldsymbol{m} = \{1, 2\}$, and $\Delta = \{0, 0, 0, 0, 2, 6, 6, 8, 9, 10, 11\}$

A.3 Properties of Spline Functions

A fundamental result in the theory of spline function is that the vectors $N_{i,m}(x)$ for $i = 1, \ldots, m+M$ are linearly independent, thus $\mathcal{B} = \{N_{i,m}(x), \text{ for } i = 1, \ldots, m + M\}$ forms a basis for a vector space $\mathcal{U}$ of dimension $\dim(\mathcal{U}) = m + M$ [424, 428]. From this, it makes sense to consider linear combinations of the functions in $\mathcal{B}$:

$$s(x) = \sum_{i=1}^{m+M} c_i N_{i,m}(x), \tag{A.2}$$

for some coefficients $c_1, \ldots, c_{m+M}$. Furthermore, from the definition of B-spline basis $N_{i,m}(x)$ in (A.1) we see that each $N_{i,m}(x)$ is piecewise polynomial of degree at most $m - 1$. From (A.2), we find that $s(x)$ itself is a piecewise polynomial function. Other fundamental results that follow from (A.1) are listed below.

Non-negativity:

$$N_{i,m}(x) > 0, \qquad \forall x \in (w_i, w_{i+m}) \text{ and } w_i < w_{i+m}. \tag{A.3a}$$

The interval $[w_i, w_{i+m})$ is called *support* of $N_{i,m}(x)$.

Local support:

$$N_{i,m}(x) = 0, \qquad \forall x \notin (w_i, w_{i+m}) \text{ and } w_i < w_{i+m}. \tag{A.3b}$$

Partition of unity:

$$\sum_{i=1}^{m+M} N_{i,m}(x) = 1, \qquad \forall x \in [a,b]. \tag{A.3c}$$

The properties listed above can be used to prove a few more results about splines. For example, by (A.3a) and (A.3c), the spline function $s(x)$ is a convex hull [424] of the coefficients $c_1, \ldots, c_{m+M}$ that appear in (A.2). In other words:

$$\min\{c_1, \ldots, c_{m+M}\} \leqslant s(x_0) \leqslant \max\{c_1, \ldots, c_{m+M}\},$$

for all $x_0 \in [a,b]$. The statement above can be further refined by using (A.3b). If $x_0 \in [w_\ell, w_{\ell+1})$, then only $N_{\ell-m+1,m}(x_0), \ldots, N_{\ell,m}(x_0)$ are non-zero and (A.2) can be rewritten as

$$s(x_0) = \sum_{i=\ell-m+1}^{\ell} c_i N_{i,m}(x_0), \tag{A.4}$$

which allows us to write

$$\min\{c_{\ell-m+1}, \ldots, c_\ell\} \leqslant s(x_0) \leqslant \max\{c_{\ell-m+1}, \ldots, c_\ell\}.$$

These results can be used to derive a fast and stable algorithm for the evaluation of $s(x)$ at any point $x = x_0$. Indeed, from (A.4), we see that if $x_0 \in [w_\ell, w_{\ell+1})$ then only m B-spline basis functions need be calculated to obtain the value $s(x_0)$. Such B-spline basis functions can be evaluated using the recursive definition of (A.1). If m were 1 then, by (A.1b), only $N_{\ell,1}(x_0)$ would be non-zero. More precisely:

$$N_{i,1}(x_0) = \begin{cases} 1 & \text{if } i = \ell, \\ 0 & \text{otherwise,} \end{cases}$$

for all $x_0 \in [w_\ell, w_{\ell+1})$. If m were 2, we can use (A.1a) and the B-spline basis functions $N_{i,1}(x_0)$—which we just calculated—to evaluate $N_{\ell-1,2}(x_0)$ and $N_{\ell,2}(x_0)$. This procedure repeats according to the following triangular scheme:

$$\begin{array}{ccccccc}
0 & \ldots & \ldots & \ldots & 0 & N_{\ell,1}(x_0) & 0 \\
0 & \ldots & \ldots & 0 & N_{\ell-1,2}(x_0) & N_{\ell,2}(x_0) & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & N_{\ell-m+2,m-1}(x_0) & \ldots & \ldots & N_{\ell-1,m-1}(x_0) & 0 \\
0 & N_{\ell-m+1,m}(x_0) & \ldots & \ldots & \ldots & N_{\ell,m}(x_0) & 0
\end{array} \tag{A.5}$$

The pseudocode for an algorithm for the evaluation of $s(x)$ at $x = x_0$ is shown below. The algorithm starts by performing a dichotomic search to first identify the value of ℓ such that $x_0 \in [w_\ell, w_{\ell+1})$. The special case $x_0 = w_{m+M+1}$ is handled separately. Once the value of ℓ is determined, the algorithm uses the triangular scheme outlined above and calculates $N_{\ell-m+1,m}(x_0), \ldots, N_{\ell,m}(x_0)$. Finally, the sum in (A.4) is evaluated. It can be shown [424] that divisions by zero are correctly handled by the algorithm below.

```
if $x_0 = w_{m+M+1}$ then
  $\ell = m + M$
else
  $n_{\text{inf}} = m$
  $n_{\text{sup}} = m + M + 1$
  $\ell = \lfloor (n_{\text{inf}} + n_{\text{sup}})/2 \rfloor$
  while $x_0 < w_\ell$ or $x_0 \geqslant w_{\ell+1}$ do
    if $x_0 < w_\ell$ then
      $n_{\text{sup}} = \ell$
    else
      $n_{\text{inf}} = \ell$
    end if
    $\ell = \lfloor (n_{\text{inf}} + n_{\text{sup}})/2 \rfloor$
  end while
end if
$N_{\ell,1} = 1$
for $n = 2$ to $m$ do
  $v = 0$
  for $i = \ell - n + 1$ to $\ell - 1$ do
    $d_1 = x_0 - w_{i+1}$
    $d_2 = w_{i+n} - x_0$
    $\alpha = N_{i+1,n-1}/(d_1 + d_2)$
    $N_{i,n} = v + d_2\alpha$
    $v = d_1\alpha$
  end for
  $N_{\ell,n} = v$
end for
$s = 0$
for $i = \ell - m + 1$ to $\ell$ do
  $s = s + c_i N_{i,m}$
end for
return $s$
```

From the pseudocode shown above, we see that the calculation of $s(x_0)$ requires a total of $\mathcal{O}(m^2)$ floating-point operations. The parameter m is usually small, for example, $m \leqslant 10$. Therefore, the algorithm above provides a fast way to calculate $s(x_0)$. The algorithm can also be modified so that it uses only a one-dimensional array with m entries: this is accomplished by overwriting rows while executing the triangular scheme pictured in (A.5) [423]. We finally note that the algorithm is stable, as the $N_{\ell-m+1,m}(x_0), \ldots, N_{\ell,m}(x_0)$ are obtained by repeatedly forming convex combinations [423].

A.4 Derivative of a Spline Function

By direct differentiation of (A.2):

$$\frac{\mathrm{d}}{\mathrm{d}x} s(x) = \sum_{i=1}^{m+M} c_i \left[\frac{\mathrm{d}}{\mathrm{d}x} N_{i,m}(x) \right],$$

where [423, 429]

$$\frac{\mathrm{d}}{\mathrm{d}x} N_{i,m}(x) = (m-1) \left[\frac{N_{i,m-1}(x)}{w_{i+m-1} - w_i} - \frac{N_{i+1,m-1}(x)}{w_{i+m} - w_{i+1}} \right].$$

Substituting and performing some manipulations

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x} s(x) = & \sum_{i=1}^{m+M} (m-1) c_i \left[\frac{N_{i,m-1}(x)}{w_{i+m-1} - w_i} - \frac{N_{i+1,m-1}(x)}{w_{i+m} - w_{i+1}} \right] = \\
= & \sum_{i=1}^{m+M} c_i \frac{m-1}{w_{i+m-1} - w_i} N_{i,m-1}(x) - \sum_{i=1}^{m+M} c_i \frac{m-1}{w_{i+m} - w_{i+1}} N_{i+1,m-1}(x) = \\
= & \sum_{i=1}^{m+M-1} c_{i+1} \frac{m-1}{w_{i+m} - w_{i+1}} N_{i+1,m-1}(x) - \sum_{i=1}^{m+M-1} c_i \frac{m-1}{w_{i+m} - w_{i+1}} N_{i+1,m-1}(x) = \\
= & \sum_{i=1}^{m+M-1} c'_{i+1} N_{i+1,m-1}(x),
\end{aligned}$$

where we have defined

$$c'_{i+1} = \begin{cases} (m-1) \dfrac{c_{i+1} - c_i}{w_{i+m} - w_{i+1}} & \text{if } w_{i+1} < w_{i+m}, \\ 0 & \text{otherwise.} \end{cases}$$

The calculation above shows that the derivative of a spline function of order m is another spline function, but of order $m-1$.

A.5 Spline Curves and Spline Surfaces

Consider the B-spline basis $\mathcal{B} = \{N_{i,m}(u), \text{ for } i = 1, \ldots, m+M\}$ associated to the knot vector $\Delta = \{w_1, \ldots, w_{2m+M}\}$ and let $\boldsymbol{R}_i = (X_i, Y_i)$ for $i = 1, \ldots, m+N$ be 2D points. We can consider the two spline functions defined as follows

$$s_X(u) = \sum_{i=1}^{m+M} X_i N_{i,m}(u), \qquad s_Y(u) = \sum_{i=1}^{m+M} Y_i N_{i,m}(u),$$

for $u \in [a,b]$. We define a 2D spline curve above as

$$\boldsymbol{s}(u) = \begin{pmatrix} s_X(u) \\ s_Y(u) \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{i=1}^{m+M} X_i N_{i,m}(u) \\ \displaystyle\sum_{i=1}^{m+M} Y_i N_{i,m}(u) \end{pmatrix}, \qquad u \in [a,b].$$

In compact form, we can express the 2D spline curve as

$$\boldsymbol{s}(u) = \sum_{i=1}^{m+M} \boldsymbol{R}_i N_{i,m}(u), \qquad u \in [a,b].$$

The construction above can be extended to the 3D case—and in general to the ND case—by simply allowing the points $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{m+M}$ to be 3D (or ND) points.

To define 3D surfaces, we will consider two B-spline bases

$$\mathcal{B}_u = \left\{N_{i,m_u}^{(u)}(u), \text{ for } i = 1, \ldots, m_u + M_u\right\},$$
$$\mathcal{B}_v = \left\{N_{i,m_v}^{(v)}(v), \text{ for } i = 1, \ldots, m_v + M_v\right\},$$

associated to knot vectors Δ_u and Δ_v, respectively. Consider a grid of 3D points $\boldsymbol{R}_{i,j} = (X_{i,j}, Y_{i,j}, Z_{i,j})$ for $i = 1, \ldots, m_u + M_u$ and $j = 1, \ldots, m_v + M_v$. Then we define the spline surface $s(u,v)$ as

$$\boldsymbol{s}(u,v)=\begin{pmatrix} s_X(u,v)\\ s_Y(u,v)\\ s_Z(u,v)\end{pmatrix}=\begin{pmatrix} \sum_{i=1}^{m_u+M_u}\sum_{j=1}^{m_v+M_v} X_{i,j}N_{i,m_u}^{(u)}(u)N_{j,m_v}^{(v)}(v)\\ \sum_{i=1}^{m_u+M_u}\sum_{j=1}^{m_v+M_v} Y_{i,j}N_{i,m_u}^{(u)}(u)N_{j,m_v}^{(v)}(v)\\ \sum_{i=1}^{m_u+M_u}\sum_{j=1}^{m_v+M_v} Z_{i,j}N_{i,m_u}^{(u)}(u)N_{j,m_v}^{(v)}(v)\end{pmatrix},$$

with $u\in[a_u,b_u]$ and $v\in[a_v,b_v]$.

The definition of $\boldsymbol{s}(u,v)$ above also shows that the restriction of $\boldsymbol{s}(u,v)$ to $u=u_0$ or $v=v_0$ is a 3D spline curve. Indeed:

$$\boldsymbol{s}(u_0,v)=\begin{pmatrix} s_X(u_0,v)\\ s_Y(u_0,v)\\ s_Z(u_0,v)\end{pmatrix}=\begin{pmatrix} \sum_{j=1}^{m_v+M_v}\left[\sum_{i=1}^{m_u+M_u} X_{i,j}N_{i,m_u}^{(u)}(u_0)\right]N_{j,m_v}^{(v)}(v)\\ \sum_{j=1}^{m_v+M_v}\left[\sum_{i=1}^{m_u+M_u} Y_{i,j}N_{i,m_u}^{(u)}(u_0)\right]N_{j,m_v}^{(v)}(v)\\ \sum_{j=1}^{m_v+M_v}\left[\sum_{i=1}^{m_u+M_u} Z_{i,j}N_{i,m_u}^{(u)}(u_0)\right]N_{j,m_v}^{(v)}(v)\end{pmatrix},$$

in which the quantities in $[\ldots]$ are the components of 3D points that define the spine curve $\boldsymbol{s}(u_0,v)$. Similarly,

$$\boldsymbol{s}(u,v_0)=\begin{pmatrix} s_X(u,v_0)\\ s_Y(u,v_0)\\ s_Z(u,v_0)\end{pmatrix}=\begin{pmatrix} \sum_{i=1}^{m_u+M_u}\left[\sum_{j=1}^{m_v+M_v} X_{i,j}N_{j,m_v}^{(v)}(v_0)\right]N_{i,m_u}^{(u)}(u)\\ \sum_{i=1}^{m_u+M_u}\left[\sum_{j=1}^{m_v+M_v} Y_{i,j}N_{j,m_v}^{(v)}(v_0)\right]N_{i,m_u}^{(u)}(u)\\ \sum_{i=1}^{m_u+M_u}\left[\sum_{j=1}^{m_v+M_v} Z_{i,j}N_{j,m_v}^{(v)}(v_0)\right]N_{i,m_u}^{(u)}(u)\end{pmatrix}.$$

A.6 Interpolation with Splines

Consider a spline basis $\mathcal{B} = \{N_{i,m}(x)$, for $i = 1, \ldots, m+M\}$ defined on $[a,b]$ along with the set of spline functions

$$s(x) = \sum_{i=1}^{m+M} c_i N_{i,m}(x) \tag{A.6}$$

spanned by the vectors in $\mathcal{B}$. Let $x_1 < x_2 < \cdots < x_{m+M-1} < x_{m+M}$ be $m+M$ points all in $[a,b]$ and consider $y_1, \ldots, y_{m+M}$ data points. In an interpolation problem, we want to find a spline function $s(x)$ such that $s(x_i) = y_i$ for $i = 1, \ldots, m+M$. Using (A.6), we can rewrite this set of conditions in matrix form as

$$\begin{pmatrix} N_{1,m}(x_1) & \ldots & N_{m+M,m}(x_1) \\ \vdots & & \vdots \\ N_{1,m}(x_{m+M}) & \ldots & N_{m+M,m}(x_{m+M}) \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_{n+M} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_{n+M} \end{pmatrix},$$

Or, more compactly,

$$\mathbf{N}_{\boldsymbol{x}} \boldsymbol{c} = \boldsymbol{y},$$

where

$$\mathbf{N}_{\boldsymbol{x}} = \begin{pmatrix} N_{1,m}(x_1) & \ldots & N_{m+M,m}(x_1) \\ \vdots & & \vdots \\ N_{1,m}(x_{m+M}) & \ldots & N_{m+M,m}(x_{m+M}) \end{pmatrix},$$

and

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_{n+M} \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{n+M} \end{pmatrix}.$$

We notice that $\mathbf{N}_{\boldsymbol{x}}$ is a square matrix of size $(m+M) \times (m+M)$ and, if it were invertible, we could obtain the coefficients $c_1, \ldots, c_{m+M}$ as

$$\boldsymbol{c} = [\mathbf{N}_{\boldsymbol{x}}]^{-1} \boldsymbol{y}.$$

In can be shown [430] that a necessary and sufficient condition for the existence and uniqueness of the coefficients $c_1, \ldots, c_{m+M}$ can be written in terms of the knot vector $\Delta = \{w_1, \ldots, w_{m+M}\}$:

$$w_i \leqslant x_i \leqslant w_{i+m},$$

with equality allowed only if the knot in question has multiplicity m. The condition above shows how to find the knot vector Δ such that the interpolation problem has a unique solution [424]. We further note that by (A.3b), each column of matrix $\mathbf{N}_{\boldsymbol{x}}$ has at most m non-zero entries, so $\mathbf{N}_{\boldsymbol{x}}$ can be stored and inverted efficiently.

The case of spline curve interpolation can be handled similarly. For example, consider 2D points $\boldsymbol{R}_i = (X_i, Y_i)$ for $i = 1, \dots, m + N$. We want to find a spline function $\boldsymbol{s}(u)$ with components $s_X(u)$ and $s_Y(u)$ such that $\boldsymbol{s}(u_i) = \boldsymbol{R}_i$ for some $u_1, \dots, u_{m+M}$ in $[a, b]$. Given some arbitrary choice of $u_1, \dots, u_{m+M}$ (for example, $u_1, \dots, u_{m+M}$ uniformly spaced in $[a, b]$; other choices are possible [424]), then the interpolation problem is solved by performing two interpolations steps: one to find $s_X(u)$ such that $s_X(u_i) = X_i$, and the other to find $s_Y(u)$ such that $s_Y(u_i) = Y_i$. Both steps are solved using the method discussed earlier. We notice that the matrix

$$\mathbf{N}_{\boldsymbol{u}} = \begin{pmatrix} N_{1,m}(u_1) & \dots & N_{m+M,m}(u_1) \\ \vdots & & \vdots \\ N_{1,m}(u_{m+M}) & \dots & N_{m+M,m}(u_{m+M}) \end{pmatrix}$$

that we would need to find and invert is the same for both interpolations, and so its inverse need be calculated only once. Notice that the choice of knot vector Δ defines the spline space. Therefore, different choices of Δ will give different spline curves $\boldsymbol{s}(u)$, all satisfying the interpolation conditions $\boldsymbol{s}(u_i) = \boldsymbol{R}_i$ [424].

For the case of spline surface interpolation, assume we have a grid of 3D points $\boldsymbol{R}_{i,j} = (X_{i,j}, Y_{i,j}, Z_{i,j})$ for $i = 1, \dots, m_u + M_u$ and $j = 1, \dots, m_v + M_v$. We want to find a spline surface $\boldsymbol{s}(u, v)$ such that $\boldsymbol{s}(u_i, v_j) = \boldsymbol{R}_{i,j}$ for $i = 1, \dots, m_u + M_u$ and $j = 1, \dots, m_v + M_v$. This condition can be rewritten as

$$\sum_{i=1}^{m_u+M_u} \sum_{j=1}^{m_v+M_v} \boldsymbol{c}_{i,j} N^{(u)}_{i,m_u}(u_i) N^{(v)}_{j,m_v}(v_j) = \boldsymbol{R}_{i,j},$$

for some coefficient vectors $\boldsymbol{c}_{i,j} = \left(c^{(X)}_{i,j}, c^{(Y)}_{i,j}, c^{(Z)}_{i,j}\right)$. Equivalently,

$$\sum_{j=1}^{m_v+M_v} \boldsymbol{d}_{i,j} N^{(v)}_{j,m_v}(v_j) = \boldsymbol{R}_{i,j}, \qquad i = 1, \dots, m_u + M_u, \tag{A.7a}$$

provided that

$$\sum_{i=1}^{m_u+M_u} \boldsymbol{c}_{i,j} N_{i,m_u}^{(u)}(u_i) = \boldsymbol{d}_{i,j}, \qquad j = 1, \ldots, m_v + M_v. \tag{A.7b}$$

The equations in (A.7) show that the coefficients $\boldsymbol{c}_{i,j}$ can be found by first solving a set of $m_u + M_u$ 3D curve interpolations of the points $\boldsymbol{R}_{i,j}$, as shown in (A.7a). All these interpolations, which share the same matrix $\mathbf{N}_{\boldsymbol{v}}$, would give 3D coefficients $\boldsymbol{d}_{i,j}$. The coefficients $\boldsymbol{d}_{i,j}$ are then interpolated as shown in (A.7b). Again, this latter step, requires a total of $m_v + M_v$ 3D curve interpolations, all of them sharing the same matrix $\mathbf{N}_{\boldsymbol{u}}$.

A.7 Approximation with Splines

Suppose we are given N numbers $x_1 < x_2 < \cdots < x_{N-1} < x_N$ in $[a, b]$ and corresponding data points $y_1, \ldots, y_N$. In an approximation problem, we look for a spline function $s(x)$ that no longer necessarily satisfies the condition $s(x_i) = y_i$ for $i = 1, \ldots, N$. Instead, we look for the spline function that minimizes the quantity

$$\varepsilon = \sum_{n=1}^{N} [y_n - s(x_n)]^2. \tag{A.8}$$

We refer to this as a least-square fit of the data $y_1, \ldots, y_N$ at points $x_1, \ldots, x_N$. The necessity for an approximation occurs when the data set we have is very large compared to the dimension of the spline space ($N \gg m + M$) or when the data are noisy. In the latter case, a low-degree function that approximates the data often gives good results. Another interpretation is that when $N > m + M$ (i.e., the number of data points exceeds the dimension of the spline space), the spline space might not posses the "flexibility" necessary to actually interpolate the data points.

Assume $s(x)$ has the form

$$s(x) = \sum_{i=1}^{m+M} c_i N_{i,m}(x). \tag{A.9}$$

We can substitute the expression above in (A.8), calculate the derivatives with respect to each coefficient c_i and set them to zero. This would give

$$\sum_{i=1}^{m+M} c_i \left(\sum_{n=1}^{N} N_{i,m}(x_n) N_{j,m}(x_n) \right) = \sum_{n=1}^{N} y_n N_{j,m}(x_n), \qquad j = 1, \ldots, m+M.$$

In matrix form, we can write this as

$$\mathbf{G}_{\boldsymbol{x}} \boldsymbol{c} = \boldsymbol{y}',$$

where

$$\mathbf{G}_{\boldsymbol{x}} = \begin{pmatrix} \sum_{n=1}^{N} N_{1,m}(x_n) N_{1,m}(x_n) & \ldots & \sum_{n=1}^{N} N_{1,m}(x_n) N_{m+M,m}(x_n) \\ \vdots & & \vdots \\ \sum_{n=1}^{N} N_{m+M,m}(x_n) N_{1,m}(x_n) & \ldots & \sum_{n=1}^{N} N_{m+M,m}(x_n) N_{m+M,m}(x_n) \end{pmatrix},$$

and

$$\boldsymbol{y}' = \begin{pmatrix} \sum_{n=1}^{N} y_n N_{1,m}(x_n) \\ \vdots \\ \sum_{n=1}^{N} y_n N_{m+M,m}(x_n) \end{pmatrix}.$$

We can express $\mathbf{G}_{\boldsymbol{x}}$ as $\mathbf{H}_{\boldsymbol{x}}^{\intercal} \mathbf{H}_{\boldsymbol{x}}$, where

$$\mathbf{H}_{\boldsymbol{x}} = \begin{pmatrix} N_{1,m}(x_1) & \ldots & N_{m+M,m}(x_1) \\ \vdots & & \vdots \\ N_{1,m}(x_N) & \ldots & N_{m+M,m}(x_N) \end{pmatrix}.$$

If $\mathbf{H}_{\boldsymbol{x}}$ has full rank, then $\mathbf{G}_{\boldsymbol{x}}$ is invertible and we can compute $\boldsymbol{c}$ as $\left[\mathbf{G}_{\boldsymbol{x}}\right]^{-1} \boldsymbol{y}'$.

APPENDIX B
The Bootstrapping Method

Bootstrapping [431, 432] is a statistical method for assessing the accuracy of sample estimates. Bootstrapping falls under the broad category of resampling methods, which includes the jackknife [433] and the cross-validation [434] methods. The bootstrap method is a nonparametric method in the sense that the data are not fit to a model—such as a parametrized normal distribution—from which one or more parameters of interest are calculated.

Assume that we are given N observed quantities $x_1, \ldots, x_N$. These quantities can represent the outcomes of an experiment repeated N times or could also be random quantities obtained via simulation codes. Given $x_1, \ldots, x_N$, consider an empirical distribution on $\{x_1, \ldots, x_N\}$ for the random variable X such that

$$\Pr(X = x_n) = \frac{1}{N},$$

for $n = 1, \ldots, N$. Let $X_1, \ldots, X_N$ be random samples of X independently drawn according to the probability distribution above and assume that $r(X_1, \ldots, X_N)$ is some quantity of interest calculated from the set of N samples $\{X_1, \ldots, X_N\}$. The premise of the bootstrap method is that the actual distribution of $r(\ldots)$ is equal to the empirical distribution of $r(X_1, \ldots, X_N)$ [431], in which samples $X_1, \ldots, X_N$ are uniformly and independently drawn from $\{x_1, \ldots, x_N\}$.

Assume now the process above is repeated M times. That results in samples $r_1, \ldots, r_M$ of $r(\ldots)$. Then, we define the "accuracy" of the bootstrap estimate $\hat{r} = \frac{1}{M}\sum_{m=1}^{M} r_m$ as the sample standard deviation s_r of $r_1, \ldots, r_M$:

$$s_r = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}(r_m - \hat{r})^2}.$$

APPENDIX C

A Theorem on the Limit of the Ratio of Two Functions

Theorem 1. *Let* $f_1 : \mathbb{R} \to \mathbb{R}^{\geqslant 0}$ *and* $f_2 : \mathbb{R} \to \mathbb{R}^{\geqslant 0}$ *be functions such that*

$$\lim_{x\to+\infty} f_1(x) = \ell_1, \qquad \lim_{x\to+\infty} f_2(x) = \ell_2,$$

where

$$0 < \ell_1 < +\infty, \qquad 0 < \ell_2 < +\infty.$$

Then

$$\lim_{x\to+\infty} \frac{f_1(x)}{f_2(x)} = \frac{\ell_1}{\ell_2}.$$

Proof. By the definition of limit,

$$\forall \varepsilon > 0 \quad \exists M_{1,\varepsilon} \quad \text{such that} \quad x > M_{1,\varepsilon} \Rightarrow |f_1(x) - \ell_1| < \varepsilon.$$

Similarly,

$$\forall \varepsilon > 0 \quad \exists M_{2,\varepsilon} \quad \text{such that} \quad x > M_{2,\varepsilon} \Rightarrow |f_2(x) - \ell_2| < \varepsilon.$$

By setting $\varepsilon' = \min\{\varepsilon, \ell_1/2, \ell_2/2\} > 0$, there exists $M_{\varepsilon'} = \max\{M_{1,\varepsilon'}, M_{2,\varepsilon'}\}$ such that

$$M_{\varepsilon'} \geqslant \max\{M_{1,\varepsilon}, M_{2,\varepsilon}\},$$

and

$$\begin{cases} 0 < \ell_1 - \varepsilon' < f_1(x) < \ell_1 + \varepsilon' \\ 0 < \ell_2 - \varepsilon' < f_2(x) < \ell_2 + \varepsilon' \end{cases}$$

for all $x > M_{\varepsilon'}$. From bounds on $f_1(x)$ and $f_2(x)$ above we can calculate bounds on the quantity $f_1(x)/f_2(x)$ as follows:

$$\frac{\ell_1 - \varepsilon'}{\ell_2 + \varepsilon'} < \frac{\ell_1 - \varepsilon'}{f_2(x)} < \frac{f_1(x)}{f_2(x)} < \frac{\ell_1 + \varepsilon'}{f_2(x)} < \frac{\ell_1 + \varepsilon'}{\ell_2 - \varepsilon'}.$$

Subtracting ℓ_1/ℓ_2 yields:

$$-\frac{\varepsilon'(\ell_1+\ell_2)}{\ell_2(\ell_2+\varepsilon')} < \frac{f_1(x)}{f_2(x)} - \frac{\ell_1}{\ell_2} < \frac{\varepsilon'(\ell_1+\ell_2)}{\ell_2(\ell_2-\varepsilon')}.$$

Now set

$$\varepsilon'' = \frac{\varepsilon'(\ell_1+\ell_2)}{\ell_2(\ell_2-\varepsilon')} > 0 \tag{C.1}$$

and note that

$$\varepsilon'' > \frac{\varepsilon'(\ell_1+\ell_2)}{\ell_2(\ell_2+\varepsilon')} > 0,$$

which allows us to write

$$\left|\frac{f_1(x)}{f_2(x)} - \frac{\ell_1}{\ell_2}\right| < \varepsilon''. \tag{C.2}$$

If, given $\varepsilon'' > 0$, we solve (C.1) for ε' and take

$$\varepsilon' = \frac{\varepsilon''\ell_2^2}{\ell_1+(1+\varepsilon'')\ell_2} < \frac{\varepsilon''\ell_2^2}{\ell_1+\ell_2},$$

then (C.2) is satisfied when $x > M_{\varepsilon'} = \max\{M_{1,\varepsilon'}, M_{2,\varepsilon'}\}$, which concludes the proof of the theorem. □

References

[1] T. Ido, C-N. Wan, V. Casella, J. S. Fowler, A. P. Wolf, M. Reivich, and D. E. Kuhl. Labeled 2-deoxy-D-glucose analogs. ^{18}F-labeled 2-deoxy-2-fluoro-D-glucose, 2-deoxy-2-fluoro-D-mannose and ^{14}C-2-deoxy-2-fluoro-D-glucose. *Journal of Labelled Compounds and Radiopharmaceuticals*, 14(2):175–183, 1978.

[2] S. Yu. Review of ^{18}F-FDG synthesis and quality control. *Biomedical Imaging and Intervention Journal*, 2(4):e57, October-December 2006.

[3] James A. Sorenson and M. E. Phelps. *Physics in Nuclear Medicine*. Grune & Stratton, Orlando, FL, 1987.

[4] Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter. Introducing Markov chain Monte Carlo. In Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, chapter 1. Chapman & Hall/CRC, Boca Raton, FL, 1996.

[5] Luca Caucci and Harrison H. Barrett. Objective assessment of image quality. V. Photon-counting detectors and list-mode data. *Journal of the Optical Society of America A*, 29(6):1003–1016, June 2012.

[6] Stephen K. Moore. *ModPET: Novel Applications of Scintillation Cameras to Preclinical PET*. PhD thesis, University of Arizona, Tucson, AZ, 2011.

[7] Luca Caucci, Matthew A. Kupinski, Melanie Freed, Lars R. Furenlid, Donald W. Wilson, and Harrison H. Barrett. Adaptive SPECT for tumor necrosis detection. In *IEEE Nuclear Science Symposium Conference Record*, pages 5548–5551, Dresden, Germany, October 2008.

[8] Douglas J. Rowland and Simon R. Cherry. Small-animal preclinical nuclear medicine instrumentation and methodology. *Seminars in Nuclear Medicine*, 38(3):209–222, May 2008.

[9] Simon R. Cherry and Sanjiv S. Gambhir. Use of positron emission tomography in animal research. *ILAR Journal*, 42(3):219–232, 2001.

[10] Miles N. Wernick and John N. Aarsvold. *Emission Tomography: The Fundamentals of PET and SPECT*. Elsevier Academic Press, San Diego, CA, 2004.

[11] Markus Rudin, Martin Rausch, and Markus Stoeckli. Molecular imaging in drug discovery and development: Potential and limitations of nonnuclear methods. *Molecular Imaging and Biology*, 7(1):5–13, November 2005.

[12] J. L. Vanderheyden. The use of imaging in preclinical drug development. *Quarterly Journal of Nuclear Medicine and Molecular Imaging*, 53(4):374–381, August 2009.

[13] Sophie Lancelot and Luc Zimmer. Small-animal positron emission tomography as a tool for neuropharmacology. *Trends in Pharmacological Sciences*, 31(9):411–417, September 2010.

[14] V. Koo, P. W. Hamilton, and K. Williamson. Non-invasive in vivo imaging in small animal research. *Analytical Cellular Pathology*, 28(4):127–139, 2006.

[15] Robert S. Balaban and Victoria A. Hampshire. Challenges in small animal noninvasive imaging. *ILAR Journal*, 42(3):248–262, 2001.

[16] David Malakoff. The rise of the mouse, biomedicine's model mammal. *Science*, 288(5464):248–253, April 2000.

[17] Yuan-Chuan Tai and Richard Laforest. Instrumentation aspects of animal SPECT. *Annual Review of Biomedical Engineering*, 7:255–285, August 2005.

[18] M. V. Green, J. Seidel, J. J. Vaquero, E. Jagoda, I. Lee, and W. C. Eckelman. High resolution PET, SPECT and projection imaging in small animals. *Computerized Medical Imaging and Graphics*, 25(2):79–86, March 2001.

[19] Michael J. Paulus, Shaun S. Gleason, Stephen J. Kennel, Patricia R. Hunsicker, and Dabney K. Johnson. High resolution X-ray computed tomography: An emerging tool for small animal cancer research. *Neoplasia*, 2(1-2):62–70, January-April 2000.

[20] Paul D. Acton and Hank F. Kung. Small animal imaging with high resolution single photon emission tomography. *Nuclear Medicine and Biology*, 30(8):889–895, November 2003.

[21] Ralph Myers and Susan Hume. Small animal PET. *European Neuropsychopharmacology*, 12(6):545–555, December 2002.

[22] Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, England, 1998.

[23] Freek J. Beekman, Frans van der Have, Brendan Vastenhouw, Annemarie J. A. van der Linden, Peter P. van Rijk, J. Peter H. Burbach, and Marten P. Smidt. U-SPECT-I: A novel system for submillimeter-resolution tomography with radiolabeled molecules in mice. *Journal of Nuclear Medicine*, 46(7):1194–1200, July 2005.

[24] Yi-Chun Chen. *System Calibration and Image Reconstruction for a new Small-Animal SPECT System*. PhD thesis, University of Arizona, Tucson, AZ, 2006.

[25] Lars R. Furenlid, Donald W. Wilson, Yi-Chun Chen, Hyunki Kim, Philip J. Pietraski, Michael J. Crawford, and Harrison H. Barrett. FastSPECT II: A second-generation high-resolution dynamic SPECT imager. *IEEE Transactions on Nuclear Science*, 51(3):631–635, June 2004.

[26] David P. McElroy, Lawrence R. MacDonald, Freek J. Beekman, Yuchuan Wang, Bradley E. Patt, Jan S. Iwanczyk, Benjamin M. W. Tsui, and Edward J. Hoffman. Performance evaluation of A-SPECT: A high resolution desktop pinhole SPECT system for imaging small animals. *IEEE Transactions on Nuclear Science*, 49(5):2139–2147, October 2002.

[27] Harrison H. Barrett. Fresnel zone plate imaging in nuclear medicine. *Journal of Nuclear Medicine*, 13(6):382–385, June 1972.

[28] Steven R. Meikle, Roger R. Fulton, Stefan Eberl, Magnus Dahlbom, Koon-Pong Wong, and Michael J. Fulham. An investigation of coded aperture imaging for small animal SPECT. *IEEE Transactions on Nuclear Science*, 48(3):816–821, June 2001.

[29] N. U. Schramm, G. Ebel, U. Engeland, T. Schurrat, M. Béhé, and T. M. Behr. High-resolution SPECT using multipinhole collimation. *IEEE Transactions on Nuclear Science*, 50(3):315–320, June 2003.

[30] Dawid Schellingerhout, Roberto Accorsi, Umar Mahmood, John Idoine, Richard C. Lanza, and Ralph Weissleder. Coded aperture nuclear scintigraphy: A novel small animal imaging technique. *Molecular Imaging*, 1(4):344–353, October 2002.

[31] F. Forrer, R. Valkema, B. Bernard, N. U. Schramm, J. W. Hoppin, E. Rolleman, E. P. Krenning, and M. de Jong. In vivo radionuclide uptake quantification using a multi-pinhole SPECT system to predict renal function in small animals. *European Journal of Nuclear Medicine and Molecular Imaging*, 33(10):1214–1217, October 2006.

[32] G. A. Kastis, H. B. Barber, H. H. Barrett, S. J. Balzer, D. Lu, D. G. Marks, G. Stevenson, J. M. Woolfenden, M. Appleby, and J. Tueller. Gamma-ray imaging using a CdZnTe pixel array and a high-resolution, parallel-hole collimator. *IEEE Transactions on Nuclear Science*, 47(6):1923–1927, December 2000.

[33] D. J. Wagenaar, J. Zhang, T. Kazules, T. Vandehei, E. Bolle, S. Chowdhury, K. Parnham, and B. E. Patt. In vivo dual-isotope SPECT imaging with improved energy resolution. In *IEEE Nuclear Science Symposium Conference Record*, pages 3821–3826, San Diego, CA, October/November 2006.

[34] Bill C. Penney, Chien-Min Kao, Xiaochuan Pan, and Nalini Bidani. Dual-isotope (In-111/Tc-99m) SPECT: Noise and artifact reduction with an analytic attenuation correction method. *IEEE Transactions on Nuclear Science*, 49(3):733–739, June 2002.

[35] G. El Fakhri, P. Maksud, M. F. Kijewski, R. E. Zimmerman, and S. C. Moore. Quantitative simultaneous ^{99m}Tc/^{123}I SPECT: Design study and validation with Monte Carlo simulations and physical acquisitions. *IEEE Transactions on Nuclear Science*, 49(5):2315–2321, October 2002.

[36] John M. Ollinger and Jeffrey A. Fessler. Positron-emission tomography. *IEEE Signal Processing Magazine*, 14(1):43–55, January 1997.

[37] Henry N. Wagner, Jr. A brief history of positron emission tomography (PET). *Seminars in Nuclear Medicine*, 28(3):213–220, July 1998.

[38] Michel M. Ter-Pogossian. The origins of positron emission tomography. *Seminars in Nuclear Medicine*, 22(3):140–149, July 1992.

[39] S. R. Cherry, Y. Shao, R. W. Silverman, K. Meadors, S. Siegel, A. Chatziioannou, J. W. Young, W. Jones, J. C. Moyers, D. Newport, A. Boutefnouchet, T. H. Farquhar, M. Andreaco, M. J. Paulus, D. M. Binkley, R. Nutt, and M. E. Phelps. MicroPET: A high resolution PET scanner for imaging small animals. *IEEE Transactions on Nuclear Science*, 44(3):1161–1166, June 1997.

[40] H. Zhang, N. T. Vu, Q. Bao, R. W. Silverman, B. N. Berry-Pusey, A. Douraghy, D. A. Williams, F. R. Rannou, D. B. Stout, and A. F. Chatziioannou. Performance characteristics of BGO detectors for a low cost preclinical PET scanner. *IEEE Transactions on Nuclear Science*, 57(3):1038–1044, June 2010.

[41] Hui Zhang, Qinan Bao, Nam T. Vu, Robert W. Silverman, Richard Taschereau, Brittany N. Berry-Pusey, Ali Douraghy, Fernando R. Rannou, David B. Stout, and Arion F. Chatziioannou. Performance evaluation of PETbox: A low cost bench top preclinical PET scanner. *Molecular Imaging and Biology*, 13(5):949–961, October 2011.

[42] Robley Dunglison Evans. *The Atomic Nucleus.* McGraw-Hill, New York, NY, 1955.

[43] L. Sodickson, W. Bowman, J. Stephenson, and R. Weinstein. Single-quantum annihilation of positrons. *Physical Review*, 124(6):1851–1861, December 1961.

[44] Michael E. Phelps, Edward J. Hoffman, Nizar A. Mullani, and Michel M. Ter-Pogossian. Application of annihilation coincidence detection to transaxial reconstruction tomography. *Journal of Nuclear Medicine*, 16(3):210–224, March 1975.

[45] Lars R. Furenlid, Jacob Y. Hesterman, and Harrison H. Barrett. Real-time data acquisition and maximum-likelihood estimation for gamma cameras. In *14th IEEE-NPSS Real Time Conference*, pages 498–501, Stockholm, Sweden, June 2005.

[46] Harrison H. Barrett, William C. J. Hunter, Brian William Miller, Stephen K. Moore, Yichun Chen, and Lars R. Furenlid. Maximum-likelihood methods for processing signals from gamma-ray detectors. *IEEE Transactions on Nuclear Science*, 56(3):725–735, June 2009.

[47] Tom K. Lewellen. Time-of-flight PET. *Seminars in Nuclear Medicine*, 28(3):268–275, July 1998.

[48] W. W. Moses. Time of flight in PET revisited. *IEEE Transactions on Nuclear Science*, 50(5):1325–1330, October 2003.

[49] Georges El Fakhri, Suleman Surti, Cathryn M. Trott, Joshua Scheuermann, and Joel S. Karp. Improvement in lesion detection with whole-body oncologic time-of-flight PET. *Journal of Nuclear Medicine*, 52(3):347–353, March 2011.

[50] J. G. Rogers, D. P. Saylor, R. Harrop, X. G. Yao, C. V. M. Leitao, and B. D. Pate. Design of an efficient position sensitive gamma ray detector for nuclear medicine. *Physics in Medicine and Biology*, 31(10):1061–1090, October 1986.

[51] W. R. Cook, M. Finger, and T. A. Prince. A thick Anger camera for gamma-ray astronomy. *IEEE Transactions on Nuclear Science*, 32(1):129–133, February 1985.

[52] P. Bartzakos and C. J. Thompson. A depth-encoded PET detector. *IEEE Transactions on Nuclear Science*, 38(2):732–738, April 1991.

[53] D. Gagnon, N. Pouliot, L. Laperrière, M. Therrien, and P. Olivier. Maximum likelihood positioning in the scintillation camera using depth of interaction. *IEEE Transactions on Medical Imaging*, 12(1):101–107, January 1993.

[54] E. Costa, E. Massaro, and L. Piro. A BGO-CsI(Tl) phoswich: A new detector for X- and γ-ray astronomy. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 243(2-3):572–577, March 1986.

[55] M. Dahlbom, L. R. MacDonald, L. Eriksson, M. Paulus, M. Andreaco, M. E. Casey, and C. Moyers. Performance of a YSO/LSO phoswich detector for use in a PET/SPECT system. *IEEE Transactions on Nuclear Science*, 44(3):1114–1119, January 1997.

[56] F. Nishikido, T. Tsuda, N. Inadama, E. Yoshida, K. Takahashi, K. Shibuya, T. Yamaya, K. Kitamura, and H. Murayama. Spatial resolution measured by a prototype system of two four-layer DOI detectors for jPET-RD. In *IEEE Nuclear Science Symposium Conference Record*, pages 3041–3044, San Diego, CA, October/November 2006.

[57] R. S. Miyaoka, T. K. Lewellen, H. Yu, and D. L. McDaniel. Design of depth of interaction (DOI) PET detector module. *IEEE Transactions on Nuclear Science*, 45(3):1069–1073, June 1998.

[58] W. W. Moses and S. E. Derenzo. Design studies for a PET detector module using a PIN photodiode to measure depth of interaction. *IEEE Transactions on Nuclear Science*, 41(4):1441–1445, August 1994.

[59] William C. J. Hunter, Harrison H. Barrett, and Lars R. Furenlid. Calibration method for ML estimation of 3D interaction position in a thick gamma-ray detector. *IEEE Transactions on Nuclear Science*, 56(1):189–196, February 2009.

[60] William Coulis Jason Hunter. *Modeling Stochastic Processes in Gamma-Ray Imaging Detectors and Evaluation of a Multi-Anode PMT Scintillation Camera for Use with Maximum-Likelihood Estimation Methods.* PhD thesis, University of Arizona, Tucson, AZ, 2007.

[61] Christoph Werner Lerche, J. M. Benlloch, F. Sánchez, N. Pavón, B. Escat, E. N. Gimenez, M. Fernández, I. Torres, M. Giménez, A. Sebastià, and J. Martínez. Depth of γ-ray interaction within continuous crystals from the width of its scintillation light-distribution. *IEEE Transactions on Nuclear Science*, 52(3):560–572, June 2005.

[62] J. Seidel, J. J. Vaquero, S. Siegel, W. R. Gandler, and M. V. Green. Depth identification accuracy of a three layer phoswich PET detector module. *IEEE Transactions on Nuclear Science*, 46(3):485–490, June 1999.

[63] T. Ling, T. K. Lewellen, and R. S. Miyaoka. Depth of interaction decoding of a continuous crystal detector module. *Physics in Medicine and Biology*, 52(8):2213–2228, April 2007.

[64] Craig S. Levin and Edward J. Hoffman. Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution. *Physics in Medicine and Biology*, 44(3):781–799, March 1999.

[65] Harrison H. Barrett, Craig K. Abbey, and Eric Clarkson. Objective assessment of image quality. III. ROC metrics, ideal observers, and likelihood-generating functions. *Journal of the Optical Society of America A*, 15(6):1520–1535, June 1998.

[66] Luca Caucci. Point detection and Hotelling discriminant: An application in adaptive optics. Master's thesis, University of Arizona, Tucson, AZ, 2007.

[67] Dale Shosa and Leon Kaufman. Methods for evaluation of diagnostic imaging instrumentation. *Physics in Medicine and Biology*, 26(1):101–112, January 1981.

[68] Zhou Wang, Alan C. Bovik, and Ligang Lu. Why is image quality assessment so difficult? In *IEEE International Conference on Acoustics, Speech, & Signal Processing*, volume 4, pages 3313–3316, Orlando, FL, May 2002.

[69] Harrison H. Barrett and Kyle J. Myers. *Foundations of Image Science*. Wiley-Interscience, Hoboken, NJ, 2004.

[70] K. M. Hanson. Method of evaluating image-recovery algorithms based on task performance. *Journal of the Optical Society of America A*, 7(7):1294–1304, July 1990.

[71] H. H. Barrett, T. Gooley, K. Girodias, J. Rolland, T. White, and J. Yao. Linear discriminants and image quality. *Image and Vision Computing*, 10(6):331–463, July/August 1992.

[72] Harrison H. Barrett. Objective assessment of image quality: Effects of quantum noise and object variability. *Journal of the Optical Society of America A*, 7(7):1266–1278, July 1990.

[73] Kyle Jean Myers. *Visual Perception in Correlated Noise*. PhD thesis, University of Arizona, Tucson, AZ, 1985.

[74] David Marvin Green and John A. Swets. *Signal Detection Theory and Psychophysics*. John Wiley and Sons, Inc., New York, NY, 1966.

[75] James O. Berger. *Statistical Decision Theory, Foundations, Concepts, and Methods*. Springer Series in Statistics. Springer-Verlag, New York, NY, 1980.

[76] Lionel Weiss. *Statistical Decision Theory*. McGraw-Hill Series in Probability and Statistics. McGraw-Hill, New York, NY, 1961.

[77] Eric Clarkson and Harrison H. Barrett. Statistical decision theory and tumor detection. In Robin N. Strickland, editor, *Image Processing Techniques for Tumor Detection*, chapter 4, pages 71–100. Marcel Dekker, New York, NY, 2002.

[78] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, January 1763.

[79] David M. Green. Application of detection theory in psychophysics. *Proceedings of the IEEE*, 58(5):713–723, May 1970.

[80] W. L. Rubin. *Radar Detection*. Prentice-Hall Electrical Engineering Series. Prentice-Hall, Englewood Cliffs, NJ, 1968.

[81] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory. Part III, Radar-Sonar Signal Processing and Gaussian Signals in Noise*. John Wiley and Sons, Inc., New York, NY, 2001.

[82] Pascal Borry, Paul Schotsmans, and Kris Dierickx. Evidence-based medicine and its role in ethical decision-making. *Journal of Evaluation in Clinical Practice*, 12(3):306–311, June 2006.

[83] Susan Bailey. Decision making in acute care: A practical framework supporting the 'best interests' principle. *Nursing Ethics*, 13(3):284–291, May 2006.

[84] James P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press Series in Cognition and Perception. Academic Press, New York, NY, 1975.

[85] Robert M. Centor. Signal detectability: The use of ROC curves and their analyses. *Medical Decision Making*, 11(2):102–106, June 1991.

[86] Charles E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, October 1978.

[87] Charles E. Metz. ROC methodology in radiologic imaging. *Investigative Radiology*, 21(9):720–733, September 1986.

[88] Arian R. van Erkel and Peter M. Th. Pattynama. Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology. *European Journal of Radiology*, 27(2):88–94, May 1998.

[89] Lee B. Lusted. Logical analysis in roentgen diagnosis. *Radiology*, 74(2):178–193, February 1960.

[90] Lee B. Lusted. Signal detectability and medical decision-making. *Science*, 171(3977):1217–1219, March 1971.

[91] Nancy A. Obuchowski. Receiver operating characteristic curves and their use in radiology. *Radiology*, 229(1):3–8, October 2003.

[92] James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.

[93] Seong Ho Park, Jin Mo Goo, and Chan-Hee Jo. Receiver operating characteristic (ROC) curve: Practical review for radiologists. *Korean Journal of Radiology*, 5(1):11–18, March 2004.

[94] Kelly H. Zou. Comparison of correlated receiver operating characteristic curves derived from repeated diagnostic test data. *Academic Radiology*, 8(3):225–233, March 2001.

[95] Robert F. Wagner, Sergey V. Beiden, Gregory Campbell, Charles E. Metz, and William M. Sacks. Assessment of medical imaging and computer-assist systems. *Academic Radiology*, 9(11):1264–1277, November 2002.

[96] K. J. Myers, H. H. Barrett, M. C. Borgstrom, D. D. Patton, and G. W. Seeley. Effect of noise correlation on detectability of disk signals in medical imaging. *Journal of the Optical Society of America A*, 2(10):1752–1759, October 1985.

[97] John A. Swets. *Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers*. Scientific Psychology Series. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.

[98] John A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293, June 1988.

[99] David Van Meter and David Middleton. Modern statistical approaches to reception in communication theory. *Transactions of the IRE Professional Group on Information Theory*, 4(4):119–145, September 1954.

[100] W. Peterson, T. Birdsall, and W. Fox. The theory of signal detectability. *Transactions of the IRE Professional Group on Information Theory*, 4(4):171–212, September 1954.

[101] T. Grettenberg. Signal selection in communication and radar systems. *IEEE Transactions on Information Theory*, 9(4):265–275, October 1963.

[102] D. McNicol. *A Primer of Signal Detection Theory*. George Allen & Unwin Ltd., London, England, 1972.

[103] A. J. Simpson and M. J. Fitter. What is the best index of detectability? *Psychological Bulletin*, 80(6):481–488, December 1973.

[104] Wilson P. Tanner, Jr. and T. G. Birdsall. Definitions of d' and η as psychophysical measures. *Journal of the Acoustical Society of America*, 30(10):922–928, October 1958.

[105] David J. Goodenough, Charles E. Metz, and Lee B. Lusted. Caveat on use of the parameter d' for evaluation of observer performance. *Radiology*, 106(3):565–566, March 1973.

[106] John A. Swets. ROC analysis applied to the evaluation of medical imaging techniques. *Investigative Radiology*, 14(2):109–121, March/April 1979.

[107] Luca Caucci, Harrison H. Barrett, Nicholas Devaney, and Jeffrey J. Rodríguez. Application of the Hotelling and ideal observers to detection and localization of exoplanets. *Journal of the Optical Society of America A*, 24(12):B13–B24, December 2007.

[108] H. C. Gilford, R. G. Wells, and M. A. King. A comparison of human observer LROC and numerical observer ROC for tumor detection in SPECT images. *IEEE Transactions on Nuclear Science*, 46(4):1032–1037, August 1999.

[109] Richard G. Swensson. Using localization data from image interpretations to improve estimates of performance accuracy. *Medical Decision Making*, 20(2):170–185, April-June 2000.

[110] Eric Clarkson. Estimation receiver operating characteristic curve and ideal observers for combined detection/estimation tasks. *Journal of the Optical Society of America A*, 24(12):B91–B98, December 2007.

[111] Eric Clarkson and Fangfang Shen. Fisher information and surrogate figures of merit for the task-based assessment of image quality. *Journal of the Optical Society of America A*, 27(10):2313–2326, October 2010.

[112] Carl W. Helstrom. *Statistical Theory of Signal Detection.* Pergamon Press, Oxford, England, 1968.

[113] H. Vincent Poor. *An Introduction to Signal Detection and Estimation.* Springer Texts in Electrical Engineering. Springer-Verlag, New York, NY, 1988.

[114] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory.* Wiley, New York, NY, 1968.

[115] Mourad Barkat. *Signal Detection and Estimation.* Artech House Radar Library. Artech House, Inc., Boston, MA, 1991.

[116] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Boston, MA, 2nd edition, 1990.

[117] Geoffrey J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley Series in Probability and Statistics. Wiley-Interscience, Hoboken, NJ, 2004.

[118] Peter A. Lachenbruch. *Discriminant Analysis.* Hafner Press, New York, NY, 1975.

[119] R. D. Fiete, H. H. Barrett, W. E. Smith, and K. J. Myers. Hotelling trace criterion and its correlation with human-observer performance. *Journal of the Optical Society of America A*, 4(5):945–953, May 1987.

[120] Harold Hotelling. The generalization of Student's ratio. *The Annals of Mathematical Statistics*, 2(3):360–378, August 1931.

[121] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[122] Ronald A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[123] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley and Sons, Inc., New York, NY, 2001.

[124] Harrison H. Barrett, Kyle J. Myers, Nicholas Devaney, and Christopher Dainty. Objective assessment of image quality. IV. Application to adaptive optics. *Journal of the Optical Society of America A*, 23(12):3080–3105, December 2006.

[125] Warren E. Smith and Harrison H. Barrett. Hotelling trace criterion as a figure of merit for the optimization of imaging systems. *Journal of the Optical Society of America A*, 3(5):717–725, May 1986.

[126] Albert N. Shiryaev. *Probability*. Graduate Texts in Mathematics. Springer-Verlag, New York, NY, 2^{nd} edition, 1994.

[127] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Boston, MA, 2002.

[128] B. Roy Frieden. *Probability, Statistical Optics, and Data Testing: A Problem Solving Approach*. Springer, Berlin, Germany, 2001.

[129] B. L. Welch. Note on discriminant functions. *Biometrika*, 31(1/2):218–220, July 1939.

[130] Luca Caucci, Harrison H. Barrett, Nicholas Devaney, and Jeffrey J. Rodríguez. Spatio-temporal Hotelling observer for signal detection from image sequences. *Optics Express*, 17(13):10946–10958, June 2009.

[131] Harrison H. Barrett, Kyle J. Myers, Brandon D. Gallas, Eric Clarkson, and Hongbin Zhang. Megalopinakophobia: Its symptoms and cures. In Larry E. Antonuk and Martin J. Yaffe, editors, *Medical Imaging 2001: Physics of Medical Imaging*, volume 4320, pages 299–307. Proceedings of the SPIE, June 2001.

[132] Kyle J. Myers and Harrison H. Barrett. Addition of a channel mechanism to the ideal-observer model. *Journal of the Optical Society of America*, 4(12):2447–2457, December 1987.

[133] Howard C. Gifford, Michael A. King, Daniel J. de Vries, and Edward J. Soares. Channelized Hotelling and human observer correlation for lesion detection in hepatic SPECT imaging. *Journal of Nuclear Medicine*, 41(3):514–521, March 2000.

[134] Jorge Oldan, Santosh Kulkarni, Yuxiang Xing, Parmeshwar Khurd, and Gene Gindi. Channelized Hotelling and human observer study of optimal smoothing in SPECT MAP reconstruction. *IEEE Transactions on Nuclear Science*, 51(3):733–741, June 2004.

[135] Yani Zhang Binh, T. Pham, and Miguel P. Eckstein. The effect of nonlinear human visual system components on performance of a channelized Hotelling observer in structured backgrounds. *IEEE Transactions on Medical Imaging*, 25(10):1348–1362, October 2006.

[136] Ljiljana Platiša, Bart Goossens, Ewout Vansteenkiste, Subok Park, Brandon D. Gallas, Aldo Badano, and Wilfried Philips. Channelized Hotelling observers for the assessment of volumetric imaging data sets. *Journal of the Optical Society of America*, 28(6):1145–1163, June 2011.

[137] A. N. Kolmogorov and S. V. Fomin. *Elements of the Theory of Functions and Functional Analysis.* Graylock Press, Rochester, NY, 1957.

[138] Ronald A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society A*, 222:309–368, May 1922.

[139] Ronald A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, July 1925.

[140] John Aldrich. R. A. Fisher and the making of maximum likelihood 1912–1922. *Statistical Science*, 12(3):162–176, August 1997.

[141] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[142] Thomas S. Ferguson. *A Course in Large Sample Theory.* Chapman & Hall, London, England, 1996.

[143] Ulf Grenander and Michael I. Miller. *Pattern Theory: From Representation to Inference.* Oxford University Press, Oxford, England, 2007.

[144] Harald Cramér. *Mathematical Methods of Statistics.* Princeton University Press, Princeton, NJ, 1946.

[145] Calyampudi Radakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–89, 1945.

[146] Peter W. Zehna. Invariance of maximum likelihood estimators. *Annals of Mathematical Statistics*, 37(3):744, June 1966.

[147] Peter Tan and Constantin Drossos. Invariance properties of maximum likelihood estimators. *Mathematics Magazine*, 48(1):37–41, January 1975.

[148] Robert V. Hogg, Joseph W. McKean, and Allen T. Craig. *Introduction to Mathematical Statistics*. Pearson Education, Upper Saddle River, NJ, 2005.

[149] D. S. Moore. Maximum likelihood and sufficient statistics. *The American Mathematical Monthly*, 78(1):50–52, January 1971.

[150] V. S. Huzurbazar. The likelihood equation, consistency and the maxima of the likelihood function. *Annals of Human Genetics*, 14(1):185–200, January 1947.

[151] F. Y. Edgeworth. Addendum on “Probable errors of frequency-constants”. *Journal of the Royal Statistical Society*, 72(1):81–90, March 1909.

[152] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis.* Academic Press, London, England, 1979.

[153] Minoru Siotani, Takesi Hayakawa, and Yasunori Fujikoshi. *Modern Multivariate Statistical Analysis: A Graduate Course and Handbook.* American Sciences Press, Columbus, OH, 1985.

[154] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, NY, 1999.

[155] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, London, England, 1981.

[156] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization.* MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, 2009.

[157] P. M. Pardalos and J. B. Rosen. Methods for global concave minimization: A bibliographic survey. *SIAM Review*, 28(3):367–379, September 1986.

[158] Aimo Törn and Antanas Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, Germany, 1989.

[159] Wenyu Sun and Ya-Xiang Yuan. *Optimization Theory and Methods: Nonlinear Programming.* Springer Optimization and its Applications. Springer, New York, NY, 2006.

[160] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Courier Dover Publications, Mineola, NY, 1998.

[161] Harold P. Benson. Deterministic algorithms for constrained concave minimization: A unified critical survey. *Naval Research Logistics*, 43(6):765–795, September 1996.

[162] A. H. G. Rinnooy Kan and G. T. Timmer. Stochastic global optimization methods part I: Clustering methods. *Mathematical Programming*, 39(1):27–56, September 1987.

[163] A. H. G. Rinnooy Kan and G. T. Timmer. Stochastic global optimization methods part II: Multi level methods. *Mathematical Programming*, 39(1):57–78, September 1987.

[164] A. O. Griewank. Generalized descent for global optimization. *Journal of Optimization Theory and Applications*, 34(1):11–39, May 1981.

[165] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[166] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986, March 1984.

[167] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, January 1985.

[168] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[169] Nils Aall Barricelli. Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(35-36):143–182, 1957.

[170] Alex S. Fraser. Simulation of genetic systems by automatic digital computers. I. Introduction. *Australian Journal of Biological Sciences*, 10:484–491, 1957.

[171] David Edward Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Reading, MA, 1989.

[172] Emanuel Falkenauer. *Genetic Algorithms and Grouping Problems*. Wiley, Chichester, NY, 1998.

[173] Jacob Y. Hesterman, Luca Caucci, Matthew A. Kupinski, Harrison H. Barrett, and Lars R. Furenlid. Maximum-likelihood estimation with a contracting-grid search algorithm. *IEEE Transactions on Nuclear Science*, 57(3):1077–1084, June 2010.

[174] Stefano Pedemonte, Alberto Gola, Andrea Abba, and Carlo Fiorini. Optimum real-time reconstruction of gamma events for high resolution Anger camera with the use of GPGPU. In *IEEE Nuclear Science Symposium Conference Record*, pages 3388–3394, Orlando, FL, October/November 2009.

[175] Rolf Sundberg. Maximum likelihood theory for incomplete data from an exponential family. *Scandinavian Journal of Statistics*, 1(2):49–58, 1974.

[176] Rolf Sundberg. An iterative method for solution of the likelihood equations for incomplete data from exponential families. *Communications in Statistics – Simulation and Computation*, 5(1):55–64, 1976.

[177] William Hadley Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, January 1972.

[178] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, June 1974.

[179] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2):113–122, October 1982.

[180] L. A. Shepp, Yehuda Vardi, Jong Beom Ra, Sadek K. Hilal, and Zang Hee Cho. Maximum likelihood PET with real data. *IEEE Transactions on Nuclear Science*, 31(2):910–913, April 1984.

[181] K. Lange and R. Carson. EM reconstruction algorithms for emission and transmission tomography. *Journal of Computer Assisted Tomography*, 8(2):306–316, April 1984.

[182] Y. Vardi, L. A. Shepp, and L. Kaufman. A statistical model for positron emission tomography. *Journal of the American Statistical Association*, 80(389):8–20, March 1985.

[183] Gerd Muehllehner, Joel S. Karp, and Albert Guvenis. A method for reconstructing images from data obtained with a hexagonal bar positron camera. *IEEE Transactions on Medical Imaging*, 4(3):134–138, September 1985.

[184] Kenneth Lange, Mark Bahn, and Roderick Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *IEEE Transactions on Medical Imaging*, 6(2):106–114, June 1987.

[185] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, December 1906.

[186] Russell A. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 45(1):47–50, 1983.

[187] C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, March 1983.

[188] A. J. Rockmore and Albert Macovski. A maximum likelihood approach to emission image reconstruction from projections. *IEEE Transactions on Nuclear Science*, 23(4):1428–1432, August 1976.

[189] A. J. Rockmore and Albert Macovski. A maximum likelihood approach to transmission image reconstruction from projections. *IEEE Transactions on Nuclear Science*, 24(3):1929–1935, June 1977.

[190] Harrison H. Barrett, Donald W. Wilson, and Benjamin M. W. Tsui. Noise properties of the EM algorithm. I. Theory. *Physics in Medicine and Biology*, 39(5):833–846, May 1994.

[191] Donald W. Wilson, Benjamin M. W. Tsui, and Harrison H. Barrett. Noise properties of the EM algorithm. II. Monte Carlo simulations. *Physics in Medicine and Biology*, 39(5):847–871, May 1994.

[192] Eric Clarkson and Harrison Barrett. A bound on null functions for digital imaging systems with positivity constraints. *Optics Letters*, 22(11):814–815, June 1997.

[193] Eric Clarkson and Harrison Barrett. Bounds on null functions of linear digital imaging systems. *Journal of the Optical Society of America A*, 15(5):1355–1360, May 1998.

[194] Craig Kendall Abbey. *Assessment of Reconstructed Images.* PhD thesis, University of Arizona, Tucson, AZ, 1998.

[195] Donald L. Snyder, Michael I. Miller, Lewis J. Thomas, and David G. Politte. Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography. *IEEE Transactions on Medical Imaging*, 6(3):228–238, September 1987.

[196] Lawrence S. Joyce and William L. Root. Precision bounds in superresolution processing. *Journal of the Optical Society of America A*, 1(2):149–168, February 1984.

[197] Llacer Jorge and Veklerov Eugene. Feasible images and practical stopping rules for iterative algorithms in emission tomography. *IEEE Transactions on Medical Imaging*, 8(2):186–193, June 1989.

[198] Nicolai Bissantz, Bernard A. Mair, and Axel Munk. A statistical stopping rule for MLEM reconstructions in PET. In *IEEE Nuclear Science Symposium Conference Record*, pages 4198–4200, Dresden, Germany, October 2008.

[199] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302):157–175, 1900.

[200] Herman Chernoff and E. L. Lehmann. The use of maximum likelihood estimates in χ^2 tests for goodness of fit. *Annals of Mathematical Statistics*, 25(3):579–586, September 1954.

[201] Αναστασιος Δ. Γαϊτανης. *Ανάπτυξη κριτηρίων παύσης των αλγορίθμων MLEM και OSEM που χρησιμοποιούνται στην ανακατασκϵυή ϵικόνας σϵ PET*. PhD thesis, University of Patras, Patras, Greece, 2010. English translation: Anastasios D. Gaitanis, *Development of stopping rule methods for the MLEM and OSEM algorithms used in PET image reconstruction*.

[202] Anastasios Gaitanis, George Kontaxakis, George Spyrou, George Panayiotakis, and George Tzanakos. PET image reconstruction: A stopping rule for the MLEM algorithm based on properties of the updating coefficients. *Computerized Medical Imaging and Graphics*, 34(2):131–141, March 2010.

[203] T. A. Gooley and H. H. Barrett. Evaluation of statistical methods of image reconstruction through ROC analysis. *IEEE Transactions on Medical Imaging*, 11(2):276–283, June 1992.

[204] H. Malcolm Hudson and Richard S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Transactions on Medical Imaging*, 13(4):601–609, December 1994.

[205] Harrison H. Barrett, Lars R. Furenlid, Melanie Freed, Jacob Y. Hesterman, Matthew A. Kupinski, Eric Clarkson, and Meredith K. Whitaker. Adaptive SPECT. *IEEE Transactions on Medical Imaging*, 27(6):775–788, June 2008.

[206] Hal O. Anger. A new instrument for mapping gamma-ray emitters. Biology and Medicine Quarterly Report UCRL-3653, University of California Radiation Laboratory, Berkeley, CA, January 1957.

[207] Hal O. Anger. Scintillation camera. *Review of Scientific Instruments*, 29(1):27–33, January 1958.

[208] Hal O. Anger. Scintillation camera with multichannel collimators. *Journal of Nuclear Medicine*, 5(7):515–532, July 1964.

[209] G. Kulberg, N. van Dijk, and G. Muehllehner. Improved resolution of the Anger scintillation camera through the use of threshold preamplifiers. *Journal of Nuclear Medicine*, 13(2):169–171, February 1972.

[210] R. Pani, F. Vittorini, M. N. Cinti, P. Bennati, R. Pellegrini, S. Ridolfi, R. Scafè, S. Lo Meo, M. Mattioli, F. L. Navarria, G. Moschini, A. Fabbri, E. D'Abramo, V. Orsolini Cencelli, and D. Sacco. Revisited position arithmetics for $\mathrm{LaBr_3}$:Ce continuous crystals. *Nuclear Physics B – Proceedings Supplements*, 197(1):383–386, December 2009.

[211] Thomas Dean Milster. *Design and Construction of a Modular Gamma Camera*. PhD thesis, University of Arizona, Tucson, AZ, 1987.

[212] T. D. Milster, J. N. Aarsvold, H. H. Barrett, A. L. Landesman, L. S. Mar, D. D. Patton, T. J. Roney, R. K. Rowe, and R. H. Seacat III. A full-field modular gamma camera. *Journal of Nuclear Medicine*, 31(5):632–639, May 1990.

[213] T. Kamae, N. Hanada, and R. Enomoto. Prototype design of multiple compton gamma-ray camera. *IEEE Transactions on Nuclear Science*, 35(1):352–355, February 1988.

[214] P. A. Schlosser, D. W. Miller, M. S. Gerber, R. F. Redmond, J. W. Harpster, W. J. Collis, and W. W. Hunter. A practical gamma-ray camera system using high-purity germanium. *IEEE Transactions on Nuclear Science*, 21(1):658–664, February 1974.

[215] Manbir Singh. An electronically collimated gamma camera for single photon emission computed tomography. Part I: Theoretical considerations and design criteria. *Medical Physics*, 10(4):421–427, July 1983.

[216] Manbir Singh and David Doria. An electronically collimated gamma camera for single photon emission computed tomography. Part II: Image reconstruction and preliminary experimental measurements. *Medical Physics*, 10(4):428–435, July 1983.

[217] F. Sánchez, J. M. Benlloch, B. Escat, N. Pavón, E. Porras, D. Kadi-Hanifi, J. A. Ruiz, F. J. Mora, and A. Sebastià. Design and tests of a portable mini gamma camera. *Medical Physics*, 31(6):1384–1397, June 2004.

[218] C. Mestais, N. Baffert, J. P. Bonnefoy, A. Chapuis, A. Koenig, O. Monnet, P. Ouvrier Buffet, J. P. Rostaing, F. Sauvage, and L. Verger. A new design for a high resolution, high efficiency CZT gamma camera detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 458(1/2):62–67, February 2001.

[219] John David Sain. *Optical Modeling, Design Optimization, and Performance Analysis of a Gamma Camera for Detection of Breast Cancer*. PhD thesis, University of Arizona, Tucson, AZ, 2001.

[220] W. P. Klein, H. H. Barrett, I. W. Pang, D. D. Patton, M. M. Rogulski, J. D. Sain, and W. E. Smith. FASTSPECT: electrical and mechanical design of a high-resolution dynamic SPECT imager. In *Nuclear Science Symposium and Medical Imaging Conference Record*, pages 931–933, San Francisco, CA, October 1995.

[221] Kenneth S. Krane and David Halliday. *Introductory Nuclear Physics*. John Wiley and Sons, Inc., New York, NY, 1988.

[222] Harrison H. Barrett and William Swindell. *Radiological Imaging: The Theory of Image Formation, Detection, and Processing.* Academic Press, New York, NY, 1981.

[223] John S. Lilley. *Nuclear Physics: Principles and Applications.* John Wiley and Sons, Inc., New York, NY, 2001.

[224] Arthur H. Compton. A quantum theory of the scattering of X-rays by light elements. *Physical Review*, 21(5):483–502, May 1923.

[225] Paolo Christillin. Nuclear Compton scattering. *Journal of Physics G: Nuclear Physics*, 12(9):837–851, September 1986.

[226] O. Klein and Y. Nishina. Über die streuung von strahlung durch freie elektronen nach der neuen relativistischen quantendynamik von Dirac. *Zeitschrift für Physik A Hadrons and Nuclei*, 52(11/12):853–868, November 1929.

[227] C. J. Everett and E. D. Cashwell. A new method of sampling the Klein-Nishina probability distribution for all incident photon energies above 1 keV. Technical Report LA-7188-MS, Department of Energy, Los Alamos Scientific Laboratory, Los Alamos, NM, March 1978.

[228] Emin N. Özmutlu. Sampling of angular distribution in Compton scattering. *International Journal of Radiation Applications and Instrumentation. Part A. Applied Radiation and Isotopes*, 43(6):713–715, June 1992.

[229] Herman Kahn. Applications of Monte Carlo. Technical Report RM-1237-AEC, RAND Corporation, Santa Monica, CA, April 1956.

[230] Mark T. Madsen. Scintillation detectors and scintillation detector counting systems. In Robert E. Henkin, editor, *Nuclear Medicine*, volume 2, chapter 6, pages 78–89. Mosby Elsevier, Philadelphia, PA, 2$^{\text{nd}}$ edition, 2006.

[231] Charles Kittel. *Introduction to Solid State Physics.* Wiley, New York, NY, 2005.

[232] Nasser Peyghambarian, Stephan W. Koch, and Andre Mysyrowicz. *Introduction to Semiconductor Optics.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[233] Jennifer Prekeges. *Nuclear Medicine Instrumentation.* Jones and Bartlett Publishers, Sudbury, MA, 2010.

[234] R. L. Heath, R. Hofstadter, and E. B. Hughes. Inorganic scintillators: A review of techniques and applications. *Nuclear Instruments and Methods*, 162(1-3):431–476, June 1979.

[235] I. Holl, E. Lorenz, and G. Mageras. A measurement of the light yield of common inorganic scintillators. *IEEE Transactions on Nuclear Science*, 35(1):105–109, February 1998.

[236] W. Mannhart and H. Vonach. Gamma-ray absorption coefficients for NaI(Tl). *Nuclear Instruments and Methods*, 134(2):347–351, April 1976.

[237] Eiji Sakai. Recent measurements on scintillator-photodetector systems. *IEEE Transactions on Nuclear Science*, 34(1):418–422, February 1987.

[238] Tom K. Lewellen. Recent developments in PET detector technology. *Physics in Medicine and Biology*, 53(17):R287–R317, September 2008.

[239] P. Dorenbos, J. T. M. de Haas, C. W. E. van Eijk, C. L. Melcher, and J. S. Schweitzer. Nonlinear response in the scintillation yield of Lu_2SiO_5:Ce^{3+}. *IEEE Transactions on Nuclear Science*, 41(4):735–737, August 1994.

[240] Carel W. E. van Eijk. Inorganic-scintillator development. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 460(1):1–14, March 2001.

[241] M. Moszyński, M. Kapusta, M. Mayhugh, D. Wolski, and S. O. Flyckt. Absolute light output of scintillators. *IEEE Transactions on Nuclear Science*, 44(3):1052–1061, June 1997.

[242] Stephen E. Derenzo and William W. Moses. Experimental efforts and results in finding new heavy scintillators. Technical Report LBL-33295, Lawrence Berkeley Laboratory, Berkeley, CA, September 1992.

[243] R. Lecomte, J. Cadorette, S. Rodrigue, D. Lapointe, D. Rouleau, M. Bentourkia, R. Yao, and P. Msaki. Initial results from the Sherbrooke avalanche photodiode positron tomograph. *IEEE Transactions on Nuclear Science*, 43(3):1952–1957, June 1996.

[244] I. Braun, S. C. Commichau, M. Rissi, M. Backes, A. Biland, T. Bretz, I. Britvitch, V. Commichau, H. von Gunten, D. Hildebrand, U. Horisberger, D. Kranich, E. Lorenz, W. Lustermann, K. Mannheim, D. Neise, F. Pauss, M. Pohl, D. Renker, W. Rhode, U. Röser, U. Straumann, and G. Viertel. First avalanche-photodiode camera test (FACT): A novel camera using G-APDs for the observation of very high-energy γ-rays with Cherenkov telescopes. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 610(1):400–403, October 2009.

[245] Harrison H. Barrett. Detectors for small-animal SPECT II. In Matthew A. Kupinski and Harrison H. Barrett, editors, *Small-Animal SPECT Imaging*, chapter 3, pages 49–86. Springer, New York, NY, 2005.

[246] U. Fano. Ionization yield of radiations. II. The fluctuations of the number of ions. *Physical Review*, 72(1):26–29, July 1947.

[247] Abdelkader Bousselham, Harrison H. Barrett, Vaibhav Bora, and Kanai Shah. Photoelectron anticorrelations and sub-Poisson statistics in scintillation detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 620(2-3):359–362, August 2010.

[248] Donald L. Snyder and David G. Politte. Image reconstruction from list-mode data in an emission tomography system having time-of-flight measurements. *IEEE Transactions on Nuclear Science*, 30(3):1843–1849, June 1983.

[249] Frank T. Solmitz. Analysis of experiments in particle physics. *Annual Review of Nuclear Science*, 14:375–402, December 1964.

[250] K. A. Bostroem, A. Aloisi, R. Bohlin, R. Diaz, V. Dixon, P. Goudfrooij, P. Hodge, D. Lennon, C. Long, S. Niemi, R. Osten, C. Proffitt, N. Walborn, T. Wheeler, M. Wolfe, B. York, and W. Zheng. *STIS Instrument Handbook, Version 10.0.* Space Telescope Science Institute, Baltimore, MD, December 2010.

[251] Lars R. Furenlid. Private communication, September 2011.

[252] Farhad Taghibakhsh, Sarah G. Cuddy, and John A. Rowlands. Novel methods of resolving energy and 3D positions of interactions in monolithic scintillator plates. In *IEEE Nuclear Science Symposium Conference Record*, pages 2549–2552, Knoxville, TN, October/November 2010.

[253] Jacob Yost Hesterman. *The Multi-Module Multi-Resolution SPECT System: A Tool for Variable-Pinhole Small-Animal Imaging.* PhD thesis, University of Arizona, Tucson, AZ, 2007.

[254] Robert S. Miyaoka, Xiaoli Li, Cate Lockhart, and Tom K. Lewellen. Design of a high resolution, monolithic crystal, PET/MRI detector with DOI positioning capability. In *IEEE Nuclear Science Symposium Conference Record*, pages 4688–4692, Dresden, Germany, October 2008.

[255] Peter Bruyndonckx, Cedric Lemaître, Dennis Schaart, Marnix Maas, D. J. (jan) van der Laan, Magalie Krieguer, Olivier Devroede, and Stefaan Tavernier. Towards a continuous crystal APD-based PET detector design. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 571(1/2):182–186, February 2007.

[256] E. Tanaka, T. Hiramoto, and N. Nohara. Scintillation cameras based on new position arithmetics. *Journal of Nuclear Medicine*, 11(9):542–547, September 1970.

[257] T. D. Milster, L. A. Selberg, H. H. Barrett, A. L. Landesman, and R. H. Seacat III. Digital position estimation for the modular scintillation camera. *IEEE Transactions on Nuclear Science*, 32(1):748–752, February 1985.

[258] Lucas C. Parra and Harrison H. Barrett. List-mode likelihood: EM algorithm and image quality estimation demonstrated on 2-D PET. *IEEE Transactions on Medical Imaging*, 17(2):228–235, April 1998.

[259] Andrew J. Reader, Roido Manavaki, Sha Zhao, Peter J. Julyan, David L. Hastings, and Jamal Zweit. Accelerated list-mode EM algorithm. *IEEE Transactions on Nuclear Science*, 49(1):42–49, February 2002.

[260] Parmeshwar Khurd, Ing-Tsung Hsiao, Anand Rangarajan, and Gene Gindi. A globally convergent regularized ordered-subset EM algorithm for list-mode reconstruction. *IEEE Transactions on Nuclear Science*, 51(3):719–725, June 2004.

[261] Harrison H. Barrett, Timothy White, and Lucas C. Parra. List-mode likelihood. *Journal of the Optical Society of America A*, 14(11):2914–2923, November 1997.

[262] Oliver C. Ibe. *Fundamentals of Applied Probability and Random Processes*. Elsevier Academic Press, Burlington, MA, 2005.

[263] Ciriaco Valdez-Flores Richard M. Feldman. *Applied Probability and Stochastic Processes*. Springer-Verlag, Berlin, Germany, 2010.

[264] A. J. Reader, K. Erlandsson, M. A. Flower, and R. J. Ott. Fast accurate iterative reconstruction for low-statistics positron volume imaging. *Physics in Medicine and Biology*, 43(4):835–846, April 1998.

[265] Ronald H. Huesman, Gregory J. Klein, William W. Moses, Jinyi Qi, Bryan W. Reutter, and Patrick R. G. Virador. List-mode maximum-likelihood reconstruction applied to positron emission mammography (PEM) with irregular sampling. *IEEE Transactions on Medical Imaging*, 19(5):532–537, May 2000.

[266] Charles Byrne. Likelihood maximization for list-mode emission tomographic image reconstruction. *IEEE Transactions on Medical Imaging*, 20(10):1084–1092, October 2001.

[267] L. Bouwens, R. Van de Walle, H. Gifford, M. King, I. Lemahieu, and R. A. Dierckx. LMIRA: List-mode iterative reconstruction algorithm for SPECT. *IEEE Transactions on Nuclear Science*, 48(4):1364–1370, August 2001.

[268] Luca Caucci, Lars R. Furenlid, and Harrison H. Barrett. Maximum likelihood event estimation and list-mode image reconstruction on GPU hardware. In *IEEE Nuclear Science Symposium Conference Record*, pages 4072–4076, Orlando, FL, October 2009.

[269] Jinyi Qi and Ronald H. Huesman. Propagation of errors from the sensitivity image in list mode reconstruction. *IEEE Transactions on Medical Imaging*, 23(9):1094–1099, September 2004.

[270] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains.* Springer-Verlag, New York, NY, 1976.

[271] Samuel Karlin and Howard M. Taylor. *A First Course in Stochastic Processes.* Academic Press, New York, NY, 1975.

[272] Kai Lai Chung. *Markov Chains with Stationary Transition Probabilities.* Springer, Berlin, Germany, 1967.

[273] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability.* Springer-Verlag, London, England, 1993.

[274] Esa Nummelin. *General Irreducible Markov Chains and Non-Negative Operators.* Cambridge University Press, Cambridge, England, 1984.

[275] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[276] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, November 1995.

[277] Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, September 1993.

[278] Adrian E. Raftery and Steven M. Lewis. Implementing MCMC. In Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, chapter 7. Chapman & Hall/CRC, Boca Raton, FL, 1996.

[279] Harrison H. Barrett, Christopher Dainty, and David Lara. Maximum-likelihood methods in wavefront sensing: stochastic models and likelihood functions. *Journal of the Optical Society of America A*, 24(2):391–414, February 2007.

[280] James C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. John Wiley and Sons, Inc., Hoboken, NJ, 2003.

[281] Peter J. Bickel and Kjell A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall, Upper Saddle River, NJ, 2001.

[282] James L. Melsa and David L. Cohn. *Decision and Estimation Theory*. McGraw-Hill, New York, NY, 1978.

[283] Andre Lehovich. *List-mode SPECT Reconstruction Using Empirical Likelihood*. PhD thesis, University of Arizona, Tucson, AZ, 2005.

[284] P. A. M. Dirac. *The Principles of Quantum Mechanics*. International Series of Monographs on Physics. Oxford University Press, Oxford, England, 1958.

[285] J. P. Rolland and H. H. Barrett. Effect of random background inhomogeneity on observer detection performance. *Journal of the Optical Society of America A*, 9(5):649–658, May 1992.

[286] Jannick Paule Yvette Rolland. *Factors Influencing Lesion Detection in Medical Imaging*. PhD thesis, University of Arizona, Tucson, AZ, 1990.

[287] Z. X. Wang and D. R. Guo. *Special Functions*. World Scientific, Singapore, 1989.

[288] A. P. Prudnikov, Yu. A. Brychkov, and O. I. Marichev. *Integrals and Series*, volume 1. Gordon and Breach Science Publishers, New York, NY, 1986.

[289] Alexander Apelblat. *Table of Definite and Infinite Integrals*. Elsevier Scientific Pub. Co., Amsterdam, Netherlands, 1983.

[290] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, New York, NY, 1980.

[291] Richard Beals and Roderick Wong. *Special Functions: A Graduate Text*. Cambridge University Press, Cambridge, England, 2010.

[292] George E. Andrews, Richard Askey, and Ranjan Roy. *Special Functions*. Cambridge University Press, Cambridge, England, 2000.

[293] J. C. Dainty and Rodney Shaw. *Image Science: Principles, Analysis and Evaluation of Photographic-Type Imaging Processes*. Academic Press, London, England, 1974.

[294] Sol M. Gruner, J. R. Milch, and Geo. T. Reynolds. Evaluation of area photon detectors by a method based on detective quantum efficiency (DQE). *IEEE Transactions on Nuclear Science*, 25(1):562–565, February 1978.

[295] C. E. Dick and J. W. Motz. Image information transfer properties of x-ray fluorescent screens. *Medical Physics*, 8(3):337–346, May/June 1981.

[296] Albert Rose. The sensitivity performance of the human eye on an absolute scale. *Journal of the Optical Society of America*, 38(2):196–208, February 1948.

[297] Harrison H. Barrett, Jie Yao, Jannick P. Rolland, and Kyle J. Myers. Model observers for assessment of image quality. *Proceedings of the National Academy of Sciences of the United States of America*, 90(21):9758–9765, November 1993.

[298] Matthew A. Kupinski, John W. Hoppin, Eric Clarkson, and Harrison H. Barrett. Ideal-observer computation in medical imaging with use of Markov-chain Monte Carlo techniques. *Journal of the Optical Society of America A*, 20(3):430–438, March 2003.

[299] Matthew A. Kupinski. Private communication, October 2011.

[300] Eric Clarkson, Matthew A. Kupinski, and John W. Hoppin. Assessing the accuracy of estimates of the likelihood ratio. In Dev P. Chakraborty and Elizabeth A. Krupinski, editors, *Medical Imaging 2003: Image Perception, Observer Performance, and Technology Assessment*, volume 5034, pages 135–143. Proceedings of the SPIE, May 2003.

[301] Eric Clarkson and Harrison H. Barrett. Approximations to ideal-observer performance on signal-detection tasks. *Applied Optics*, 39(11):1783–1793, April 2000.

[302] Eric Clarkson. Bounds on the area under the receiver operating characteristic curve for the ideal observer. *Journal of the Optical Society of America A*, 19(10):1963–1968, October 2002.

[303] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, December 1945.

[304] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, March 1947.

[305] Jeffrey H. Shapiro. Bounds on the area under the ROC curve. *Journal of the Optical Society of America A*, 16(1):53–57, January 1999.

[306] Solomon Kullback. *Information theory and statistics.* Courier Dover Publications, Mineola, NY, 1997.

[307] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., New York, NY, 1990.

[308] Arthur E. Burgess. Comparison of receiver operating characteristic and forced choice observer performance measurement methods. *Medical Physics*, 22(5):643–655, May 1995.

[309] Eric Clarkson, Matthew A. Kupinski, and Harrison H. Barrett. A probabilistic model for the MRMC method, part 1: Theoretical development. *Academic radiology*, 13(11):1410–1421, November 2006.

[310] Matthew A. Kupinski, Eric Clarkson, and Harrison H. Barrett. A probabilistic model for the MRMC method, part 2: Validation and applications. *Academic radiology*, 13(11):1422–1430, November 2006.

[311] Harrison H. Barrett, Matthew A. Kupinski, and Eric Clarkson. Probabilistic foundations of the MRMC method. In Miguel P. Eckstein and Yulei Jiang, editors, *Medical Imaging 2005: Image Perception, Observer Performance, and Technology Assessment*, volume 5749, pages 21–31. Proceedings of the SPIE, April 2005.

[312] Brandon D. Gallas. One-shot estimate of MRMC variance: AUC. *Academic radiology*, 13(3):353–362, March 2006.

[313] Sergei Winitzki. A handy approximation for the error function and its inverse. Unpublished, February 2008.

[314] Ivar Stakgold. *Green's Functions and Boundary Value Problems.* Wiley, New York, NY, 1979.

[315] Harrison H. Barrett. Private communication, March 2011.

[316] Arthur E. Albert. *Regression and the Moore-Penrose Pseudoinverse.* Academic Press, New York, NY, 1972.

[317] Luca Caucci, William C. J. Hunter, Lars R. Furenlid, and Harrison H. Barrett. List-mode MLEM image reconstruction from 3D ML position estimates. In *IEEE Nuclear Science Symposium Conference Record*, pages 2643–2647, Knoxville, TN, October/November 2010.

[318] Harrison H. Barrett, J. L. Denny, Robert F. Wagner, and Kyle J. Myers. Objective assessment of image quality. II. Fisher information, Fourier crosstalk, and figures of merit for task performance. *Journal of the Optical Society of America A*, 12(5):834–852, May 1995.

[319] David E. Culler, Jaswinder Pal Singh, and Anoop Gupta. *Parallel Computer Architecture: A Hardware/Software Approach.* Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[320] Kai Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability.* McGraw-Hill, New York, NY, 1993.

[321] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolić. *Digital Integrated Circuits: A Design Perspective.* Pearson Education, Upper Saddle River, NJ, 2003.

[322] Kai Hwang and Zhiwei Xu. *Scalable Parallel Computing: Technology, Architecture, Programming.* WCB/McGraw-Hill, Boston, MA, 1998.

[323] Lina J. Karam, Ismail AlKamal, Alan Gatherer, Gene A. Frantz, David V. Anderson, and Brian L. Evans. Trends in multi-core DSP platforms. *IEEE Signal Processing Magazine*, 26(6):38–49, November 2009.

[324] Daniel E. Lenoski and Wolf-Dietrich Weber. *Scalable Shared-Memory Multiprocessing.* Morgan Kaufmann Publishers, San Francisco, CA, 1995.

[325] Peter S. Pacheco. *An Introduction to Parallel Programming.* Morgan Kaufmann Publishers, San Francisco, CA, 2011.

[326] Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming.* Elsevier/Morgan Kaufmann, Amsterdam, Netherlands, 2008.

[327] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, Atlantic City, NJ, April 1967. AFIPS Press.

[328] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy. Introduction to the Cell Multiprocessor. *IBM Journal of Research and Development*, 49(4/5):589–604, July/September 2005.

[329] H. Peter Hofstee. Introduction to the Cell Broadband Engine. Technical report, IBM Corporation, Riverton, NJ, 2005.

[330] Samuel Williams, John Shalf, Leonid Oliker, Shoaib Kamil, Parry Husbands, and Katherine Yelick. The potential of the Cell Processor for scientific computing. In *Proceedings of the 3^{rd} Conference on Computing Frontiers*, pages 9–20, Ischia, Italy, May 2006. Association for Computing Machinery.

[331] David A. Bader, Virat Agarwal, Kamesh Madduri, and Seunghwa Kang. High performance combinatorial algorithm design on the Cell Broadband Engine processor. *Parallel Computing*, 33(10/11):720–740, November 2007.

[332] Ingo Wald Carsten Benthin, Michael Scherbaum, and Heiko Friedrich. Ray tracing on the Cell Processor. In *IEEE Symposium on Interactive Ray Tracing*, pages 15–23, Salt Lake City, UT, September 2006.

[333] David A. Bader, Virat Agarwal, and Seunghwa Kang. Computing discrete transforms on the Cell Broadband Engine. *Parallel Computing*, 35(3):119–137, March 2009.

[334] Marc Kachelrieß, Michael Knaup, and Oliver Bockenbach. Hyperfast parallel-beam and cone-beam backprojection using the Cell general purpose hardware. *Medical Physics*, 34(4):1474–1486, April 2007.

[335] M. Sakamoto and M. Murase. A parallel implementation of 3-D CT image reconstruction on the Cell Broadband Engine. *International Journal of Adaptive Control Signal Process*, 24(2):117–127, February 2010.

[336] M. Sakamoto and M. Murase. Parallel implementation for 3-D CT image reconstruction on Cell Broadband Engine™. In *IEEE International Conference on Multimedia and Expo*, pages 276–279, July 2007.

[337] Dac C. Pham, Tony Aipperspach, David Boerstler, Mark Bolliger, Rajat Chaudhry, Dennis Cox, Paul Harvey, Paul M. Harvey, H. Peter Hofstee, Charles Johns, Jim Kahle, Atsushi Kameyama, John Keaty, Yoshio Masubuchi, Mydung Pham, Jürgen Pille, Stephen Posluszny, Mack Riley, Daniel L. Stasiak, Masakazu Suzuoki, Osamu Takahashi, James Warnock, Stephen Weitzel, Dieter Wendel, and Kazuaki Yazawa. Overview of the architecture, circuit design, and physical implementation of a first-generation Cell Processor. *IEEE Journal of Solid-State Circuits*, 41(1):179–196, January 2006.

[338] *Cell Broadband Engine Programming Handbook*, May 2008.

[339] Michael D. McCool. Data-parallel programming on the Cell BE and the GPU using the RapidMind Development Platform. In *Proceedings of the GSPx Multicore Applications Conference*, Santa Clara, CA, October/November 2006.

[340] *NVIDIA CUDA C Programming Guide*, April 2012. Version 4.2.

[341] Tom R. Halfhill. Parallel processing with CUDA. *Microprocessor Journal*, January 2008.

[342] Guillem Pratx, Garry Chinn, Peter D. Olcott, and Craig S. Levin. Fast, accurate and shift-varying line projections for iterative reconstruction using the GPU. *IEEE Transactions on Medical Imaging*, 28(3):435–445, March 2009.

[343] Sanjiv S. Samant, Junyi Xia, Pınar Muyan-Özçelik, and John D. Owens. High performance computing for deformable image registration: Towards a new paradigm in adaptive radiotherapy. *Medical Physics*, 35(8):3546–3553, August 2008.

[344] Wenyuan Bi, Zhiqiang Chen, Li Zhang, and Yuxiang Xing. Accelerate helical cone-beam CT with graphics hardware. In Jiang Hsieh and Ehsan Samei, editors, *Medical Imaging 2008: Physics of Medical Imaging*, volume 6913, page 69132T. Proceedings of the SPIE, March 2008.

[345] Christoph Vetter and Rüdiger Westermann. SPECT reconstruction on the GPU. In Jiang Hsieh and Ehsan Samei, editors, *Medical Imaging 2008: Physics of Medical Imaging*, volume 6913, page 69132R. Proceedings of the SPIE, March 2008.

[346] Friedemann Rößler, Ralf P. Botchen, and Thomas Ertl. Dynamic shader generation for GPU-based multi-volume ray casting. *IEEE Computer Graphics and Applications*, 28(5):66–77, September/October 2008.

[347] Thomas Schiwietz, Ti chiun Chang, Peter Speier, and Rüdiger Westermann. MR image reconstruction using the GPU. In Michael J. Flynn and Jiang Hsieh, editors, *Medical Imaging 2006: Physics of Medical Imaging*, volume 6142, page 61423T. Proceedings of the SPIE, March 2006.

[348] Jingyu Cui, Guillem Pratx, Sven Prevrhal, Lingxiong Shao, and Craig S. Levin. Fully 3-D list-mode positron emission tomography image reconstruction on GPU using CUDA. In *IEEE Nuclear Science Symposium Conference Record*, pages 2635–2637, Knoxville, TN, October/November 2010.

[349] S. Pedemonte, A. Bousse, K. Erlandsson, M. Modat, S. Arridge, B. F. Hutton, and S. Ourselin. GPU accelerated rotation-based emission tomography reconstruction. In *IEEE Nuclear Science Symposium Conference Record*, pages 2657–2661, Knoxville, TN, October/November 2010.

[350] Timothy J. Purcell, Ian Buck, William R. Mark, and Pat Hanrahan. Ray tracing on programmable graphics hardware. *ACM Transactions on Graphics*, 21(3):703–712, July 2002.

[351] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, March 2007.

[352] Takashi Kanai. Fragment-based evaluation of non-uniform B-spline surfaces on GPUs. *Computer-Aided Design and Applications*, 4(1-4):287–294, 2007.

[353] Daniel Ruijters, Bart M. ter Haar Romeny, and Paul Suetens. Efficient GPU-based texture interpolation using uniform B-splines. *Journal of Graphics, GPU, and Game Tools*, 13(4):61–69, 2008.

[354] Stephen H. Unger. Hazards, critical races, and metastability. *IEEE Transactions on Computers*, 44(6):754–768, June 1995.

[355] W. C. J. Hunter, H. H. Barrett, L. R. Furenlid, and S. K. Moore. Method of calibrating response statistics for ML estimation of 3D interaction position in a thick-detector gamma camera. In *IEEE Nuclear Science Symposium Conference Record*, pages 4359–4363, Honolulu, HI, October/November 2007.

[356] O. Barndorff-Nielsen. On a formula for the distribution of the maximum likelihood estimator. *Biometrika*, 70(2):343–365, August 1983.

[357] Craig K. Abbey, Eric Clarkson, Harrison H. Barrett, Stefan P. Müller, and Frank J. Rybicki. A method for approximating the density of maximum-likelihood and maximum *a posteriori* estimates under a Gaussian noise model. *Medical Image Analysis*, 2(4):395–403, December 1998.

[358] Lars R. Furenlid. Private communication, July 2012.

[359] M. J. Berger, J. H. Hubbell, S. M. Seltzer, J. Chang, J. S. Coursey, R. Sukumar, D. S. Zucker, and K. Olsen. XCOM: Photon cross sections database, 1998. `http://www.nist.gov/pml/data/xcom/`.

[360] Robert L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, 12(2):252–255, March/April 1985.

[361] Andrew J. Reader, Stijn Ally, Filippos Bakatselos, Roido Manavaki, Richard J. Walledge, Peter J. Julyan Alan P. Jeavons, Sha Zhao, David L. Hastings, and Jamal Zweit. One-pass list-mode EM algorithm for high-resolution 3-D PET image reconstruction into large arrays. *IEEE Transactions on Nuclear Science*, 49(3):693–699, June 2002.

[362] Ronald H. Huesman, Jinyi Qi, Dimitris Samaras, Yang Wang, Liu Yang, Wei Zhang, Richard Ferrieri, Paul Vaska, and David Schlyer. Direct list mode reconstruction for motion compensation. In *IEEE Nuclear Science Symposium Conference Record*, pages 3953–3955, Rome, Italy, October 2004.

[363] Richard E. Carson, W. Craig Barker, Jeih-San Liow, and Calvin A. Johnson. Design of a motion-compensation OSEM list-mode algorithm for resolution-recovery reconstruction for the HRRT. In *IEEE Nuclear Science Symposium Conference Record*, pages 3281–3285, Portland, OR, October 2003.

[364] Roger Fulton, Ingo Nickel, Lutz Tellmann, Steven Meikle, Uwe Pietrzyk, and Hans Herzog. Event-by-event motion compensation in 3D PET. In *IEEE Nuclear Science Symposium Conference Record*, pages 3286–3289, Portland, OR, October 2003.

[365] Arman Rahmim, Peter Bloomfield, Sylvain Houle, Mark Lenox, Christian Michel, Kenneth R. Buckley, Thomas J. Ruth, and Vesna Sossi. Motion compensation in histogram-mode and list-mode EM reconstructions: Beyond the event-driven approach. *IEEE Transactions on Nuclear Science*, 51(5):2588–2596, October 2004.

[366] A. Rahmim, M. Lenox, A. J. Reader, C. Michel, Z. Burbar, T. J. Ruth, and V. Sossi. Statistical list-mode image reconstruction for the high resolution research tomograph. *Physics in Medicine and Biology*, 49(18):4239–4258, September 2004.

[367] Robert M. Lewitt and Gerd Muehllehner. Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimation. *IEEE Transactions on Medical Imaging*, 5(1):16–22, March 1986.

[368] Filip Jacobs, Erik Sundermann, Bjorn De Sutter, Mark Christiaens, and Ignace Lemahieu. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. *Journal of Computing and Information Technology*, 6(1):89–94, 1998.

[369] John Michael Hammersley and David Christopher Handscomb. *Monte Carlo Methods*. Methuen & Co., Ltd., London, England, 1964.

[370] George S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, New York, NY, 1995.

[371] Ido Kanter, Yaara Aviad, Igor Reidler, Elad Cohen, and Michael Rosenbluh. An optical ultrafast random bit generator. *Nature Photonics*, 4(1):58–61, January 2010.

[372] Benjamin Jun and Paul Kocher. The Intel® random number generator. Technical report, Cryptography Research, San Francisco, CA, 1999.

[373] Ray C. C. Cheung, Dong-U Lee, Wayne Luk, and John D. Villasenor. Hardware generation of arbitrary random number distributions from uniform distributions via the inversion method. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(8):952–962, August 2007.

[374] F. James. A review of pseudorandom number generators. *Computer Physics Communications*, 60(3):329–344, October 1990.

[375] Donald Ervin Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley, Reading, MA, 1981.

[376] Stephen K. Park and Keith W. Miller. Random number generators: Good ones are hard to find. *Communications of the ACM*, 31(10):1192–1201, October 1988.

[377] Pierre L'Ecuyer and Serge Côté. Implementing a random number package with splitting facilities. *ACM Transactions on Mathematical Software*, 17(1):98–111, March 1991.

[378] Pierre L'Ecuyer. Tables of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation*, 68(225):249–260, January 1999.

[379] George Marsaglia. Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences of the United States of America*, 61(1):25–28, September 1968.

[380] Pierre L'Ecuyer and Richard Simard. TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, 33(4):Article 22, August 2007.

[381] George Marsaglia. The Marsaglia random number CDROM including the Diehard battery of tests of randomness, 2005.

[382] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation – Special Issue on Uniform Random Number Generation*, 8(1):3–30, January 1998.

[383] George Marsaglia and Arif Zaman. A new class of random number generators. *The Annals of Applied Probability*, 1(3):462–480, August 1991.

[384] George Marsaglia and Arif Zaman. Some portable very-long-period random number generators. *Computers in Physics*, 8(1):117–121, January/February 1994.

[385] George Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14), July 2003.

[386] François Panneton and Pierre L'Ecuyer. On the Xorshift random number generators. *ACM Transactions on Modeling and Computer Simulation*, 15(4):346–361, October 2005.

[387] W. B. Langdon. A fast high quality pseudo random number generator for graphics processing units. In *IEEE Congress on Evolutionary Computation*, pages 459–465, Hong Kong, China, June 2008.

[388] Myles Sussman, William Crutchfield, and Matthew Papakipos. Pseudorandom number generation on the GPU. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 87–94, Vienna, Austria, September 2006.

[389] David Barrie Thomas, Lee Howes, and Wayne Luk. A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation. In *Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 63–72, Monterey, CA, February 2009.

[390] Stanley Tzeng and Li-Yi Wei. Parallel white noise generation on a GPU via cryptographic hash. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 79–88, Redwood City, CA, February 2008.

[391] Victor Podlozhnyuk. Parallel Mersenne twister. Technical report, NVIDIA Corporation, Santa Clara, CA, 2007.

[392] Wai-Man Pang, Tien-Tsin Wong, and Pheng-Ann Heng. Generating massive high-quality random numbers using GPU. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 841–847, Hong Kong, China, June 2008.

[393] Lee Howes and David Thomas. Efficient random number generation and application using CUDA. In Hubert Nguyen, editor, *GPU Gems 3*, chapter 37. Addison-Wesley Professional, 2007.

[394] K. A. Hawick, A. Leist, D. P. Playne, and M. J. Johnson. Speed and portability issues for random number generation on graphical processing units with CUDA and other processing accelerators. In Jinjun Chen and Rajiv Ranjan, editors, *Proceedings of the 9th Australasian Symposium on Parallel and Distributed Computing*, pages 3–12, Perth, Australia, January 2011.

[395] Sami Hissoiny, Benoît Ozell, and Philippe Després. Using graphics processing units to generate random numbers. http://arxiv.org/abs/1101.1846v1, January 2011.

[396] Pierre L'Ecuyer. Efficient and portable combined random number generators. *Communications of the ACM*, 31(6):742–751, June 1998.

[397] A. C. Atkinson. The computer generation of Poisson random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):29–35, 1979.

[398] G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, June 1958.

[399] Joachim H. Ahrens and Ulrich Dieter. Computer methods for sampling from gamma, beta, Poisson and bionomial distributions. *Computing*, 12(3):223–246, September 1974.

[400] Etta D. Pisano, Martin J. Yaffe, and Cherie M. Kuzmiak. *Digital Mammography.* Lippincott Williams & Wilkins, Philadelphia, PA, 2004.

[401] L. R. Furenlid, E. Clarkson, D. G. Marks, and H. H. Barrett. Spatial pileup considerations for pixellated gamma-ray detectors. *IEEE Transactions on Nuclear Science*, 47(4):1399–1403, August 2000.

[402] Andrew G. Webb. *Introduction to Biomedical Imaging.* John Wiley and Sons, Inc., Hoboken, NJ, 2003.

[403] François O. Bochud, Craig K. Abbey, and Miguel P. Eckstein. Statistical texture synthesis of mammographic images with clustered lumpy backgrounds. *Optics Express*, 4(1):33–42, January 1999.

[404] Cyril Castella, Karen Kinkel, François Descombes, Miguel P. Eckstein, Pierre-Edouard Sottas, Francis R. Verdun, and François O. Bochud. Mammographic texture synthesis: secondgeneration clustered lumpy backgrounds using a genetic algorithm. *Optics Express*, 16(11):7595–7607, May 2008.

[405] K. Bliznakova, Z. Bliznakov, V. Bravou, Z. Kolitsi, and N. Pallikarakis. A three-dimensional breast software phantom for mammography simulation. *Physics in Medicine and Biology*, 48(22):3699–3719, November 2003.

[406] Predrag R. Bakic, Michael Albert, Dragana Brzakovic, and Andrew D. A. Maidment. Mammogram synthesis using a 3D simulation. I. Breast tissue model and image acquisition simulation. *Medical Physics*, 29(9):2131–2139, September 2002.

[407] Predrag R. Bakic, Michael Albert, Dragana Brzakovic, and Andrew D. A. Maidment. Mammogram synthesis using a 3D simulation. II. Evaluation of synthetic mammogram texture. *Medical Physics*, 29(9):2140–2151, September 2002.

[408] Richard E. Greenblatt, Alexei Ossadtchi, and Mark E. Pflieger. Local linear estimators for the bioelectromagnetic inverse problem. *IEEE Transactions on Signal Processing*, 53(9):3403–3412, September 2005.

[409] Jun Shao. *Mathematical Statistics*. Springer-Verlag, New York, NY, 2003.

[410] Meredith Kathryn Whitaker, Eric Clarkson, and Harrison H. Barrett. Estimating random signal parameters from noisy images with nuisance parameters: linear and scanning-linear methods. *Optics Express*, 16(11):8150–8173, May 2008.

[411] L. W. Nolte and David Jaarsma. More on the detection of one of M orthogonal signals. *Journal of the Acoustical Society of America*, 41(2):497–505, February 1967.

[412] David G. Brown, Michael F. Insana, and Markku Tapiovaara. Detection performance of the ideal decision function and its McLaurin expansion: Signal position unknown. *Journal of the Acoustical Society of America*, 97(1):379–398, January 1995.

[413] E. Clarkson, M. A. Kupinski, and H. H. Barrett. Transformation of characteristic functionals through imaging systems. *Optics Express*, 10(13):536–539, July 2002.

[414] Robert L. Parker. Understanding inverse theory. *Annual Review of Earth and Planetary Science*, 5:35–64, May 1977.

[415] George Backus and Freeman Gilbert. The resolving power of gross Earth data. *Geophysical Journal of the Royal Astronomical Society*, 16(2):169–205, October 1968.

[416] George Backus and Freeman Gilbert. Uniqueness in the inversion of inaccurate gross Earth data. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 266(1173):123–192, March 1970.

[417] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1992.

[418] Thomas J. Loredo and Richard I. Epstein. Analyzing gamma-ray burst spectral data. *Astrophysical Journal*, 336:896–919, January 1989.

[419] Carl de Boor. *A Practical Guide to Splines.* Springer-Verlag, New York, NY, 1978.

[420] Michael Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999.

[421] Les Piegl. On NURBS: A survey. *IEEE Computer Graphics and Applications*, 11(1):55–71, January 1991.

[422] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions, Part A: On the problem of smoothing of graduation, a first class of analytic approximation. *Quarterly of Applied Mathematics*, 4:45–99, 1946.

[423] Carl de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, July 1972.

[424] Les A. Piegl and Wayne Tiller. *The $\mathcal{NURBS}$ Book.* Springer, Berlin, Germany, 1997.

[425] M. G. Cox. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, October 1972.

[426] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann Publishers, Los Altos, CA, 1987.

[427] L. Benini, R. Grottesi, S. Morigi, and M. Ruggiero. Parallel rendering and animation of subdivision surfaces on the Cell BE processor. *International Journal of Parallel Programming*, 39(4):494–521, August 2011.

[428] Larry L. Schumaker. *Spline Functions: Basic Theory.* Wiley, New York, NY, 1981.

[429] Jana Procházková. Derivatives of B-spline function. In *Sborník 25. Konference o Geometrii a Počítačové Grafice*, pages 199–204, Janov nad Nisou, Czech Republic, September 2005.

[430] I. J. Schoenberg and Anne Whitney. On Pólya frequency function. III. The positivity of translation determinants with an application to the interpolation problem by spline curves. *Transactions of the American Mathematical Society*, 74(2):246–259, March 1953.

[431] Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall/CRC, New York, NY, 1994.

[432] Bradley Efron. Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599, December 1981.

[433] Rupert G. Miller. The jackknife—a review. *Biometrika*, 61(1):1–15, April 1974.

[434] Richard R. Picard and R. Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, September 1984.