COLLABORATIVE COMPRESSION AND TRANSMISSION OF
DISTRIBUTED SENSOR IMAGERY

by

Joseph C. Dagher

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
2006
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by JOSEPH DAGHER entitled Collaborative Compression and Transmission of Distributed Sensor Imagery and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

_________________________________________ Date: 8/11/2006
Mark A. Neifeld

_________________________________________ Date: 8/11/2006
Michael W. Marcellin

_________________________________________ Date: 8/11/2006
Arthur F. Gmitro

_________________________________________ Date: 8/11/2006
Matthew A. Kupinski

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.

_________________________________________ Date: 8/11/2006
Dissertation Director: Mark A. Neifeld
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: Joseph C. Dagher
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my dissertation advisors, Prof. M. W. Marcellin and Prof. M. A. Neifeld, for guiding me throughout my degree. I am thankful to Prof. Marcellin for his patience and confidence. His insight and knowledge provided tremendous support throughout my graduate life. I thank Prof. Neifeld for his commitment and dedication to his task as an advisor. His insight and knowledge in the field of Optics offered great help in my work and deeply affected my view of my post-graduate goals. I feel extremely lucky to have been able to work with Profs. Marcellin and Neifeld. Their confidence fueled my motivation and my curiosity to approach difficult tasks. Their brilliant intuition helped this work come to light.

I also would like to express my gratitude for my committee members, Prof. A. F. Gmitro and Dr. M. A. Kupinski, for their valuable input and commitment to improve the quality of the dissertation.

Last but not least, I would like to thank a great friend, a great mentor and a great colleague, Ali Bilgin, whose advice and experience helped me throughout my graduate life.
To my parents, Camille and Najah,

To my brothers, Jihad and Milad,

and

To my best friend, Zeina.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIST OF FIGURES</strong></td>
<td>8</td>
</tr>
<tr>
<td><strong>LIST OF TABLES</strong></td>
<td>11</td>
</tr>
<tr>
<td><strong>ABSTRACT</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>1 INTRODUCTION</strong></td>
<td>14</td>
</tr>
<tr>
<td><strong>2 SINGLE-HOP TRANSMISSION WITH COLLABORATIVE COMPRESSION</strong></td>
<td>21</td>
</tr>
<tr>
<td>2.1 Physical Model</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Problem statement</td>
<td>24</td>
</tr>
<tr>
<td>2.3 Theoretical Treatment</td>
<td>26</td>
</tr>
<tr>
<td>2.3.1 Region allocation problem</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2 Network Lifetime</td>
<td>32</td>
</tr>
<tr>
<td>2.3.3 Discrete algorithm</td>
<td>34</td>
</tr>
<tr>
<td>2.4 Exploiting intra-SFOV correlation structure</td>
<td>35</td>
</tr>
<tr>
<td>2.5 Experimental Results</td>
<td>36</td>
</tr>
<tr>
<td>2.5.1 Simulation Issues</td>
<td>36</td>
</tr>
<tr>
<td>2.5.2 Lossless representation</td>
<td>38</td>
</tr>
<tr>
<td>2.5.3 Lossy representation</td>
<td>47</td>
</tr>
<tr>
<td>2.6 Concluding remarks</td>
<td>50</td>
</tr>
<tr>
<td><strong>3 A THEORY FOR MAXIMIZING THE LIFETIME OF SENSOR NETWORKS</strong></td>
<td>55</td>
</tr>
<tr>
<td>3.1 Background on network lifetime</td>
<td>55</td>
</tr>
<tr>
<td>3.2 Problem statement</td>
<td>58</td>
</tr>
<tr>
<td>3.2.1 Physical model and definitions</td>
<td>58</td>
</tr>
<tr>
<td>3.2.2 Formulation</td>
<td>61</td>
</tr>
<tr>
<td>3.3 An optimal algorithm</td>
<td>65</td>
</tr>
<tr>
<td>3.3.1 Computational complexity</td>
<td>69</td>
</tr>
<tr>
<td>3.4 Theory</td>
<td>71</td>
</tr>
<tr>
<td>3.5 Experimental results</td>
<td>78</td>
</tr>
<tr>
<td>3.6 Concluding remarks</td>
<td>83</td>
</tr>
</tbody>
</table>
## TABLE OF CONTENTS – Continued

4 MULTI-HOP TRANSMISSION WITH COLLABORATIVE IMAGE COMPRESSION .................................................. 87  
  4.1 Physical model .......................................................... 87  
  4.2 Problem formulation and solution ...................................... 88  
    4.2.1 Exploiting spatial redundancy through collaboration .............. 89  
    4.2.2 Exploiting intra-SFOV correlation structure .................. 94  
  4.3 Simulation Results ..................................................... 94  
    4.3.1 Alternate definitions of network lifetime ...................... 102  

5 REAL-LIFE PERFORMANCE ................................................. 107  
  5.1 Direct application of CMT algorithm to real images .................. 107  
  5.2 A realistic optics model .............................................. 109  
  5.3 Modified CMT and a reconstruction algorithm ..................... 119  
  5.4 mCMTR Performance .................................................. 122  
  5.5 Concluding remarks ................................................ 135  

6 CONCLUSION AND FUTURE RESEARCH ............................ 137  

APPENDIX A. ................................................................. 140  

REFERENCES ................................................................. 143
LIST OF FIGURES

2.1 An imaging sensor network model. .......................... 22
2.2 A simplified imaging network model in which the sensors must transmit the union of the SFOV$_i$........................................... 24
2.3 The overall network field of view can be divided into non-overlapping largest regions. .................................................. 26
2.4 The problem can be represented with this graph. ................. 27
2.5 The coastal image map from which the scenes are sampled. ....... 40
2.6 Energy remaining in each sensor versus transmission instance for the CRA algorithm. ..................................................... 41
2.7 Energy remaining in each sensor versus transmission instance for the D-DRA algorithm. ..................................................... 42
2.8 Lifetime of the network as a function of scene coverage for 3 sensors. 43
2.9 Lifetime of the network as a function of scene coverage for 2 sensors. 45
2.10 Lifetime of the network as a function of scene coverage for 4 sensors. 46
2.11 Lifetime of the network as a function of scene coverage for 8 sensors. 47
2.12 Lifetime of the network as a function of scene coverage for 12 sensors. 48
2.13 Average lifetime of the network versus the sensor density. .......... 49
2.14 Rural landscape. .......................................................... 50
2.15 Urban landscape. ......................................................... 50
2.16 Average lifetime of the network versus sensor density, for the rural image map. .......................................................... 51
2.17 Average lifetime of the network versus sensor density, for the urban image map. .......................................................... 52
2.18 Lifetime of the network as a function of scene coverage for 3 sensors. The reconstructed images have a normalized RMSE of 0.78%. .... 52
2.19 Original FOV sampled from the coastline landscape image. ........ 53
2.20 Same FOV as in Figure 2.19 but transmitted at a normalized RMSE of 0.78%. ............................................................... 53
2.21 Average lifetime of the network plotted versus the sensor density, for the coastline image map, with images transmitted at different normalized RMSE levels of 0.39%, 0.78%, 1.56% and 2.35%. ..................... 54
3.1 A sensor network model is illustrated here using 3 sensors. .......... 59
3.2 Plot of sensor lifetimes versus $x_{1,0}$ for the example network of 2 sensors. 63
3.3 Example of a 4-sensor network deployment showing the optimal solution obtained using our method. The algorithm converged in one iteration. ................................................................. 79
3.4 Another example of a 4-sensor network deployment. The solution was obtained in four iterations. ................................................................. 80
3.5 Probability of the percentage gain $g_{n:N}$ shown on log-log scale for (a) $n=1$, (b) $n=2$, (c) $n=3$, and (d) $n=4$. ................................................................. 85
3.6 Probability of attaining a positive gain in any of the minimum lifetimes as a function of number of sensors. ................................................................. 86

4.1 A simplified imaging network model in which the sensors must transmit the union of the SFOV$_i$. ................................................................. 88
4.2 The overall network field of view can be divided into largest non-overlapping regions. ................................................................. 89
4.3 The problem for the network of imagers in Figure 4.2 can be represented using this graph. ................................................................. 92
4.4 The optimal transmission strategy for one transmission event from the network deployment illustrated by Figure 4.2. ................................................................. 95
4.5 Remaining energy in each sensor vs. transmission instance for the example deployment of Figure 4.4. Note that the curves for sensors 1 and 2 overlap in the baseline approach. ................................................................. 97
4.6 Lifetime of the network as a function of scene coverage for 3 sensors. 98
4.7 Gain of our CMT algorithm over the CST, MT and baseline approaches as a function of scene coverage for 3 sensors. ................................................................. 100
4.8 Gain of the CMT algorithm over the CST, MT and baseline approaches as a function of scene coverage for (a) 4 (b) 5 and (c) 6 sensors.104
4.9 Gain of the CMT algorithm over the CST, MT and baseline approaches is averaged over all $\eta$ for each sensor density, $\rho$, and then plotted here versus $\rho$. ................................................................. 105
4.10 Gain of our CMT algorithm over the CST and baseline methods for $\rho = 0.44$ when CMT utilizes a transmission strategy that maximizes the 4th conditional lifetime. The results show the gains achieved in the (a) first, (b) second, (c) third and (d) fourth minimum lifetimes. . . . . 106

5.1 Original SFOV captured at one time instant by sensor (a) 1, (b) 2, and (c) 3. ................................................................. 110
5.2 SFOV portions to be sent by sensor (a) 1, (b) 2, and (c) 3. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
5.3 The images sent from each sensor can be combined at the base station.112
5.4 Geometry of the real model. ................................................................. 113
5.5 Images measured by imagers (a) 1: \((x, y, z) = (-1.14, 1.22, -79.56)\)m, 
\(\theta_x = -0.81^\circ, \theta_y = 0.68^\circ\), (b) 2: \((x, y, z) = (-1.70, 2.24, -69.69)\)m, \(\theta_x = -0.76^\circ, \theta_y = 1.44^\circ\) and (c) 3: \((x, y, z) = (-1.97, -1.32, -126.17)\)m, 
\(\theta_x = -0.50^\circ, \theta_y = -0.42^\circ\).

5.6 Object intensity to be imaged by the network of sensors.

5.7 Object-domain map showing the largest non-overlapping regions for the deployment of Figures 5.5a-c.

5.8 An object-domain map showing which imager(s) contribute to every object pixel, for the deployment of Figures 5.5a-c.

5.9 Reconstructed object obtained using our mCMTR algorithm. The resulting fidelity is 35dB.

5.10 Lifetime as a function of coverage for a network of 3 sensors.

5.11 Power spectra of images from Figures 5.5a-c shown using a logarithmic scale. Note that the spectrum of Imager 1 contains higher frequencies than the proximate Imager 2. It is also obvious from the spectrum of the farthest Imager 3 that the main spatial frequencies have shifted outwards to induce even higher spectral content.

5.12 RMSE obtained using our mCMTR algorithm as a function of coverage for the same network of 3 sensors.

5.13 Example utilities for various deployments. (a) \(N = 3, \zeta = 0.0717\), (b) \(N = 1, \zeta = 0.0161\), (c) \(N = 10, \zeta = 0.1545\) and (d) \(N = 25, \zeta = 0.28\).

5.14 Lifetime as a function of the utility function \(\zeta\) for a network of 3 imagers.

5.15 Lifetime as a function of deployment utility \(\zeta\) for \(N=3,4,5,\) and 6 imagers.

5.16 Average lifetime as a function of number of imagers.

5.17 Average utility as a function of number of imagers.

5.18 Lifetime as a function of deployment utility \(\eta_w\) for \(N=3,4,5,\) and 6 imagers.
LIST OF TABLES

3.1 Summary of results for \( N = 3, 4, 5 \) and 6 ............................... 83
5.1 mCMTR Algorithm......................................................... 120
ABSTRACT

Distributed imaging using sensor arrays is gaining popularity among various research and development communities. A common bottleneck within such an imaging sensor network is the large resulting data load. In applications for which transmission power and/or bandwidth are constrained, this can drastically decrease the network lifetime. In this dissertation, we consider a network of imaging sensors. We address the problem of energy-efficient communication of the resulting measurements.

First, we develop a heuristic-based method that exploits the redundancy in the measurements of imaging sensors. The algorithm attempts to maximize the lifetime of the network without utilizing inter-sensor communication. Gains in network lifetime up to 114% are obtained when using the suggested algorithm with lossless compression. Our results also demonstrate that when lossy compression is employed, much larger gains are achieved. For example, when a normalized Root-Mean-Squared-Error of 0.78% can be tolerated in the received measurements, the network lifetime increases by a factor of 2.8, as compared to the lossless case.

Second, we develop a novel theory for maximizing the lifetime of unicast multi-hop wireless sensor networks. An optimal centralized solution is presented in the form of an iterative algorithm. The algorithm attempts to find a Pareto Optimal solution. In the first iteration, the minimum lifetime of the network is maximized. If
the solution is not Pareto Optimal a second iteration is performed which maximizes the second minimum lifetime subject to the minimum lifetime being maximum. At the \( n^{th} \) iteration, the algorithm maximizes the \( n^{th} \) minimum lifetime subject to the \((n-1)^{th}\) minimum lifetime being maximum, subject to the \((n-2)^{th}\) minimum lifetime being maximum, etc. The algorithm can be stopped at any iteration \( n \).

Third, we present a novel algorithm for the purpose of exploiting the inherent inter- and intra-sensor correlation in a network of imaging sensors while utilizing inter-sensor communication. This algorithm combines a collaborative compression method in conjunction with our cooperative multi-hop routing strategy in order to maximize the lifetime of the network. This CMT algorithm is demonstrated to achieve average gain in lifetime as high as 3.2 over previous methods.

Finally, we discuss practical implementation considerations of our CMT algorithm. We first present some experimental results that illustrate the practicality of our method. Next, we develop a realistic optical model that permits us to consider a more heterogeneous network of cameras by allowing for varying resolution, intrinsic and extrinsic parameters, point-spread function and detector size. We show that our previous CMT algorithm can be extended to successfully operate in such a diverse imaging model. We propose new object-domain quality metrics and show that our proposed method is able to balance lifetime and fidelity according to expectations.
CHAPTER 1
INTRODUCTION

Imaging using a distributed array of sensors has been shown to provide important capabilities in various applications such as video surveillance, security, medical imaging, traffic control, environmental monitoring and distributed robotics [11, 23, 54, 72]. In contrast with more traditional centralized schemes, distributed sensing provides for the accuracy, rate, scalability and robustness needed in systems that monitor and measure the physical world. However, because of the nature of the environments in which they are likely to operate, sensors must deal with resource constraints such as energy [17, 53], bandwidth and processing power [68]. This dynamic behavior can create challenges in data processing [33, 55], communication and network management [26, 30, 43]. A common bottleneck within the imaging sensor network framework is the large resulting data size [26, 40]. In applications where transmission power and/or bandwidth are constrained, this can quickly deplete the energy resources of the imaging sensor network.

It is thus of importance to address the following two questions: (a) how much information needs to be transmitted by each sensor in order to guarantee a given reconstruction quality at the receiver and, (b) what is the most resource-efficient manner in which the sensors could transmit this information? Note that the term
“reconstruction quality” is a task-specific measure. For example, if the task of the network is to estimate the location of a target, then a reasonable quality measure at the receiver would be the probability distribution on the target location. In this work, the sensors are assumed to measure the reflected intensity from the scene. Thus, one quality measure used for this work is Root Mean Squared Error (RMSE).

In order to treat the first problem, we must consider the data redundancy inherent in a sensor network. One simple approach would ignore inter-sensor redundancy and simply utilize various available source coding techniques to independently compress the images at each sensor [64]. Such an approach would take advantage of the correlation inherent in a single sensor’s measurement without acknowledging the common nature of the measurements across the array. On the other hand, distributed compression methods utilize scene correlation structure across a sensor array in order to reduce the dimensionality of the data transmitted to the receiver.

Inter-sensor compression methods can be divided into 2 classes: statistical and collaborative. The statistical methods are based on results obtained by Slepian-Wolf [61] and Wyner-Ziv [70] who showed that jointly encoding multiple sources yields the same coding gains as separately encoding the same sources. This important result rests heavily on the following assumptions: (a) joint decoding is utilized and (b) the correlation structure of the sources is known a priori [61,70]. The latter constraint is inconvenient to impose in an imaging sensor network: the correlation between images taken from neighboring sensors is not easily described by a simple joint-process. So,
the few efforts that have considered image compression in a sensor network chose to follow a more deterministic “collaborative” approach. Baraniuk *et. al.* [67] used a method that has each imager transmit a low-resolution version of the maximum area of overlap. Super-resolution methods are then applied at the receiver in order to reconstruct an estimate of the original high-resolution image of that area of overlap. Chen *et. al.* [69] utilized inter-sensor communication in order to transmit the encoded difference between images taken at multiple sensors. The method relies on transforming each image into a reference frame using image registration techniques. Finally, Girod *et. al.* [76] utilizes geometrical side information at the encoders in order to encode the scene using key (reference) frames and Wyner-Ziv (disparity) frames. The method requires prior-knowledge of camera parameters, scene geometry and necessitates an object with extraneous background. Note that, in order for any of these aforementioned methods to achieve any compression gain (and hence reduce the energy consumption), the network needs to be populated with enough imagers so as to have a large region of common overlap. In this dissertation, we present a collaborative method that exploits any level of overlap between sensor network imagery. Furthermore, none of the previous algorithms take into account the energy consumption associated with the transmission/routing of information.
A “route” could consist of a single-hop transmission (all the data sent directly to the receiver), and/or multi-hop transmissions (some portion of the data sent indirectly via other sensors). When inter-sensor communication is not allowed, transmission strategies consist solely of single-hop routes. Multi-hop routes, on the other hand, could be sought in the case when inter-sensor communication is permitted.

In general, we distinguish between two routing strategies: minimum-energy routing and maximum-lifetime routing. The former searches for a transmission strategy that consumes the least amount of energy across all sensors [75]. The latter attempts to maximize the lifetime of the sensor network [10, 13] where “lifetime” is defined as the time until the first sensor runs out of (battery) power. Unlike maximum-lifetime routing where the operation of every sensor is considered equally important to the network as a whole, minimum-energy routing can quickly exhaust the battery of certain sensors. Hereafter, we only consider maximum-lifetime routing strategies. Nevertheless, we should note that our work could be easily extended to incorporate other transmission strategies.

To the best of our knowledge, energy efficient collaborative image compression and transmission in a sensor network has not been studied in the literature. In an earlier work [14]¹, we developed novel methods for compressing volumetric imagery that has been generated by single platform (mobile) range sensors. In that work, we exploit the correlation structure inherent in multiple LASER Radar (LADAR) views in order to improve compression efficiency. We show that, for lossless compression,
three-dimensional volumes compress more efficiently than multiple two-dimensional (2D) images by a factor of 60%. Furthermore, our error metric for lossy compression suggests that accumulating more than 9 range images in one volume before compression yields up to a 99% improvement in compression performance over 2D compression.

Finally, we should mention methods in the literature that deal with data aggregation in a sensor network. Although these methods [2, 37, 50, 51] do not consider image measurements explicitly, they combine aggregation and routing algorithms in order to achieve energy efficient transmission.

In this dissertation, we consider a network of imaging sensors. We present novel collaborative compression methods as well as a novel multi-hop routing algorithm for the purpose of maximizing the lifetime of the network.

Specifically, in Chapter 2, we develop a heuristic-based method that exploits the redundancy in the measurements of imaging sensors in order to maximize the lifetime of the network without utilizing inter-sensor communication [15]. The results show gains in network lifetime up to 114% over traditional approaches. However, as we will see next, there exist instances in which it is not possible to balance the load over the network without inter-sensor communication. In those cases, the method in Chapter 2 does not provide any gain in lifetime.

\footnote{Recipient of the Best Student Paper Award at SPIE’s Visual Communication and Image Processing conference, VCIP2003, Lugano, Switzerland, July 2003 and, recipient of the Best Student Paper, second place award, at IFT’s International Telemetry Conference, ITC 2003, Las Vegas, Nevada, October 2003.}
Thus, we ask the following fundamental question: how can we utilize multi-hop routing (inter-sensor communication) in order to maximize the lifetime of a network of imagers?

We answer this question in 2 parts. We first consider sensors with no correlation structure between their measurements. We develop in Chapter 3 a novel theory for maximizing the lifetime of unicast multi-hop wireless sensor networks [13]. An optimal centralized solution is presented in the form of an iterative algorithm.

In Chapter 4, however, we allow for intra- and inter-sensor correlation structure by specifically treating imaging sensors. We again ask the same question as in Chapter 3 but, we aim at increasing the lifetime of the network further by additionally exploiting this redundancy. We propose a novel collaborative compression algorithm to be used in conjunction with a cooperative multi-hop transmission (CMT) strategy in order to maximize the lifetime of the network. The algorithm is demonstrated to achieve average gain in lifetime as high as 3.2 over previous methods [12].

Throughout Chapters 2 and 4, we assume a simple imaging model. In Chapter 5, we first describe an experimental procedure that applies our proposed CMT algorithm directly. This procedure assumes the simple imaging model described in Chapter 2. The result illustrates that this assumption yields good performance in some applications. Next, we develop a more realistic optical model that takes into account a more heterogeneous network of cameras by allowing for varying resolution, intrinsic and extrinsic parameters, point-spread function, detector size, etc. We show
that our CMT algorithm presented in Chapter 4 can be extended to successfully operate in such a diverse imaging model. We propose new object-domain quality metrics and show that our CMT algorithm is still able to balance lifetime and utility according to expectations.

We identify 3 main components in our algorithms of Chapters 4 and 5: exploiting intra-sensor redundancy via source coding (or compression), exploiting inter-sensor redundancy via overlapping fields-of-view (or collaboration) and, optimal transmission of the resulting loads (or routing). The main contributions of this dissertation consist of the following: (a) presenting novel collaboration methods (in Chapters 2 and 4), (b) developing a novel theory and a corresponding optimal algorithm for the purpose of optimal routing (Chapter 3), (c) combining the above algorithms for the purpose of collaborative compression and routing (Chapter 4) and, (d) developing a practical and comprehensive framework, and showing that the resulting algorithm performs according to expectations in realistic scenarios whereby different cameras possess different utilities (Chapter 5).

Chapter 6 summarizes these contributions, and outlines future exciting work.
CHAPTER 2
SINGLE-HOP TRANSMISSION WITH COLLABORATIVE COMPRESSION

In this chapter, we develop a heuristic-based method that exploits the redundancy in the measurements of imaging sensors in order to maximize the lifetime of the network without utilizing inter-sensor communication [15]. We first start by presenting the physical model under consideration. This simple model will be utilized in Chapter 4 as well. Chapter 5 will extend the model to handle more realistic scenarios. Section 2.2 formulates the collaborative compression problem. We present theoretical treatments and a heuristic-based solution in Section 2.3. Simulation results are finally presented in 2.5.

2.1 Physical Model

In this section, we describe a model of the imaging sensor network under consideration. In this model, $N$ imaging sensors are deployed over a scene of interest some distance away from a central base station. This is illustrated in Figure 2.1. We treat here the case of a unique receiver (base station) but, this work could be extended to include multiple receivers. The positioning of the sensors depends on the application of interest. For the purpose of our treatment, we assume that the location of each
sensor is a random variable that obeys a probability distribution function governed by a physical law (e.g., diffusion). Thus, the field of view of sensor $i$ (SFOV$_i$), defined as the set of object voxels that are seen by sensor $i$, is random as well.

![Figure 2.1: An imaging sensor network model.](image)

Various novel applications demand the capabilities of this random deployment model [42,60]. One example is the military battlefield in which images must be transmitted to a remote station for information processing (target identification, tactical planning, etc). Environment and habitat monitoring is another example in which sensors are distributed over a large area and are needed to take images and transmit them to a central processing station. Other applications range from industrial sensing to traffic control and infrastructure security [11].

For a given random deployment, the task of the network consists of transmitting the union of the resulting Sensors’ Field of View (SFOV $\equiv \bigcup_{i=1}^{N}$ SFOV$_i$) to the base
station. We define this operation as a “transmission event.” As mentioned previously, constraints such as limited transmission bandwidth and/or power can quickly reduce (or prevent) transmission events. Thus, in this chapter, the problem is to maximize the number of transmission events by reducing the overall data load through exploiting the inherent inter- and intra-sensor redundancy via collaboration.

Solving this problem for a three-dimensional (3D) object is challenging. Occlusion effects as well as perspective differences complicate the problem. This is the topic of ongoing work. In Chapters 2 and 4, we assume the following for our initial imaging model:

1. Each sensor images a two-dimensional (2D) world. Occlusion and perspective effects are neglected for now.  
2. All sensors are identical.  
3. The optical axes of all sensors are parallel so the pixel shape is the same (square) for all sensors.  
4. The sensors are at the same distance from the scene, so that the pixel spatial dimensions are the same for all sensors.  
5. Sensors take noise-free measurements. 

So, a measurement of a spatial region taken by different sensors yields the same pixel values. 

Figure 2.2 illustrates the simplified model described above. In this figure, three sensors are randomly deployed over the scene of interest. The shaded regions in the figure represent the SFOV. Together, all the sensors cover a certain area of the scene. The task is to transmit the union of the SFOV to the base station.
2.2 Problem statement

Before presenting the problem, we shall make use of Figure 2.2 to introduce some of the terminology used throughout this chapter. This figure illustrates the deployment of 3 imaging sensors according to our simplified model described in Section 2.1. Each SFOV \( i \) covers a certain area of the scene. The scene coverage \( \eta \) is defined as the fraction of the total scene area covered by all the sensors. Clearly, \( \eta \) is a random variable that depends on the deployment process and the number of deployed sensors.

We define sensor density \( \rho \) to be the ratio of the number of deployed sensors to the minimum number of sensors required to cover the entire scene. In the example shown in Figure 2.2, \( \eta = 0.2 \) and \( \rho = 0.33 \). The third quantity that we will define is the lifetime of the sensor network. In general, this is a task-specific measure. In this chapter, and unless noted otherwise, we choose to define the network lifetime \( \alpha \) as
the number of scenes transmitted to the base station (i.e., transmission events) before one or more sensors run out of battery power.

Given a random deployment of the network along with the corresponding energy of each sensor, we would like to find a transmission strategy that maximizes the lifetime of the sensor network by exploiting any spatial redundancy. The transmission strategy consists of assigning scene elements (pixels) to sensors while satisfying the following constraints:

- Constraint A: a sensor can only send pixels that fall within its field of view (no inter-sensor communication is employed) and,
- Constraint B: all elements (pixels) constituting the union of the sensor fields of view must be sent to the base station.

The first step towards exploiting the spatial redundancy within the network is to divide the SFOV into largest non-overlapping regions. Or, more formally, we decompose the overall field of view into a set of areas that are mutually exclusive and collectively exhaustive. This is illustrated in Figure 2.3 where the fields of view are divided into a set of 7 non-overlapping regions, $R_1, R_2, \ldots, R_7$. Our goal consists of transmitting these regions according to constraints A and B mentioned above while maximizing the lifetime of the network $\alpha$. 
2.3 Theoretical Treatment

In what follows, we develop a mathematical formulation of the problem presented in the previous section. Our theoretical treatment is divided into 2 parts. In the first part, we formulate the problem and outline a solution. In the second part, we derive an upper-bound on the lifetime of the sensor network as defined in Section 2.2.

2.3.1 Region allocation problem

The region allocation problem can be represented using the graph shown in Figure 2.4. What follows is a brief description of the various quantities in the graph. We consider here the general case of $N$ deployed sensors in the network. We denote by $B_i$, \(i = 1, 2, ..., N\), the remaining transmit energy (in bits) of the $i^{th}$ sensor at any
transmission instant. It should be noted that the set \( \{B_i\} \) is a function of: (a) the remaining energy (in Joules) in each battery, (b) the path loss model of signal power versus transmit distance [56] and, (c) the distance from the sensor to the base station \(^1\). In the graph, each region labeled by \( R_j \) has a corresponding transmission cost \( C_j \) (in bits) associated with it, \( \{j = 1, 2, ..., P\} \), where \( P \) is the total number of largest non-overlapping regions. We defer the discussion on how to compute the cost \( C_j \) to Section 2.4. The branches of the graph, labeled \( t_{i,j} \), denote the fraction of region \( R_j \) to be sent by sensor \( i \). Our problem then is to determine the set of non-negative variables \( \{t_{i,j}\} \).

![Graph representation](image)

**Figure 2.4:** The problem can be represented with this graph.

So, from these definitions, we can write the expected lifetime of a sensor \( i \) as,

\[
\alpha_i = \frac{B_i}{\sum_{j=1}^{P} t_{i,j} C_j},
\]

where the denominator simply represents the total number of bits to be sent by sensor \( i \).

\(^1\)A more detailed description of energy consumption in a sensor network is deferred to Chapter 3.
Now, according to our definition of the network lifetime $\alpha$ (Section 2.2), we want to maximize the number of transmissions before one or more sensors fail. In other words, what is desired is to maximize the minimum expected lifetime $\alpha_i$ across all $N$ sensors. Mathematically, we want to find the optimum set $\{t_i^{opt}\}$ such that

$$\{t_i^{opt}\} = \arg\max_{t_{i,j}} \left\{ \min_i \left( \frac{B_i}{\sum_{j=1}^{P} t_{i,j} C_j} \right) \right\}. \quad (2.2)$$

This is to be achieved subject to the constraints that a sensor only sends scene elements that fall within its field of view (Constraint A) and, all elements constituting the entire scene are sent (Constraint B).

Constraints A and B translate to the following Equations (2.3) and (2.4), respectively:

$$t_{i,j} = 0, \quad \text{if sensor } i \text{ can not see region } j, \quad (2.3)$$

$$\sum_i t_{i,j} = 1, \quad \forall \ j = 1, 2, \ldots, P, \quad i = 1, 2, \ldots, N. \quad (2.4)$$

This problem can be transformed into a linear least-squares problem. To see this, we first solve the problem with Constraint A relaxed. In this case, we claim that a reasonable solution to Equation (2.2) (subject to Constraint B only) would make the lifetimes $\alpha_i$ of all the sensors equal$^2$, i.e., $\alpha_1 = \alpha_2 = \ldots = \alpha_N$ or, as Equation (2.1) implies:

$$\frac{B_1}{\sum_{j=1}^{P} t_{1,j} C_j} = \frac{B_2}{\sum_{j=1}^{P} t_{2,j} C_j} = \ldots = \frac{B_N}{\sum_{j=1}^{P} t_{N,j} C_j}, \quad (2.5)$$
which implies
\[
\frac{B_i}{B_1} \sum_{j=1}^{P} t_{1,j} C_j = \sum_{j=1}^{P} t_{i,j} C_j, \quad \forall \ i = 1, 2, \ldots, N. \tag{2.6}
\]

Taking the sum of Equation (2.6) on both sides over all \(i\) yields,
\[
\sum_{i=1}^{N} \left\{ \frac{B_i}{B_1} \sum_{j=1}^{P} t_{1,j} C_j \right\} = \sum_{i=1}^{N} \sum_{j=1}^{P} t_{i,j} C_j
\]
\[
\Rightarrow \frac{B}{B_1} \sum_{j=1}^{P} t_{1,j} C_j = \sum_{j=1}^{P} C_j \sum_{i=1}^{N} t_{i,j} = \sum_{j=1}^{P} C_j, \tag{2.7}
\]
where \(B\) is defined as \(B \equiv \sum_{i=1}^{N} B_i\) and the last equality follows directly from Equation (2.4).

Finally, putting Equations (2.6) and (2.7) together yields,
\[
\sum_{j=1}^{P} t_{i,j} C_j = \frac{B_i}{B} \sum_{j=1}^{P} C_j, \quad \forall i = 1, 2, \ldots, N. \tag{2.8}
\]

This result describes a set of equations linear in \(t_{i,j}\) and thus could also be represented in the following matrix form,
\[
\begin{bmatrix}
  t_{1,1} & t_{1,2} & \ldots & t_{1,p} \\
  t_{2,1} & t_{2,2} & \ldots & t_{2,p} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{N,1} & t_{N,2} & \ldots & t_{N,p}
\end{bmatrix}
\begin{bmatrix}
  C_1 \\
  C_2 \\
  \vdots \\
  C_p
\end{bmatrix}
= \frac{B}{B_1}
\begin{bmatrix}
  \sum_{j=1}^{P} C_j \\
  B_1 \\
  \vdots \\
  B_N
\end{bmatrix}
\]

or more compactly,
\[
\tilde{\mathbf{T}} \mathbf{C} = \mathbf{d} \tag{2.9}
\]

\(^2\text{A more careful solution to maximizing the lifetime of a sensor network is deferred until Chapter 3.}\)
where,

\[ C^\dagger \equiv [C_1 C_2 \ldots C_p], \]

\[ \tilde{T} \equiv \begin{bmatrix}
  t_{1,1} & t_{1,2} & \ldots & t_{1,P} \\
  t_{2,1} & t_{2,2} & \ldots & t_{2,P} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{N,1} & t_{N,2} & \ldots & t_{N,P}
\end{bmatrix} \]

and,

\[ d^\dagger \equiv \frac{\sum_{j=1}^{P} C_j}{B} [B_1 B_2 \ldots B_N]. \]

The solution to our problem can be then obtained by solving Equation (2.9) along with Constraint B. Given that we have relaxed Constraint A here, the resulting solution rests on the assumption that any sensor \( i \) can send any region \( R_j \) at the original corresponding transmit cost \( C_j \). In other words, the solution thus obtained represents the case when inter-sensor communication is allowed at no communication cost. Including Constraint A will yield a solution that is suboptimal compared to that obtained by Equation (2.9). What is desired then is to minimize this deviation from optimality.
So, our problem presented in Equation (2.2) can be restated as:

\[
\{ t_{i,j}^{\text{opt}} \} = \arg\min_{t_{i,j}} \left\| \tilde{T} C - d \right\|^2, \text{subject to constraints A and B.} \tag{2.10}
\]

Conveniently, the constraints can also be written as a set of equations linear in \( t_{i,j} \).

More precisely, we can combine Equations (2.3) and (2.4) into the following form:

\[
A_j T_j^\dagger = 1, \quad \forall j = 1, 2, \ldots, P, \tag{2.11}
\]

where,

\[
T_j = [t_{1,j} t_{2,j} \ldots t_{N,j}],
\]

and,

\[
A_j = [a_{1,j} a_{2,j} \ldots a_{N,j}],
\]

with,

\[
a_{i,j} = \begin{cases} 
1, & \text{if sensor } i \text{ can “see” region } j; \\
0, & \text{otherwise.}
\end{cases}
\]

Finally, we summarize here the problem:

\[
\{ t_{i,j}^{\text{opt}} \} = \arg\min_{t_{i,j}} \left\| \tilde{T} C - d \right\|^2, \text{ such that,}
\]

\[
A_j T_j^\dagger = 1, \forall j \quad \text{and,} \quad t_{i,j} \geq 0, \forall i, j. \tag{2.12}
\]

Our problem is now stated in terms of minimizing a quadratic criterion function subject to linear equality and inequality constraints. The solution to such a problem can be obtained using various efficient numerical methods [25, 66].
We should note that the solution \( \{ t_{i,j}^{\text{opt}} \} \) thus obtained assumes that attempting to equalize the lifetimes of the sensors would maximize the lifetime of the network, as defined earlier. A more careful treatment in Chapter 3 would show that this heuristic is well-founded. Moreover, note that our definition of sensor lifetime in Equation (2.1) assumes that the current regions’ cost set \( \{ C_j \} \) is stationary. If the cost set changes then, obviously, the solution \( \{ t_{i,j}^{\text{opt}} \} \) would change as well. Given no prior knowledge about the temporal evolution of the regions’ cost set \( \{ C_j \} \), it is reasonable to assume that, on average, the cost per sensor \( i \) is given by \( \sum_{j=1}^{P} t_{i,j} C_j \).

2.3.2 Network Lifetime

In this section, we present a theoretical answer to the following question: What is the expected network lifetime that would result when applying our algorithm?

Consider the following simplified 2D scene model comprising \( M \times M \) pixels of data. Assume that the average encoded cost of each pixel is \( b \) bits. Assume also that the \( N \) sensors are randomly deployed over the scene of interest. Specifically, let the location of each sensor field of view (SFOV) be random and uniform within the \( M \times M \) scene area (SFOV are assumed to lie inside the scene area). Also, let the dimensions of each SFOV be \( f \times f \) pixels, where \( f \leq M \).

Given this model, we can now express sensor density \( \rho \) as: \( \rho = \frac{N f^2}{M^2} \). So, for a fixed number of sensors \( N \), \( \rho \) is a constant while the scene coverage \( \eta \) is random over the range \( \frac{f}{N} \leq \eta \leq \min\{\rho, 1\} \). The lower-bound corresponds to the case when all SFOVs
fully overlap (intersection = union) while maximum $\eta$ is attained when there is no overlap at all (intersection $=\emptyset$).

Assuming temporal stationarity, i.e. the average cost $b$ bits per pixel does not change across time for the same network geometry (configuration), we can write:

$$\alpha_{opt} = \frac{B_i}{\sum_{j=1}^{P} c_{i,j} C_j}, \ \forall \ i = 1, 2, \ldots, N.$$

If we relax Constraint A, we can make use of Equation (2.8) to write,

$$\alpha_{opt} = \frac{B}{\sum_{j=1}^{P} C_j}.$$ The term in the denominator, $\sum_{j=1}^{P} C_j$, is in fact the total number of bits to be transmitted which, under our model, can also be written as $\sum_{j=1}^{P} C_j = M^2 b\eta$. Finally, this allows us to express the maximum lifetime that would result from our algorithm as:

$$\alpha_{opt} = \frac{B}{M^2 b\eta}. \quad (2.13)$$

Equation (2.13) indicates that $\alpha_{opt}$ increases as $\eta$ decreases. This is expected: given that a low value of $\eta$ implies more overlap between the SFOVs, this also indicates a higher degree of spatial redundancy to exploit. In the extreme case of total SFOVs overlap, $\eta = \frac{\xi}{N}$ is minimum and the lifetime becomes $\alpha_{opt} = \frac{N B}{M^2 b\eta}$. The other extreme limit occurs for maximum coverage, $\eta = \min\{\rho, 1\}$. In this case, there is no overlap between the sensors thus, there are no common regions that we could use to balance the load on the network. In this case, lifetime is at a minimum.

Because the scene coverage $\eta$ is the result of a random deployment, it is a random variable. We can therefore ask the following question: given a fixed number of sensors $N$, what is the probability of attaining a given coverage? In Appendix A we show...
that $\eta$ can be approximated by a doubly truncated normal distribution with mean $\langle \eta \rangle = 1 - \left(1 - \frac{f^2}{M^2}\right)^N$. We can now average Equation (2.13) to deduce the expected lifetime of the network.

\[
\langle \alpha_{opt} \rangle = \frac{B}{M^2 b} \frac{1}{\langle \eta \rangle},
\]

\[
= \frac{B}{M^2 b} \frac{1}{\langle \eta \rangle},
\]

\[
= \frac{B}{M^2 b \left[1 - \left(1 - \frac{f^2}{M^2}\right)^N \right]}. \tag{2.15}
\]

Equation 2.14 follows from the fact that for the positively-truncated Gaussian random variable $\langle \frac{1}{\eta} \rangle = \frac{1}{\langle \eta \rangle}$ [45].

So, for a collection of random sensor deployments, Equation (2.15) enables us to compute the expected lifetime of the sensor network.

2.3.3 Discrete algorithm

Thus far, we have been concerned with determining the values of $\{t_{i,j}\}$ that maximize the lifetime of the sensor network. The values of $\{t_{i,j}\}$ are continuous, which means that sensors can share the transmit cost of sending a region. This might not be desired. We may therefore impose the additional constraint that each region be sent by one sensor only. This additional restriction constrains $\{t_{i,j}\}$ to be a set of binary numbers.

Solving this problem can be more difficult [49]. Here, we propose two methods to approach the solution for this problem. The first one is trivial and consists of an
exhaustive search for the optimal scenario. However straightforward, this method is not computationally efficient because the number of regions ($P$) increases almost exponentially with the number of sensors\textsuperscript{3}. The second method is based upon the solution of the continuous optimization problem presented in Section 2.3.1. The algorithm starts from the continuous solution set $\{t_{i,j}^{\text{opt}}\}$. This set is then discretized subject to Constraints A and B. Then, a steepest descent approach is followed to find a solution closest (in the mean-squared error sense) to the solution obtained by the continuous algorithm [49].

2.4 Exploiting intra-SFOV correlation structure

As mentioned previously, an important part of maximizing the lifetime of a sensor network is the ability to exploit the inter- and intra-sensor redundancy. In Section 2.3 we have developed a collaborative single-hop method for exploiting the former. However, as can readily be seen from Equation (2.1), the lifetime $\alpha_i$ is increased as the load $C_j$ is decreased. Thus, it is of interest to exploit the intra-sensor redundancy in order to decrease the region costs $C_j$. This can be achieved through compression of the image data. Lossless compression of the data guarantees an RMSE of zero. Further increase in lifetime could be achieved at the cost of a higher value of RMSE. Therefore, each sensor should be equipped with an encoding (compression) engine that computes the corresponding compressed costs of the regions associated with that

\textsuperscript{3}It can be easily shown that the maximum number of regions is given by $P_{\text{max}} = 2^N - 1$. 
sensor. In this work we have used JPEG2000 [64] as the encoding engine. Results for other compression systems will be similar.

2.5 Experimental Results

2.5.1 Simulation Issues

In what follows, we present an overview of the experimental procedures adopted in the simulations. Important implementation considerations are discussed in this section as well. The following implementation considerations are also relevant for the algorithm presented in Chapter 4.

- The scene over which the sensors are deployed is 76.8 m × 76.8 m (Figure 2.3). The physical dimensions of each SFOV, is 25.6 m × 25.6 m, ∀i. Assuming an imager resolution of 0.2 m/pixel, each SFOV, (image) corresponds to 128 × 128 pixels.

- The scene is assumed to be some distance d away from the base station. This measure sets a lower bound on the distance between any sensor and the base station.

- The sensors are randomly deployed over the scene of interest. We should note that there could be some circumstances that will favor a pre-arranged and/or optimized deployment. However, in many situations, deterministic deployment is either impractical or impossible. For example, in applications where the environment is unknown or hostile, the sensors can not be precisely positioned. Instead, random deployment is adopted (e.g., sensors dropped from an aircraft).
As discussed earlier, a central part in our formulation is the sensors' knowledge of the network constellation. In our simplified imaging model (Figure 2.2), each sensor only needs to know the coordinates of other sensors. This can be determined, with sufficient accuracy, with the aid of GPS data. Alternatively, each sensor $i$ may initially transmit one full image of its entire SFOV$_i$. The base station can then determine correspondences from the set of all such images. Given this knowledge along with the 2D assumption, the set of largest non-overlapping regions can be determined.

- Each region $R_i$ is encoded with JPEG2000 using 2 levels of wavelet transform. The corresponding transmit cost $C_j$ is then computed and transmitted to the base station.

- The base station then computes the transmission strategy $\{t_{i,j}^{opt}\}$ which will maximize the lifetime of the network.

- Each sensor receives the set $\{t_{i,j}^{opt}\}$. This completely determines the number of bits (or equivalently pixels) that should be sent by each sensor. It is now needed to know the answer to the following: what pixels should each sensor transmit? This is only an issue for shared regions. We have elected to have sensors simply “skip” over pixels assigned to previous (e.g., lower index) sensors (assuming the usual raster ordering).
• In many instances, the regions that need to be coded will not be rectangular. See for example $R_2$ in Figure 2.3. In order to compress such a region, we simply zero pad to the smallest enclosing rectangle. Since JPEG2000 codes zeros very efficiently, negligible coding efficiency is lost. Alternatively, it is possible to use the spatial scalability feature in JPEG2000 [64] to transmit only those pixels that were not zero-padded.

• At the base-station, the data from all sensors are decoded independently before reconstructing the final scene. These steps are then repeated until one or more sensors run out of battery power.

• Although we consider perfect channel and network conditions in our experiments, our model can allow for any value for the link costs. The link costs used assume that the radio dissipates enough power per bit in order to achieve acceptable quality at the receiver. Larger link costs could mimic, for example, higher noise environments, higher interference, etc. We hereafter assume that a simple TCP-based messaging protocol is appropriate to use in these network conditions.

2.5.2 Lossless representation

We present experimental results illustrating the impact of our algorithm on the lifetime of sensor networks. In this section, we only consider the lossless representation (compression) of sensor imagery.
First, we apply our algorithm on the example 3-sensor network deployment of Figure 2.3. In this specific configuration, \( \rho = 0.33 \) and \( \eta = 0.2 \). The transmit energy model employed here is such that sensors 1, 2 and 3 possess the capability of transmitting a total of \( 9 \times 10^5 \) bits, \( 10 \times 10^5 \) bits and \( 11 \times 10^5 \) bits, respectively.

We want to look at the transmit energy remaining in each sensor after each transmission instance. To do this, we present the sensors with a new scene before each transmission instance, while maintaining a fixed sensor network geometry. The \( 384 \times 384 \) scene is randomly chosen from a large image map. The image map from which our scenes are “sampled” in this example is a \( 4450 \times 4450 \) grayscale image representing a coastline landscape (low-resolution version shown in Figure 2.5). Our algorithm is then applied to each scene. In Figure 2.6, we show the impact of such a procedure on the energy remaining (in bits) in each sensor. The transmission instance number is indicated along the horizontal axis and the vertical axis represents the remaining bit budget. The 3 different curves represent the bit budget (energy) left for each of the 3 different sensors. For example, we can see that initially Sensor 3 had the largest bit budget (e.g., it is the closest to the base station). Our algorithm is then applied at each transmission instance in order to optimally allocate regions so as to maximize the lifetime of the network. Eventually, as can be seen, all sensors run out of power at the same time, which is what we expected (Equation (2.5)). For this case, the corresponding lifetime of the network is 24 transmissions. The algorithm employed in this example allows for a common region to be shared by more than one
sensor, i.e., the $\{\ell^\text{opt}_{i,j}\}$ set is continuous. We refer to this as the Continuous Region Allocation (CRA) problem.

![Image](image1.png)

Figure 2.5: The coastal image map from which the scenes are sampled.

We also present the performance of the Discrete Region Allocation (DRA) algorithm discussed in Section 2.3.3 whereby the entirety of each region is sent by only one sensor. The results are shown in Figure 2.7 for the same network configuration as before. The axes labels are the same as in Figure 2.6. Again, the sensors start with different transmission capabilities. We can observe from Figure 2.7 that in DRA the load balancing is more visible across transmissions. This makes sense because the allocation is not as “fine” as in the CRA case. Eventually, all the sensors run out of energy at the same time. The lifetime of the network in this case is also 24 transmissions. We should note that in both examples (Figures 2.6 and 2.7) the sensors were able to run out of energy at the same time. This is what we set to achieve. However,
as explained in Section 2.3, this is not always possible. As we shall see in subsequent results, there are cases in which we might not be able to achieve this optimum.

The results presented in Figures 2.6 and 2.7 are for a sensor density of $\rho = 0.33$ with a specific scene coverage of $\eta = 0.2$. Next, we fix the number of sensors and vary the scene coverage $\eta$. This is done for the purpose of observing how the lifetime of the network changes with $\eta$. To do so, we randomly deploy the same number of sensors over a scene and, for each occurring configuration, we employ our algorithm at every transmission instance. We then average the lifetime $\alpha$ over the deployments having the same coverage $\eta$. We note here that, as a result of the random deployment process, the transmit energy model employed should reflect the random distances between sensors and the base-station. Thus, hereafter, we assign to every
sensor the capability of transmitting on average $10^6 \pm 10^4$ bits of data. The result is shown in Figure 2.8. The horizontal and vertical axes represent the coverage $\eta$ and lifetime $\alpha$, respectively. The dashed curve is the lifetime of the network using the CRA algorithm, while the dotted and marked curves represent the 2 different implementations of the DRA algorithm presented in Section 2.3.3. The first implementation, indicated by the marked curve, uses the exhaustive Search-based DRA (S-DRA) algorithm. The second implementation, indicated by the dotted curve, uses our descent-based implementation of the DRA (D-DRA) algorithm. We also include in the figure a plot of the maximum achievable lifetime as predicted by our theoretical derivations (Equation (2.13)). This is shown with the solid bold curve. Finally, for comparison purposes, we include in the same graph (solid line) the performance of

Figure 2.7: Energy remaining in each sensor versus transmission instance for the D-DRA algorithm.
the baseline approach that does not exploit inter-sensor correlation, whereby every sensor sends everything in its SFOV to the base station.

Figure 2.8: Lifetime of the network as a function of scene coverage for 3 sensors.

We note the following from Figure 2.8. Firstly, we point out the general predicted trend: as coverage increases, the lifetime of the network decreases. At minimum coverage ($\eta = 0.11$), all SFOVs overlap and we are able to fully exploit the spatial redundancy and hence, the lifetime is at a maximum. At maximum coverage ($\eta = 0.33$), the SFOVs do not overlap at all, and the performance is identical to the baseline approach where the lifetime is at a minimum. Second, we note the performance of the baseline approach. Given that each sensor in this case is transmitting, on average, the same number of bits for all coverage then, lifetime $\alpha$ does not vary with $\eta$. Finally, we see from Figure 2.8 that the theoretical curve given by Equation (2.13) does indeed
represent the upper-bound for the lifetime performance. As mentioned earlier, this theoretical result represents the case when inter-sensor communication is allowed at no communication cost. In practice, any deviation from this upper-bound is due to the fact that it is not possible to always allocate the optimal number of bits to every sensor. One extreme example occurs when we have an “isolated” sensor, i.e., a sensor whose SFOV does not intersect any of the remaining SFOVs. Such a sensor must transmit its entire SFOV at each instance. This scenario drastically decreases the lifetime of the whole network.

Again, the results shown in Figure 2.8 are for a fixed number of sensors, or equivalently a fixed sensor density $\rho$. However, it is also important to see how the performance changes as the sensor density $\rho$ varies. To this end, we repeat the experiment for different numbers of sensors. The results for 2, 4, 8, and 12 sensors ($\rho = 0.22, 0.44, 0.88, 1.33$) are shown in Figures 2.9, 2.10, 2.11 and 2.12, respectively. We make the following remarks from these plots. First, we note that the range of coverage $\eta$ that occurred experimentally is different for the different numbers of sensors employed. For example, no coverage below 0.4 occurred with 12 sensors. In principle, the minimum expected coverage is 0.11 but, the probability of occurrence of this event (12 sensors fully overlapping) is negligible ($(384 \times 384)^{-12}$). Second, we note the absence of the S-DRA algorithm results from Figures 2.11 and 2.12. For that number of sensors, the number of possible configurations to search can be much larger than $2^{12}$ for each transmission. For these cases, exhaustive search becomes
infeasible. Finally, we observe that for the same $\eta$, $\alpha$ increases with $\rho$. To see this effect more clearly, we average $\alpha$ over all coverage $\eta$ for a fixed sensor density $\rho$, and plot the result $\langle \alpha \rangle$ versus $\rho$. This is shown in Figure 2.13. The horizontal and vertical axes represent the sensor density and the averaged network lifetime, respectively. The solid bold curve is the upper bound obtained using our theoretical treatment (Equation (2.15)). The result obtained using our CRA and D-DRA algorithms are represented by the dashed and dotted lines, respectively. The baseline approach is indicated by the marked line on the same graph.

![Figure 2.9: Lifetime of the network as a function of scene coverage for 2 sensors.](image)

The following remarks can be drawn from Figure 2.13. First, at low sensor densities $\rho$, we note the departure of the experimental results from those obtained from theory. This is expected. As mentioned earlier, a low sensor density increases the
Figure 2.10: Lifetime of the network as a function of scene coverage for 4 sensors. The possibility of occurrence of the “isolated” sensor event. However, this event is less likely to occur as the sensor density increases. This is manifested in the figure for $\rho \geq 1.0$. Finally we note that, when using our proposed algorithms, a gain in network lifetime up to 114% is eventually obtained.

All of the results presented thus far are obtained using imagery taken from Figure 2.5. We repeat the same process for 2 different types of imagery that show urban and rural landscapes. Low-resolution versions of the $4450 \times 4450$ grayscale images are shown in Figures 2.14 and 2.15, respectively. Here, we only reproduce the comprehensive plots of the expected network lifetime $\langle \alpha \rangle$ versus sensor density $\rho$. Figures 2.16 and 2.17 show the results for urban and rural landscapes, respectively. Comparing these 2 figures to Figure 2.13 we see that the general shape of the curve is the
Figure 2.11: Lifetime of the network as a function of scene coverage for 8 sensors.

same but simply shifted up or down. An easier to compress image (e.g. rural) has a lower per-pixel transmit cost \( b \) compared to the coastline image and hence, a higher expected lifetime. Whereas a harder to compress image (e.g. urban) has a higher \( b \) and thus a lower lifetime compared to the coastline image.

2.5.3 Lossy representation

The sensors in our experiments have been encoding all regions \( R_j \) in a lossless fashion. It can readily be seen from Equation (2.13) that we can increase the lifetime \( \alpha_{opt} \) by further decreasing the bit cost per pixel, \( b \). We can thus trade-off the fidelity of the received measurements against the lifetime of the sensor network. Here, we choose to quantify fidelity in terms of Root Mean Squared Error (RMSE). A higher
Figure 2.12: Lifetime of the network as a function of scene coverage for 12 sensors.

RMSE implies a lower fidelity, which in turn yields a higher number of transmissions for the network (for the same initial battery power).

To illustrate this point, we show in Figure 2.18 a plot of lifetime $\alpha$ versus scene coverage $\eta$ for our example 3-sensors network deployment. The figure shows the results of the algorithms when applied to the coastline landscape image. The resulting received images have an RMSE of 2 (or 0.78%). We can compare this figure with the corresponding Figure 2.8 where the RMSE of the received measurements is 0 (lossless). We readily see an overall upward shift in all the curves and, on average, an increase in the lifetime of the network by a factor of 2.82 over the (optimized) lossless case. For the visual assessment of the results, we show in Figures 2.19 and 2.20 an SFOV reconstructed at RMSE levels of 0 and 0.78%, respectively.
Figure 2.13: Average lifetime of the network versus the sensor density.

Finally, Figure 2.21 shows a plot of the average lifetime $\langle \alpha \rangle$ versus sensor density $\rho$ for various RMSE levels (1, 2, 4 and 6). The 4 families of curves correspond to the 4 different RMSE levels and are indicated by the 4 different symbol types. Each family of curves includes a plot of the derived theory (solid curve) and both the CRA (dashed) and D-DRA (dotted) algorithms. The latter 2 curves are almost indistinguishable. Again, we note the increase in network lifetime for the same sensor density compared to the lossless case. For example, for an RMSE of 6, the expected lifetime of the network at $\rho = 1$ increased by a factor of 17.8 compared to the lossless (optimized) case. For applications where a certain level of fidelity loss at the receiver is tolerated, this approach can offer tremendous factors of improvements in network lifetime.
2.6 Concluding remarks

In this chapter, we developed a heuristic-based method that exploits the redundancy in the measurements of imaging sensors in order to maximize the lifetime of the network without utilizing inter-sensor communication [15]. The results showed gains in network lifetime up to 114% over traditional approaches. However, we have seen that in instances with small amounts of overlap, the proposed algorithm does not
Figure 2.16: Average lifetime of the network versus sensor density, for the rural image map.

offer significant advantages over baseline methods. In those cases of little inter-sensor correlation, we need to resort to multi-hop routing (inter-sensor communication) in order to balance the load over the network. Thus, we ask the following fundamental question: how can we utilize multi-hop routing (inter-sensor communication) in order to maximize the lifetime of a network of imagers? In Chapter 3, we consider sensors with no spatial inter-sensor correlation structure and focus on the topic of multi-hop routing in a wireless sensor network.
Figure 2.17: Average lifetime of the network versus sensor density, for the urban image map.

Figure 2.18: Lifetime of the network as a function of scene coverage for 3 sensors. The reconstructed images have a normalized RMSE of 0.78%.
Figure 2.19: Original FOV sampled from the coastline landscape image.

Figure 2.20: Same FOV as in Figure 2.19 but transmitted at a normalized RMSE of 0.78%.
Figure 2.21: Average lifetime of the network plotted versus the sensor density, for the coastline image map, with images transmitted at different normalized RMSE levels of 0.39%, 0.78%, 1.56% and 2.35%.
CHAPTER 3

A THEORY FOR MAXIMIZING THE LIFETIME OF SENSOR NETWORKS

A novel theory is developed in this chapter for maximizing the lifetime of unicast multi-hop wireless sensor networks [13]. The sensors treated in this chapter are assumed to have no correlation structure amongst their measurements. Section 3.1 reviews prior work on maximum lifetime routing in sensor networks. We formulate our problem in Section 3.2. We then present the proposed solution. Section 3.3 outlines our algorithm and the intuition behind it. The theorems that prove the optimality of our algorithm are presented in Section 3.4. We present experimental results in Section 3.5 and compare with results from prior work.

3.1 Background on network lifetime

As we pointed out in Chapter 1, the scope of problems covered by sensor network research is vast and encompasses numerous aspects of distributed sensing including networking [23, 24], data fusion [37, 41, 51], signal processing [31, 52, 55], power control [26, 29, 53, 58, 73], communication theory [29, 38, 58, 71], etc. An important issue in wireless sensor networks is the limited energy and bandwidth resources [22].
This fact has motivated recent efforts that aim to maximize the lifetime of a network of sensors.

In a unicast network routing model, Chang and Tassiulas [9] considered the problem of maximizing the network lifetime, which they defined to be the time until the first node runs out of battery power. In fact, the authors in [9] were the first to treat this problem as a linear programming problem. Subsequent work in the literature [10,39,57] proposed distributed algorithms to solve this linear programming problem. In multicast networks, authors in [16,35] presented an optimal routing strategy that maximizes the network lifetime, as defined above. Other definitions of network lifetime have been proposed in the literature. These definitions include (but are not limited to), the time until a fraction of sensors run out of battery [22], the average sensor lifetime [65], the time until the first loss of some desired coverage [6], etc. We should also note that although most of the methods proposed to solve the lifetime maximization problem are numerical in nature, some analytical upper bounds on network lifetime could be derived as in [6,27,28]. Finally, we should point out that the research efforts discussed above have mainly focused on optimizing the energy consumption in data communication in order to control the sensor lifetime. However, it has been demonstrated that controlling other network parameters, such as the “sleep” and “wake-up” schedule of sensor nodes [7,8], could also be utilized to further increase the lifetime of the network. For a more complete survey on network
lifetime maximization problems in various energy consumption models, the reader is referred to [20].

In this chapter, we consider the problem of maximizing the lifetime of a unicast network. Specifically, we develop a theory and a corresponding algorithm that maximize the $n^{th}$ minimum sensor lifetime in the network subject to the $(n-1)^{th}$ minimum lifetime being maximum . . . subject to the minimum lifetime being maximum, where $n$ is an arbitrary positive integer less than or equal to the total number of sensors in the network. The choice of $n$ depends on the application of interest.

Note that in the case when $n = 1$, our lifetime definition becomes identical to that introduced in [9]. In order to motivate the need to maximize sensor lifetimes beyond that of the minimum, consider a case in which the network is limited by a single “weak” sensor, i.e., a sensor that has a much higher energy consumption per bit as compared to the rest of the sensors. No matter how the energy of various sensors is employed, that particular “weak” sensor will always run out of energy at the same time. The lifetime of the network, as defined in [9], is thus equal to the lifetime of that “weak” sensor. Thus, in this case, there exist multiple solutions that will yield the same network lifetime. However, each of those solutions will result in different lifetimes for the remaining sensors. One can readily foresee many applications in which it is desired to proceed with the operation of the sensor network well after the first sensor runs out of battery. Hence, it is essential to consider transmission
strategies that take into account sensor lifetimes well after the battery depletion of
one or more sensors.

3.2 Problem statement

3.2.1 Physical model and definitions

The sensor network model employed in this chapter is illustrated in Figure 3.1. This model is similar to the one introduced in the previous chapter. We will use this figure to introduce some of the terminology used throughout this chapter. Specifically, $N$ sensors are deployed some distance away from a central base station. We treat here the case of a unique receiver (base station) but, this work could be easily extended to include multiple receivers. We denote by $B_i, \{i = 1, 2, ..., N\}$, the remaining energy (in Joules) of the $i^{th}$ sensor prior to a transmission event. For this transmission event, sensor $i$ has an initial load $x_{i,i}$ (bits) that it needs to transmit to the base station via some route. A route could consist of a single-hop transmission (all the load sent directly to the base station), and/or multi-hop transmissions (some portion of the load sent indirectly via other sensors). The branches of the graph, labeled $x_{i,j}$, denote the number of bits that will be routed from sensor $i$ to sensor $j, \{i, j = 1, 2, ..., N; i \neq j\}$. $x_{i,0}$ indicates the number of bits that will be directly transmitted from sensor $i$ to the base station.
A transmission along any given route can be assigned a cost. The communication cost (in Joules/bit) along a given path depends on multiple factors including the path-loss of the channel, the length of the path, the cost of processing a bit, the cost of receiver electronics, etc. Thus, different sensors could have different transmission capabilities depending on their respective communication costs. We shall denote by \( t_{i,j} \) and \( r_{j,i} \) the costs incurred by sensor \( i \) for transmitting and receiving one bit to/from sensor \( j \), respectively. Also define \( t_{i,0} \) as the cost incurred by sensor \( i \) for transmitting one bit to the base station. Finally, it is convenient to define \( t_{i,i} \equiv 0 \) and \( r_{i,i} \equiv 0 \).

From these definitions, we can write the total amount of energy spent by a given sensor \( i \) for a given transmission event as:

\[
\xi_i = \sum_{j=0}^{N} x_{i,j} t_{i,j} + \sum_{j=1}^{N} x_{j,i} r_{j,i}, \quad (J)
\]
where the first sum represents the total cost of the transmitted data and the second sum represents the total cost of the received data. Using vector notation we can rewrite Equation (3.1) in the following form:

$$\xi_i(x) = x \left( t_i^\dagger + r_i^\dagger \right),$$

(3.2)

where, $x$ is the $N(N + 1)$ vector of variables given by

$$x \equiv [x_{1,0} \ x_{1,1} \ \ldots \ x_{1,N} \ \ldots \ x_{N,0} \ x_{N,1} \ \ldots \ x_{N,N}],$$

(3.3)

and $t_i$ and $r_i$ are respectively the transmission and receipt cost vectors associated with sensor $i$, whose $k^{th}$ element ($k = 1, \ldots, N(N + 1)$) is given by:

$$ (t_i)_k = \begin{cases} 
  t_{i,j} & \text{if } k = (i - 1)(N + 1) + (j + 1), \\
  0, & \text{otherwise,}
  
\end{cases}$$

and,

$$ (r_i)_k = \begin{cases} 
  r_{j,i} & \text{if } k = (j - 1)(N + 1) + (i + 1), \\
  0, & \text{otherwise.}
  
\end{cases}$$
The lifetime of sensor \( i \) will obviously be a function of the data transmission strategy and can now be written as,

\[
\alpha_i(x) = \frac{B_i}{\xi_i(x)},
\]

where \( \xi_i(x) \) is given by Equation (3.2). Note that Equation (3.4) assumes that the network is stationary. This is a reasonable assumption given no prior knowledge about the variation of sensor loads or network distances over time. Moreover, this assumption is particularly useful in instances where data are acquired at a fixed rate (e.g., images acquired at a fixed coding rate [12]). The dynamic version of the problem, in which the loads change over time, is not considered here.

The task of each sensor is to transmit its load to the base station via some route. Thus, we must ensure that at every sensor node \( i \), the total transmitted load equals the sum of sensor \( i \)'s initial load and the total load received from other sensors. This translates to the following “conservation of load” constraint:

\[
\sum_{j=1}^{N} x_{j,i} = \sum_{j=0, j \neq i}^{N} x_{i,j}, \forall i = 1, 2, ..., N.
\]

**Definition:** Define \( C \) to be the constraint set. That is, \( x \in C \) if the elements of \( x \) are non-negative and satisfy Equation (3.5).

### 3.2.2 Formulation

We assume that the operation of every sensor is equally important to the network as a whole. Thus, it would be desirable to maximize the lifetime of every sensor in
this network. However, this is not always possible to achieve because the lifetime functions are not independent of one another. This is a result of the imposed constraint (Equation (3.5)): reducing traffic along a given route inherently increases the traffic along other routes. We illustrate this with an example. For simplicity, we assume here the case of \( N = 2 \) sensors with equal initial batteries \( B_1 = B_2 = 1 \) (J) and with initial loads, \( x_{1,1} = 10^3 \) bits, \( x_{2,2} = 10^2 \) bits. The transmit costs chosen here are \( t_{1,0} = 2 \times 10^{-3} \), \( t_{2,0} = 7 \times 10^{-4} \) (J/bit) and, \( t_{1,2} = t_{2,1} = 0 \) (no inter-sensor communication cost). The cost of receiving data is also chosen to be zero here. Note that, in this example, specifying \( x_{1,0} \) completely specifies \( x_{2,0} \) via the conservation of load constraint, and vice versa. So, only one of those variables is needed to determine the sensor lifetimes. Figure 3.2 shows the plot of \( \alpha_1(x) \) (solid line) and \( \alpha_2(x) \) (dotted line) versus \( x_{1,0} \). The vertical axis is shown on a log-scale for clarity. As we can see from this example, it is not possible here to simultaneously maximize the lifetimes of both sensors.

In general, this is a multi-objective (MO) optimization problem in which the objectives are the respective sensor lifetimes. In this paradigm, one notion of optimality is that of Pareto Optimal (PO) solutions. A vector \( x^* \in C \) is said to be Pareto Optimal if it is not possible to improve one objective without making at least one other objective worse. Stated in more formal terms [63]:

**Definition:** A vector \( x^* \in C \) is said to be Pareto Optimal if there exists no \( x' \in C \)
Figure 3.2: Plot of sensor lifetimes versus $x_{1,0}$ for the example network of 2 sensors.  

such that (s.t.) $\alpha_i(x') \geq \alpha_i(x^*), \forall i = 1, \ldots, N$, with at least one strict inequality.

**Definition:** Let $\mathcal{P}$ be the subset of solutions in $\mathcal{C}$ that are PO.

In general, there could exist (infinitely) many PO solutions. As can be seen from Figure 3.2, any $x \in \mathcal{C}$ is PO: $\alpha_1(x)$ decreases as $\alpha_2(x)$ increases for all $x_{1,0} \in \mathcal{C}$.

Among the set of PO solutions, we want to choose the solution that is most suitable for our application. For example, if the network is useful only when all the sensors are simultaneously “alive” then, we should maximize the minimum lifetime among all sensors. However it may also be important to maintain the network until the next sensor runs out of battery. In this case, we would want to maximize the second minimum lifetime subject to the minimum lifetime being maximum.
In this chapter, we treat the following problem: we want to find the PO solution that maximizes the $n^{th}$ minimum lifetime such that the $(n - 1)^{th}$ minimum lifetime is maximized ... such that the $2^{nd}$ minimum lifetime is maximized such that the minimum lifetime is maximum, where $n$ is any integer between 1 and $N$. Hereafter, we refer to this desired solution as the solution that maximizes the “$n^{th}$ conditional lifetime.”

Formally, for any integer $n$, $1 \leq n \leq N$, we want to find:

$$x^* = \underset{x \in C}{\text{argmax}} \left[\min_{i} \{\alpha_i(x)\}\right] \quad s.t. \quad \min_{i} \{\alpha_i(x^*)\} \text{ is maximum s.t.}$$

$$\ldots$$

$$\min_{i} \{\alpha_i(x^*)\} \text{ is maximum,}$$

where $\min_{i} \{\alpha_i(x)\}$ denotes the $k^{th}$ minimum lifetime amongst $\{\alpha_i(x)\}, \forall i = 1, \ldots, N$.

We should note that a similar problem considered in the networking community is that of max-min fairness [5, 59]. The concept of max-min fairness arises mainly in problems that aim at allocating bandwidth to a set of pre-determined routes in a wired network [5, 32, 44]. Informally, in that context, an allocation is called max-min fair if one can not increase the bandwidth of one route without decreasing the bandwidth allocated to another route with lower or equal bandwidth [5]. Several algorithms have been shown to achieve the max-min fair allocation in specific wired network topologies [44]. To the best of our knowledge, none of those algorithms
have yet been shown to solve the wireless network lifetime maximization problem defined in (3.6). In this work, we present an algorithm that can exactly achieve the optimal solution to (3.6) without any constraints on the network topology. Note that we consider here the case where the allocations (loads) can be arbitrarily split among multiple paths. The integral counterpart of the problem, shown to be NP-hard in [3], is not considered in this dissertation. Finally, we should note that our algorithm specifies a centralized solution: the base station is responsible for the necessary computations. Distributed algorithms that attempt to solve the problem in (3.6) are beyond the scope of this dissertation.

3.3 An optimal algorithm

We have derived a theory, and a corresponding algorithm, to solve the problem defined above. For the purpose of clarity, we choose to first introduce the optimal algorithm. The theorems that prove the optimality of the algorithm will be presented in Section 3.4 along with their proofs.

Before proceeding further, we first outline a basic result from Section 3.4 that offers some intuition into the algorithm. We show in Theorem 1 that if there exists a Pareto Optimal solution in the constraint set $C$ that makes all lifetimes equal then, this solution maximizes the $N^{th}$ conditional lifetime. This is easily seen for the simple example of Figure 3.2. In that example, the solution that maximizes the second minimum lifetime such that the minimum lifetime is at its maximum is the
one where the two curves cross; or equivalently, when both lifetimes are equal. This solution, like all solutions in that example, is PO. This fact provides motivation for methods found in the literature that try to equalize the lifetimes of the sensors in an attempt to maximize the minimum lifetime \([15, 65, 74]\). Unfortunately, we are not always guaranteed to find a solution in \(P\) that equalizes all lifetimes. In what follows, we present an iterative algorithm that initially searches for the PO solution that equalizes all lifetimes. If this sought after solution does not exist, the algorithm then proceeds in an iterative fashion. We show that the algorithm eventually attains the desired objective of maximizing the \(n^{th}\) conditional lifetime. The number of steps required for convergence is upper-bounded by \(n\).

Specifically, in the first iteration, our algorithm finds the maximum value to which all lifetimes can be set equal. This is possible because, as Theorem 2 shows, we can always find for any sensor \(i\) a vector \(x_e \in C\) such that \(\alpha_i(x_e) = \alpha_0\), where \(\alpha_0\) is any constant such that \(0 \leq \alpha_0 \leq \max_{x_e \in C} \{\alpha_i(x_e)\}\).

Hereafter, we will use the superscript notation \((n)\) to denote quantities obtained at iteration \(n\), \(\forall n = 1, \ldots, N\). Thus, let \(x_e^{(1)}\) be a solution that makes all lifetimes equal to a maximum value \(\alpha^{(1)}\). If the solution thus obtained is PO then, as we argued above, we can stop and declare that we found a solution that maximizes the “\(N^{th}\) conditional lifetime.” Otherwise, in the case where \(x_e^{(1)}\) is not PO then, it is obvious that we could improve the lifetime of at least one sensor without decreasing the lifetimes of the remaining sensors. However, we show in Lemma 2 that there
exists a specific sensor $j$ whose lifetime can not be increased without decreasing the minimum lifetime $\alpha^{(1)}$. We show in Section 3.4 that this important fact has two immediate implications. First, it means that $\alpha^{(1)}$ is the maximum minimum lifetime ($Theorem 3$). Second, it also implies that, should we need to look for solutions that improve other sensor lifetimes, we have to do so while keeping the lifetime of sensor $j$ constant.

Next, we “set aside” this sensor $j$ in a set $E^{(1)}$. In the case where we have more than one choice of such sensors, we arbitrarily choose the sensor with the minimum index. We then proceed to the next iteration where the task is to maximize the minimum lifetime of those sensors not in $E^{(1)}$. We apply similar steps as in the first iteration. Specifically, we find the maximum value $\alpha^{(2)}$ to which we can set equal all lifetimes of those sensors not in $E^{(1)}$. We prove in $Theorem 4$ that at the end of this second iteration we have a solution that maximizes the second minimum lifetime such that the minimum lifetime is maximum. We can iterate the same steps until we find the first $x^{(n)} \in \mathcal{P}$, where $n$ is the iteration index. The algorithm is outlined below.
1- Initialization: set $E^{(l)} = \{\}, l = 1, 2, \ldots, N$; and set $n = 1$,

2- Choose $x_e^{(n)} \in C$ such that $\alpha^{(n)}$ is maximum, where

$$\alpha_i(x_e^{(n)}) = \alpha_k(x_e^{(n)}) \equiv \alpha^{(n)} \forall i, k \notin \bigcup_{m=1}^{n-1} E^{(m)}$$

subject to $\alpha_j(x_e^{(n)}) = \alpha^{(m)}$, if $j \in E^{(m)}$, $m = 1, \ldots, n - 1$

3- If $x_e^{(n)} \in \mathcal{P}$, done.

4- Else, define set $E^{(n)}$ such that,

$$E^{(n)} = \left\{ \min\{j\} : \exists x_e \in C \text{ s.t. } \alpha_j(x_e) > \alpha^{(n)} \text{ subject to} \begin{align*} 
\alpha_i(x_e) &\geq \alpha^{(n)}, \forall i \notin \bigcup_{m=1}^{n-1} E^{(m)}, i \neq j \\
\alpha_l(x_e) &= \alpha^{(m)}, \text{ if } l \in E^{(m)}. 
\end{align*} \right\}$$

5- $n = n + 1$

6- Go to 2.

We make the following remarks about the algorithm in (3.7):

1. The maximum number of iterations required to find the solution that maximizes the $N^{th}$ conditional lifetime is $N$, the number of sensors in the network. However, note that this is not a tight upper-bound. We could have achieved the same result with fewer iterations had we excluded (in set $E^{(n)}$) more than just one sensor at a time. Nevertheless, we chose to present the algorithm as it is for the purpose of clarity. Also, note that we do not need to maintain separate sets $E^{(n)}$, $\forall n = 1, \ldots, N$, but chose to do so for the sake of clarity.

2. The algorithm is guaranteed to find a PO solution. To see this, consider the
last iteration \( n = N \). This iteration will find a vector \( \mathbf{x}^{(N)}_e \in \mathcal{C} \) that maximizes \( \alpha_i(\mathbf{x}^{(N)}_e) \) for only that index \( i \notin \bigcup_{m=1}^{N-1} E^{(m)} \). This is done such that \( \alpha_j(\mathbf{x}^{(N)}_e) = \alpha^{(m)}_j \), for \( j \in E^{(m)}, \forall m = 1, \ldots, N - 1 \), where \( \alpha^{(m)}_j \) is the maximum \( m \)th conditional lifetime. Hence, \( \nexists \mathbf{x}' \) s.t. \( \alpha_i(\mathbf{x}') \geq \alpha_i(\mathbf{x}^{(N)}_e) \forall i \) with at least one strict inequality.

3. We note also that, depending on the application of interest, we can stop the algorithm at any iteration \( n \) and then obtain a solution that maximizes the \( n \)th conditional lifetime.

3.3.1 Computational complexity

We discuss here some of the implementation considerations of the algorithm as outlined in (3.7). Using Equation (3.4), Step 2 of the algorithm could be easily converted to one linear programming (LP) problem. Step 3 involves determining whether a given solution \( \mathbf{x}^{(n)}_e \) is Pareto Optimal. To this end, we check each sensor \( k \) to determine if it is possible to find a better lifetime \( \alpha_k(\mathbf{x}^{(n)}_e) > \alpha_k(\mathbf{x}^{(n)}_e) \) subject to other sensor lifetimes being at least as good \( \alpha_i(\mathbf{x}'_e) \geq \alpha_i(\mathbf{x}^{(n)}_e), i \neq k \). To do so, we
implement the following algorithm:

\[
\text{for each } k \notin \bigcup_{m=1}^{n} E(m)
\]

\[\rightarrow \text{find } x'_e \text{ s.t.}
\]

\[x'_e \in C\]

\[x'_e ((t_e)_k + (r_e)_k)^\top < \frac{B_k}{\alpha_k(x^{(n)}_e)}\]  \hspace{1cm} (3.8)

\[x'_e ((t_e)_i + (r_e)_i)^\top \leq \frac{B_i}{\alpha_i(x^{(n)}_e)}, \forall i = 1, \ldots, N, i \neq k.
\]

\[\rightarrow \text{If a solution exists, then stop loop and}
\]

\[\text{declare that } x^{(n)}_e \text{ is not PO.}
\]

end

Each \(k\) in (3.8) can be tested with one LP instance. We don’t need to check those sensors that belong to any of the \(E^{(m)}\) sets. The lifetime of such sensors can not be increased without decreasing the lifetime of at least one other sensor. Thus, no more than \(N - n + 1\) LP instances are required for iteration \(n\).

Step 4 of the algorithm involves finding the first sensor \(j \notin \bigcup_{m=1}^{n} E(m)\) whose lifetime can not be increased without decreasing the lifetime of at least one other sensor. Similar to (3.8), it can be shown that one LP instance is needed for each \(k \notin \bigcup_{m=1}^{n} E^{(m)}\) until the first such sensor is found. The corresponding index \(j\) is added to \(E^{(n)}\). Again, no more than \(N - n + 1\) LP instances are required at iteration \(n\).

From the discussion above, the maximum number of LP instances at iteration \(n\) is \(2(N - n + 1) + 1\). If all iterations are required, \(n = 1, 2, \ldots, N\), the maximum total
LP instances is upper bounded by $N(N + 1) + 1$. However, this number is not a tight upper bound. On average, fewer than half this many LP instances are required to achieve the solution that maximizes the $N^{th}$ conditional lifetime.

Finally, we note the following important result. If it is desired to maximize only the minimum lifetime, then we only need to apply Step 2 of the algorithm once. Hence, according to our formulation, only one LP instance is needed to maximize the minimum lifetime.

3.4 Theory

In this section, we present the theorems that show the optimality of the algorithm presented above.

The following theorem motivates Step 2 of the first iteration.

**Theorem 1**: If $\exists x \in P$ such that $\alpha_1(x) = \alpha_2(x) = \ldots = \alpha_N(x) \equiv \alpha$ then $x$ maximizes the $N^{th}$ conditional lifetime.

**proof**: $x \in P \Rightarrow \exists x' \in C$ s.t. $\alpha_i(x') > \alpha$ for some $i$, then we must have $\alpha_j(x') < \alpha$ for some $j$. Thus, $\exists x' \in C$ such that $\min_i \{\alpha_i(x')\} > \alpha$ so, $x$ maximizes the minimum lifetime. Similarly, if we can find any $x' \in C$ s.t. $\min_i \{\alpha_i(x')\} = \alpha$ and $\alpha_k(x') > \alpha$ for any $k$ then, there must be at least one $j$ such that $\alpha_j(x') < \alpha$. So,
x maximizes the second minimum lifetime such that the minimum lifetime is maximum. Proceeding in a similar manner, it is straightforward to show that \( \alpha \) is the maximum \( N^{th} \) conditional lifetime and hence, \( x \) is the desired solution.

The main implication of Theorem 1 asserts that if we can find a Pareto Optimal solution in our constraint set \( \mathcal{C} \) that makes all lifetimes equal then we have attained our objective.

Now, in order to completely validate the second step of our algorithm, we need to argue that it is always possible to find a solution in \( \mathcal{C} \) that makes all lifetimes equal. To do so, we start by introducing in the network of \( N \) sensors a virtual node. The idea is to use this node as an “energy sink”: each sensor can independently waste an arbitrary amount of its energy by communicating with this energy sink. The energy sink, now node element \( N + 1 \) in the network, has the following properties:

1. The energy sink node has no initial load, \( x_{N+1,N+1} = 0 \),
2. The energy sink node sends back to sensor \( i \) the same load it had received from it, \( x_{N+1,i} = x_{i,N+1} \),
3. The energy sink node transmits nothing to the base station, \( x_{N+1,0} = 0 \).
4. The cost of transmitting 1 bit from sensor \( i \) to the energy sink node (denoted by \( t_{i,N+1} \)) is non-zero.

It should be clear then that sending (and receiving back) data to (from) node \( N + 1 \) is just a way to “waste” energy at sensor \( i \). Next, we need to incorporate the addition of this energy sink node into the problem formulation of Section 3.2.2. This can be
done simply by increasing the number of sensors by one, \( M \equiv N + 1 \). The number of variables now becomes \( M \times (M + 1) \). Thus, similar to Equation (3.4), we can write the lifetime of sensor \( i, i = 1, \ldots, N \), as:

\[
\alpha_i(x_e) = \frac{B_i}{x_e ((t_e)_i + (r_e)_i)^\top} = \frac{B_i}{x (t_i + r_i)^\top + (x_{i,N+1} t_{i,N+1}) + (x_{N+1,i} r_{N+1,i})}, \tag{3.9}
\]

where \( x_e, t_e \) and \( r_e \) denote the “extended version” of vectors \( x, t \) and \( r \), respectively. We should clarify that the additional variables \( x_{i,N+1} \) and \( x_{N+1,i} \) have no physical meaning and can be disregarded once the final solution is obtained. It will become clear next that if the extended vector \( x_e \) maximizes a given conditional lifetime then, the corresponding vector \( x \) maximizes that same conditional lifetime as well.

We will show next that the addition of this extra node does not change the constraint set \( C \).

**Lemma 1**: \( x_e \in C \iff x \in C \).

**Proof**: If \( x_e \in C \) then, by definition, we have

\[
\sum_{j=1}^{N} x_{j,i} = \sum_{j=0,j\neq i}^{N+1} x_{i,j}, \quad \forall \ i = 1, 2, \ldots, N + 1, \Rightarrow \sum_{j=1}^{N} x_{j,i} + x_{N+1,i} = \sum_{j=0,j\neq i}^{N} x_{i,j} + x_{i,N+1}. \]

But, \( x_{N+1,i} = x_{i,N+1} \) by virtue of Property 2 above so, \( \sum_{j=1}^{N} x_{j,i} = \sum_{j=0,j\neq i}^{N} x_{i,j} \Rightarrow x \in C \). Similarly, we could show that \( x \in C \Rightarrow x_e \in C \). □

**Definition**: We say \( x^*_e \in \mathcal{P} \) if \( x^*_e \in C \) and, there exists no \( x' \in C \) such that

\[
\alpha_i(x') \geq \alpha_i(x^*_e), \quad \forall i = 1, \ldots, N, \end{equation}

with at least one strict inequality. Note that if \( x^*_e \in \mathcal{P} \) then, \( x^* \in \mathcal{P} \).

Now, we are ready to show that we can always make the sensor lifetimes equal.
Theorem 2: Let $x \in C$ and let $\alpha_0 = \min_i \{\alpha_i(x)\}$. Then, $\exists x' \in C$ s.t. $\alpha_1(x') = \alpha_2(x') = \cdots = \alpha_N(x') = \alpha_0$.

Proof: Let $x' = x$. The aim is to determine the parameters $x'_{i,N+1}$ of $x'$ so as to make $\alpha_i(x') = \alpha_0, \forall i = 1, \ldots, N$. We can readily do that simply by setting $\alpha_i(x_e) = \alpha_0$ in Equation (3.9) and then proceeding to solve for the desired $x'_{i,N+1}$ for each sensor $i$. Thus, we can find $x'_e \in C$ s.t. $\alpha_i(x'_e) = \alpha_0, \forall i = 1, \ldots, N$. \[\blacksquare\]

Next, we will present the theorem that completes Step 2 of the first iteration of our algorithm. Specifically, we will show that this initial step will yield a solution that maximizes the minimum lifetime of the network. Hereafter, we use the superscript $(n)$ to denote quantities obtained at iteration $n, n = 1, 2, \ldots, N$.

Theorem 3: If $\alpha^{(1)}$ is the maximum lifetime such that $\alpha_1(x^{(1)}_e) = \alpha_2(x^{(1)}_e) = \cdots = \alpha_N(x^{(1)}_e) \equiv \alpha^{(1)}$, then $\alpha^{(1)}$ is the maximum minimum lifetime.

Proof:

- If $x^{(1)}_e \in P$ then $x_{i,N+1} = 0, \forall i$ by definition of Pareto Optimality. It then follows that $\alpha_1(x^{(1)}) = \alpha_2(x^{(1)}) = \cdots = \alpha_N(x^{(1)}) = \alpha^{(1)}$ and, because $x^{(1)} \in P$ then, Theorem 1 implies that $x^{(1)}$ is the desired solution that maximizes the $N^{th}$ conditional lifetime.

- Else, if $x^{(1)}_e \notin P$ then $\exists x' \in C$ s.t. $\alpha_i(x') \geq \alpha^{(1)}$, with at least one strict inequality. But, if strict inequality holds for all $i$ then $\alpha' \equiv \min_i \{\alpha_i(x')\} > \alpha^{(1)}$. Theorem 2 then implies that there exists $x'_e \in C$ s.t. $\alpha_1(x'_e) = \alpha_2(x'_e) = \cdots = \alpha_N(x'_e) = \alpha'_e$. \[\blacksquare\]
\( \alpha' > \alpha^{(1)} \), which is contradictory to our hypothesis. So, there does not exist any \( \mathbf{x}' \in \mathcal{C} \) s.t. \( \min_i \{ \alpha_i (\mathbf{x}') \} > \alpha^{(1)} \).

If \( \mathbf{x}^{(1)} \) is not PO, then we could improve the lifetime of at least one sensor without decreasing the lifetimes of the remaining sensors. On the other hand, Step 4 of the algorithm claims that there exists a specific sensor \( j \) whose lifetime can not be increased without decreasing the minimum lifetime \( \alpha^{(1)} \). We prove this next.

**Lemma 2:** \( \exists j \) for which \( \nexists \mathbf{x}_e \in \mathcal{C} \) such that \( \alpha_j (\mathbf{x}_e) > \alpha^{(1)} \) and \( \alpha_i (\mathbf{x}_e) \geq \alpha^{(1)} , \forall i \neq j \).

**Proof:** Assume that for each \( j = 1, \ldots, N \), it is possible to find a vector \( \mathbf{x}_{ej} \in \mathcal{C} \) such that

\[
\alpha_j (\mathbf{x}_{ej}) > \alpha^{(1)}
\]

and,

\[
\alpha_i (\mathbf{x}_{ej}) \geq \alpha^{(1)}, \forall i \neq j.
\]

Combining Equation (3.9) and Equation (3.10) yields

\[
\mathbf{x}_{ej} \left( (t_e)_j + (r_e)_j \right)^\dagger < \xi_j^{(1)}
\]

and,

\[
\mathbf{x}_{ej} \left( (t_e)_i + (r_e)_i \right)^\dagger \leq \xi_i^{(1)}, \forall i \neq j,
\]
where $\xi_k^{(1)} = \frac{R_k}{\alpha^{(1)}}$. Now, for any sensor $k$, consider the following

$$\frac{1}{N} \sum_{j=1}^{N} x_{ej} \left( (t_e)_k + (r_e)_k \right)^\dagger =$$

$$\frac{1}{N} \sum_{j=1,j\neq k}^{N} x_{ej} \left( (t_e)_k + (r_e)_k \right)^\dagger +$$

$$\frac{1}{N} x_{ek} \left( (t_e)_k + (r_e)_k \right)^\dagger <$$

$$\frac{1}{N} \sum_{j=1}^{N} \xi_k^{(1)} = \xi_k^{(1)},$$

where strict inequality follows directly from Equation (3.11). Furthermore, the first term in Equation (3.12) can be written as:

$$\frac{1}{N} \sum_{j=1}^{N} x_{ej} \left( (t_e)_k + (r_e)_k \right)^\dagger = x'_e \left( (t_e)_k + (r_e)_k \right)^\dagger,$$

where $x'_e = \frac{1}{N} \sum_{j=1}^{N} x_{ej}$. Equations (3.12) and (3.13) then imply that,

$$x'_e \left( (t_e)_k + (r_e)_k \right)^\dagger < \xi_k^{(1)}, \forall k = 1, \ldots, N,$$

or, equivalently,

$$\alpha_k (x'_e) > \alpha^{(1)}, \forall k.$$

Now, since $x_{ej} \in \mathcal{C}$ and noting that $\mathcal{C}$ is clearly convex then, we conclude that $x'_e \in \mathcal{C}$. Thus, inequality (3.14) $\Rightarrow \exists x'_e \in \mathcal{C}$ s.t. $\alpha_i (x'_e) > \alpha^{(1)}, \forall i = 1, \ldots, N$, which is not possible because $\alpha^{(1)}$ is the maximum minimum lifetime. Thus, there exists $j$ for which we can not find $x_e \in \mathcal{C}$ satisfying the inequalities of Equation (3.10).

Step 4 of the algorithm then “sets aside” in a set $E^{(1)}$ (defined in (3.7)) any one sensor $j$ that satisfies the condition of Lemma 2. We are ready to proceed to the
second iteration where the task is to maximize the minimum lifetime of those sensors not in \(E^{(1)}\). We apply similar steps as in the first iteration.

Next, we show that the solution obtained at Step 2 of the second iteration maximizes the second conditional lifetime.

**Theorem 4:** Let \(x_e^{(2)}\) be a solution that maximizes \(\alpha^{(2)} \equiv \alpha_i(x_e^{(2)}) = \alpha_k(x_e^{(2)}), \forall i, k \notin E^{(1)}\), such that \(\alpha_j(x_e^{(2)}) = \alpha^{(1)}\) if \(j \in E^{(1)}\). Then, \(x_e^{(2)}\) maximizes the second minimum lifetime such that the minimum lifetime is maximum.

**Proof:**

- If \(x_e^{(2)} \in P\) then, using similar arguments to Theorem 1, we can proceed as follows. If \(\exists x' \text{ s.t. } \alpha_j(x') = \alpha^{(1)}\) for \(j \in E^{(1)}\) and \(\alpha_i(x') > \alpha^{(2)}\) for any \(i \notin E^{(1)}\) then \(\exists k \notin E^{(1)} \text{ s.t. } \alpha_k(x') < \alpha^{(2)}\). Thus, \(x_e^{(2)}\) (and equivalently, \(x^{(2)}\)) maximizes the second minimum lifetime such that the minimum lifetime is maximum. Similarly, any attempt to increase the 3\(^{rd}\), 4\(^{th}\), \ldots, or \(N^{th}\) minimum lifetime subject to the minimum lifetime being maximum will decrease the 2\(^{nd}\) minimum lifetime. So, \(x^{(2)}\) is the solution that maximizes the \(N^{th}\) conditional lifetime and, we can stop the iterations here.

- If \(x_e^{(2)} \notin P\) \(\Rightarrow \exists x' \text{ s.t. } \alpha_j(x'_e) = \alpha^{(1)}\) for \(j \in E^{(1)}\) and \(\alpha_i(x'_e) \geq \alpha^{(2)}\forall i \notin E^{(1)}\) with at least one strict inequality. But, if \(\min_{i \notin E^{(1)}} \{\alpha_i(x'_e)\} > \alpha^{(2)}\) then Theorem 2 implies that \(\exists x'' \in C \text{ s.t. } \alpha_i(x''_e) = \alpha_k(x''_e) \equiv \alpha_0 > \alpha^{(2)}, \forall i, k \notin E^{(1)}\) and \(\alpha_j(x''_e) = \alpha^{(1)}\) for \(j \in E^{(1)}\). This contradicts our hypothesis. So, \(\alpha^{(2)}\) is the maximum 2\(^{nd}\) minimum lifetime subject to the minimum lifetime being maximum. \(\blacksquare\)
Next, we create a new set, $E^{(2)}$, that contains the index of one sensor whose lifetime can not be increased without decreasing the second minimum lifetime (subject to the minimum lifetime being maximum). Using similar arguments as in Lemma 2, we can show that this set is well defined.

At the end of this second iteration, we have a solution that maximizes the second minimum lifetime such that the minimum lifetime is maximum. We can iterate the same steps until we find the first $x^{(n)}_e \in \mathcal{P}$, where $n$ is the iteration index.

3.5 Experimental results

In this section we present some experimental results that illustrate the practicality and the efficacy of our theoretical analysis. The theoretical model used in this chapter is general, so we can easily adopt any convenient sensor network model. Here, we assume that $N$ sensors are deployed randomly in a $100m \times 100m$ square area. The location of the base station is also random within this square. We only account for communication costs and neglect all processing costs. We use the following radio model for data transmission, whereby the cost for transmitting one bit along a path of length $d_{i,j}$ is given by, $t_{i,j} = 50 \times 10^{-9} + 100 \times 10^{-12}d_{i,j}^2$ (J/bit), while the cost of receiving a bit is given by, $r_{i,j} = 50 \times 10^{-9}$ (J/bit), where $d_{i,j}$ is the distance (in meters) between node $i$ and node $j$ [18].
Consider the 4-sensor network deployment illustrated in Figure 3.3. The circles on this grid represent the sensor nodes while the rectangular mark represents the base-station. Assume all sensors possess the same initial energy of $B_i = 1J$. The bracketed number next to each sensor represents the initial load $x_{i,i}$ (in bits). The distances are to scale.

![Figure 3.3: Example of a 4-sensor network deployment showing the optimal solution obtained using our method. The algorithm converged in one iteration.](image)

We apply our algorithm, and solve for the vector $\mathbf{x}$ that maximizes all conditional lifetimes. The resulting optimal values of $\{x_{i,j}\}$ are shown on the links between sensors $i$ and $j$. Using Equation (3.4) we get the following sensor lifetimes: $\alpha_1 = 64.74, \alpha_2 = 64.74, \alpha_3 = 64.74$ and, $\alpha_4 = 64.74$. In this case, the final result was obtained with just one iteration.
Consider another 4-sensor network deployment shown in Figure 3.4. Again, we solve here for the \( \{x_{i,j}\} \) that maximizes all conditional lifetimes. Applying our algorithm in this case yields the following sensor lifetimes: \( \alpha_1 = 95.33 \), \( \alpha_2 = 95.33 \), \( \alpha_3 = 95.33 \) and, \( \alpha_4 = 505.25 \). In this case, we obtained the final result with 4 iterations of our algorithm. A modified algorithm that adds multiple sensors to \( E^{(m)} \) could achieve this result in two iterations.

Figure 3.4: Another example of a 4-sensor network deployment. The solution was obtained in four iterations.

An interesting question is the following: how often does our algorithm perform better than methods that only aim at maximizing the minimum lifetime? In order to answer this question, we opt to compare with an algorithm from the literature. Chang and Tassiulas [10] were the first to address the problem of maximizing the minimum
lifetime. In that work, the authors treat this problem as a linear programming problem.

To obtain results for comparison, a given number of sensors are randomly deployed as outlined previously. For each deployment instance, we apply our algorithm as well as the algorithm of [10]. We then compare the values of the $n^{th}$ minimum lifetime as obtained using both algorithms, $\forall n = 1, \ldots, N$. Specifically, we compute the percentage gain in the $n^{th}$ minimum lifetime defined as,

$$g_{n:N} = 100 \times \frac{\min_i \{\alpha_i\} - \min_i \{\alpha_i'\}}{\min_i \{\alpha_i'\}},$$

(3.15)

where $\alpha_i$ and $\alpha_i'$ are the lifetimes for sensor $i$ obtained using our algorithm and the algorithm of [10], respectively. Finally, we plot the estimated probability of gain for all $g_{n:N}$.

We note the following about the quantity $g_{n:N}$. First, we expect $g_{1:N}$ to be always non-negative. This follows from Section 3.4, where our algorithm was shown to maximize the minimum lifetime. Second, recall that we are not interested in maximizing the “unconditional” $n^{th}$ minimum lifetime, for $n \geq 1$. Instead, our objective is to maximize the $n^{th}$ “conditional” minimum lifetime (such that the $(n - 1)^{th}$ minimum lifetime is maximum, etc.) So, it is possible that there could be instances in which $g_{n:N} < 0$, for $n \geq 2$. But, this can only happen if $g_{m:N} > 0$ for some $m < n$.

Figures 3.5a-d show the estimated probability of a given gain $g_{n:4}$ for $n = 1, 2, 3$ and 4, respectively.\(^1\) For example, Figure 3.5b shows that the probability of obtaining
a gain in the second minimum lifetime in the interval $[10\%-20\%]$ is about 0.05. On average, the expected gain in the second minimum lifetime due to our algorithm is around 7.9%. Similarly, we can see from Figure 3.5c that the probability of obtaining a gain in the third minimum lifetime in the interval $[10\%-20\%]$ lifetime is about 0.07. The expected gain in the third minimum lifetime, denoted $\bar{g}_{3:4}$, is 10.75%.

It is interesting to note the result in Figure 3.5a. As we can see, the gain of our 1st minimum lifetime over that obtained using our implementation of the LP program of [10] is not always zero. In fact, applying that LP program to the example of Figure 3.4 results in the following lifetimes: $\alpha'_1 = 85.12$, $\alpha'_2 = 85.12$, $\alpha'_3 = 85.12$, and $\alpha'_4 = 162.44$. This results in a minimum lifetime 11% smaller than the one obtained using our algorithm. The simulations here also show that there are instances in which the gain is as high as 46.7%. On average, $\bar{g}_{1:4} = 0.5\%$. These observations imply that our algorithm is able to attain higher 1st minimum lifetimes than the LP algorithm described in [10]. The reason behind this is the assumption mentioned in that paper. The linear programming treatment in [10] requires an assumption that the lifetime of the network is independent from the variables $x_{i,j}$.

Table 3.1 summarizes the aforementioned results for $N = 4$. We note in particular the high value of average percentage gain for the fourth minimum lifetime. The experiments are repeated for differing numbers of sensors with Table 3.1 summarizing

\footnote{Note that we don't show here the negative values of $g_{n,N}$, to allow plotting on a log scale. However, we include those negative values in our computations.}
the results. Finally, we report the probability of achieving a gain, $P_{g_N}$, defined as the rate at which our algorithm succeeds in finding a larger value for any of the $n^{th}$ minimum lifetimes. We plot $P_{g_N}$ versus the number of sensors $N$ in Figure 3.6. For example, note that the probability of obtaining a gain in any of the lifetimes for 7 sensors is around 0.57.

<table>
<thead>
<tr>
<th>Gain</th>
<th>Average value</th>
<th>Maximum observed value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_{1:3}$</td>
<td>0.82%</td>
<td>63.26%</td>
</tr>
<tr>
<td>$g_{2:3}$</td>
<td>9.11%</td>
<td>$1.7 \times 10^3%$</td>
</tr>
<tr>
<td>$g_{3:3}$</td>
<td>122.23%</td>
<td>$1.4 \times 10^6%$</td>
</tr>
<tr>
<td>$g_{1:4}$</td>
<td>0.5%</td>
<td>41.7%</td>
</tr>
<tr>
<td>$g_{2:4}$</td>
<td>7.9%</td>
<td>677%</td>
</tr>
<tr>
<td>$g_{3:4}$</td>
<td>10.75%</td>
<td>$8.83 \times 10^3%$</td>
</tr>
<tr>
<td>$g_{4:4}$</td>
<td>260%</td>
<td>$4.3 \times 10^6%$</td>
</tr>
<tr>
<td>$g_{1:5}$</td>
<td>0.25%</td>
<td>37.9%</td>
</tr>
<tr>
<td>$g_{2:5}$</td>
<td>6.62%</td>
<td>562.8%</td>
</tr>
<tr>
<td>$g_{3:5}$</td>
<td>8.78%</td>
<td>811.75%</td>
</tr>
<tr>
<td>$g_{4:5}$</td>
<td>9.02%</td>
<td>$3.77 \times 10^3%$</td>
</tr>
<tr>
<td>$g_{5:5}$</td>
<td>222.61%</td>
<td>$2.53 \times 10^6%$</td>
</tr>
<tr>
<td>$g_{1:6}$</td>
<td>0.15%</td>
<td>30%</td>
</tr>
<tr>
<td>$g_{2:6}$</td>
<td>5.2%</td>
<td>385.5%</td>
</tr>
<tr>
<td>$g_{3:6}$</td>
<td>7.3%</td>
<td>457.46%</td>
</tr>
<tr>
<td>$g_{4:6}$</td>
<td>8.2%</td>
<td>$1.1 \times 10^3%$</td>
</tr>
<tr>
<td>$g_{5:6}$</td>
<td>9.48%</td>
<td>$3.48 \times 10^3%$</td>
</tr>
<tr>
<td>$g_{6:6}$</td>
<td>292.51%</td>
<td>$5.17 \times 10^6%$</td>
</tr>
</tbody>
</table>

3.6 Concluding remarks

In this chapter, we developed a novel theory and a corresponding algorithm for maximizing the lifetime of unicast sensor networks via multi-hop routing. The sensors
considered here did not have any correlation between their measurements. When considering imaging sensors, however, we demonstrated in Chapter 2 that we could also increase the lifetime via collaborative compression. We then ask the following obvious question: how can we utilize multi-hop routing (inter-sensor communication) in order to maximize the lifetime of a network of imagers? Chapter 4 presents a solution to this problem.
Figure 3.5: Probability of the percentage gain $g_{n:N}$ shown on log-log scale for (a) $n=1$, (b) $n=2$, (c) $n=3$, and (d) $n=4$. 
Figure 3.6: Probability of attaining a positive gain in any of the minimum lifetimes as a function of number of sensors.
CHAPTER 4
MULTI-HOP TRANSMISSION WITH COLLABORATIVE IMAGE COMPRESSION

In this chapter, we reconsider our network of imaging sensors and aim at increasing the lifetime of the network by exploiting intra- and inter-sensor correlation. We propose a novel collaborative compression algorithm to be used in conjunction with a cooperative multi-hop routing strategy in order to maximize the lifetime of the network [12]. Section 4.1 presents an overview of the general sensor network model employed. Section 4.2 formulates the problem and develops the solution along with the proposed algorithm. The simulation results are presented in Section 4.3.

4.1 Physical model

In this section, we describe a model of the imaging sensor network under consideration. The imaging model illustrated in 4.1 is identical to the one introduced in Section 2.1. \( N \) imaging sensors are randomly deployed over a scene of interest some distance away from a central base station. For a given random deployment, the task of the network consists of transmitting the union of the resulting Sensors' Field of View (\( \text{SFOV} \equiv \bigcup_{i=1}^{N} \text{SFOV}_i \)) to the base station subject to power constraints.
4.2 Problem formulation and solution

In this chapter, the problem is to maximize the number of transmission events by:
(a) reducing the overall data load through exploiting the inherent inter- and intra-
sensor redundancy via collaboration and, (b) choosing a transmission strategy that
routes the data in an energy-efficient manner.

In Chapter 3, a general model for solving a similar problem was developed. However,
we treated the case of general sensor loads without acknowledging the correlation
present both within the load of sensor $i$ ($x_{i,i}$) and between the loads of different
sensors. In this section we present a method that exploits the inherent redundancy
in a network of imaging sensors in order to further increase the lifetime.
4.2.1 Exploiting spatial redundancy through collaboration

Figure 4.2 illustrates the deployment of 3 imaging sensors according to our simplified model described in Section 4.1. Each SFOV\(_i\) covers a certain area of the scene.

![Figure 4.2: The overall network field of view can be divided into largest non-overlapping regions.](image)

The first step towards exploiting the spatial redundancy within the network is to divide the SFOV into largest non-overlapping regions. Or, more formally, we decompose the overall field of view into a set of areas that are mutually exclusive and collectively exhaustive. This is illustrated in Figure 4.2 where the fields of view are divided into a set of 7 non-overlapping regions, \(R_1, R_2, \ldots, R_7\). Our goal consists of transmitting these regions using energy-efficient transmission strategies (e.g., maximum-lifetime routing). In analogy with our previous model of Chapter 3, these regions correspond to the total load (or bits) that need to be communicated to the
base station. One major difference here is that there exist bits which are now common to more than one sensor. Solving this problem using our formalism of Chapter 3 involves re-writing the constraint equations and objective functions to incorporate the redundant nature of the data load (i.e., overlapping SFOV<sub>i</sub>) across the sensors. Specifically, the new formulae should reflect the fact that: 1) bits (i.e., pixels) shared amongst a set of sensors only need to be communicated by one of those sensors and, 2) the union of the bits (rather than their sum) must be transmitted to the base station.

Although such an approach is possible, it is rather complicated. Instead, we develop a simpler and more intuitive approach. Define first two classes of imagers: real and virtual. The load of a real imager corresponds to a region seen only by one sensor. A virtual imager has a load that corresponds to a region shared by more than one sensor. For example, in Figure 4.2, regions R<sub>1</sub>, R<sub>2</sub> and R<sub>3</sub> would define 3 different real imagers, S<sub>1</sub>, S<sub>2</sub> and S<sub>3</sub>, respectively. The same figure also shows that there are 4 virtual imagers (S<sub>4</sub>-S<sub>7</sub>) corresponding to 4 different shared regions (labeled R<sub>4</sub> to R<sub>7</sub>). A virtual imager S<sub>j</sub> is said to be associated with real imager S<sub>i</sub> if region R<sub>j</sub> is “seen by” real imager S<sub>i</sub>. For example, virtual imager S<sub>4</sub> is associated with real imagers S<sub>1</sub> and S<sub>2</sub>. Define V<sub>i</sub> as the set of virtual imagers associated with real imager S<sub>i</sub>. The idea is then that virtual imagers can transmit to their associated real imagers free of charge.
Using this classification, we can utilize our formalism of Chapter 3 to represent this problem using the graph of Figure 4.3. We illustrate in this graph the example of Figure 4.2: nodes $S_1$, $S_2$ and $S_3$ represent the real imagers while the virtual imagers correspond to (shaded) nodes $S_4$, $S_5$, $S_6$ and $S_7$. Similar to the graph of Figure 3.1, we denote by $B_i \{i = 1, 2, ..., N\}$ the initial battery (in Joules) of real imager $i$ and by $x_{j,j}$ the initial load (in bits) of imager $j$, $\{j = 1, 2, ..., N + V\}$, where $N$ and $V$ are the total number of real and virtual imagers, respectively. The branches of the graph, labeled $x_{j,k}$, denote the number of bits that are routed from imager $j$ to imager $k$, $\{j, k = 1, 2, ..., N + V\}, j \neq k$. $x_{j,0}$ indicates the number of bits that will eventually be transmitted from imager $j$ directly to the base station. The initial imager loads $\{x_{j,j}\}$ correspond to the number of bits required to represent the pixels of region $R_j$ at a given RMSE. Again, if a region is shared by more than one sensor, then its load is assigned to a unique virtual imager; otherwise, it is assigned to a unique real imager. We clarify here that only real imagers are physically deployed in the network. The virtual imagers are just a convenient artifice to help solve the problem. Accordingly, the locations of real imagers in the network correspond to the physical locations of the sensor nodes. As we shall see next, the physical locations of the virtual sensors do not matter.

We can differentiate here between 3 types of links: physical (shown using dashed lines), virtual (shown with dotted lines) and forbidden (indicated by the absence of a connector line). The physical links are dedicated to communications involving only
Figure 4.3: The problem for the network of imagers in Figure 4.2 can be represented using this graph.

real imagers and/or the base station. The real imagers are the nodes responsible for transmitting the union of all loads to the base station. Virtual imagers can not transmit over a real physical link. Instead, a virtual imager can only transmit data via a virtual link to a real imager with which it is associated. The cost of such a transmission is zero Joules. All other links are forbidden. Specifically, a virtual imager can not transmit data directly to the base station, to another virtual imager, nor to a real imager with which it is not associated.

We choose to incorporate such “link constraints” into the following equality constraints. First, the conservation of load constraint (c.f. Equation 3.5) at real imager $i$ could be modified as follows:

$$
\sum_{j=1}^{N} x_{j,i} + \sum_{l \in V_i} x_{l,i} = \sum_{j=0, j \neq i}^{N} x_{i,j}, \forall \ i = 1, 2, \ldots, N.
$$

(4.1)
Next, we need to guarantee that the sum of all the load transmitted from virtual imager \( l \) to all sensors with which it is associated equals the initial load. Namely,

\[
\sum_{i \in V, l \in V} x_{l,i} = x_{l,l}, \forall l = N + 1, N + 2, \ldots, N + V.
\] (4.2)

Note that, by taking the sum of Equation (4.1) over all \( i \), we can show using Equation (4.2) that,

\[
\sum_{i=1}^{N} x_{i,i} + \sum_{l=N+1}^{N+N} x_{l,l} = \sum_{i=1}^{N} x_{i,0},
\] (4.3)

or, in other words, the sum of all loads transmitted from the real imagers to the base station equals the sum of all initial loads, as desired. So, Equations (4.1) and (4.2) completely characterize the constraint set.

**Definition:** Define \( C' \) to be the constraint set. That is, \( x \in C' \) if the elements of \( x \) are non-negative and satisfy Equations (4.1) and (4.2).

This structuring of the graph of Figure 4.3 allows us to simultaneously exploit the intra-sensor correlation and solve the problem of energy-efficient routing using tools developed in Chapter 3. The maximum-minimum lifetime routing formulation represented by Equation (3.6) could be utilized to solve the problem. The only difference here is the constraint set \( C' \).

We state the problem now as,

\[
x^* = \arg\max_{x \in C'} \left[ \min_{i} \{ \alpha_i (x) \} \right].
\] (4.4)

This linear-programming problem can be solved using methods referenced in the previous chapter, including our suggested optimal routing algorithm.
4.2.2 Exploiting intra-SFOV correlation structure

As discussed in Section 2.4, an important part of maximizing the lifetime of a sensor network is the ability to exploit the inter- and intra-sensor redundancy. In the previous section, we have developed a collaborative multi-hop method for exploiting inter-sensor correlation. It is of interest to exploit the intra-sensor redundancy in order to further increase the lifetime of the network. Therefore, each sensor should be equipped with an encoding (compression) engine that computes the corresponding compressed costs of the regions associated with that sensor. In this work we have used JPEG2000 [64] as the encoding engine. Results for other compression systems will be similar.

4.3 Simulation Results

This section presents experimental results\(^1\) illustrating the impact of our algorithm on the lifetime of imaging sensor networks.

As a first example, we apply our algorithm on the 3-sensor network deployment of Figure 4.2. In this specific configuration, the sensor density \( \rho = 0.33 \) and the scene coverage \( \eta = 0.2 \). Each sensor here is assumed to have an energy of \( 0.5J \). The deployment can be represented using Figure 4.4. The clear circles on this grid represent the real imagers, the coordinates of which correspond to the physical locations

---

\(^1\)The implementation issues here are identical to the ones discussed in Section 2.5.1.
of the sensors. The shaded circles represent the virtual imagers while the square mark represents the base-station. The bracketed number next to each sensor node represents the initial load \((x_{i,i})\) (in bits). The distances (between the real imagers and the base-station) are to scale.

Figure 4.4: The optimal transmission strategy for one transmission event from the network deployment illustrated by Figure 4.2.

We first look at the transmit energy remaining in each sensor after each transmission instance. To do this, we present the sensors with a new scene before each transmission event, while maintaining a fixed sensor network geometry. The \(384 \times 384\) scene is randomly chosen from a large image map. The image map from which our scenes are “sampled” in this example is a \(4450 \times 4450\) grayscale image representing a coastline landscape (low-resolution version shown in Figure 2.5).

We show a solution obtained for one scene in Figure 4.4. The resulting optimal values of \(x_{i,j}\) are shown on each link between sensors \(i\) and \(j\). As can be seen, sensors
take advantage of the free communication cost available via their associated virtual imagers (shaded circles) to transmit their load to more “capable” sensors.

This procedure is then applied to each acquired scene until one sensor runs out of battery. Figure 4.5 shows the energy remaining in each sensor as a function of transmission instance. The performance of our collaborative multi-hop transmission (CMT) algorithm is shown using solid lines. Eventually, as can be seen, all sensors run out of power at the same time, which is what we expect. For this case, the corresponding lifetime of the network is \( \alpha = 36 \) transmissions. We also include on the same plot the remaining energy in each sensor had we used a non-optimized single-hop approach, whereby each sensor \( i \) independently compresses everything in SFOV\(_i\) and sends it directly to the base station. Hereafter, we shall refer to such a method as the baseline approach. The lifetime of the network for that case is \( \beta_b = 19 \) transmissions. Thus the proposed approach offers a 90% gain over the baseline approach. This gain is due to both collaboration (the exploitation of overlapping SFOV\(_i\)) and the optimal multi-hop routing of the resulting data. The compression (source coding) algorithm is the same in both cases.

The results presented in Figure 4.5 are for a sensor density of \( \rho = 0.33 \) with a specific scene coverage of \( \eta = 0.2 \). Next, we fix the number of sensors (i.e., constant \( \rho \)) and vary the scene coverage \( \eta \). This is done for the purpose of observing how the lifetime of the network changes with coverage at constant density. To do so, we randomly deploy a constant number of sensors (with equal initial battery of 1J)
Figure 4.5: Remaining energy in each sensor vs. transmission instance for the example deployment of Figure 4.4. Note that the curves for sensors 1 and 2 overlap in the baseline approach.

over a scene and, for each occurring configuration, we employ our algorithm at every transmission instance. We then average the resulting lifetime $\alpha$ over all deployments that yield the same coverage $\eta$. The result is shown in Figure 4.6. The horizontal and vertical axes represent the coverage $\eta$ and lifetime $\alpha$, respectively. The solid curve is the lifetime of the network using the CMT algorithm. The dashed curve represents the lifetime obtained using the baseline approach. In order to individually attribute the gain in lifetime due to the routing and collaboration components, we respectively compare the performance of our CMT algorithm with the following 2 algorithms: 1 the collaborative single-hop transmission (CST) algorithm described in [15] (shown hereafter using the dash-dotted curve) and 2 a multi-hop transmission (MT) algorithm that does not utilize any collaboration between sensors (shown hereafter using
the dash-crossed curve). The MT algorithm simply transmits the compressed SFOV\(_i\) to the base station using the route that maximizes the lifetime of the network.

Figure 4.6: Lifetime of the network as a function of scene coverage for 3 sensors.

We note the following from Figure 4.6. First, we point out the general trend of our proposed CMT algorithm: as coverage decreases, the lifetime of the network increases. At minimum coverage \((\eta = 0.11)\), all SFOV\(_i\) overlap and we thus are able to fully exploit the spatial redundancy and hence, the lifetime is at a maximum. On the other hand, at maximum coverage \((\eta = 0.33)\), the SFOV\(_i\) do not overlap at all, and thus multi-hop communication is exclusively relied on in order to balance the load across the network. Second, we note the performance of the baseline approach. In this case, each sensor is transmitting, on average, the same number of bits for every coverage. The baseline lifetime, \(\beta_b\), is slightly decreasing with \(\eta\). This is a result of
the random deployment: it is more likely to obtain a sensor with large distance to the base-station when coverage is high as compared to when coverage is low. So, because the transmission capability of a sensor is also a function of its distance to the base-station, a network with large $\eta$ is more likely to run out of battery before a network with small $\eta$ when employing the baseline method. Third, note the performance of the CST algorithm. We see that, at minimum coverage, the CST and CMT methods have identical performance. This is expected. At low coverage, both algorithms are able to exploit correlation without using multi-hop routing. As the coverage increases, however, the overlaps between the $\text{SFOV}_i$, $\forall i$, decrease and thus the CST algorithm is no longer capable of balancing the load over the network. At maximum coverage (no overlap), the performance of the CST algorithm is identical to the baseline approach. The CMT algorithm in this case outperforms the single-hop methods by a factor of 1.52. Finally, we note the constant performance of the MT algorithm. This is expected as well. The MT algorithm transmits, on average, the same amount of data for any given value of coverage. At high coverage, there is little overlap between the $\text{SFOV}_i$, $\forall i$ and thus both the MT and CMT algorithms transmit about the same amount of data using optimal multi-hop routing. At maximum coverage, the MT and CMT algorithms are identical.

It is of interest to show the gain of the CMT algorithm over the other method as a function of coverage. We show this in Figure 4.7. The solid line represents the gain of CMT over the baseline approach, $\gamma_b \equiv \frac{\alpha}{\beta_b}$. The dashed line represents the
gain of CMT over CST, $\gamma_{CST} \equiv \frac{\alpha}{\beta_{CST}}$, where $\beta_{CST}$ denotes the lifetime of the CST algorithm. The dash-crossed line represents the gain of CMT over MT, $\gamma_{MT} \equiv \frac{\alpha}{\beta_{MT}}$, where $\beta_{MT}$ denotes the lifetime of the MT algorithm. It can be seen that at low coverage, the performance of the CST algorithm is close to that of CMT, while the latter outperforms the baseline method by a factor as high as 2.6. As coverage increases, the gain of CMT over both single-hop methods converges to the same value of 1.52. On the other hand, at high coverage, the performance of the MT algorithm is close to that of CMT, while the latter outperforms the former method at low coverage by a factor as high as 2.6. It is evident from these curves that our CMT algorithm combines the gains due to multi-hop transmission (at high coverage) and inter-sensor collaboration (at low coverage).

Figure 4.7: Gain of our CMT algorithm over the CST, MT and baseline approaches as a function of scene coverage for 3 sensors.
The results shown in Figures 4.6 and 4.7 are for a fixed number of sensors, or equivalently a fixed sensor density $\rho$. However, it is also of interest to investigate how the performance of all these methods changes as the sensor density $\rho$ varies. To this end, we repeat the experiment for different numbers of sensors. The results for 4, 5, and 6 sensors ($\rho = 0.44, 0.55, 0.66$) are shown in Figures 4.8a-c, respectively. We make the following remarks from these plots. First, we note that the range of coverage $\eta$ that occurred experimentally is different for different $\rho$. For example, no coverage below 0.15 occurred with 4 sensors. In principle, the minimum expected coverage is 0.11 but, the probability of occurrence of this event (4 sensors fully overlapping) is negligible ($(384)^{-6}$). Second, we observe that for the same $\eta$, the gain increases with $\rho$. To see this effect more clearly, we average the gain over all coverage $\eta$ for a fixed sensor density $\rho$, and plot the result versus $\rho$. This is shown in Figure 4.9. The horizontal and vertical axes represent the sensor density and the averaged gain in lifetime, respectively. The solid curve is the average gain obtained using our CMT algorithm over the baseline approach, while the dashed line and dash-crossed lines represent the average gain of CMT over CST and MT, respectively.

From the data in Figure 4.9, we note the general behavior of the gain curves: as sensor density increases, the gain over all methods increases. This is expected. The denser the network is, the less energy it takes to “hop” from one sensor to the next and, in turn, the cheaper it gets to balance the load over the network via optimal routing. Also, a dense network has more overlaps and thus, more potential gain from
collaboration. For example, for a density of 0.66, the average gain offered by our proposed CMT method is a factor 2.47 over the baseline approach, a factor of 1.61 over the CST algorithm and, a factor of 1.52 over the MT algorithm. It is interesting to note that, in this figure, the overall average gain \( \langle \gamma_k \rangle \approx \langle \gamma_{CST} \rangle \times \langle \gamma_{MT} \rangle \). This is shown using the line marked with triangles.

4.3.1 Alternate definitions of network lifetime

We mentioned in Section 4.2 that the proposed method for exploiting the inter- and intra-SFOV correlation structure could be utilized along with other routing strategies. The strategy that we have adopted so far maximizes the minimum lifetime in the network. It is interesting, however, to consider transmission strategies that take into account sensor lifetimes after the death of one or more sensors. Specifically, it is interesting to maximize the “\( n^{th} \) conditional lifetime,” as defined in Chapter 3. Formally, for any integer \( n, 1 \leq n \leq N \), we want to find:

\[
\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{C}} \left[ \min_{i}^{n} \{ \alpha_i(\mathbf{x}) \} \right] \quad s.t.
\]

\[
\min_{i}^{n-1} \{ \alpha_i(\mathbf{x}^*) \} \quad \text{is maximum} \quad s.t.
\]

\[
\cdots 
\]

\[
\min_{i}^{k} \{ \alpha_i(\mathbf{x}^*) \} \quad \text{is maximum,}
\]

where \( \min_{i}^{k} \{ \alpha_i(\mathbf{x}) \} \) denotes the \( k^{th} \) minimum lifetime amongst \( \{ \alpha_i(\mathbf{x}) \} \), \( \forall i = 1, \ldots, N \).

We can easily use our provably optimal algorithm proposed in Chapter 3 with our formulation of Section 4.2.1 and repeat the same experimental steps outlined above.
Figures 4.10a-d plot both gains $\gamma_b$ and $\gamma_{CST}$ at $\rho = 0.44$ versus scene coverage for the 1st, 2nd, 3rd and 4th minimum lifetimes, respectively. The routing algorithm here was set to maximize the 4th conditional lifetime. We note that, as expected, the gain curves in Figure 4.10a-d have the same trend as those obtained in Section 4.3.
Figure 4.8: Gain of the CMT algorithm over the CST, MT and baseline approaches as a function of scene coverage for (a) 4 (b) 5 and (c) 6 sensors.
Figure 4.9: Gain of the CMT algorithm over the CST, MT and baseline approaches is averaged over all $\eta$ for each sensor density, $\rho$, and then plotted here versus $\rho$. 
Figure 4.10: Gain of our CMT algorithm over the CST and baseline methods for $\rho = 0.44$ when CMT utilizes a transmission strategy that maximizes the $4^{th}$ conditional lifetime. The results show the gains achieved in the (a) first, (b) second, (c) third and (d) fourth minimum lifetimes.
CHAPTER 5

REAL-LIFE PERFORMANCE

Throughout Chapters 2 and 4, we have assumed a simple imaging model. In this chapter, we first describe an experimental procedure that applies our proposed algorithm directly. This procedure assumes the simple 2D imaging model described in Chapter 2. The result illustrates that the simple optical system model could be utilized in some applications. Next, we develop a more realistic optical system model that takes into account a more heterogeneous network of cameras by allowing for varying resolution, camera parameters, point-spread function, detector size, etc. We show that our CMT algorithm presented in Chapter 4 can be extended to successfully operate in such diverse domain. We propose new object-domain quality metrics and show that our slightly modified CMT algorithm is still able to balance lifetime and fidelity according to expectations.

5.1 Direct application of CMT algorithm to real images

Note that there are several assumptions made in the imaging model of Section 2.1 that need to be relaxed when considering real world scenarios. In particular, the 3D nature of real scenes will add some additional level of complexity to various elements of this problem (e.g., the task of classifying the largest regions). However, in many
scenarios, the 2D assumptions are well justified. As we will see next, in the case
where the object resides in the far-field of the sensors, a direct application of our
proposed CMT algorithm yields good results.

In fact, we were able to successfully implement an experimental apparatus based
on our proposed CMT/CST algorithms. We deployed a number of cameras attached
to battery powered iPAQ PocketPCs [1]. We then calibrated the cameras to image
a distant scene (about 6m away). According to our model, the cameras need to
operate at the same object resolution (magnification) so, the iPAQs were placed at
the same distance from the object. Also the cameras’ optical axes were oriented
parallel to one another. Each camera captures a 320 × 240 grayscale image. These
images are to be wirelessly transmitted by the iPAQs to the base-station (a remote
terminal) according to our CMT algorithm. The iPAQs represent the sensor nodes
whose battery lifetime we want to maximize. All iPAQs originally start with an equal
battery capacity of 1Ah. The iPAQs use re-chargeable Lithium-Ion batteries with a
nominal voltage of 3.7V. We present here the results obtained with a network of 3
iPAQs. Figures 5.1a, 5.1b and 5.1c show the original SFOVs at one time instant as
seen by iPAQ cameras 1, 2 and 3, respectively. Using the baseline approach, whereby
each iPAQ communicates its image (SFOV) directly to the base station, the measured
lifetime of the network is 247 transmissions. Next, we apply our algorithm. Each
sensor in this case has to transmit portions of its SFOV via routes as computed by
the base station. An example of such regions captured at one time instant is shown in
Figures 5.2a, 5.2b and 5.2c. Once at the base station, we can combine these images according to our 2D assumptions. The result is shown in Figure 5.3. This whole process is repeated until the first sensor (iPAQ) runs out of battery power. Using our optimized approach, the measured lifetime of the network is 701 transmissions. This is a factor of 184% improvement over the baseline approach. Note the artifacts in the reconstructed object in Figure 5.3. This is due to factors such as the changing level of illumination across the imagers and mis-registration. Recall that the task here is not to reconstruct a “pretty picture” of the object. Instead, we are only interested in transmitting image measurements (pixels) that constitute the scene of interest. Reconstruction/fusion algorithms could be utilized at the base station, along with finer registration methods, in order to achieve better quality pictures.

We note the following important remark. For our algorithm to work, there should be some level of communication between the sensors and the base station. The information required from each sensor is: (i) its initial battery power (sent only once) and, (ii) the compressed cost for each image. However, as we have seen in the example above, the cost of sending this information is negligible compared to that of sending the actual measurements.

5.2 A realistic optics model

The previous section showed that direct application of our algorithm to a simple experimental apparatus yields good results. However, more practical applications
Figure 5.1: Original SFOV captured at one time instant by sensor (a) 1, (b) 2, and (c) 3.
Figure 5.2: SFOV portions to be sent by sensor (a) 1, (b) 2, and (c) 3.
Figure 5.3: The images sent from each sensor can be combined at the base station.

would allow the sensors/imagers to be deployed at different distances from the object. Also, the optical axes of these imagers are not usually guaranteed to be aligned. Thus, we present here a more realistic forward imaging model. The model presented next will allow for imagers to possess arbitrarily different lenses (or, point-spread functions (PSFs)), positions, tilts and image/detector size. This will allow us to model a more “heterogeneous” network of imagers: different imagers measuring images at different qualities. The forward model will also account for the optical blur introduced by the lens aperture and the pixel blur introduced by the detector array.

Before presenting the imaging model, we first discuss some of the object properties that we assume hereafter. We maintain that, for our purposes, the object is 2D. This is a reasonable assumption for the intended applications. Recall that the imagers are assumed to be randomly dropped off from a remote location above a scene of interest. In that context, it is reasonable to assume that there is a minimum separation
distance, $Z_{\text{min}}$, to be maintained between the imagers and the object and that, at that distance, the object could be modeled as a 2D field.

Another implication of this minimum separation $Z_{\text{min}}$ is that, for a given imager (lens PSF, detector pixel size), a continuous object can be modeled as an array of discrete pixels $(P_x \times P_y)$ with a spatial resolution $\Delta x \times \Delta y$. Given prior knowledge about the imagers as well as the separation $Z_{\text{min}}$, we can choose $\Delta x$ and $\Delta y$ in a manner that guarantees no imager pixel can ever “over-sample” the object pixels. We say that an imager “critically samples” the object when any pixel in the imager maps to exactly one object pixel. A reasonable object geometry model is shown in Figure 5.4.

Figure 5.4: Geometry of the real model.

We discuss next the imaging operation that we model in this chapter. As shown in Figure 5.4, imagers are randomly distributed over a scene of interest. The imagers
are allowed to occupy any position in the $x - y$ plane, while the height (in the $z$ plane) of any given imager is restricted within the range $Z_{\text{min}} < Z < Z_{\text{max}}$. Moreover, the imagers are randomly tilted in the $x - z$ and $y - z$ planes. The random tilts are chosen such that any imager always "sees" regions entirely inscribed within the object. Imager $i$ is assumed to be composed of the following: a simple lens of focal length $f$ and diameter $D$ and, $L_x \times L_y$ detector elements each of resolution $\Delta x'' \times \Delta y''$. Obviously, the tilt of the imager renders the system highly shift-variant: different object regions are imaged via different spatially variant PSFs. However, we assume that the PSF variation within one object pixel is negligible. Thus, the object pixels can be divided into $P_x \times P_y$ regions over which the system is shift-invariant.

Thus, the image-plane intensity distribution at coordinate $x''$ can be described as the superposition of outputs from several Linear Shift Invariant (LSI) systems. We can show that the measurement at imager pixel $l$ can be written as\(^1\):

$$m_l = \sum_{p=1}^{P_x} g_p C(l, p),$$

(5.1)

where, $g_p$ is the object intensity value at pixel position $p$, and $C(l, p)$ is the imaging matrix given by\(^2\):

$$C(l, p) = \frac{1}{\Delta x} \int dx'' \int dx h_p(x'' - x) \text{rect}\left(\frac{x - x_p}{\Delta x}\right) \text{rect}\left(\frac{x'' - x''_l}{\Delta x''}\right).$$

(5.2)

---

\(^1\)Hereafter we only discuss the 2D geometry (along the $x - z$ dimension). Although taken into consideration in the implementation, the extension to the more complicated 3D geometry has been omitted here for clarity.

\(^2\)We adopt the following notation throughout this chapter: unprimed, primed and doubly-primed symbols denote object space, pupil space and image space quantities, respectively.
In Equation (5.2), $h_p$ is the local PSF of the imager near object region $p$. Implementing the double-convolution in Equation (5.2) on a computer can be time consuming. Moreover, sampling the rect functions in space domain would always induce aliasing. Using some Fourier Transform manipulations, we were able to re-write Equation (5.2) as follows:

$$C(l, p) = \Delta x'' F^{-1}\{H_p(-\sigma)sinc(\Delta x\sigma)sinc(\Delta x'\sigma)\}_{x=x_p-x''_l}$$  \hspace{1cm} (5.3)

where $H_p(\sigma)$ is the Optical Transfer Function (OTF) for object pixel region $p$ and $F^{-1}$ is the inverse Fourier Transform operator. Thus, we just transformed the 2D integral of Equation (5.2) into a much simpler 1D inverse FFT operation. Moreover, sampling the sinc functions here can be achieved without aliasing. Sampling the bandlimited OTF, however, induces aliasing. Nevertheless, we could truncate the PSF (usually a sinc$^2$), $h_p(x) = F^{-1}\{H_p(\sigma)\}$, at a width high enough so as to minimize the effects of aliasing.

So, in the absence of noise, each imager $i$, $i = 1, 2, \ldots, N$, makes the following measurement:

$$m_i = C_i g,$$  \hspace{1cm} (5.4)

where $m_i$ and $g$ are the measurement and object vectors of length $L$ and $P$, respectively, and $C_i$ is the imaging matrix of imager $i$. We should note that the dynamic range of both object and image is between 0 and 255 (8-bit representation).
The model presented through Equations (5.1)-(5.4) treats 1D objects. We have extended the model to 2D objects but chose to omit the details for clarity. The results hereafter include the outcome of such an extension.

Figures 5.5a-c show three images of an object taken from different angles/positions. The $512 \times 512$ object (shown in Figure 5.6) is a 2D sinusoidal function whose spatial frequencies are chosen to have non-zero response in the MTF of all imagers at all distances of interest. The dimension of the object pixel size is chosen to be $0.01\text{m}$ in both directions (the pixel size is chosen so as to have no aliasing when representing the sinusoidal spatial frequencies).

All three imagers possess similar intrinsic parameters ($f = 0.05\text{m}$, $D = 0.05\text{m}$, $L_x = L_y = 64$, $\Delta x'' = \Delta y'' = 10\mu\text{m}$). The locations and angles of the imagers (extrinsic parameters) are indicated in Figures 5.5a-c. As can be seen, the area seen by each imager depends highly on the values of these extrinsic parameters. The “imager resolution” (defined as the smallest resolvable object area with the imager) varies with the extrinsic parameters as well. Thus, it is vague to define “areas of overlap” between the imagers in the image domain.

Recall that our collaborative compression algorithm of Chapter 4 trades image-domain pixels between sensors for the purpose of exploiting spatial correlation. Thus, we need to modify our CMT algorithm so as to perform the collaboration in the common object space. The first step towards doing so requires inverting Equation (5.4) for each imager. This task is not straight-forward mainly because the matrix
Figure 5.5: Images measured by imagers (a) 1: \((x, y, z) = (-1.14, 1.22, -79.56)\) m, \(\theta_x = -0.81^\circ, \theta_y = 0.68^\circ\), (b) 2: \((x, y, z) = (-1.70, 2.24, -69.69)\) m, \(\theta_x = -0.76^\circ, \theta_y = 1.44^\circ\) and (c) 3: \((x, y, z) = (-1.97, -1.32, -126.17)\) m, \(\theta_x = -0.50^\circ, \theta_y = -0.42^\circ\).
$C_i$ possesses a large number of all-zero rows. This renders the problem ill-posed. Hereafter, we choose a simpler inversion method that ignores the blur due to the optics and detector pixels and attempts to reconstruct an estimate of the object by using only geometrical side-information. The point of this exercise is to quantify the performance of our CMT algorithm when combined with a simple reconstruction method. The task of the CMT algorithm is to balance the load across the network via compression and routing. The reconstruction algorithm would demonstrate that it is possible to utilize the measurements transmitted by the imagers in order to reconstruct a good estimate of the object. The result is a practical method that could be easily implemented on a real-life network of imagers.
5.3 Modified CMT and a reconstruction algorithm

We present in this section the reconstruction algorithm that we use with the realistic model of Section 5.2. We also present a slight modification to our CMT algorithm that enables us to integrate both procedures.

Table 5.1 describes this modified CMT and reconstruction (m-CMTR) algorithm. We discuss in more detail the following non-trivial steps of the table:

• **Step 2**: Images are encoded losslessly at each imager. The resulting file size is transmitted to the base-station.

• **Step 3**: The base-station utilizes information about the 5 extrinsic \((x, y, z, \theta_x, \theta_y)\) and 2 intrinsic \((f, D)\) camera parameters of each imager in order to determine the corresponding FOV. The PSF blur spot is ignored here.

• **Step 5**: The base-station utilizes prior-information about the imager detector size \((\Delta x'', \Delta y'')\) in order to determine which image pixels see which region(s). In this computation, it is allowed for an image pixel to “see” more than one region.

• **Step 6**: Recall that our algorithm determines the optimal transmission strategy, i.e. \(\{x_{i,j}\}\), such as to maximize the lifetime of the network. The initial loads of the sensors (i.e. regions) are one of the factors defining the constraint set \(C'\) of the optimization problem, (c.f. Equations (4.1) and (4.2)). The regions (and hence the real and virtual imagers) here are defined in object domain. The cost of transmitting one region should thus be reflected back into the imager domain.
1- Randomly deploy $N$ imagers.

2- Take images and determine the encoding rate, $R_i$ (in bits/pixel), of imager $i$.

3- for each imager $i$, utilize geometry to determine the set of object pixels seen by imager $i$.

4- Determine the “largest non-overlapping regions.” Define $K$ to be the number of those mutually exclusive regions.

5- Determine the number of image pixels from imager $i$ that belong to region $k$, $c_{i,k}$.

6- Estimate encoding costs of each region, i.e. initial loads $x_{k,k}$:
   i. If region $k$ is seen by only 1 sensor $i$, then $x_{k,k} = c_{i,k}R_i$ (bits).
   ii. If region $k$ is seen by more than 1 sensor, then set $x_{k,k} =$ size (in pixels) of region $k$.

7- Modify equality constraints and apply our CMT algorithm.

8- Create an object-space 2D grid (denoted by XMT) at the base station that specifies which sensor sends each object pixel.

9- Begin the reconstruction step:

   for each imager $i$
      for each imager pixel, $(m_i)_{l,j}$
         i. Determine the set of object pixels seen by each imager pixel, $G_{l,j,i}$.
         ii. if any object pixel $\{p,q\}$ is such that $\{p,q\} \in G_{l,j,i}$ and $\frac{XMT_{p,q}}{2} = 1$ then, replicate that pixel $(m_i)_{l,j}$ over the entire $G_{l,j,i}$ area.

10- Average those object pixels that saw contributions from more than one sensor.

Table 5.1: mCMTR Algorithm.
• **Step 6-i**: The initial load for a real imager, i.e. a region seen by one sensor $i$ only, is obviously the number of pixels (bits) from imager $i$ contributing to the region defining the real imager.

• **Step 6-ii**: The initial load for a virtual imager is not as trivial. A region in object space seen by more than one imager does not necessarily cost the same amount of bits for each of those imagers. That is because our model allows the same object region to be imaged at different resolutions/quality. So, we set the initial load of a virtual imager to be the number of object domain pixels defining that shared region. Then, we scale that number appropriately depending on who is set to transmit any portion of that region. This scaling factor is quantified in Step 7.

• **Step 7**: Note that we are solving for the number of bits in (image domain) that need to be communicated between any 2 nodes. The only modification to our algorithm is made to Equation (4.1). As explained in the above bullet, we perform the following conversion:

$$
\sum_{j=1}^{N} x_{j,i} + \sum_{l \in \mathcal{V}_i} \frac{x_{l,i}}{x_{l,l}} R_{i,l} c_{i,l} = \sum_{j=0,j \neq i}^{N} x_{i,j}, \forall \, i = 1, 2, ..., N.
$$

(5.5)

As can be seen, the second term on the left side of the equality is the only term that requires modification. Note that $x_{l,i}, l \in \mathcal{V}_i$, is simply the number of object domain pixels to be transmitted from virtual imager $l$ to its associated real imager $i$ (c.f. Equation (4.2): $\sum_{i \text{ s.t. } l \in \mathcal{V}_i} x_{l,i} = x_{l,l}, \forall \, l = N + 1, N + 2, ..., N + V$.) This number divided by the total number of object domain pixels $x_{l,l}$ constituting that shared region simply
yields the fraction of region \( l \) to be transmitted to real imager \( i \). Multiplying this fraction by the total number of bits that this region costs to imager \( i \), \( R_i c_{i,l} \), yields the cost of this allocation (in bits) to imager \( i \).

- **Step 8:** Determine the number of image domain pixels that each imager should send from each region. It is obvious that a region associated with a real imager should be entirely sent by that same imager. Imagers sharing a region utilize a raster scan order to transmit the required pixels \( \left( \frac{x_{li}}{x_{li}}, c_{i,l} \right) \) from that region. Then, use the forward geometry model to determine the corresponding object pixels that each image pixel would contribute to. Encode this information on a grid: \( XMT_{p,q}+ = 2^i \) if imager \( i \) will contribute to object pixel \( \{p, q\} \).

5.4 mCMTR Performance

We present some example reconstructions of our mCMTR algorithm described in Table 5.1.

We use the example of the three imagers deployed in Figures 5.5a-c. Each imager is assumed to have an initial energy of 1J. The base-station is assumed to be located at \( x_b = y_b = 100 \text{m} \) and \( z_b = Z_{\text{max}} = 135 \text{m} \), as shown in Figure 5.4. This places the base-station outside of the object boundaries and above all imagers\(^3\). Figure 5.7 shows the regions in object domain resulting from this deployment. Each region is colored with a different level of gray, the darkest region (0) being seen by no imager, and the brightest region (grayscale of 7) being seen by all 3 imagers. The
total number of object pixels covered by this deployment is 37,610 which results in a coverage \( \eta = \frac{37610}{512^2} = 0.1435 \). The solution obtained by our collaborative compression algorithm is shown in Figure 5.8. This figure shows which object pixels were assigned to which imagers. Using our mCMTR algorithm, the resulting network lifetime of this deployment is equal to 40 transmissions. The lifetime of this same deployment when using the baseline algorithm, whereby every imager sends its entire image, is equal to 31 transmissions.

![Figure 5.7: Object-domain map showing the largest non-overlapping regions for the deployment of Figures 5.5a-c.](image)

The resulting reconstruction using our mCMTR algorithm is shown in Figure 5.9. It is clear from this figure that the algorithm was able to reconstruct a "visually

\[ ((d_{i,0})_{\text{min}} \approx 141m, (d_{i,0})_{\text{max}} \approx 155m) \]. This slight difference in distance only yields a slight increase in a sensor's lifetime between heights \( Z_{\text{max}} \) and \( Z_{\text{min}} \).
Figure 5.8: An object-domain map showing which imager(s) contribute to every object pixel, for the deployment of Figures 5.5a-c

pleasing” estimate of the object. One important question at this point is, what is the quality of the reconstruction? Although the answer is task-specific, one straightforward fidelity measure is the RMSE. Comparing only the reconstructed object regions in Figure 5.9 to the original yields an RMSE of 4.5, which is equivalent to a 1.76% error.

In order to observe the interdependence of lifetime, coverage, and RMSE, we repeat the same random deployment procedure used in Chapters 2 and 4. Figure 5.10 plots the lifetime $\alpha$ versus coverage $\eta$. The solid and dashed lines correspond to the performance of the mCMTR and baseline algorithms, respectively. We see here a trend in the performance of the mCMTR algorithm similar to what has been observed in Chapters 2 and 4: as coverage increases, lifetime decreases. In
earlier chapters, the reason behind this trend was attributed to the increasing inter-sensor correlation with decreasing coverage. However, in our more realistic model here, a given value of coverage (for a fixed number of sensors), is a function of 2 quantities: (a) the amount of overlaps (just like before) and (b) the distances between the imagers and the object. Thus, the minimum value of coverage we expect to observe here is when all imagers’ FOV overlap and when the all imagers critically sample the object (i.e. \( \frac{\Delta x'_{l,j}}{\Delta x} = 1, \forall \text{ imagers } i \text{ and imager pixels } (l,j) \)). In that case, 

\[
\eta_{\text{min}} = \frac{64^2}{512^2} = 0.0156.
\]

As the distance between the imager and the object increases, coverage increases but, both magnification and imager resolution decrease. This implies that, a given spatial frequency in the object would “appear” in the image
plane of a given imager at higher spatial frequencies when imaged from a farther position. This can be observed from the power spectra of the images in Figures 5.5a-c shown in Figures 5.11a-c, respectively. We then expect more distant imagers to produce images that are harder to compress. In fact, the lossless compression cost of the images in Figures 5.5a-c are $x_{1,1} = 7641$, $x_{2,2} = 5255$ and $x_{3,3} = 11456$ bits, respectively. Thus, far-away imagers would transmit “more-bits” than closer imagers. This also contributes to the decrease in lifetime with increasing coverage. This can be clearly seen from the performance of the baseline approach: even though every imager here is transmitting its entire image, the lifetime of the network changes (decreases) as coverage increases. Finally note that our mCMTR algorithm provides
a gain in the lifetime of the 3-imager network as high as 25% compared to the baseline approach.

The quality (resolution) degradation with coverage can be observed by plotting the quality of the reconstructed object (RMSE) as a function of coverage. Figure 5.12 illustrates that, as expected, RMSE increases with coverage.

Figures 5.10 and 5.12 show that there is a certain trade-off between the transmission capability of the network (lifetime) and the quality at which it transmits the images. A high coverage of the object is naturally desirable, as it is the initial purpose of deploying the imagers. However, a given coverage \( \eta \) could be induced at different qualities, e.g. one distant imager “covering” the entire object versus many close imagers covering the same area. So, it is important at this stage to define a “utility” function that combines both scene coverage and image quality in order to quantify the goodness of a given deployment. The utility function should favor both high coverage and good image quality as well. Since, for a given imager, these quantities are inversely proportional, we propose a measure that maximizes their product [62]. Namely,

\[
\zeta (\{\text{FOV}_i\}) = \eta \sum_i \frac{1}{N} \left[ \frac{r_o}{r_i} \right]^b
\]  

(5.6)

where

\[
r_o \equiv \min \{\Delta x, \Delta y\}
\]

(5.7)

\(^4\)This assumes (a) no aliasing in the detector sampling and (b) the object frequency to be within the pass-band of the MTF of the imager at all distances of interest.
Figure 5.11: Power spectra of images from Figures 5.5a-c shown using a logarithmic scale. Note that the spectrum of Imager 1 contains higher frequencies than the proximate Imager 2. It is also obvious from the spectrum of the farthest Imager 3 that the main spatial frequencies have shifted outwards to induce even higher spectral content.
Figure 5.12: RMSE obtained using our mCMTR algorithm as a function of coverage for the same network of 3 sensors.

denotes the object pixel size and,

\[ r_i \equiv \max \left\{ \frac{\Delta x''_i}{M_i}, \frac{\Delta y''_i}{M_i} \right\} \quad (5.8) \]

denotes imager \( i \)'s resolution, with \( M_i \) being the magnification of the imager. Note that, since \( r_i \leq r_o \), the maximum (desired) value the utility function \( \zeta \) can take is 1. This occurs when all imagers cover the entire object and, the imager resolution matches the pixel size of the object. The exponential factor \( b \) determines the relative importance of resolution compared to coverage. An infinitely large \( b \) corresponds to a measure that does not “reward” any imager with a resolution lower than that of the object. Hereafter, we pick \( b = 2 \). Finally, note that the measure \( \zeta \) also depends on the number of imagers deployed. For example, for our deployment of Figure 5.9,
$\zeta = 0.0511$. More example deployments along with their utilities are shown in Figures 5.13a-d.

Figure 5.14 shows a plot of the network lifetime $\alpha$ as a function of $\zeta$ for a network of 3 imagers. The solid and dashed lines correspond to the mCMTR and baseline algorithms, respectively. Note first the performance of the mCMTR algorithm. As expected, for a fixed number of sensors, a higher value of utility results in a lower network lifetime. The performance of the baseline algorithm, however, is constant over all observed utilities. For example, at a utility of 0.04, the lifetime obtained using our mCMTR algorithm is 31% larger than that achieved with the baseline approach.

Next, we plot the network lifetime as a function of $\zeta$ for different numbers of imagers. Figure 5.15 shows the result for $N = 3, 4, 5$ and 6 imagers. We note the following from the performance of the mCMTR (solid lines). First, the range of $\zeta$ that occurred varies with the number of imagers. As can be seen, both the minimum and maximum values of $\zeta$ increase with $N$. This is expected. As the number of imagers increases, it is more likely to obtain (a) a deployment with higher coverage and/or (b) imagers with better resolution. Also, note that for a fixed lifetime, the utility $\zeta$ increases with the number of imagers. Or, equivalently, for a fixed utility $\zeta$, the lifetime increases with $N$. This is expected as well. As the number of imagers increases, it becomes “cheaper” to transmit a given scene at a given coverage/resolution. For example, for a utility of $\zeta = 0.06$, the lifetime achievable with 6 imagers is on average
Figure 5.13: Example utilities for various deployments. (a) $N = 3$, $\zeta = 0.0717$, (b) $N = 1$, $\zeta = 0.0161$, (c) $N = 10$, $\zeta = 0.1545$ and (d) $N = 25$, $\zeta = 0.28$. 
Figure 5.14: Lifetime as a function of the utility function $\zeta$ for a network of 3 imagers. 1.77 times greater than that obtained with 3 imagers. Finally, note the constant lifetime obtained when using the baseline approach (dashed lines). For example, for a utility of 0.07, the lifetime obtained using our mCMTR algorithm with 6 imagers is 66% larger than that achieved with the baseline approach.

In order to better observe the performance of our mCMTR algorithm as a function of the number of imagers, we perform the following. For a given $N$, we compute the average lifetime values obtained. We repeat for different values of $N$ and then plot the resulting quantities. The result is shown in Figures 5.16. Also included in this Figure 5.16 is the performance of the baseline algorithm (dashed line). As can be noted, when using 10 imagers, the average lifetime obtained using our algorithm is 62% larger than that obtained with the baseline algorithm.
Figure 5.15: Lifetime as a function of deployment utility $\zeta$ for $N=3,4,5,$ and 6 imagers.

We repeat the same averaging procedure over all utility values obtained for a given number of imagers. Figure 5.17 plots the result for a various $N$. As can be seen from the figure, $\zeta$ also increases with the number of imagers, as expected. Note that the utility $\zeta$ does not depend on the reconstruction, collaborative compression or, transmission algorithms. In fact, $\zeta$ only quantifies the utility of the deployment. So, plotting the values of $\zeta$ for a given $N$ for the baseline algorithm would yield the same curve as in Figure 5.17. The idea behind this choice of utility function is to show that, for a desired “goodness” of deployment, we can balance the load across the imagers in a manner that maximizes the lifetime of the network.

Other utility functions that depend upon the reconstruction algorithm could be introduced as well. One such trivial utility function is the weighted coverage, $\eta_w$. 
defined as:

\[ \eta_w = \frac{\eta}{RMSE}. \] (5.9)

Note that \( \eta_w \) is desired to be large (i.e., small RMSE and large \( \eta \)). We then repeat the same procedures outlined above and plot the resulting lifetime as a function of \( \eta_w \) for various \( N \). The result is shown in Figure 5.18. We observe a similar trend as in Figure 5.15. In other words, when using the mCMTR algorithm (solid lines), (i) lifetime decreases with the utility for a fixed \( N \), (ii) lifetime increases with \( N \) for a fixed utility. The result also shows that our algorithm is able to achieve lifetime gains over the baseline approach (dashed curves). For example, for \( \eta_w = 0.04 \), our algorithm is able to attain a 60% gain over the baseline method with 6 imagers.
5.5 Concluding remarks

In this chapter, we presented two practical procedures. The first method applies our CMT algorithm without any modification. This method yields good results. However, this direct method also assumes our simple 2D imaging model and thus necessitates restrictive calibration procedures for its successful application. The second method is based upon a much more realistic imaging model. When combined with a simple reconstruction method, we have shown that this mCMTR method can still balance the load across the network and also yield a “useful” reconstruction. The utility of a reconstruction was defined in a manner that combines both coverage and resolution. The presented results agree with expectations.
Figure 5.18: Lifetime as a function of deployment utility $\eta_w$ for $N=3, 4, 5,$ and 6 imagers.
CHAPTER 6
CONCLUSION AND FUTURE RESEARCH

In this dissertation, we considered a network of imaging sensors. We presented novel collaborative compression methods as well as a novel multi-hop routing algorithm for the purpose of maximizing the lifetime of the network.

In Chapter 2, we developed a heuristic-based method that exploits the redundancy in the measurements of imaging sensors in order to maximize the lifetime of the network without utilizing inter-sensor communication [15]. Gains in network lifetime up to 114% are obtained when using the suggested algorithm with lossless compression.

In Chapter 3 we developed a novel theory for maximizing the lifetime of unicast multi-hop wireless sensor networks [13]. An optimal centralized solution was presented in the form of an iterative algorithm. The sensors considered here had no correlation amongst their measurements.

In Chapter 4, we presented a novel collaborative algorithm that was used in conjunction with a cooperative multi-hop routing strategy in order to further increase the lifetime of the imaging network [12]. The algorithm is demonstrated to achieve average gain in lifetime as high as 3.2 over previous methods.
Finally, in Chapter 5, we presented two practical procedures. The first method applies our CMT algorithm without any modification and thus illustrates its practicality. This method, however, assumes our simple 2D imaging model and thus necessitates certain conditions for its successful application. The second procedure modifies slightly our existing CMT algorithm. When combined with a simple reconstruction method, we have shown that this second mCMTR method can still balance the load across the network and also yield a “useful” reconstruction. The utility of a reconstruction was defined in a manner that combines both coverage and resolution. The presented results agree with expectations.

Imaging in a sensor network is still an emerging field. Our work developed a solid and practical foundation for an exciting framework. There are several issues that need to be further considered, however.

One such issue that should be studied is the inherent assumption about the noise-free measurements. In this work, we focus on maximizing the network lifetime at the inherent resolution of the sensors/cameras. Nevertheless, due to real world issues such as sensor noise, different sensor impulse responses, different illumination, etc., our simple spatial partitioning may not be optimal. In fact, combining the measured information in overlapping regions could improve resolution in those regions via some fusion or super-resolution algorithm [21].

Another important issue is the task of the sensor-network. In this dissertation, we have assumed that the task of the network is to transmit the union of the FOVs
to the base-station. However, if the task is of a lower-dimensionality, e.g. detection of a target, then the transmission of the union of the FOVs might not be the most energy-efficient procedure. A valid question is then the following: What is the trade-off between network lifetime and task performance in that lower-dimensional domain?

The topic of task-specificity brings out another actively studied issue: Compressive Imaging (CI). Compressive Imaging refers to any system in which the number of measurements is far less than the number of reconstructed pixels. In contrast with conventional imaging, compressive imagers offer (a) reduced camera cost, weight, size and (most importantly here) power due to the reduction in the number of photodetectors, (b) improved signal-to-noise-ratio because the same total number of photons can be measured using fewer photodetectors [36, 46–48]. Combining Compressive Imaging with energy-efficient compression and transmission methods could yield tremendous increases in the lifetime of sensing networks.
In this appendix, we show that the probability distribution function (pdf) of the coverage $\eta$ can be approximated by a doubly-truncated Gaussian distribution.

This fact becomes apparent when we realize that the scene coverage is actually formed by the random superposition over space of the SFOVs. In our model introduced in Section 2.3.2, we postulated a deployment process which obeys the following statistical model:

- The SFOVs can be modelled by a 2D disturbance function $h(x, y)$ such that:
  $$h(x, y) = \begin{cases} 
  1, & \text{for } x = 0, 1, \ldots, f - 1 \text{ and } y = 0, 1, \ldots, f - 1; \\
  0, & \text{otherwise.}
  \end{cases}$$

- The set of $N$ disturbance functions (which correspond to $N$ sensors) are scattered over $N$ random locations with upper left corners $\{(x_i, y_i), i = 1, 2, \ldots, N\}$ such that $x_i = 0, 1, \ldots, M - f$ and $y_i = 0, 1, \ldots, M - f$.

- The locations $\{(x_i, y_i)\}$ are independent realizations of a random variable (RV) that obeys a given bivariate probability law $p_{X,Y}(x, y)$. For our purposes, $p_{X,Y}(x, y)$ is chosen to be uniform over the range $x = 0, 1, \ldots, M - f$ and $y = 0, 1, \ldots, M - f$.

Much of the following analysis will be simplified due to the uniformity assumption.
However, it is possible to develop expressions that hold for any arbitrary \( p_{X,Y}(x,y) \).

From this model, we can then write the scene coverage as a function of the RVs \((x_i, y_i)\) using the following transformations:

- Let \( s(x,y) \) represent the sum of the disturbance functions
  \[
  s(x,y) = \sum_{i=1}^{N} h(x - x_i, y - y_i). \tag{A.1}
  \]
  Thus, \( s(x_0, y_0) \) represents the total number of overlapping SFOVs at position \((x_0, y_0)\).

- Next, we define another discrete valued random variable \( \bar{s}(x,y) \) such as
  \[
  \bar{s}(x,y) = \begin{cases} 
  1, & \text{if } s(x,y) > 0; \\
  0, & \text{if } s(x,y) = 0. 
  \end{cases} \tag{A.2}
  \]

- Finally, the scene coverage \( \eta \) can be written as
  \[
  \eta = \frac{1}{M^2} \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \bar{s}(x,y). \tag{A.3}
  \]

To determine the pdf of \( \eta \) we first find the pdfs of \( s(x,y) \) and \( \bar{s}(x,y) \). We start by finding the distribution of \( s(x,y) \). Obviously, \( s(x_0, y_0) = \sum_{i=1}^{N} h(x_0 - x_i, y_0 - y_i) \) represents the sum of independent experiments. Each experiment consists of dropping a sensor. The probability \( p \) that \( h(x,y) \) “covers” a point \((x_0, y_0)\) is simply \( p = \frac{f^2}{M^2} \) (assuming \( M \gg f \), so that edge effects can be ignored). After \( N \) such independent experiments, the probability that \( s(x_0, y_0) = n \), where \( n = 0, 1, \ldots, N \), is simply
expressed as a binomial,

\[ P_s^n = \binom{N}{n} \left( \frac{f^2}{M^2} \right)^n \left( 1 - \frac{f^2}{M^2} \right)^{N-n}. \] (A.4)

From this, it is trivial to find the pdf of \( \bar{s}(x, y) \). In fact, we can clearly see from Equation (A.2) that the distribution of \( \bar{s}(x, y) \) is Bernoulli with probability \( \bar{p} \) given by:

\[
\bar{p} = Pr(s(x_0, y_0) > 0) = 1 - Pr(s(x_0, y_0) = 0) = 1 - P_s^n = 1 - \left( 1 - \frac{f^2}{M^2} \right)^{N-n}. \] (A.5)

Finally, we see from Equation (A.3) that \( \eta \) is in fact a finite sum of dependent identically distributed random variables. The pdf of such a random variable can be approximated by a (positively-truncated) Gaussian distribution [4,19,34]. Here, the mean of \( \eta \) is simply given by:

\[ \langle \eta \rangle = \bar{p} = 1 - \left( 1 - \frac{f^2}{M^2} \right)^N. \] (A.6)
REFERENCES


