University of California
Los Angeles

Energy Efficient Sampling, Source Coding, and Data Routing in Wireless Sensor Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Huiyu Luo

2005

The dissertation of Huiyu Luo is approved.

Kung Yao

William Kaiser

Deborah Estrin

Gregory Pottie, Committee Chair

University of California, Los Angeles

2005

To the memory of my father

Table of Contents

List of Figures

List of Tables

Acknowledgments

My greatest and heartfelt gratitude goes to my advisor Professor Gregory Pottie, who is instrumental in providing a comfortable and stimulating research environment for his students. Gregory is always motivating, encouraging, and inspiring. Without his constant support, none of this work would have been possible.

I also thank Professor Kung Yao, Professor William Kaiser, and Professor Deborah Estrin for taking time to serve on my dissertation committee and provide me encouragement and insightful advice.

I am grateful to all my teachers at UCLA for their valuable teachings. In particular, I would like to thank Professor Izhak Rubin for guiding me in my Master studies and Professor Ali Sayed for allowing me to work as his teaching assistant in multiple occasions. Without the pleasant and stimulating interaction with my colleagues and friends, my experience at UCLA would not have been so memorable and rewarding. In particular, I am indebted to Ameesh N. Prandya, Yu-Ching Tong, Xiangming Kong, and Zhaoyu Zhang.

During my journey to the Ph.D, my family have always been there watching over me. Without their love and care, nothing will be possible for me. Lastly, I would like to thank Ms. Ping Zhang for her unconditional support in all these years. She has my deepest love and gratitude.

Vita

1976	Born, Pingxiang, Jiangxi, China
1999	B.S. Mechanical Engineering, University of Science and Technology of China, Hefei, Anhui, China.
2002	M.S. Electrical Engineering, University of California at Los Angeles, Ca, USA.
2005	Ph.D Electrical Engineering, University of California at Los Angeles, Ca, USA.

Publications

Huiyu Luo and Gregory Pottie, "Designing routes for source coding with explicit side information in wireless sensor networks," submitted to *IEEE/ACM Transactions on Networking,* 2005.

Huiyu Luo, Xiangming Kong, and Gregory Pottie, "An adaptive algorithm for sampling the sunlight field under a forest canopy using mobile sensors," submitted to *ACM Transactions on Sensor Networks,* 2005.

Huiyu Luo and Gregory Pottie, "A study on combined routing and source coding with explicit side information in sensor networks," *IEEE Global Telecommunications Conference,* St. Louis, Mo, USA, November, 2005.

Huiyu Luo and Gregory Pottie, "Balanced aggregation trees for routing correlated data in wireless sensor networks," *IEEE International Symposium on Information Theory,* Adelaide, Australia, September, 2005.

Huiyu Luo and Gregory Pottie, "Routing explicit side information for data compression in wireless sensor networks," *International Conference on Distributed Computing in Sensor Systems,* Marina del Rey, Ca, USA, June–July, 2005.

Huiyu Luo, Yuching Tong, and Gregory Pottie, "A two-stage DPCM scheme for wireless sensor networks," *IEEE International Conference on Acoustic, Speech, and Signal Processing,* Philadelphia, Pa, USA, March, 2005.

Gregory Pottie, Huiyu Luo, and Ameesh Pandya, "Sensor network information theory," chapter in *Encyclopedia of Sensors,* Edited by Craig A. Grimes, Elizabeth C. Dickey, and Michael V. Pishko, American Scientific Publishers, 2005.

Ameesh Pandya, Huiyu Luo, and Gregory Pottie, "Spatial fidelity and estimation in sensor networks," *The 38th Annual Asilomar Conference on Signals, Systems, and Computers,* Pacific Grove, Ca, USA, November, 2004.

Ameesh Pandya, Huiyu Luo, and Gregory Pottie, "Characterizing sensor networks," *IEEE International Symposium on Information Theory, Poster Session on Recent Results,* Chicago, Il, USA, June–July, 2004.

Huiyu Luo, Ameesh Pandya, and Gregory Pottie, "Detection fidelity in distributed wireless sensor networks," *UCLA CENS Technical Report #20,* 2003.

Izhak Rubin, Arash Behzad, Runhe Zhang, Huiyu Luo, and Eric Caballero, "TBONE: a mobile-backbone protocol for ad hoc wireless networks," *IEEE Aerospace Conference,* 2002.

Abstract of the Dissertation

Energy Efficient Sampling, Source Coding, and Data Routing in Wireless Sensor Networks

by

Huiyu Luo

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2005

Professor Gregory Pottie, Chair

One important problem in wireless sensor networks is how to efficiently utilize the limited network resources to observe and estimate physical phenomena. In this dissertation, we take a divided approach to this problem and separately devise efficient algorithms for field sampling, source coding, and data routing.

Before we start, various types of distortion during the process of sensing, quantization, communication, and reconstruction are examined. It is observed that the bounds on these errors are fundamentally tied to the scarce network resources, e.g. node density, sensor energy, and communication capacity.

Nodes with limited and controlled mobility have been proposed recently for use in wireless sensor networks. The problem of efficiently relocating sensors to sample a distributed field is investigated. We propose an adaptive algorithm based on the Bayesian framework. This scheme maintains an estimate of how well the current reconstructed field approximates the true field based on all collected samples, while iteratively sampling the field by picking the most desirable set of sampling sites from a candidate pool. With minor modifications, this method can also be used in a distributed implementation where static sensors are woken

up from sleep to collect measurements.

Due to the high data correlation in a sensor network, source coding should be used to remove redundancy among data streams from different sensors even before they are transmitted to the fusion center to reduce communication cost. We give a brief overview of distributed source coding, where sensors independently conduct data compression without interacting with one another. Then, our attention is shifted to source coding with explicit side information. A two-stage DPCM (differential pulse coded modulation) coding scheme is proposed. It can continuously monitor the additional coding gain provided by correlated side information from other sensors, and hence can be used in joint data aggregation/routing optimization.

The last topic we take up is the data-centric routing. We proposed a data aggregation model for source coding with explicit side information. In this model, data transmissions are decomposed into individual flows originating at different sensors, and a data rate function is defined for each flow. The full optimization problem is formulated and discovered to be NP hard, which indicates that efficient algorithms for finding the exact solution are unlikely to exist. We turn to heuristics subsequently. Several routing schemes are examined. Among them, the BAS (balanced aggregation scheme) and DSIT (designated side information transmission) hold the highest promise as they yield good performance when data correlation is high and converges to SPT (shortest path tree) when coding gain diminishes.

CHAPTER 1

Introduction

1.1 Wireless Sensor Networks

1.1.1 The emergence of wireless sensor networks

In the last several decades, advances in solid-state physics, integrated circuitry, MEMS (micro electro-mechanical systems), wireless communications, and digital signal processing technologies have resulted in the development of powerful hardware platforms designed for distributed sensing applications. Sensors incorporating wireless transceivers and signal processing modules have been produced at decreasing cost due to progress in fabrication technologies. These wirelessly networked sensors can be used in a variety of applications such as security surveillance, disaster relief, ecosystem monitoring, manufacturing control, inventory tracking, entertainment, performance arts, and education [EGP01, PK00, SS02a, EGH00, CEE01, MPS02, BMK02, SMP01]. Combining the sensing capability of micro-sensors, computing power of processors, and wireless networking, wireless sensor networks have the potential to change the way we interact with the physical world, and are becoming one of the most exciting frontiers of computer science and engineering.

The future of computing, as envisioned in [Wei91] and [Ten00], is to shift from traditional human-interactive computers to proactive and pervasive embed-

ded systems. The realization of this vision of pervasive computing requires the computing system to quickly respond to external stimuli and extensively interact with the physical world, in which distributed networks of embedded sensors, controls, and processors promise to be an essential ingredient.

1.1.2 Why distributed?

The property and distribution of the sources that the sensing systems are to observe and interact with necessitate that we take a distributed approach in designing such systems. The sensing object of a sensor network can be random fire sparks in the forest, the chemical composition of the water in the ocean, the mine location at a battlefield, or the temperature field of a certain area etc. It can be an isolated event or a distributed physical phenomenon. When it is an isolated event, we may not have the precise location of the event except for knowing that it is confined to a certain region. Therefore, in either case, a relatively large region often needs to be covered, and sometimes the source may even reside at places where no human beings and any established infrastructure are present. On the other hand, the strength of signals that are emitted from the source generally decays rapidly with distance in the space. For example, the power of electromagnetic waves decays as the square of the distance in free space as a result of wavefront propagation. In practice, the situation can be even worse if we take into account various dispersive and absorptive surfaces that the wavefront may encounter before it reaches the sensor. To make matters worse, multi-path may occur as well. Hence, to have reliable observations of the source, it is desirable to place the sensor as close to the target as possible so that signals are detected with the highest SNR (signal to noise ratio). Consequently, sensors need to be deployed distributively in space, and the distribution preferably follows that of

the sensing objects.

When wired networks of distributed sensors are possible, it is often the more advantageous approach. If sensor nodes can be connected to wired renewable energy sources and high speed communication links, the system design and operation are greatly simplified. However, in many applications, the environment that is being monitored has no established infrastructure or human presence, and it is too expensive to install a wired sensor network. Hence, untethered and unattended nodes with limited energy reserve must be relied on to carry out sensing tasks. Advances in technologies have made available low-power data processing units and reliable wireless communication links suitable for micro-sensors. This makes it possible to deploy wireless networks for sensing applications in places where no installed infrastructure is available at relatively low cost. With wireless connections, the sensor network also has the advantage of being reconfigurable and easy to deploy. However, at the same time, stringent constraints are placed on sensors' design and operations. Some researchers have observed that the communication capacity available to each node diminishes as the number of nodes increase in a dense wireless network [GK00]. Furthermore, the strong propagation loss of radio power makes wireless communication a power-hungry exercise. The energy that is needed to maintain the connectivity all the time will soon drain the batteries of typical wireless sensor nodes. These constraints prevent us from building wireless sensor networks using traditional centralized fusion schemes in which all the raw data are transmitted to the global fusion center, and the fusion center carries out all the data aggregation and decision making. Substantial changes to the sensing strategy, network architecture and data processing are needed. In other words, distributed approaches in sensing, communication, and processing are inevitable in wireless sensor networks [PK00].

1.1.3 Characteristics and challenges

In this section, we give an overview of some characteristics of wireless sensor networks and the resulting challenges.

- Nodes in a wireless sensor network are deployed distributively in space. If the nodes are static, the spatial distribution of nodes in the area of interest may not be precisely controlled due to the massive and quick deployment, in which accurately placing and calibrating each individual sensor in the network often represents a high cost. In addition, node failures, which may occur frequently in the network, also play a important role in shaping the network configuration.

- The size of a wireless sensor network may vary greatly from one to another. For instance, the network may consist of less than 10 sensors in a small sub-array used to determine the incoming wavefront's angle of arrival, or tens of sensors in a smart house, or hundreds of sensors in a seismic application, or thousands of small nodes envisioned for use in a battle ground. General algorithms designed to accommodate various sensor networks must be scalable.

- The capability of sensor nodes in various networks differs greatly. Individual sensor nodes range from tiny mica nodes [mic] with limited processing and communicating capability to NIMS nodes [KPS04a] that are equipped with powerful processors and can move in its patrol area to collect samples. Different strategies must be employed for such disparate networks.

- Sensor nodes, especially in these massively deployed and low-cost sensor networks, are often powered by batteries and unattended after initial deployment, so sensors become defunct as soon as their batteries are drained.

Therefore, to prolong the network life, it is desirable to employ strategies with high energy efficiency. Even in applications where renewable energy sources are available, energy conservation is recommended.

- Network components in sensor networks are usually untethered, unattended, and unreliable. Algorithms designed for such systems should be fault tolerant so that the networks are able to continue functioning even if some nodes fail. This often requires that certain quality of service (QoS) indicators, such as the estimation fidelity in field reconstruction, to be associated with the result of data fusion, so we will know as soon as the networks are unable to meet the prescribed service requirements.

- Although communication links, which are usually wireless, exist among sensors, data transmission is often considered a higher cost operation than data processing in that communications consume more energy. Moreover, while the power consumption of VLSI chips have been continuously scaled down, the fundamental limits in wireless communication (propagation loss of radio power and Shannon's theory) sets fundamental limits on the transmission power and data rate. Consequently, local processing that can reduce the communication rate is strongly encouraged in wireless sensor networks.

- In practice, since sensors are often observing some common physical phenomena, the data streams produced at different sensors, especially the ones that are close to one another, are generally correlated. Hence, local collaborative signal processing should be employed to exploit the redundancy among different data streams before transmitting them to the fusion center.

- Senor nodes are generally static. However, sensors with limited but well-controlled mobility have been developed recently. For example, [KKP05]

tries to reduce sensing uncertainty by allowing cameras to move along a straight line and adjust the angle of viewing; [RHS04] uses cableways to relocate sensors to designated locations of their patrol area. In either cases, the controlled mobility significantly enhences sensors' capability of monitoring the area of interest and adapting to the dynamic environment.

- In many sensing applications, fast responses from sensor nodes are required either because the fusion decision is time-critical or the monitored phenomenon undergoes slow changes in time. Together with the goal of energy conservation, this demands algorithms in some wireless sensor networks to have low complexity and be able to respond to the environmental stimuli quickly.

- In this dissertation, we assume there is a global fusion center, to which all sensors in the network are required to transmit their data. As a result the transmission pattern in the network is all to one. This differs from general ad hoc networks, where any node can be an end user.

The list above presents us the challenge of designing scalable, distributive, energy-efficient, fault-tolerant, fast-adapting, and low-complexity algorithms for wireless sensor networks.

1.1.4 Distributed Sensing and Fusion Paradigm

In sensor networks, to reduce the communication rate and preserve energy, data processing is best carried out distributively, and is often tightly coupled with communications. This new paradigm of sensing, communication, and data fusion is depicted in Fig. 1.1.

In Fig. 1.1, n sensors are deployed in the field to observe a distributed physical

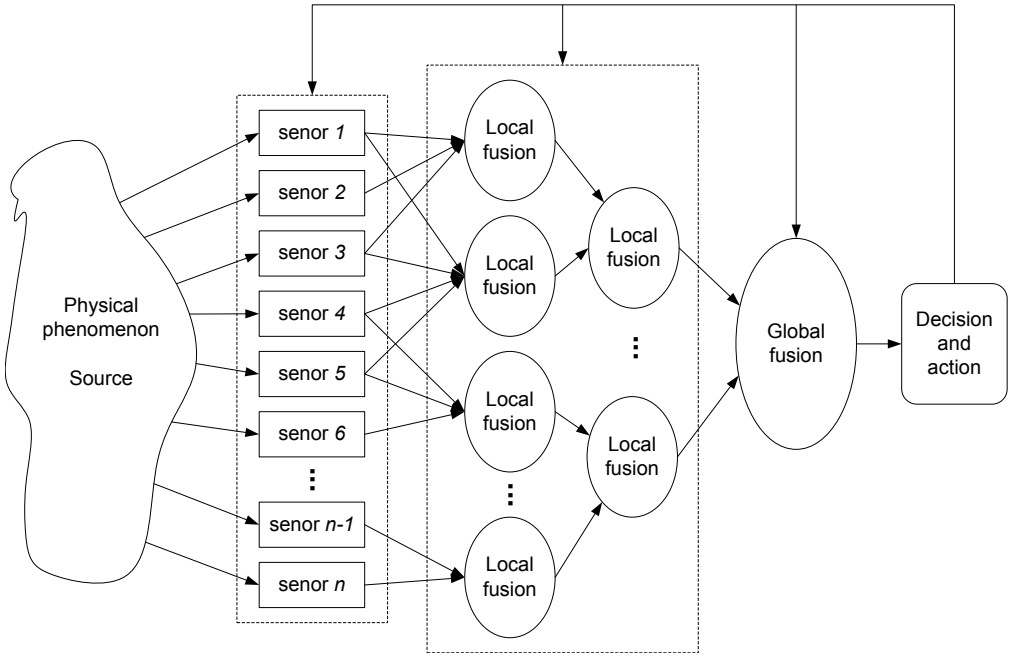


Figure 1.1: A distributed sensing paradigm.

phenomenon, and need to transmit their data to the global center to generate a fusion result, based on which actions are taken, and the sensor network is adjusted accordingly. Before converging to the global fusion center, the data streams originating from different sensors merge into small local fusion cells based on, for example, geographical proximity and the data correlation structure. Several levels of local fusion may occur before data are finally transmitted to the global fusion center. Intensive communication and data aggregation occur within each local fusion cell. The communication cost of this local data aggregation is relatively low compared to that of the transmissions to the global fusion center due to the short-ranged transmissions involved in local cooperation. Further, collaboration within these fusion cells is often the most productive in terms of reducing communication rate because in most physical phenomena the correlation among sources decrease rapidly as the spatial separation increases.

1.2 Information Theory

While designing energy efficient algorithms for wireless sensor networks, a recurring scene is that the fundamental limits on data aggregation and communication are often set by the information theory. Here, we we give a quick overview of some basic results and unsolved problems in information theory. For more details, readers are referred to [Gal68, CT91]. Additionally, the articles in IEEE transaction on information theory, vol. 44, no. 6, October 1998 (the special commemorative issue that celebrates the 50th anniversary of C. E. Shannon's landmark paper [Sha48]) provide great reviews on the subject.

1.2.1 Single user information theory

Information theory gives answers to two fundamental questions in communication theory: what is the limit of ultimate data compression, and what is the limit of error-free transmission rate across a communication channel.

The classical single channel communication system is depicted in Fig. 1.2. We want to transmit the sequence of symbols $V^n = \{V_1, V_2, \cdots, V_n\}$ across the

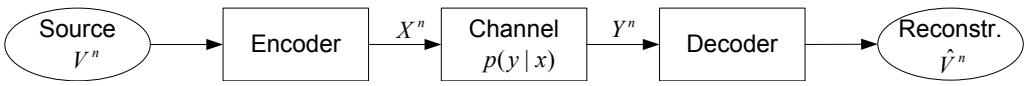


Figure 1.2: Joint source and channel coding in a classical single channel communication system.

channel, which can be described by the probability distribution $p(y|x)$. To do this, we map V^n into codeword X^n, and transmit X^n over the channel. The receiver receives Y^n, which is the codeword X^n altered by the channel, and from Y^n, the receiver reconstructs the original set of symbols $\hat{V}^n = \{\hat{V}_1, \hat{V}_2, \cdots, \hat{V}_n\}$. We define the probability of error $P_e^{(n)} = \Pr(V^n \neq \hat{V}^n)$. The joint source channel

coding theorem for such a communication system is stated as follows.

Theorem 1.2.1 (Source-Channel Coding Theorem) *If V^n is a finite alphabet stochastic process that satisfies the AEP, then a source channel code exists such that $P_e^{(n)} \to 0$ if $H(\mathcal{V}) < C$. Conversely, if $H(\mathcal{V}) > C$, then it is impossible to send the process V^n over the channel with arbitrarily low probability of error. Here,*

$$H(\mathcal{V}) = \lim_{n\to\infty} \frac{1}{n} H(V_1, V_2, \cdots, V_n) \tag{1.1}$$

is the entropy rate of the stochastic process V^n, and

$$C = \max_{p(x)} I(X;Y) \tag{1.2}$$

is the capacity of the communication channel.

This theorem also implies that we can split a single channel communication system into two parts: source coding and channel coding. In other words, we can design the most efficient representation of the source, while separately devising the best channel codeword for the specific channel. Designing the two systems independently will be just as efficient as considering the joint source-channel coding. Unfortunately, this decomposition does not preserve the efficiency in a multi-terminal communication system.

The entropy rate is defined for sources with discrete alphabets. However, most physical signals in the world are analog. The question is how to represent a continuous variable given that it is impossible to exactly describe an arbitrary analog signal in a finite length codeword. To render such a question meaningful, we have to define some distortion constraint, and reframe the problem as follows: given a source distribution and distortion measure, what is the minimum data rate required to achieve certain distortion. The answer to this question is given by the rate-distortion theory.

Theorem 1.2.2 (Rate-Distortion Function) *Given an i.i.d. source X with distribution $p(x)$ and a distortion function $d(x, \hat{x})$, the minimum rate required to represent such a source under the constraint $d(x, \hat{x}) \leq D$ is*

$$R(D) = \min_{p(\hat{x}|x):\sum_{(x,\hat{x})} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \tag{1.3}$$

where $\hat{X}$ is the reconstructed source based on the codeword.

Among all communication systems, the most important continuous alphabet channel is the Gaussian channel described in Fig. 1.3. The input X_i is corrupted

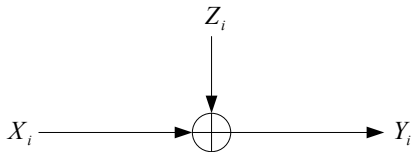


Figure 1.3: The Gaussian channel.

by the Gaussian noise $Z_i \sim \mathcal{N}(0, N)$, which gives rise to the output $Y_i = X_i + Z_i$. The capacity of this channel is given by the following theorem.

Theorem 1.2.3 (Gaussian Channel Capacity) *The capacity of a Gaussian channel with power constraint P and noise variance N is given by:*

$$C = \frac{1}{2} \log \left(1 + \frac{P}{N} \right) \tag{1.4}$$

Of particular interest to wireless communication is the capacity of a multiple antenna Gaussian channel, which has spurred intensive research on the MIMO system. We refer readers to [Tel99] for more discussions on the subject.

1.2.2 Network information theory

Since the foundations of information theory were laid by the classical paper of C. E. Shannon [Sha48] in 1948, more than five decades has passed. While tremen-

dous advances in communication theory and practice have been observed, many problems remain unsolved. Network information theory is one such area where complete answers to only a few special cases are known.

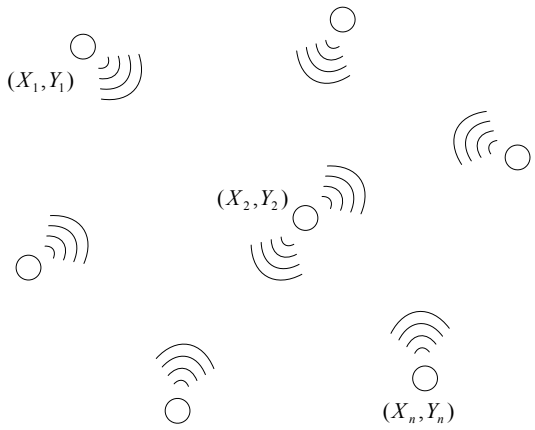
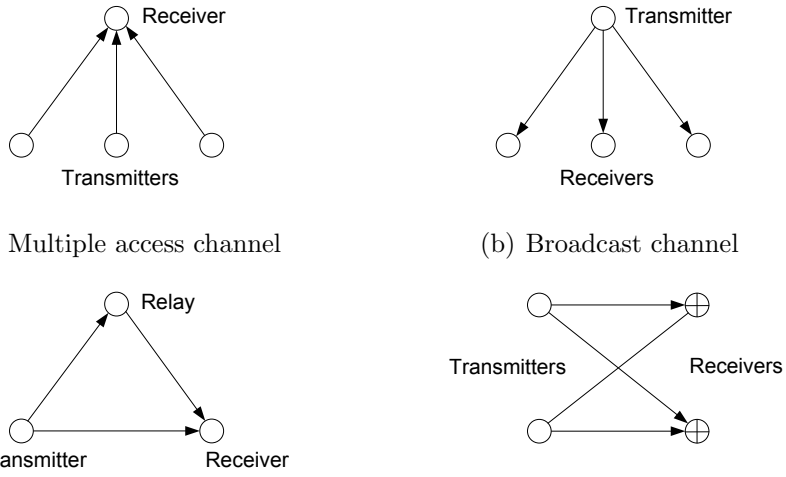


Figure 1.4: A general communication network.

In a general communication network setup depicted in Fig. 1.4, nodes in the network simultaneously send symbols X_i, $i = 1, \cdots, n$ and receive symbols Y_i, $i = 1, \cdots, n$. Given the probability distribution of the source data at senders and the channel transition matrix that describes the effect of noise and interference in the network, the question is whether or not the sources can be transmitted from the senders over the channel to the destination receivers with appropriate distortion. This general problem involves distributed source coding and distributed communication, and is extremely difficult. Unlike in a single channel communication system, separately considering source and channel coding in a network setup is known to be suboptimal. Moreover, not only does a complete network information theory remain a distant goal, the distributed source and channel coding problems when considered separately result in performance bounds rather than complete definition of achievable rates apart from a small number of special cases. In the rest of this section, we present some special instances of network

information theory.



(a) Multiple access channel

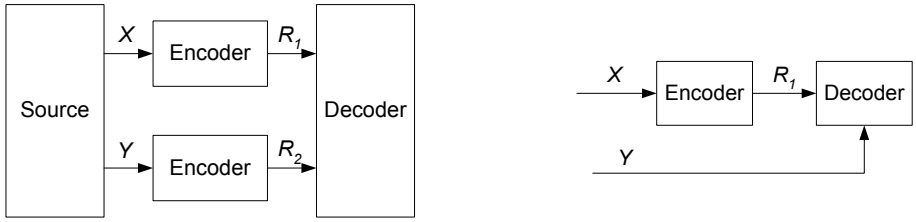
(b) Broadcast channel

(c) Relay channel

(d) Interference channel

Figure 1.5: Some special instances of channel coding. Noise exists in all the transmissions

Some instances of channel coding are depicted in Fig. 1.5. In a multiple access channel, Fig. 1.5(a), several terminals try to transmit to a common receiver. This is the best understood instance of multi-terminal channel coding, and the general capacity region is known. In Fig. 1.5(b), one transmitter attempts to communicate to multiple receivers. This problem is not yet solved, but the result is known when the channel is physically degraded. In a relay channel, Fig. 1.5(c), besides the transmitter and receiver, there are intermediate nodes to help relay the messages. The capacity region for the general problem is also unknown. The interference channel is described in Fig. 1.5(d), where each receiver wants to decode the messages from one of the receivers while treating those from the other as interference. The answer to this instance is unknown. The problem is not yet solved even under Gaussian noise. For extensive discussions, readers are referred to [CT91, GC80, Cov98]



(a) Correlated source coding (b) Source coding with side info.

Figure 1.6: Some special instances of source coding.

Some instances of distributed source coding is depicted in Fig. 1.6. In Fig. 1.6(a), two encoders independently compress two correlated sources and transmit to a common receiver. When the sources have discrete alphabets, the rate region of this instance is given by Slepian and Wolf in [SW73]. However, if the source alphabets are continuous, it becomes the rate distortion problem for correlated sources, which is unsolved. One special case of this instance is the rate distortion coding with side information, which is described in Fig. 1.6(b). Information provided by Y is used as side information to help recover source X. The rate region for this problem is given in [WZ76]. More discussions on various other special instances of the rate distortion coding problem can be found in [BY89, ZB99, Ooh97, Oza80]

1.3 A Unified Approach

Consider the general problem of coding and routing correlated data to the fusion center with minimum power in wireless sensor networks. One example is depicted in Fig. 1.7. Sensor $i = 1, 2, \cdots, n$ produces the data stream X_i, and transmits it to the fusion center through the network. The data is the result of observing certain physical phenomena. We assume that it satisfies the ergodic condition so that the results of statistical probability theory can be applied here. The objective is to optimize some cost function C while recovering these data streams

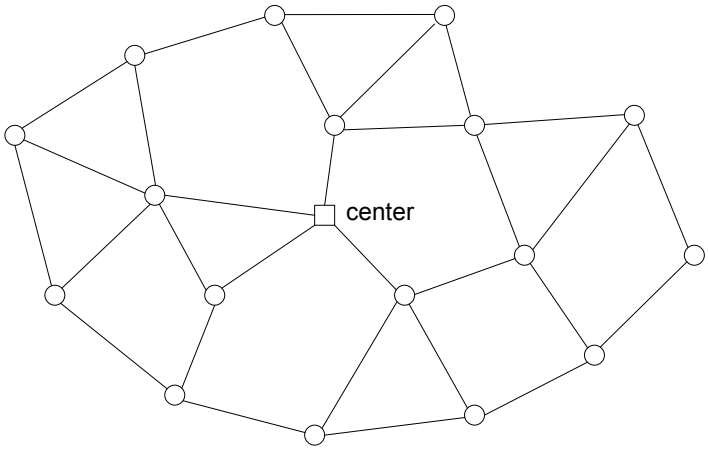


Figure 1.7: A sensor network with a single fusion center.

subject to certain distortion constraint $d(X_1, \cdots, X_n; \hat{X}_1, \cdots, \hat{X}_n) \leq D$. For the particular instance of minimizing the aggregate transmission power, the problem can be formulated as follows:

$$\begin{aligned} & \min \textstyle\sum_{i=1}^{n} P_i \\ \text{subject to:} \quad & d(X_1, \cdots, X_n; \hat{X}_1, \cdots, \hat{X}_n) \leq D \text{ is} \\ & \text{achievable under power budget } (P_1, \cdots, P_n). \end{aligned}$$

This constitutes an optimization problem, with bounds on the set of admissible power allocations determined by the complete network information theory. Unfortunately, an exact solution is unavailable. In addition, information theoretic encoders and decoders usually have high complexity and incur long delays. As a result, sub-optimal approaches that consider sampling, channel coding, source coding, and routing separately, are generally taken in practice.

1.4 Organization of the Thesis

The rest of the dissertation is organized as follows.

Chapter 2 examines the errors in sensing, quantization and source reconstruction when the sensor network is relied on to observe and reconstruct physical phenomena. The sensing error depends on signal attenuation and measurement noise, which can be lessened by reducing source-sensor separation and increasing the number of independent observations. The rate of data transmission to the global fusion center, which requires the most energy and capacity due to its long range, can be brought down by exploiting correlation among sensors through local fusion. However, the rate ultimately constrains the number of quantization levels, and is bounded by the rate-distortion theory. In this chapter, the distributed phenomenon is modeled as correlated point sources, and bounds on the total error using a localized reconstruction algorithm based on cubic splines are derived. In particular, the interpolation error in cubic spline fitting is shown to converge at least on the same order as the sensing and quantization noise given appropriate mesh sizes.

The introduction of mobility in sensor networks has generated a variety of research topics. Chapter 3 studies the problem of sampling and reconstructing a two-dimensional sunlight field under a forest canopy using robotic sensors that can quickly move to designated locations to collect light intensity measurements. The sampling and reconstruction process is carried out in adaptive steps. During each step, the most desirable sampling sites are selected from a pool of site candidates based on a maximum a posteriori (MAP) test. Source statistical models and field roughness are used to further account for heterogeneity. The adaptive algorithm is compared to other schemes, and shown to work effectively. Although developed as a centralized scheme, with minor modifications, the method is also suitable for distributed implementation, in which static sensors in a network are woken up from sleep to perform sensing tasks.

Chapter 4 discuss source coding in sensor networks. After a quick overview on distributed source coding, out attention turns to source coding with explicit side information. We implement a two-stage DPCM coding scheme for wireless sensor networks. The scheme consists of temporal and spatial stages that compress data by making predictions based on samples from the past and helping sensors. It continuously monitors the additional gain provided by samples from other sensors, which indicates the level of correlation among different data streams. Therefore, this scheme can be combined with data-centric routing algorithms for joint data compression/routing optimization. Backward ϵ-NLMS adaptation is used to better track changing environments and avoid coefficient transmissions. Several simulations based on different sets of data are conducted to demonstrate the effectiveness of this coding scheme.

Chapter 5 studies the problem of combined routing and source coding with explicit side information in wireless sensor networks. Two difficulties in designing such data-centric routes [KEW02, CBV04, GE03, IGE03] are the lack of reasonably practical data aggregation models and the high computational complexity resulting from the coupling of routing and in-network data fusion. In this chapter, the data flows are decomposed into individual flows originating from different sensors, and the data aggregation model is built upon the observation that in many physical situations the side information that provides the most coding gain comes from a small number of nearby sensors. Based on this model, we formulate an optimization problem to minimize the communication cost of routing all the data to the fusion center, and show that finding the exact solution of this problem is NP-hard. Subsequently, a mixed integer program is formulated for one sub-instance of the general CRSC (combined routing and source coding) problem.

Chapter 6 is a continuation of chapter 5, and discusses heuristic algorithms to solve the CRSC problem formulated in the previous chapter. The shortest path tree and clustering methods are first presented. Then two suboptimal algorithms are proposed. One is inspired by the balanced trees that have small total weights and reasonable distance from each sensor to the fusion center [KRY95]. The other separately routes the explicit side information to achieve data compression and cost minimization. The performances of both algorithms are analyzed. Simulations are conducted to compare the average performance of different heuristic algorithms.

Chapter 7 concludes the thesis and offers some suggestions on future research directions.

CHAPTER 2

Estimation Fidelity in Wireless Sensor Networks

2.1 Introduction

Recent advances in technology have made a broad set of applications of sensor networks possible [PK00] [EGP01]. In these applications, it is often required to deploy large scale, distributed, wireless networks. One interesting set of problems is the distortion bounds in these networks under various constraints such as sensing noise and limited network resources. As one of the basic QoS (quality of service) metrics, these distortion bounds indicate on how well the network is able to cover the area of interest. Based on this information and the fusion requirement, it can be determined whether more resources should be added to the system.

Usually, sensors observe distributed phenomena rather than single isolated events that are considered point sources. In engineering practice, however, real distributed continuous processes are never fully observable. A typical approach in sensing is to sample the processes in time and space, in which the distributed phenomena are reasonably modeled as sets of correlated point sources. The reconstruction of the source is then done by interpolation, for example, spline fitting. In this chapter, we study the distortion bounds of sensor networks based on this approach.

First of all, the distortion of sensor networks is circumscribed by the sensing

capabilities of sensors. The accuracy of sensing is affected by the amount of signal attenuation and noise corruption. While noise is an unavoidable process, attenuation is strongly related to the distance between the sensor and the source, which in turn is a function of sensor coverage.

When observing a distributed phenomenon, the rate used to quantize the signal is usually constrained by the network capacity owing to the large amount of information embedded in the source. In [GK00], using a point-point transmission model, it was found that the capacity per node decreases as the number of nodes in the network increases, which may be attributed to the fact that the destination is randomly chosen among all the nodes in the network. In [Ser02], on the other hand, it was argued that the situation in sensor networks is not at all as pessimistic since the problem is often that of coding a correlated source. The minimum rate required for correlated sources under a distortion constraint, i.e. rate-distortion problem, has been studied for decades. In this case what is bounded is the rate needed to communicate to the global fusion center under a quantization distortion constraint. Although approaching this bound may entail intensive local cooperation and fusion among sensors and local fusion centers, this is still beneficial because local transmission is far less constrained due to the bounded transmission ranges [PP03]. Also note that in contrast to [PP03], where the problem is to observe distinct point sources, the perspective here is that there is one distributed process, which is modeled using correlated point sources.

For a distributed source with a continuous sample path, the source is usually reconstructed at the fusion center by interpolating from measured points. This gives rise to interpolation error. Conventionally, interpolation error has been analyzed with the assumption that the data at prescribed points has no error. Here, we consider the combined distortion of sensing, quantization and interpolation.

Also we would like to point out that similar problems have been dealt with in image processing [Jai89], and many results may be applied here except for the distributed nature of sensor networks. This results in communication cost, and hence a set of constraints on the rate. Additionally, the sensors may be irregularly deployed and heterogeneous.

The rest of the chapter is organized as follows. In section 2.2, we consider sensing, quantizing and estimating point sources. The discussion is then extended to distributed sources. In section 2.3, the minimum rate needed for correlated sources is given based on rate-distortion theory under certain assumptions for the source distribution. In section 2.4, the total error due to sensing, quantization and interpolation when reconstructing continuous sources using spline fitting is discussed. The chapter concludes in section 2.5.

2.2 Point Sources

Point sources are useful abstractions since many phenomena can be reasonably modelled as either a single point source, or constructed via interpolation from a set of point sources. We begin with a single point source. The data sent to a fusion center by sensors is the quantized version of attenuated and noise corrupted signals radiated from the source. The sensing error, which is dependent on the strength of signal and noise at the sensor, persists even if errors due to quantization and communication are zero. In this section, we use a simple model to capture the processing of sensing a point source.

2.2.1 Sensing model

The signal radiated from the source X, corrupted by noise Z_i, and received by sensor i is modeled as:

$$Y_i = a_i X + Z_i$$

with

$$X \sim \mathcal{N}(0, \sigma_X^2), \quad Z_i \sim \mathcal{N}(0, \sigma_Z^2), \quad a_i = \frac{1}{1 + \kappa r_i^2}.$$

The attenuation factor a_i is a function of the distance r_i between the source and the sensor. κ is a constant that models how strongly the distance affects the signal attenuation. For convenience, we assume that the data sequence X^n produced by the source are i.i.d. Gaussian random variables. The noise Z_i at sensor i is assumed Gaussian and is independent of X. Also the noise is assumed i.i.d. at different sensors, which implies that they are the same type of sensor nodes and affected by similar ambient noise processes.

2.2.2 Field coverage by static sensors

We assume that the sensors are to be deployed in a unit area to monitor a point source. In addition, we are tempted to presume that the location of the source is uniformly distributed in such a general context due to the following reasons. First, we often have no prior knowledge about the probability distribution of the source location except that it is confined to a certain region. Second, the monitored area in many applications can often be considered homogeneous or comprising homogeneous sections, in which the source appears anywhere with equal probability. Therefore, it is natural to distribute sensors in an equally homogeneous fashion. Third, uniformly distributed sources and sensors are amenable to theoretical analysis.

Two ways to deploy n static sensors are considered in this chapter. One is deterministic: divide the unit area into n identical square cells, then place one sensor at the center of each cell. It should be noted that this often necessitates approximation since the boundaries of an arbitrary area seldom conform to the collection of square cells. Thus the approximation is more reasonable for networks of large size. The other way is random: independently and randomly place each sensor according to the uniform distribution assuming that the region of interest is a unit disc. If only the sensor that is closest to the source transmits its observation to the fusion center, the source-sensor separation r is $R_{\min}$, the distance between the source and the closest sensor. For either way of deploying sensors, the probability density functions of $R_{\min}$ are computed, and the mean values are found to decrease with $\sqrt{n}$.

$$\mathrm{E}(R_{\min}) \propto \frac{1}{\sqrt{n}} \tag{2.1}$$

A detailed derivation of the above relation is given in Appendix A. Placing sensors on a predetermined grid results in a lower $\mathrm{E}(R_{\min})$ than distributing sensors randomly. However, if node failures are taken into account, we surmise that $R_{\min}$ may deteriorate more drastically in a deterministic scheme after a certain number of sensor nodes become defunct. This subject is not pursued further in this dissertation.

2.2.3 Minimum mean square error estimation

The distortion measure is defined over n estimations, $d(X, \hat{X}) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2$. The raw data Y_i observed by the sensor that is closest to the source is sent back to the fusion center with the quantization distortion constraint D_q. When optimal coding that attains the rate-distortion bound is used, the quantization noise Z_q is a Gaussian random variable with variance D_q, independent of X and

Z_i. Assume that the signal received at the fusion center is Y. Arbitrarily low communication error can be achieved as long as the source coding rate R is less than the link capacity W.

$$Z_q \sim \mathcal{N}(0, D_q)$$

$$Y = Y_i + Z_q = a_i X + Z_i + Z_q$$

$$R = \frac{1}{2} \log \left(\frac{\sigma_{Y_i}^2}{D_q} \right) \leq W \tag{2.2}$$

The optimal estimator at the fusion center is a linear estimator [Say03], which is also the best that can be done given that the source X and received signal Y are both Gaussian.

$$\hat{X} = \frac{a_i \sigma_X^2 Y}{a_i^2 \sigma_X^2 + \sigma_Z^2 + D_q} \tag{2.3}$$

$$D(a_i) = \mathrm{E}_{X,Z_i,Z_q}[(\hat{X} - X)^2] = \frac{\sigma_X^2(\sigma_Z^2 + D_q)}{a_i^2 \sigma_X^2 + \sigma_Z^2 + D_q}$$

As the data from the sensor that is closest to the source is used, $r_i = R_{\min}$, the distortion is computed by taking the average with respect to $R_{\min}$.

$$D(n) = \mathrm{E}_{R_{\min}} \left[\frac{\sigma_X^2(\sigma_Z^2 + D_q)}{a_i^2 \sigma_X^2 + \sigma_Z^2 + D_q} \right]. \tag{2.4}$$

with a communication cost of $R_g = R$, Eq. (2.2), where R_g indicates the global transmission rate between the sensor and fusion center. This is distinguished from R_l, the rate among sensors and local fusion centers with bounded transmission range. Note that it is usually desirable to make Z_q and Z_i about the same. Therefore, we can select

$$R \approx \frac{1}{2} \log \left(\frac{\sigma_Y^2}{\sigma_Z^2} \right)$$

2.2.4 Local fusion to reduce sensing error

When the error due to sensing is much larger than the achievable quantization error D_q, more than one sensor's observations can be used to estimate X such

that a lower distortion bound is achieved. Consider N_m sensors that are in the vicinity of the source. Each sensor has an observation of the source. Instead of sending all the observations to the global fusion center, we obtain a refined estimate locally, and transmit this value. The resulting estimate, mean square error and transmission cost are as follows assuming that the local fusion center knows the a_i's of the surrounding sensors.

$$\hat{X} = \frac{\sigma_X^2 \sum_{i=1}^{N_m} a_i (Y_i + Z_{li})}{D_l + \sigma_Z^2 + \sigma_X^2 \sum_{i=1}^{N_m} a_i^2} \tag{2.5}$$

$$D = \frac{(D_l + \sigma_Z^2)\sigma_X^2}{D_l + \sigma_Z^2 + \sigma_X^2 \sum_{i=1}^{N_m} a_i^2} + D_g \tag{2.6}$$

$$R_g = \frac{1}{2} \log \left(\frac{\sigma_{\hat{X}}^2}{D_g} \right) \tag{2.7}$$

$$R_l = \sum_{i=1}^{N_m} \frac{1}{2} \log \left(\frac{\sigma_{Y_i}^2}{D_l} \right) \tag{2.8}$$

R_g is the rate required to transmit to the global fusion center with distortion constraint D_g, and R_l is the total rate for transmitting to the local fusion center with distortion constraint D_l. Also, we assumed every Y_i is corrupted by the quantization noise Z_{li}, which is zero mean and has variance D_l. The average D here is related to the distances between the source and the N_m neighboring sensors. In principle, we can find the probability density function of all r_i's and evaluate the mean distortion.

The local transmission cost can be reduced by noticing that the observations at separate sensors are correlated (Section IV, [PP03]). This is hence a problem of rate distortion coding with side information [WZ76]. Specifically, for a Gaussian source [Ooh97], the following local transmission rate is achievable assuming the first node serves as the local fusion center.

$$R_l = \sum_{i=2}^{N_m} \frac{1}{2} \log \left[\frac{\sigma_{Y_i}^2}{D} (1 - \rho_i^2) \right] \tag{2.9}$$

$$\rho_i = \frac{a_1 a_i \sigma_X^2}{\sqrt{(a_1^2\sigma_X^2 + \sigma_Z^2)(a_i^2\sigma_X^2 + \sigma_Z^2)}}$$

Note that the rate can be further reduced, if received information from other sensors is used as side information as well. As an example, consider $\sigma_X^2 = 1$, $\sigma_Z^2 = 0.003162$, and $a_1 = a_i = 0.1$. This results in $\rho_i = 0.756$ and an approximate rate reduction of 1.231 bits per sample.

2.2.5 Simulations

Considering a single point source appearing in a unit disc, we numerically evaluate Equation (2.4) to show the dependence of optimal distortion on the size of the sensor network and quantization rate. In addition, a practical scheme based on an optimal scalar quantizer [Max60] is simulated, and the resulting distortion is compared to the optimal distortion. In practice, since it is impossible to make the quantization noise behave like $\mathcal{N}(0, D_q)$, we choose to obtain a local estimate of the source at the sensor, then send it to the global fusion center. The result is shown in Figure 2.1. It can be seen that in a relatively sparse sensor network ($n \leq 50$), the sensing error is the major contribution to the total distortion. Hence the increase in sensing accuracy (by deploying more sensors) leads to a significant drop in distortion. As the sensor network gets denser, quantization error due to insufficient rate starts to dominate, and a rate increase affects the distortion more. Besides, as the quantization rate increases the distortion gap between Max and optimal quantizers diminishes.

Fig. 2.2 compares the estimation distortion when single and multiple observations are used to compute $\hat{X}$. As shown, the distortion can be effectively reduced by using multiple independent observations. It is also obvious that the observations from the initial few additional sensors provides the most gain, and the rate of distortion reduction reduces as the number of observations increases.

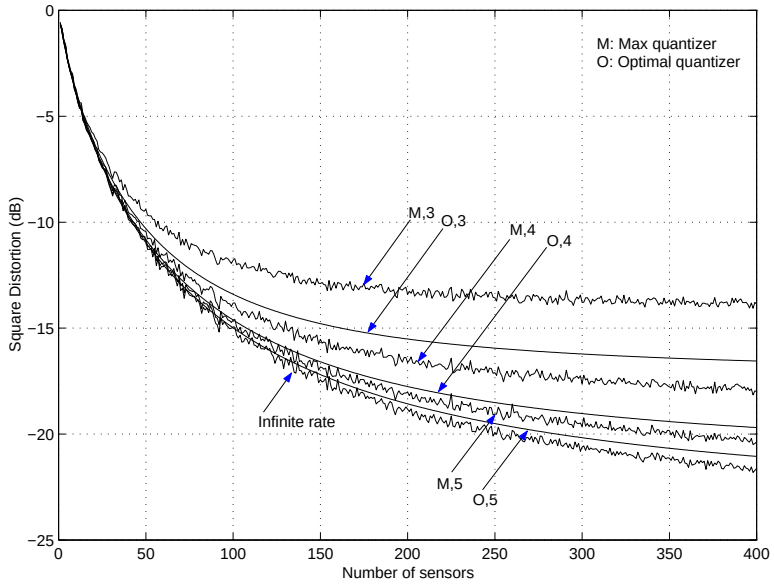


Figure 2.1: Comparison of the distortion of a Max quantizer and the optimal quantizer: $\sigma_X^2 = 1$, $\sigma_Z^2 = 0.0032$, $\kappa = 500$. Labels are interpreted as follows. "M/O, r": Max/optimal quantizer with rate r bits per sample; "Infinite rate": qantizer with infinite rate.

2.3 Distributed Sources: Transmitting Data Across the Network

We now look at the distortion due to quantization when transmitting to the fusion center the sensor observations made for a distributed source. When observing a single point source, the sensing error often dominates since a relatively small amount of information needs to be transmitted to the global fusion center. However, there is a great amount of information embedded in a distributed source. Thus a compromise often needs to be made between the reduction of quantization error and the constrained network resources. On the other hand, due to the spatial correlation embedded in a distributed source, it is possible to

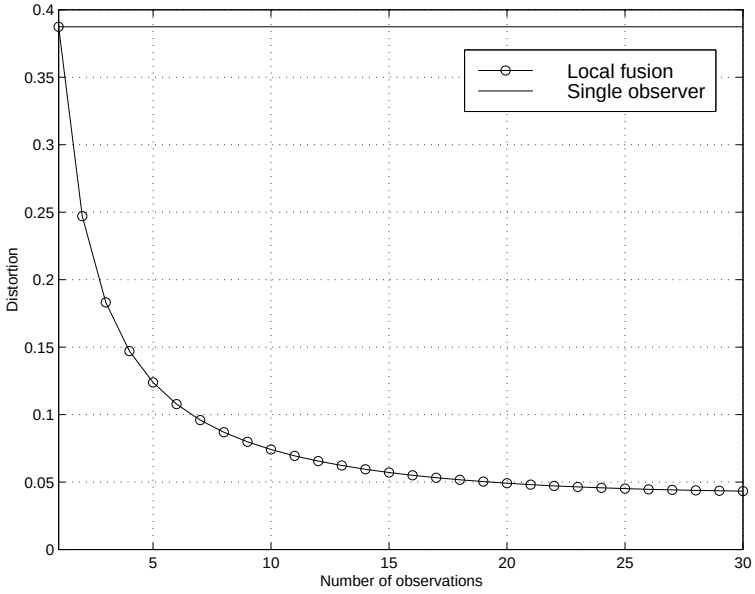


Figure 2.2: Quantization error using single and multiple observations: $\sigma_X^2 = 1$, $\sigma_Z^2 = 0.003162$, $D_l = D_g = \sigma_Z^2$, $a_i = [0.1 - 0.001(i-1)]$.

significantly reduce the rate at which sensors transmit to the global fusion center, instead of transmitting at the raw information rate accumulated at sensors. Usually attaining this reduction entails intensive local interactions among sensors. This kind of tradeoff is desirable because the energy or capacity constraint on global transmissions (between sensors and global fusion center) is far more severe than local transmissions (among sensors and local fusion centers), whose range is bounded [PP03]. The minimum rate required for distributed sources under certain error constraints, i.e. the rate-distortion problem, has received a fair amount of research focus in the image processing literature [Sak71] [Dav72]. A number of results can be borrowed from there. However, we should point out that the minimum rates are harder to achieve in sensor networks because of their distributed nature. That is, the encoder at each sensor compress its data independently without knowing what happens at other sensors.

In engineering practice, real continuous distributed sources are never fully observable. The data streams collected by sensors and processed by digital computers are always discrete, and comprise a finite collection of points. In this section, we consider the problem of coding N correlated point source observations, which can be considered as the individual observations of a distributed source made by N sensors, given certain distortion constraints.

2.3.1 Compressing correlated point sources

Consider N point sources $\mathbf{X} = \{X_i, i = 1, 2, \ldots, N\}$ in the space. Each source is monitored by a sensor, and the measurement at sensor i is given by $Y_i = a_i X_i + Z_i$, which is the attenuated signal plus noise. Collectively, we have $\mathbf{Y} = \{Y_i, i = 1, 2, \ldots, N\}$. At discrete times, the measurements $Y_i^1, Y_i^2, Y_i^3 \ldots$ at each sensor are zero-mean i.i.d. random variables. The samples $Y_1^n, Y_2^n, \ldots, Y_N^n$ measured at the same time are correlated with covariance matrix $\mathbf{Q}_N$, whose positive eigenvalues are given by $\lambda_1, \lambda_2, \ldots, \lambda_N$. We assume nothing else is known of the probability distribution of $\mathbf{Y}$. Therefore, we are facing the problem of jointly coding the i.i.d. blocks of N samples generated from a class of random variables with the distortion requirement

$$\mathrm{E}[d(\mathbf{Y}, \hat{\mathbf{Y}})] \leq D_q. \tag{2.10}$$

For the distortion measure defined as $d(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$, it can be shown that the superior of the minimum rate of this class of random variables is attained when $\mathbf{Y}$ is Gaussian [Sak70] [Tel99]. Thus, to be able to code this whole class of random variables and satisfy the given constraint, the minimum rate required is [Dav72] [Sak70]:

$$R_g = \sum_{i=1}^{N} \frac{1}{2} \log \left(\frac{\lambda_i}{\min[\lambda_i, D^*]} \right); \tag{2.11}$$

$$D_q = \frac{1}{N}\sum_{i=k}^{N} \min(\lambda_i, D^*) \tag{2.12}$$

in which D_q is the distortion bound in Eq. (2.10). The value of D^* needs to be appropriately set according to Eq. (2.11) and (2.12). Consider the case when the distortion is small such that $D_q \leq \min_{i=1,\cdots,N} \lambda_i$. Define $\mathbf{D}_N = D_q\mathbf{I}$. where $\mathbf{I}$ is an N by N identity matrix. This is to impose the same quantization distortion constraint at all sensors.

$$R_g = \sum_{i=1}^{N} \frac{1}{2}\log\frac{\lambda_i}{D_q} = \frac{1}{2}\log\frac{\mid \mathbf{Q}_N \mid}{\mid \mathbf{D}_N \mid} \tag{2.13}$$

where

$$\mathbf{Q}_N = \mathrm{E}\left(\mathbf{Y}\mathbf{Y}^t\right), \quad \mathbf{Y} = \begin{bmatrix} a_1X_1 + Z_1 \\ a_2X_2 + Z_2 \\ \vdots \\ a_NX_N + Z_N \end{bmatrix}.$$

2.3.2 One dimensional Brownian field

As a special case, we consider a one dimensional Brownian field $\mathbf{X}_u(k)$ defined on $u \in [0,1]$, using the same setting as in [Ser02]. For fixed k, $\mathbf{X}_u(k)$ is a Brownian motion with σ_X^2; for fixed u, $\mathbf{X}_u(k)$, $k = 1, 2, \cdots$, are i.i.d. Gaussian random variables $\sim \mathcal{N}(0, \sigma_X^2 u)$. The N measuring points are uniformly placed on $[0,1]$, and the attenuation factor is assumed to be unity.

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} + \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{bmatrix} = \mathbf{H}\mathbf{W} + \mathbf{Z}.$$

$$w_i \sim \mathcal{N}(0, \frac{\sigma_X^2}{N}), \quad Z_i \sim \mathcal{N}(0, \sigma_Z^2), \quad i = 1, 2, \cdots, N$$

$$\mathbf{Q}_N = \frac{\sigma_X^2}{N}\mathbf{H}\mathbf{H}^t + \sigma_Z^2\mathbf{I}$$

In this case, if we require $D_q \leq \sigma_Z^2$, which means that the quantization error is no more than the noise variance, we have $D_q < \lambda_i$. The determinant of the covariance matrix can be evaluated to be the following.

$$|\mathbf{Q}_N| = \left(\frac{\sigma_X^2}{N}\right)^N \left(Ar_1^N + Br_2^N\right) \quad \text{for } N \geq 1. \tag{2.14}$$

where

$$A = \frac{1}{r_1}\left(\frac{1+\nu}{2} + \frac{1+3\nu}{2\sqrt{1+4\nu}}\right), \quad B = \frac{1+\nu-A}{r_2},$$
$$r_{1,2} = \frac{(1+2\nu) \pm \sqrt{1+4\nu}}{2}, \quad \nu = \frac{N}{\sigma_X^2/\sigma_Z^2}.$$

So the minimum distortion due to quantization is:

$$D_q = \frac{\sigma_X^2}{N}\left(Ar_1^N + Br_2^N\right)^{1/N} 2^{-2R_g/N} \tag{2.15}$$

To achieve $D_q \leq \sigma_Z^2$, the rate R_g needs to grow at least linearly with N, the number of measurements. On the other hand, N should be appropriately chosen according to the sensing noise level σ_Z^2. R_g's linear growth with N is expected because at each sensor, at least one bit is required to transmit the noise Z_i, given $D_q \leq Z_i$. If such R_g is not available, appropriate D_q should be designed to satisfy the capacity constraint in Eq. (2.11).

2.3.3 A two dimensional isotropic random field

The rate-distortion function is evaluated for a two dimensional isotropic random field with correlation function $e^{-|r|/d_c}$, where d_c is the coherence distance. Nine sensors are placed on a square grid, and s is defined as depicted in Fig. 2.3. The minimum rates required for transmitting all the data collected at these nine sensors to the global fusion center are plotted against the distortion constraint

in Fig. 2.4. It can be seen that as the distortion requirement loosens, the data rate drops, which is expected. In addition, when sensors become closer to each other (with s decreasing), the correlation among sensors increases. As a result, the rate at which data are transmitted to the global fusion center decreases. This rate reduction is accomplished by local fusion.

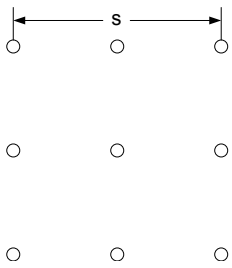


Figure 2.3: Nine sensors in a random field.

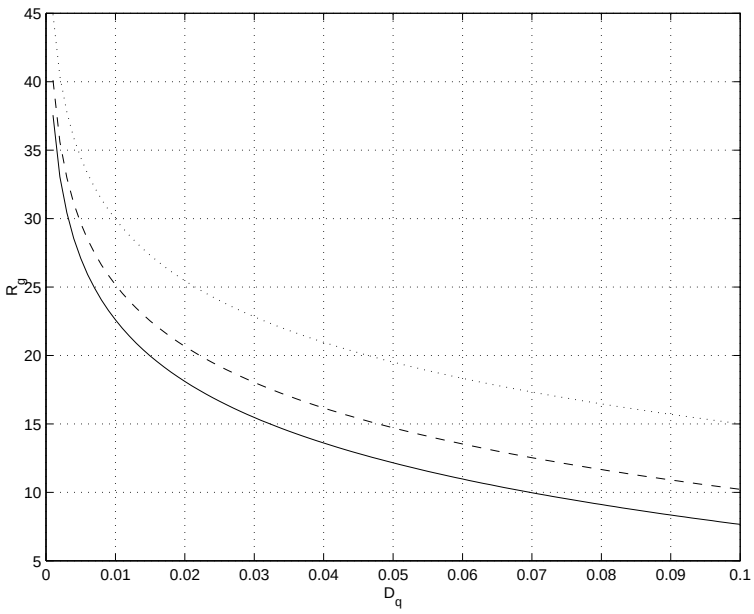


Figure 2.4: The rate-distortion functions under different coherence distance: $a_i = 1$, $\sigma_X^2 = 1, \sigma_Z^2 = 0.01$. Solid line: $s/d_c = 0.4$; dashed line: $s/d_c = 0.7$; dotted line: $s/d_c = \infty$.

2.4 Distributed Source: Reconstruction Using Cubic Spline

2.4.1 Overview

Assume that the distributed source in space to be observed is continuous or at least piecewise continuous (for the latter, we consider one continuous piece of the sample path). We consider using splines to fit the source based on the observations at discrete points. While it is generally assumed that the measurements at prescribed points have no error, and the distortion is all due to the interpolation process, in this section, we examine how sensing and quantization errors at these sampling points will affect the outcome of interpolation.

By reconstructing the source using splines, we implicitly assume that the source is deterministic. However, when seeking the limits of source coding rate in the previous section, we considered the source to be a random process. The reconciliation of these two viewpoints is achieved as follows. First, minimum source rates are obtained only by considering jointly coding long blocks of i.i.d. realizations of sources, while source reconstruction is performed for one particular realization of the source. Second, evaluating the minimum rate demands certain knowledge about the distribution of the source field. However spline fitting is able to take advantage of the correlation among local observations embedded in the continuity of the source. Third, information embedded in a distributed source is never completely transmitted to the fusion center. For a continuous sample path, spline fitting makes reasonable estimation on the missing data providing that the samples are closely spaced.

The derivation of this section is based mostly on the cubic spline theory presented in [Ahl67]. Two types of distortion measure are considered here: $d(X, \hat{X}) = |X - \hat{X}|$ and $d(X, \hat{X}) = (X - \hat{X})^2$. For simplicity, we first con-

sider the cubic spline in a one-dimensional setting. The result is then extended to two-dimensional reconstruction.

2.4.2 Cubic spline fitting

We first consider a one dimensional cubic spline. Given the locations of $(N+1)$ sampling points and the set of associated ordinates

$$\Delta: \quad a = x_0 < x_1 < \cdots < x_N = b.$$

$$Y: \quad y_0, y_1, \cdots, y_N.$$

the spline function on $[x_{j-1}, x_j]$, $(j = 1, 2, \ldots, N)$ is defined as follows

$$\begin{aligned} S_\Delta =& M_{j-1}\frac{(x_j - x)^3}{6h_j} + M_j\frac{(x - x_j)^3}{6h_j} + \left(y_{j-1} - \frac{M_{j-1}h_j^2}{6}\right)\frac{x_j - x}{h_j} \\ &+ \left(y_j - \frac{M_j h_j^2}{6}\right)\frac{x - x_{j-1}}{h_j} \end{aligned} \tag{2.16}$$

in which $h_j = x_j - x_{j-1}$. $M_j = S''_\Delta(x_j)$ are the moments of the spline, and they satisfy the following set of equations [Ahl67]:

$$\begin{bmatrix} 2 & \lambda_0 & 0 & \ldots & 0 \\ \mu_1 & 2 & \lambda_1 & \ldots & 0 \\ 0 & \mu_2 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_N \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \tag{2.17}$$

2.4.3 Approximation error

We assume that the curve to be fitted belongs to $C^n[a, b]$, $n = 0, 1, 2$ and 3, i.e. having nth continuous derivative (by 0, we mean the curve is continuous), and satisfies Hölder's condition to the order of α $(0 < \alpha \leq 1)$. Given converging

meshes Δ_k ($\lim_{k\to\infty} \parallel \Delta_k \parallel = 0$) and appropriate end conditions, the interpolation error uniformly converges with respect to x in $[a, b]$ as follows (Theorem 2.3.1, 2, 3 and 4 [Ahl67]):

$$e_1 = |f(x) - S_\Delta(x)| \leq K_1 \parallel \Delta_k \parallel^{n+\alpha}, \quad \text{for some constant } K_1$$

However, the spline reconstructed at the fusion center is not $S_\Delta(x)$ but a shifted spline $S^e_\Delta(x)$ due to the sensing and quantization error at the sampling points. It remains to show how much the interpolation deteriorates given that the noise at prescribed points is bounded by: $\mathrm{E}|\delta y_i| \leq e_{sq}$ and $\mathrm{E}(\delta y_i)^2 \leq D_{sq}$. We consider the absolute error first.

$$e = \mathrm{E}|f(x) - S^e_\Delta(x)| \leq |f(x) - S_\Delta(x)| + \mathrm{E}|S_\Delta(x) - S^e_\Delta(x)| = e_1 + e_2 \tag{2.18}$$

Note that since both $f(x)$ and $S_\Delta(x)$ are considered deterministic, the mean operation disappears for e_1. As for e_2, we have the following:

$$e_2 = \mathrm{E}\left|\sum_{i=0}^{N} \frac{\partial S_\Delta}{\partial y_i}\, \delta y_i\right| \leq e_{sq} \sum_{i=0}^{N} \left|\frac{\partial S_\Delta}{\partial y_i}\right| \tag{2.19}$$

In Appendix B, it is shown that for proper end conditions $\lambda_0, \mu_N < 2$ and evenly distributed meshes, which means that h_j/h_{j+1}, $j = 1, \cdots, N-1$ are bounded, the following relation holds.

$$\beta = \sum_{i=0}^{N} \left|\frac{\partial S_\Delta}{\partial y_i}\right| \leq K_2, \quad \text{for some finite number } K_2$$

Hence the total absolute error is bounded by

$$e \leq e_1 + \beta e_{sq} \tag{2.20}$$

Next we consider the mean square error. Since data is locally fused before it is transmitted to the global fusion center, the correlation between the noise δy_i at different sensors is not necessarily zero, but it is bounded by the following:

$$\mathrm{E}(\delta y_i \delta y_j) \leq \sqrt{E(\delta y_i)^2 E(\delta y_j)^2} \leq D_{sq}$$

We first find a bound on the mean square error between the original and noise corrupted splines with respect to $x \in [a, b]$.

$$\begin{aligned}
\mathrm{E}\left[S_\Delta(x) - S^e_\Delta(x)\right]^2 &= \mathrm{E}\left(\sum_i \frac{\partial S_\Delta}{\partial y_i}\,\delta y_i\right)^2 \\
&= \sum_i \left(\frac{\partial S_\Delta}{\partial y_i}\right)^2 \mathrm{E}(\delta y_i)^2 + 2\sum_{i\neq j}\left(\frac{\partial S_\Delta}{\partial y_i}\frac{\partial S_\Delta}{\partial y_j}\right)\mathrm{E}\delta y_i \delta y_j \\
&\leq \left(\sum_i \frac{\partial S_\Delta}{\partial y_i}\right)^2 D_{sq} \\
&\leq \beta^2 D_{sq}
\end{aligned}$$

Now, we evaluate the total mean square error.

$$\begin{aligned}
D &= \mathrm{E}\left[(f(x) - S_\Delta(x)) + (S_\Delta(x) - S^e_\Delta(x))\right]^2 \\
&= \left[f(x) - S_\Delta(x)\right]^2 + 2\left[f(x) - S_\Delta(x)\right]\mathrm{E}\left[S_\Delta(x) - S^e_\Delta(x)\right] \\
&\quad + \mathrm{E}\left[S_\Delta(x) - S^e_\Delta(x)\right]^2 \\
&\leq e_1^2 + 2\beta e_1 e_{sq} + \beta^2 D_{sq}
\end{aligned}$$

2.4.4 Simulation and extension

In the following simulation, we use a cubic spline to fit a sinusoid function based on noise corrupted data. Figure 2.5 shows error e_2 plotted against the noise due to sensing and quantization e_{sq}. The relation in this case appears to be approximately linear. In the simulation, we have kept mesh size relatively small so that $e \approx e_2$.

This result is readily extensible to a two-dimensional doubly cubic spline defined on a rectangular grid, $\Delta_t : a = t_0 < t_1 < \cdots < t_N = b$, $\Delta_s : c = s_0 < s_1 < \cdots < s_M = d$, noticing that a doubly cubic spline can be obtained by partial splines on t and s (p. 238 [Ahl67]). The resulting error after twice

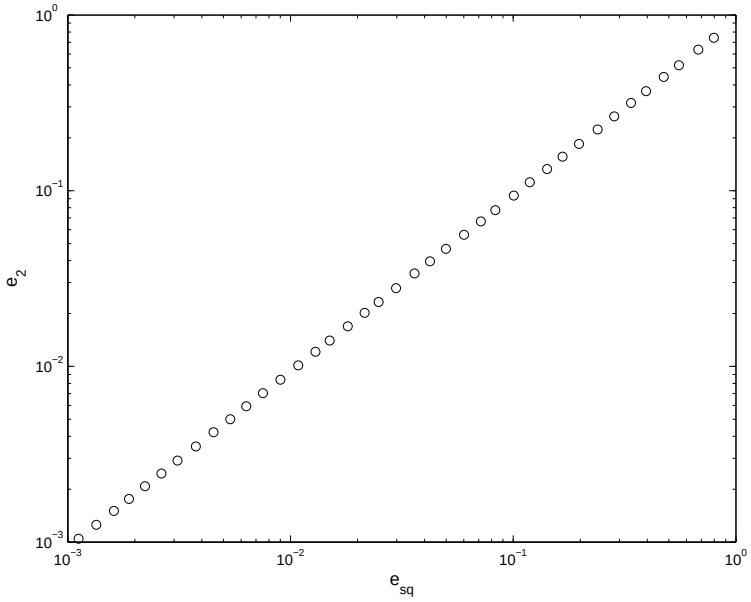


Figure 2.5: e_2 plotted against sensing and quantizing error e_{sq}.

one-dimensional interpolation is thus bounded by:

$$e \leq e_{1s} + \beta_s(e_{1t} + \beta_t e_{sq}) \tag{2.21}$$

where, $\beta_i, e_{1i}, (i = s, t)$ are the corresponding parameters on s and t coordinates. A similar derivation applies to the mean square error.

2.5 Conclusion

In this chapter, we discussed the distortions due to sensing, quantization and interpolation in sensor networks by modelling distributed phenomena as correlated point sources.

Sensing error is determined by measurement noise and signal attenuation, which can be reduced by improving sensor coverage and increasing the number of observations. Achieving a reasonable coverage for a large sensing field often

incurs a enormous number of sensors. An increasingly popular practice is to employ mobile sensors. Such sensors can move in their patrol area and take measurements at specified locations, which enable them to move close to the subject of interest and take high SNR (signal noise ratio) measurements.

When observing a distributed source, network capacity may become severely strained due to the high raw information rate. In this case, local cooperation and fusion based on correlations among nearby observations can be used to bring down the communication rate, which is bounded by the rate-distortion theory according to appropriate distortion requirements. Despite its importance in resource conservation, achieving a high level of data compression in sensor networks is far more challenging than in image and video processing owing to the distributed nature of the problem.

Cubic splines are used as the algorithm to reconstruct distributed and continuous sources. The total reconstruction error was found to converge at least on the same order of sensing and quantization error given appropriate mesh sizes.

Thus, the scalability of distributed sensor networks where certain physical phenomena are observed and reconstructed is closely tied to the fidelity constraints demanded by the system. Furthermore, correlation among data streams produced at different sensors offers the opportunity of local processing, which can greatly reduce power/bandwidth consuming long-range data transmissions.

CHAPTER 3

Adaptively Sampling Distributed Fields with Mobile Sensors

3.1 Introduction

In this chapter, we consider the problem of sampling distributed phenomena. The introduction of mobile sensors, for example the NIMS (Networked InfoMechanical System) [KPS04b], has generated an interesting set of new problems in wireless sensor networks. Advantages of the mobility have been explored in, for example communication [GT02], localization [SH03], security [CHB03], and system reliability [KKP05]. As we discussed in the previous chapter, it is difficult to cover a relatively large region using static sensors. However, when equipped with mobility, a few sensors can carry out the task of collecting measurements within their patrol area, which would otherwise require a large number of static sensors. Moreover, mobile sensors are especially suitable for observing heterogeneous and slowly time-varying phenomena owing to their ability to sample the field at adjustable and arbitrary spatial density.

A typical NIMS setup in a natural environment is illustrated in Fig. 3.1 [RPK04]. The sensor node can move vertically and horizontally to prescribed locations within its patrol area to collect samples. Three-dimensional mobility is possible by devising more sophisticated cableways. In such an ecological system,

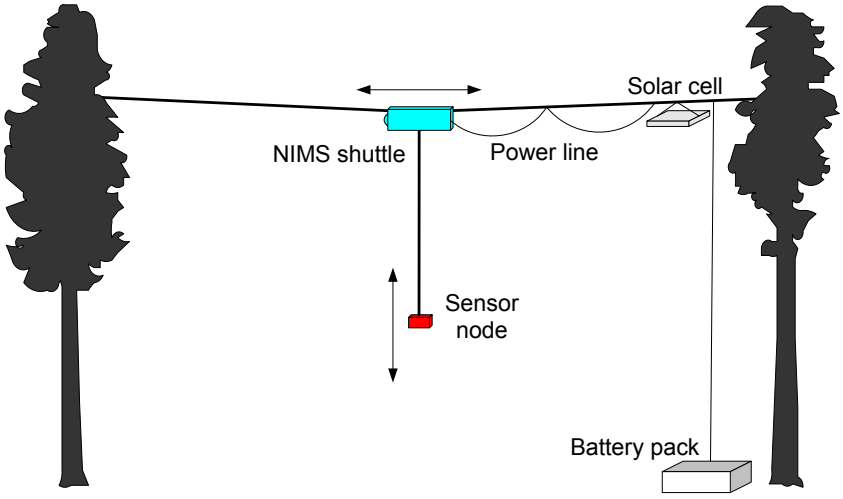


Figure 3.1: A NIMS platform in a natural environment.

several parameters, such as temperature, humidity, and sunlight illumination, are of interest. Here, we concentrate on measuring sunlight intensity and reconstructing its two-dimensional field subject to a fidelity constraint. Besides offering an interesting realization of a multi-dimensional statistical field, the distribution of sunlight under forest canopies plays a crucial role in plant growth, and has been extensively researched for many years [CP94]. In addition, it is relatively easy to reproduce the light distribution in a two-dimensional space, with which we can evaluate the sampling algorithm's error performance directly.

In a mobile sampling system, major energy expenditure often results from sensor movement and sample collection. Hence, the total number of measurements provides a first-order indication of the resource that the system consumes. On the other hand, when the field varies slowly in time (for example, caused by the movement of the sun) the total number of samples that can be collected for estimating a field snapshot is restricted by some critical time constant. Therefore, in addition to meeting the fidelity constraint, the need to build a low-cost and fast responding system leads to the goal of minimizing the total number of

samples the sensor needs to collect for a certain task.

Traditional problems in signal processing often begin with a panorama of the field, and then proceed to compress the complete set of data without violating the distortion requirement. For instance, image compression and the approach adopted in [NMW04] fall in this category. Here, the process is reversed. Unless the source is exhaustively sampled, which is prohibitively expensive in most cases, we generally possess only partial knowledge about the true field. Therefore, it is more appropriate to take a statistical approach and consider the probability of satisfying the fidelity constraint given the incomplete information at hand. New levels of confidence on whether the fidelity goal has been met can be gained by collecting more samples. Herein lies the fundamental compromise between our confidence of achieving the fidelity goal and our willingness to consume more resource. Furthermore, the heterogeneous nature of the field presents us the challenge of wisely allocating our limited resource (sampling sites) such that a high confidence level can be reached as efficiently as possible.

Besides being used for mobile sensors, our adaptive algorithm can also be applied to distributed networks where sensor nodes are static. In such a network, the sensors are initially set at standby mode. As the algorithm executes, appropriate sensors are woken up to take measurements at the most desirable locations. The system resource is preserved by waking up as few sensors as possible to carry out the sensing task. In contrast to the case in the mobile setting, the algorithm designed for a static network must be suitable for distributed implementation. During the development of our adaptive scheme in this chapter, we will point out at appropriate places what modifications are needed such that the algorithm can be used in such a distributed fashion.

The ecological and environmental importance of solar energy transfer through

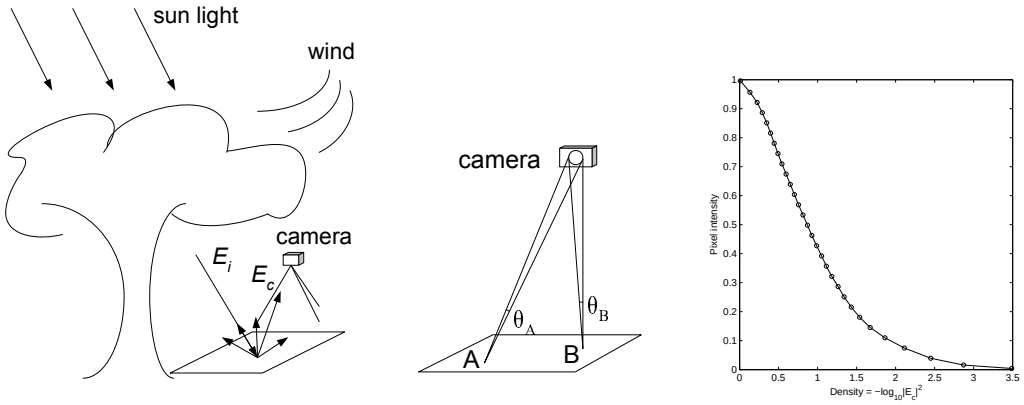
forest canopies has prompted extensive studies on the subject [MRA89] [VP99]. Many papers are devoted to measuring and modelling the distribution of solar radiation under various canopies [CP94] [RSS98]. Statistical approaches have been instrumental in the characterization of spatial variation [Cre93] and interpolation of sampled data [MM02]. In a sense, the sequential sampling process can be viewed as an optimal experimental design [Fed72], in which the input variables (sampling locations) of a series of tests (measurements) are purposefully adjusted such that the system response (the field to be reconstructed) is observed efficiently. [Raf86] explores this idea in choosing optimal trajectories of moving sensors based on the Fisher information matrix. However, in this chapter we have adopted a Bayesian approach. A sequential method for estimating discontinuities in curves and surfaces is discussed in [HM03]. In sensor network community, [BRY04] and [RPK04] represent preliminary efforts on using mobile sensors to adaptively sample distributed phenomena.

The rest of the chapter is organized as follows. Section 3.2 describes the experimental setup that we use to record the two-dimensional sunlight field. Section 3.3 presents the adaptive sampling algorithm in detail. Simulation results are displayed in section 3.4. Section 3.5 concludes the chapter.

3.2 Experimental Setup

Obtaining a complete and accurate account of a two-dimensional sunlight field using light sensors is very difficult due to the enormous number of sampling points involved. In our experiment, we use a camera to capture snapshots of the instantaneous light field, then convert pixel intensity to incident light intensity.

The experimental setup is depicted in Fig. 3.2(a). A flat screen (approxi-



(a) Experimental setup (b) Path loss (c) Characteristic curve

Figure 3.2: Experimental setup.

mately 100cm×80cm) is placed on the ground. The screen has roughly the same reflectance ρ over its surface, and provides good scatter reflection. The incident light $E_i(x, y)$ impinges on the small patch at (x, y), and only $\rho E_i(x, y)$ is reflected to the half space. After suffering some path loss as depicted in Fig. 3.2(b), $E_c(x, y)$ enters camera lens, and is recorded as the pixel intensity $I_p(x, y)$:

$$I_p(x, y) = f_c(|E_c|^2) = f_c[g_p(\rho^2|E_i(x, y)|^2, \theta(x, y))] \tag{3.1}$$

where $f_c(\cdot)$ is the camera characteristic, $g_p(\cdot)$ accounts for the path loss sustained by the reflected light, and θ is the solid angle depicted in Fig. 3.2(b). The characteristic curve of our camera, shown in Fig. 3.2(c), is obtained with the help of a Kodak gray scale. The experimental configuration is recreated in the lab, and a picture is taken while the screen is placed under uniform illumination. This image is divided by the images taken under the forest canopy to compensate for the path loss and nonuniform reflectance of the screen. Experiments were conducted at the UCLA Sunset Canyon, where the canopy consists of a mixture of conifers and broadleaves. Note that this is not a practical way of accurately measuring sunlight distribution in the wild since 1) the natural environment in

a forest generally does not provide a homogeneous reflecting surface; 2) The camera's spectral response to solar radiation may be different from what we desire. Nevertheless, we consider these recorded fields good enough for testing our algorithms.

As far as evaluating the performance of adaptive algorithms is concerned, there is no need to convert $I_p(x, y)$ to $|E_i(x, y)|^2$. The algorithm can operate as if $I_p(x, y)$ is some real distributed phenomenon. However, working on $|E_i(x, y)|^2$ makes the source statistical model that is to be developed reusable and helps to simplify the field implementation of algorithms in the future.

The sunlight field in our experiments was discovered to be fairly volatile even under mild wind conditions, which makes it less meaningful to estimate the instantaneous field using mobile sensors. Instead, we attempt to reconstruct the mean field averaged over a short period of time ($5 \sim 15$ minutes). In practice, this requires that mobile sensors take multiple readings at each data site to obtain a sample of the mean field. Fortunately, in many applications the average light intensity received by a small patch of space is more important than the instantaneous value. Together with the penumbra, this averaging process tends to create a fairly smooth field. Fig. 3.3 displays two sunlight fields captured in our experiments. Our ensuing models and simulations are based on these mean fields reproduced in the computer.

3.3 Adaptive Sampling Algorithm

Fig. 3.4 is the block diagram of our adaptive sampling algorithm. It runs in iterations. A pool of sampling candidates is maintained and updated each time a new sampling point is picked. At the beginning of each iteration, a number of data

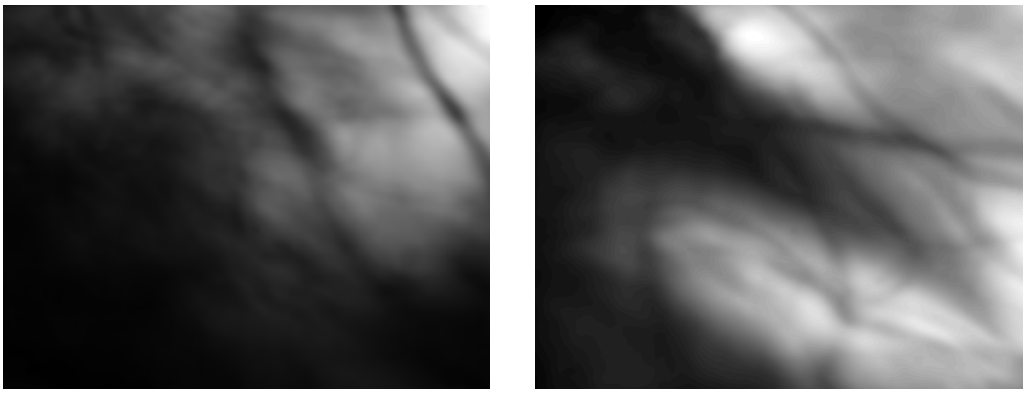


Figure 3.3: Two sunlight fields captured in our experiments.

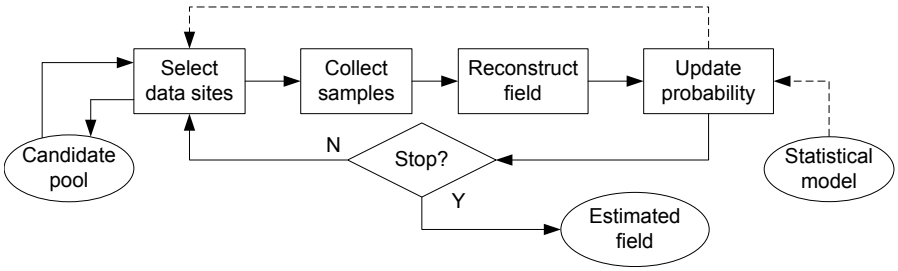


Figure 3.4: The block diagram of the adaptive sampling algorithm.

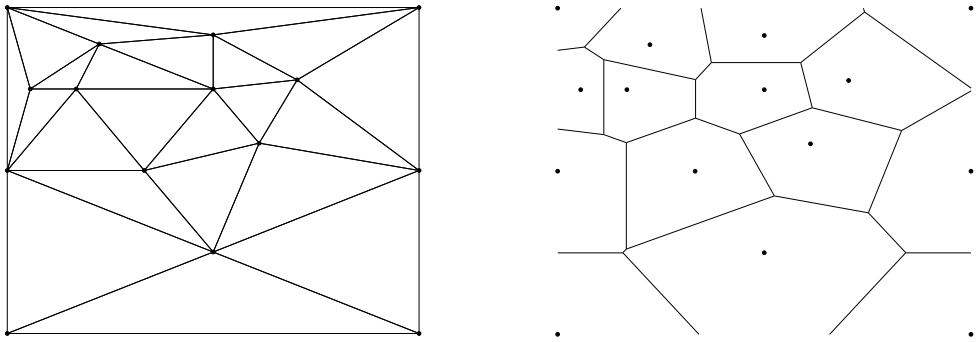
sites are selected from the pool based on a MAP criterion. The sensor then moves to collect measurements at these locations, and reconstructs the field. Based on these latest samples, the probability of the newly reconstructed field satisfying the fidelity constraint is evaluated. The algorithm iterates until the confidence of meeting the distortion requirement is high enough or sufficient samples have been taken. The rest of this section describes key functional blocks and the algorithm implementation in detail.

3.3.1 Sampling candidates

In a two-dimensional space, apart from a few scattered sites where measurements have been taken, there is a large number of potential sampling locations. The

complexity of optimizing over all potential sites can be rather high, so we maintain a small pool of site candidates, from which new sampling points are picked.

Given a set of points $\mathcal{S}$, a Delaunay tessellation DT($\mathcal{S}$) is obtained by connecting any two points $p, q \in \mathcal{S}$ with a line segment if there exists a circle that passes through p, q and contains no other sites of $\mathcal{S}$. We call the edges of DT($\mathcal{S}$) Delaunay edges. DT($\mathcal{S}$) is the graph-theoretic dual of V($\mathcal{S}$), the Voronoi diagram with respect to $\mathcal{S}$, in that two points of $\mathcal{S}$ are connected by a Delaunay edge if and only if their Voronoi regions are edge-adjacent [SU00]. As an example, Fig. 3.5 shows the DT($\mathcal{S}$) and V($\mathcal{S}$) of a set of data sites.



(a) Delaunay tessellation (b) Voronoi diagram

Figure 3.5: Delaunay and Voronoi cells.

Denoting by $\mathcal{S}_k$ the set of existing sampling sites at iteration k and n_k the size of $\mathcal{S}_k$, we construct DT($\mathcal{S}_k$), and use as site candidates the centers of Delaunay cells' circumcircles. Our choice of sampling candidates is justified by some of their nice properties. For detailed derivations and other appealing properties, readers are referred to, for example, [SU00].

First, with little knowledge about the true field in the gaps between existing samples, the minimum distances from candidates to existing sampling sites should be as large as possible to yield maximum information based on the maximin

design [JMY90]. If we consider the circumcircles of Delaunay cells the largest circles that can be fitted in such gaps, our scheme follows this principle by placing potential sites at the centers of these sampling gaps.

The size of the candidate pool strongly affects the complexity of subsequent pruning processes. The total number of Delaunay cells is no more than $(2n_k - 5)$ at iteration k. In addition, the density of potential sites in different areas follows that of existing samples. This is desirable assuming that the distribution of current samples correctly reflects the field heterogeneity.

Parallel algorithms that run in $O(n_k \log n_k)$ time and suit distributed implementation exist for finding $\mathrm{DT}(\mathcal{S}_k)$. Moreover, when a new candidate is added to an existing base, incremental schemes, that update the pool in the neighborhood of the new site, require even less computation.

For the Delaunay cell corresponding to the mth candidate in $\mathrm{DT}(\mathcal{S}_k)$, we define $\mathcal{O}_m^k$ its set of vertices, i.e. the sampling points that are closest to the mth candidate. Denoting by $\mathcal{V}_j^k$ the Voronoi cell corresponding to the jth sampling site during iteration k, then the mth candidate at iteration k is also the common vertex of Voronoi cells $\mathcal{V}_j^k$, $j \in \mathcal{O}_m^k$, which can be easily seen from Fig. 3.6.

One minor complaint about the above choice of potential data sites might be that the candidate sometimes falls outside the corresponding Delaunay cell. This adds some complexity in a distributed implementation where every data site corresponds to a sensor and each Delaunay cell acts also as a small cluster of local cooperation. An alternative is to use the centroids of Delaunay cells as potential sites. This violates the maximin design principle, but retains most of the desirable properties of the Delaunay cells. Lastly, in our experiment, simple rules as in Eq. (3.2) are employed when the potential site (x_c, y_c) falls outside the boundaries of the sensor's rectangular patrol area $x \in [x_{\min}, x_{\max}]$

and $y \in [y_{\min}, y_{\max}]$.

$$x'_c = \begin{cases} x_{\min} & \text{if } x_c < x_{\min} \\ x_{\max} & \text{if } x_c > x_{\max} \end{cases}; \qquad y'_c = \begin{cases} y_{\min} & \text{if } y_c < y_{\min} \\ y_{\max} & \text{if } y_c > y_{\max} \end{cases} \tag{3.2}$$

3.3.2 Field reconstruction

In this chapter, we ignore the measurement error, and interpolation is used to reconstruct the field. Two methods are presented. One is the thin plate spline that requires the complete data set and the other is a piece-wise linear interpolation suitable for distributed implementation.

3.3.2.1 Thin plate spline

The thin plate spline is the extension of the cubic spline to two dimensions. It is suitable for scattered data [HD72, Boo89]. Supposing the true field is $f(x, y)$, we approximate it with

$$s(x, y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{N} w_i \phi(r_i) \tag{3.3}$$

where

$$\phi(r_i) = r_i^2 \log r_i^2, \quad r_i^2 = (x - x_i)^2 + (y - y_i)^2$$

The coefficients a_0, a_1, a_2, and $w_i, i = 1, \cdots, n$ are determined by interpolating the spline at n scattered data points (x_i, y_i), $i = 1, \cdots, n$

$$s(x_i, y_i) = f(x_i, y_i), \quad i = 1, \cdots, n$$

and enforcing the equilibrium equations:

$$\sum_{i=1}^{n} w_i = 0, \quad \sum_{i=1}^{n} x_i w_i = 0, \quad \sum_{i=1}^{n} y_i w_i = 0$$

This set of linear equations can be concisely expressed in the matrix form

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{3.4}$$

where

$$\Phi_{ij} = \phi(r_{ij}), \quad r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2, \quad i, j = 1, \cdots, n$$

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}, \quad a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad F = \begin{bmatrix} f(x_1, y_1) \\ \vdots \\ f(x_n, y_n) \end{bmatrix}$$

Suppose there are a total of n_k samples after N new samples are collected at iteration k. At first glance, constructing the thin plate spline demands the inversion of a matrix of size $(n_k + 3)$, which requires $O[(n_k + 3)^3]$ computation. However, since only N new data sites are added at iteration k, a careful implementation that makes use of the results from earlier stages runs in $O[(n_k + 3)^2]$ time when $N \ll n_k$.

The bending energy defined as follows has been used by many to characterize the roughness of a two-dimensional function.

$$I(s) = \iint_{R^2} \left[\left(\frac{\partial^2 s}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 s}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 s}{\partial y^2} \right)^2 \right] dx\, dy \tag{3.5}$$

One interesting property of the thin plate spline $s(x, y)$ is that its bending energy, given as follows, is minimum among all functions that agree with $f(x, y)$ at all sampling points (x_i, y_i), $i = 1, \cdots, n$ [Pow94].

$$I(s) = \sum_{i=1}^{N} w_i s(x_i, y_i) = w \Phi w^T \tag{3.6}$$

This is a result of $s(x, y)$ being the physical shape that a weightless elastic plate takes when it is bent by point forces w_i at (x_i, y_i) [LL86]. Consequently, the

bending energy of reconstructed fields during successive iterations of our adaptive algorithm is non-decreasing if the thin plate spline is used. Moreover, this non-decreasing roughness is closely related to the thin plate spline's convergence to the true field as is observed by this result from [Pow94].

Theorem 3.3.1 (Uniform Convergence of Thin Plate Spline) *Suppose* (x, y) *is inside or on an edge of a triangle, whose vertices are any three of the interpolation points. Denoting by* h *the length of the longest side of the triangle, the error of the thin plate spline is bounded by*

$$|f(x,y) - s(x,y)| \leq h\sqrt{[I(f) - I(s)](\log 3)/(24\pi)} \tag{3.7}$$

This suggests that we densely sample the region where the roughness is high so that $I(s)$ is maximized given the total number of samples. Intuitively, this is equivalent to placing more samples at regions where the field has high spatial variation.

3.3.2.2 Piece-wise linear interpolation

Although piece-wise linear interpolation is less precise than the thin plate spline, it is suitable for distributed implementation. In this method, a Delaunay triangulation is first constructed from the Delaunay tessellation by arbitrarily splitting the non-triangle Delaunay cells into triangles [OBK00]. Each Delaunay triangle then forms an reconstruction element, and a linear function of x and y are fitted in the triangle based on the samples at triangle vertices. In this local element, only the information from sensors on the vertices of the corresponding Delaunay triangle is needed for the interpolation. Specifically, in triangle i, the interpolating function is given by:

$$z_i(x, y) = a_i x + b_i y + c_i$$

Since Delaunay triangles are used, this piece-wise linear interpolation minimizes the following roughness measure among all possible triangulations [SU00].

$$\sum_{\Delta \in T} |\Delta|(\alpha^2 + \beta^2)$$

with $|\Delta|$ being the area of triangle Δ, and α and β being the slopes of the corresponding triangle in 3-space.

3.3.3 Adaptive sample selection

3.3.3.1 MAP sample selection

When continuous analog signals are to be recorded in finite-length digital codes, certain fidelity constraints must be furnished for the problem to be tractable. Denoting by Dom the two-dimensional domain where sampling takes place, we consider following distortion requirements:

$$\max_{(x,y)\in \mathrm{Dom}} |f(x,y) - s(x,y)| \leq D_{\max} \tag{3.8}$$

$$\iint_{\mathrm{Dom}} [f(x,y) - s(x,y)]^2 \, dx \, dy \leq D_{\mathrm{ave}} \cdot \mathrm{Area(Dom)} \tag{3.9}$$

where $f(x,y)$ is the actual field, and $s(x,y)$ is the reconstructed function. At kth iteration, we impose the same error requirements on each Voronoi cell ($\mathcal{V}_i^k$, $i = 1, 2, \cdots, n_k$), and define the following events:

$$U_i^k \quad : \quad \text{Fidelity constraint is unsatisfied in } \mathcal{V}_i^k.$$

Fidelity constraints are satisfied in Dom if they are met in all Voronoi regions. Note that the inverse of this statement is not true. Hence, requiring fidelity constraints to be satisfied in all Voronoi regions imposes more stringent conditions than what Eq. (3.8) and (3.9) imply.

Denote by $s_k(x, y)$ the reconstructed field at iteration k, and ϵ a small quantity that is appropriately set according to the fidelity constraint. When a sample is collected at (x_i, y_i) during iteration k, besides being used to obtain a new approximation of the true field, this sample also reveals important information on how well $s_{k-1}(x, y)$ approximates $f(x, y)$ in the vicinity of (x_i, y_i). To formalize the idea, we say a test T_i^k is conducted at (x_i, y_i), and define

$$T_i^k = \begin{cases} G & |f(x_i, y_i) - s_{k-1}(x_i, y_i)| > \epsilon \\ L & |f(x_i, y_i) - s_{k-1}(x_i, y_i)| \leq \epsilon \end{cases}$$

Since multiple samples are often collected during one iteration, T^k generally consists of several tests at different data sites.

$$T^k = \{T_{i_1}^k, \cdots, T_{i_N}^k\}$$

in which $i_1, \cdots, i_N$ are the corresponding candidates where the tests are conducted, and N is the number of new samples collected during each iteration. Accumulating all the tests up to time k, we define:

$$Z^k = \{T^k, T^{k-1}, \cdots, T^1\}$$

At iteration k, due to insufficient knowledge about the true field, we are generally not completely sure about whether the fidelity constraint is satisfied in $\mathcal{V}_j^k$. However, we can define $P(U_j^k|Z^k)$ i.e. the probability of U_j^k given all the past tests. If we decide to continue sampling, appropriate new data sites need to be picked from the candidate pool. Since each selected sampling site will create a new Voronoi cell at the next iteration, a MAP criterion prompts us to choose the potential data site with maximum $P(U_i^{k+1}|Z^k)$ among all candidate sites, which simply indicates that the probability of cell $\mathcal{V}_i^{k+1}$ fails the fidelity requirement is the most given our partial information up to iteration k. Supposing the new cell

$\mathcal{V}_i^{k+1}$ corresponds to the candidate m at iteration k, then $\mathcal{V}_i^{k+1}$ lies mostly within $\mathcal{V}_j^k$, $j \in \mathcal{O}_m^k$, (recall that candidate m is the common vertex of $\mathcal{V}_j^k$, $j \in \mathcal{O}_m^k$). Therefore, a weighted sum is used to compute this unknown probability.

$$P(U_i^{k+1}|Z^k) = \sum_{j \in \mathcal{O}_m^k} \mu_j P(U_j^k|Z^k) \tag{3.10}$$

The weight μ_j characterizes the influence of $\mathcal{V}_j^k$ on $\mathcal{V}_i^{k+1}$, and is defined as

$$\sum_{j \in \mathcal{O}_m^k} \mu_j = 1 \quad \text{and} \quad \mu_j \propto (d_{j1} + d_{j2})/r_j \tag{3.11}$$

in which d_{j1}, d_{j2}, and r_j are defined as in Fig. 3.6. Note that r_j is the same for all vertices unless the center of the Delaunay cell falls outside Dom.

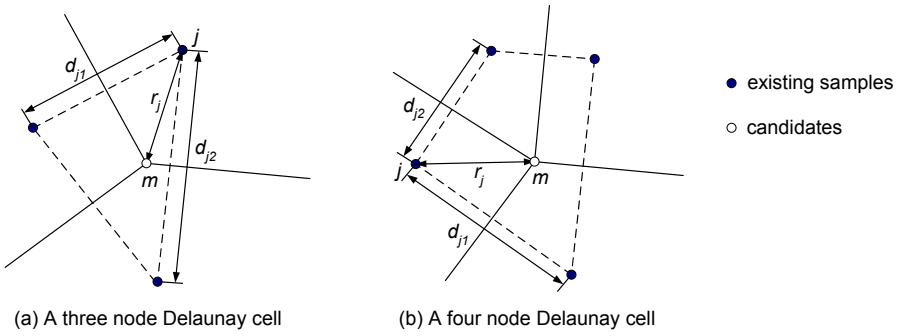


Figure 3.6: Delaunay cells are enclosed by dashed lines. Solid lines without arrows are the boundaries of Voronoi cells. $j \in \mathcal{O}_m^k$.

Once new data sits are chosen, samples are collected, and the spline is computed. Before proceeding to the next iteration, it remains to update $P(U_i^{k+1}|Z^{k+1})$ by assimilating the new information revealed from the evaluation of T^{k+1}. The importance of this procedure invites a careful analysis.

3.3.3.2 Probability update

The probability is updated as follows based on the Bayesian framework:

$$\begin{aligned} P(U_i^{k+1}|Z^{k+1}) &= P(U_i^{k+1}|T^{k+1}, Z^k) \\ &= \frac{P(U_i^{k+1}, T^{k+1}|Z^k)}{P(T^{k+1}|Z^k)} \\ &= \frac{P(T^{k+1}|Z^k, U_i^{k+1})P(U_i^{k+1}|Z^k)}{P(T^{k+1}|Z^k)} \end{aligned} \tag{3.12}$$

It is very difficult to compute the exact value of each quantity in Eq. (3.12). Instead, we design effective schemes to approximate this updating procedure.

If $\mathcal{V}_i^{k+1}$ corresponds to a newly collected sample at iteration $k+1$, $P(U_i^{k+1}|Z^k)$ computed from Eq. (3.10) is used. Otherwise, we set $P(U_i^{k+1}|Z^k) = P(U_j^k|Z^k)$, where $\mathcal{V}_i^{k+1}$ contains the same data site as $\mathcal{V}_j^k$.

Literally, $P(T^{k+1}|Z^k, U_i^{k+1})$ is the probability of T^{k+1} taking a certain set of values (Gs and Ls) given all the samples up to iteration k and the knowledge that the fidelity constraint is unsatisfied in $\mathcal{V}_i^{k+1}$. If $\mathcal{V}_i^{k+1}$ is far away from the new data sites sampled at iteration $k+1$, the status of $\mathcal{V}_i^{k+1}$ is expected to exert little influence on the outcome of T^{k+1}. Hence, $P(T^{k+1}|Z^k, U_i^{k+1}) \approx P(T^{k+1}|Z^k)$, and $P(U_i^{k+1}|Z^{k+1})$ is roughly the same as $P(U_i^{k+1}|Z^k)$. Intuitively, this means that T^{k+1} sheds little information on Voronoi cells far away from the testing sites.

Assume $\mathcal{V}_i^{k+1}$ is created by a new sample collected at candidate m during iteration $k+1$, and T_m^{k+1} is the corresponding test conducted on the site. Fixing the values of the remaining tests in T^{k+1}, we examine the effect of T_m^{k+1} on $\mathcal{V}_i^{k+1}$. For simplicity, we keep only T_m^{k+1} in writing T^{k+1}. If the error bound ϵ is properly set according to the fidelity constraint, we expect

$$P(T_m^{k+1} = G|Z^k, U_i^{k+1}) \approx 1; \quad P(T_m^{k+1} = L|Z^k, U_i^{k+1}) \approx 0.$$

Therefore, $P(U_i^{k+1}|Z^{k+1})$ increases when $T_m^{k+1} = G$, and decreases when $T_m^{k+1} =$

L. The outcome of T_m^{k+1} has similar but less significant effects on Voronoi cells inherited from $\mathcal{V}_j^k$, $j \in \mathcal{O}_m^k$, since they are adjacent to V_i^{k+1}.

In summary, for each test T_m^{k+1}, we update the probability as follows. If $\mathcal{V}_i^{k+1}$ is the new cell created at candidate site m,

$$P(U_i^{k+1}|Z^{k+1}) = \begin{cases} \kappa_g \sum_{j \in \mathcal{O}_m^k} \mu_j P(U_j^k|Z^k) & \text{if } T_m^{k+1} = G; \\ \kappa_l \sum_{j \in \mathcal{O}_m^k} \mu_j P(U_j^k|Z^k) & \text{if } T_m^{k+1} = L. \end{cases} \tag{3.13}$$

Suppose the data site of $\mathcal{V}_i^{k+1}$ is inherited from $\mathcal{V}_j^k$. If $j \in \mathcal{O}_m^k$,

$$P(U_i^{k+1}|Z^{k+1}) = \begin{cases} \kappa_g^{\mu_j} P(U_j^k|Z^k) & \text{if } T_m^{k+1} = G; \\ \kappa_l^{\mu_j} P(U_j^k|Z^k) & \text{if } T_m^{k+1} = L; \end{cases} \tag{3.14}$$

Otherwise,

$$P(U_i^{k+1}|Z^{k+1}) = P(U_j^k|Z^k) \tag{3.15}$$

where μ_j are determined by Eq. (3.11), and $\kappa_g > 1$, $\kappa_l < 1$ are parameters properly set according to ϵ and the fidelity constraint. Using the same κ_g and κ_l for all tests in T^{k+1} implicitly assumes that cells are uniform. However, $\mathcal{V}_i^{k+1}$, $i = 1, \cdots, n_{k+1}$, may differ from one another significantly in size and local variation. To account for such heterogeneity, we consider the source statistical model and field roughness.

3.3.3.3 Heterogeneity

Since it is more convenient to gauge the effect of heterogeneity on $P(U_i^{k+1}|Z^k)$ directly, we have elected to apply a compensating factor h_i to Eq. (3.10) and avoid interfering with the probability update. Define

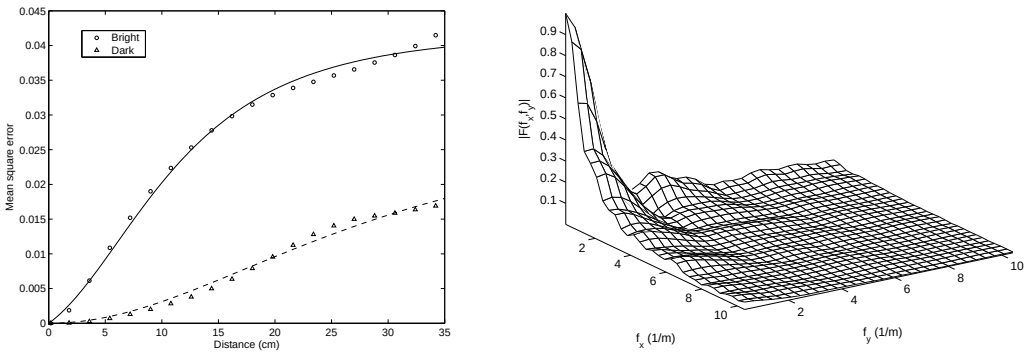
$$C_i = h_i P(U_i^{k+1}|Z^k) \tag{3.16}$$

We pick the candidate that maximizes C_i as the new sampling site.

The source statistical model is constructed to characterize the uncertainty in the gaps between scattered sampling points. Supposing p and q are two points that are d away from one another, and the light intensity at p is known, we use $\hat{f}(x_q, y_q) = f(x_p, y_p)$ to estimate the light intensity at q. The mean square error $\gamma(d)$ of this simple estimator is used to quantify the spatial correlation:

$$\gamma(d) = \mathrm{E}[f(x_p, y_p) - f(x_q, y_q)]^2, \quad \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} = d$$

The light environment under forest canopies consists of low-intensity background punctuated by sunlight flecks. The areas where light intensity is above certain threshold are often near the transition region where spatial variation tends to be high. This is confirmed by Fig. 3.7(a), where $\gamma(d)$ is estimated based on the data



(a) $\gamma(d)$ is plotted against d (b) 2D FFT of a sunlight field

Figure 3.7: Source models.

of eight sets of experiments, and we distinguish 1) dark: $f(x_p, y_p) \leq I_t$; 2) bright: $f(x_p, y_p) > I_t$. In both cases, $\gamma(d)$ is fitted using a rational quadratic model:

$$\gamma_i(d) = \frac{a_i d^2 + b_i d}{1 + c_i d^2} \tag{3.17}$$

where $i = 1$ or 2 depending on whether $f(x_p, y_p)$ is dark or bright. The statistical

model is incorporated in our algorithm by defining the compensating coefficient:

$$h_i^s = \sum_{j \in \mathcal{O}_m^k} \mu_j \gamma_j(d_{ij}, f(x_j, y_j), \quad d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.18}$$

where m is the candidate corresponding to $\mathcal{V}_i^{k+1}$, μ_j are the weights in Eq. (3.10), and $f(x_j, y_j)$ determines which model in Eq. (3.17) is used.

Another way of quantifying the heterogeneity is inspired by our observation in section 3.3.2.1. The field roughness estimated by multiplying the second order derivative of the reconstructed field at site candidate (x_i, y_i) with the cell area Δ_i is used as a compensating factor.

$$h_i^r = \left[\left(\frac{\partial^2 s(x_i, y_i)}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 s(x_i, y_i)}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 s(x_i, y_i)}{\partial y^2} \right)^2 \right] \Delta_i \tag{3.19}$$

Computing h_i^r is easy when the thin plate spline is used. When the second order derivative is not readily available (say if we use the piecewise linear interpolation for field reconstruction), techniques in [FJ89] can be used.

When source model and field roughness are both used, it is unclear what their relative weights should be. In our algorithm, two sets of samples are selected at each iteration: one is compensated by the source model and the other by roughness. Maintaining such diversity also helps to decrease the probability of large bias in case one approach fails miserably.

In addition, we often have some prior knowledge about the smoothness of the field. In Fig. 3.7(b), the two-dimensional FFT of a sunlight field is plotted. It is evident that most of its energy is concentrated in a low frequency band. If this piece of information is available, some cutoff rate can be specified as an upper bound on the spatial sampling density. This can result in considerable saving of system resources because otherwise we have to oversample at least once in each cell to reach a high level of confidence in cell fidelity.

3.3.4 Algorithm implementation

The algorithm starts with an initial set of sampling sites. Set $k = 1$, and execute the following steps.

1. Collect samples, and reconstruct the field. Set $P(U^1|Z^1)$ for initial data sites. Determine the sampling candidates.

2. Compute $P(U_i^{k+1}|Z^k)$ for all candidates according to Eq. (3.10).

3. Compute h_i^s and $C_i^s = h_i^s P(U_i^{k+1}|Z^k)$ according to the source statistical model.

   (1) Pick the candidate that maximizes C_i^s, and $N_s = N_s - 1$.

   (2) Update the candidate pool and C_i^s by considering the newly picked data site as an existing sampling point. If $N_s > 0$, go back to (1).

4. Compute h_i^r and $C_i^r = h_i^r P(U_i^{k+1}|Z^k)$ based on the reconstructed field at k.

   (1) Pick the candidate that maximizes C_i^r, and $N_r = N_r - 1$.

   (2) Update the candidate pool and C_i^r by considering the newly picked data site as an existing sampling point. If $N_r > 0$, go back to (1).

5. Collect samples at selected locations and reconstruct the field.

6. Evaluate T^{k+1}, and update probabilities using Eq. (3.13), (3.14), and (3.15).

7. If $P(U_i^{k+1}|Z^{k+1})$, $i = 1, \cdots, n_{k+1}$, reach required values or the total number of samples reaches a prescribed bound, exit the algorithm. Otherwise, $k = k + 1$, and go back to step 2.

We use the same $P(U^1|Z^1)$ for all initial sites. When multiple samples are collected during one iteration, the candidate pool and C_i is updated each time a data site is selected to avoid picking two sites too close to one another.

N_s and N_r are the numbers of new samples based on the source model and field roughness during each iteration. When the field is vastly undersampled, the roughness estimation generally deviates badly from its real value, so we may want to rely more on the source model. As we approach the critical sampling density, the estimation accuracy improves, and field roughness becomes more important. Hence, N_s and N_r can be varied to reflect such a shift of strategies.

When one sensor is responsible for collecting multiple samples in the field, a route design step can be inserted before the sampling action takes place. The problem of finding the optimal route to cover a set of sites, for example, the traveling salesman problem [LLK85], is a well-researched topic in optimization. Alternatively, we can incorporate the routing cost into our sample-selecting cost function such that the chance of selecting distant data sites in the same iteration is reduced. This naturally brings up the question of whether to employ strategies of a depth-first or breadth-first nature in picking new data sites. At each iteration, the sensor faces the decision of repeatedly taking measurements in a small region until the $P(U_i^{k+1}|Z^{k+1})$ in the area is small enough, or traveling a long distance to collect the most probable samples in the whole field. There is no easy answer as for which scheme is better. In general, if logistic cost is the main concern, a depth-first search is preferred. In this chapter, since the number of samples is considered a major constraint, a breadth-first search is used.

The probability updating process is suitable for distributed implementation owing to its nature of local operation. Hence, it can be applied to a network where static sensors are woken up to take measurements. As for the overall scheme, if

we use reconstruction schemes that require only local knowledge of samples, for example, the piece-wise linear interpolation, distributed algorithms can be easily devised.

3.4 Simulations

In this section, we first describe two sampling methods that our adaptive algorithm will be compared with. Then, a set of simulation results is presented.

3.4.1 Two sampling methods

To better evaluate our adaptive algorithm, we compare its performance with that of two other schemes. The first one is based on the maximin design. In this method, a pool of sampling candidates is maintained using the same method in section 3.3.1. At each step, the candidate that has the maximum distance to existing samples is selected as the new sampling site. This algorithm has the tendency of spreading the sampling sites uniformly over Dom, and is therefore called the uniform sampling.

The second scheme, which we call the Q method, is inspired by the stratified approach in [RPK04], but a few important modifications were made. Assuming Dom is a near-square rectangle, execute the following steps:

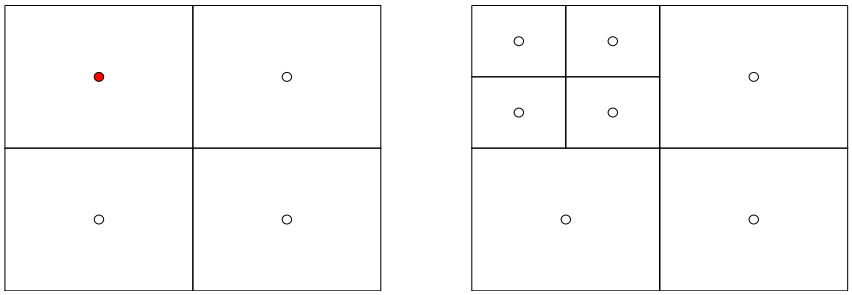
1. As illustrated in Fig. 3.8(a), divide Dom into four identical rectangular cells, and a sampling site is placed at the center of each cell.

2. Collect samples at new data sites and reconstruct the field.

3. Compare the reconstructed field with the true field in each cell, and compute the error. If fidelity constraint is satisfied, stop here. Otherwise continue

to 4.

4. Determine the cell with maximum error and denote it by m_e. Remove the sample at the center of m_e, and divide it into four identical rectangular cells in the same manner as in Fig. 3.8(a). Place four new sampling sites at the centers of new cells. Go back to step 2.



(a) (b)

Figure 3.8: Add new sampling sites in Q method. (a) Cell m_e is indicated by the solid sampling point at its center. (b) The old sample is removed and four new ones are added.

The way new sampling points are added to m_e is illustrated in Fig. 3.8. In the above algorithm, we have assumed that the true field is available for the evaluation of error in each cell, and old samples can be removed to allow for optimal distribution of data sites. Neither will be possible in practice. However, this scheme serves to provide an indication of the best performance that the stratified approach can deliver. During each iteration, since one sample is removed and four new samples are added, the total number of samples increases by three. In addition, thin plate spline interpolation is used for field reconstruction in both schemes.

3.4.2 Simulation results

The adaptive sampling algorithm is implemented and tested with various sunlight fields captured using the experimental setup in section 3.2. With carefully tuned parameters, the scheme works effectively and its performance is comparable and sometimes superior to that of Q method. Here we present one set of simulation results based on a particular sunlight field.

The true field is depicted in Fig. 3.9. About three quarters of the Dom are covered by a low-intensity light field, which can be adequately described by a few sampling points. In contrast, the shade of a steady tree branch cuts through the sunlight patch in the lower left corner, producing drastic spatial variations. The reconstructed fields using uniform sampling, Q method, and adaptive sampling algorithm are plotted in Fig. 3.10, 3.11, and 3.12 respectively.

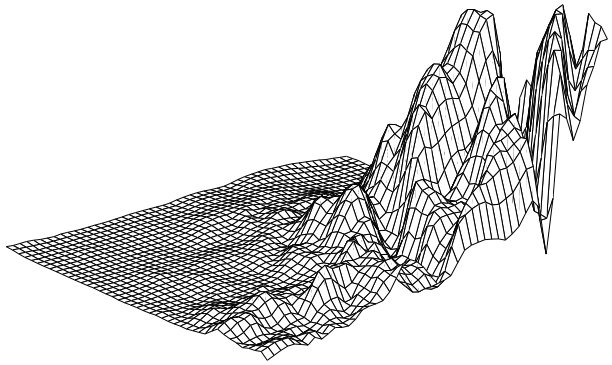


Figure 3.9: The true sunlight field.

We can see that both the Q method and adaptive sampling algorithm reasonably capture the shape of sunlight patch and the shade cast by the tree branch, while the uniform method smears the cut badly due to under-sampling in the region. The reconstructed field in Fig. 3.11 is slightly bent near the boundary of the low-intensity region since the Q method always places data sites at the

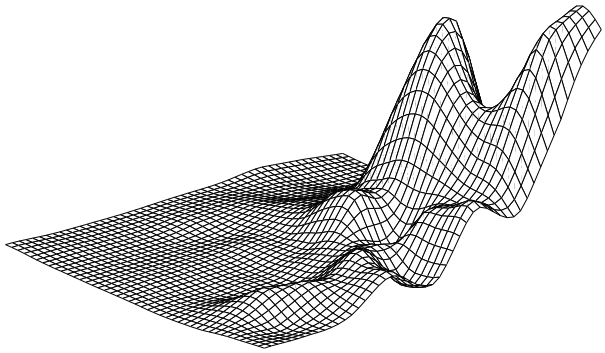


Figure 3.10: The reconstructed sunlight field using uniform sampling method with 102 samples.

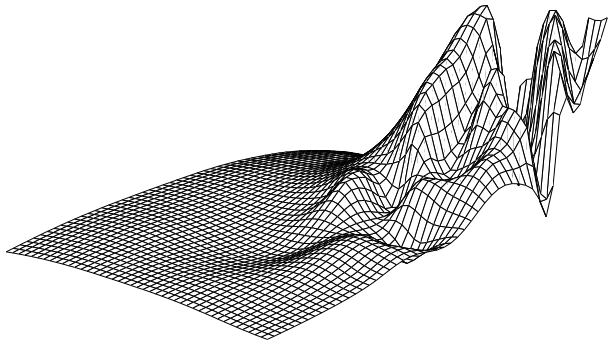


Figure 3.11: The reconstructed sunlight field using the Q method with 103 samples.

centers of cells.

The sampling sites based on these three schemes are shown in Fig. 3.13, 3.14, and 3.15 respectively. The distribution of sampling sites in Fig. 3.15 shows that the adaptive algorithm has followed the heterogeneity of the field nicely.

The bending energy and mean square error of the reconstructed thin plate splines at successive iterations are plotted in Fig. 3.16 and 3.17 respectively. For uniform and adaptive algorithms, we start with eight samples and set $N_s = N_r = 1$. Hence, two new samples are added at each step. In this case, the

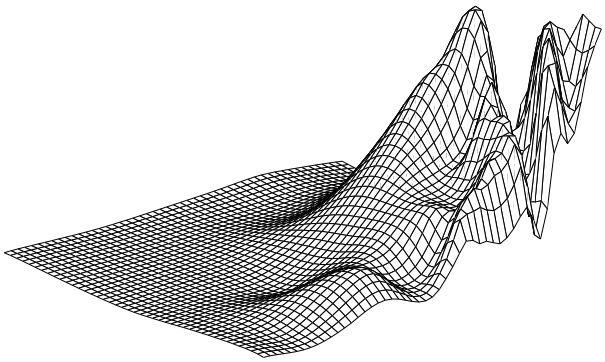


Figure 3.12: The reconstructed sunlight field using adaptive sampling method with 102 samples.

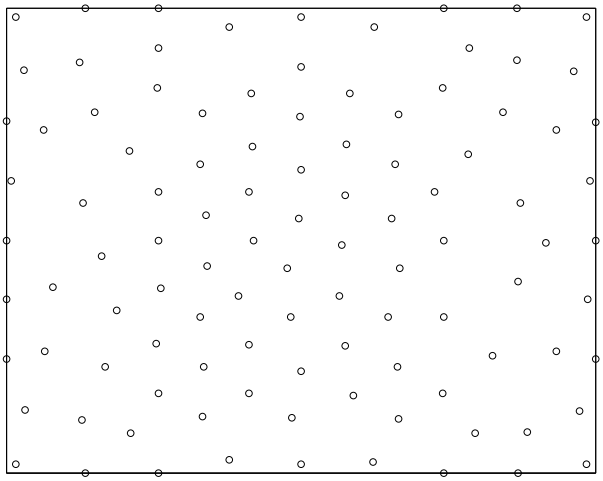


Figure 3.13: The distribution of 102 sampling sites in the uniform sampling method.

adaptive sampling method has the highest bending energy and outperforms the Q method during most iterations. It is reasoned that although the Q method has the advantage of knowing the true field, the uniform and adaptive algorithms appear to fill the space more efficiently than the square cells.

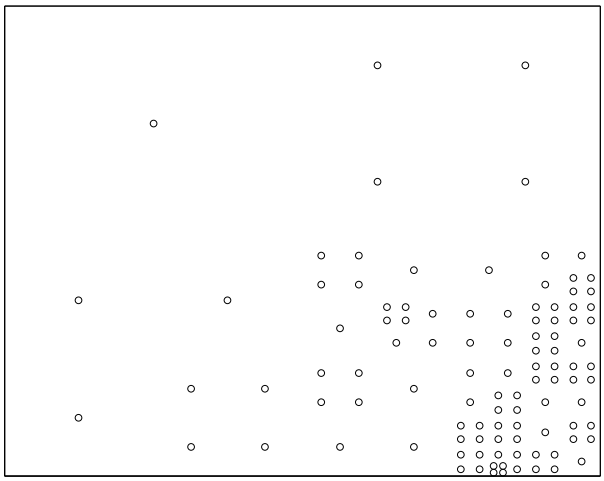


Figure 3.14: The distribution of 103 sampling sites in the Q method.

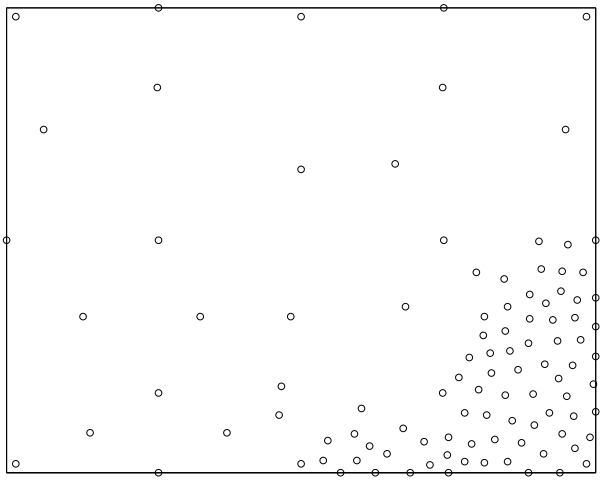


Figure 3.15: The distribution of 102 sampling sites in the adaptive sampling method.

3.5 Conclusion

In this chapter, we developed an Bayesian algorithm for adaptively sampling and reconstructing distributed phenomena. Simulations with field data show that the method works effectively and has competitive performance.

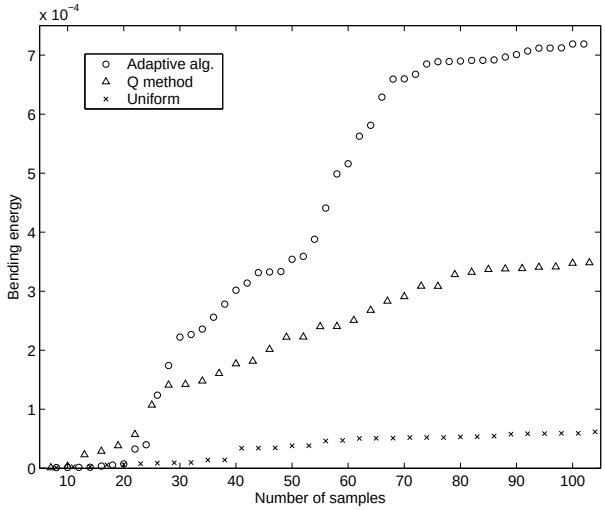


Figure 3.16: The field bending energy in successive steps.

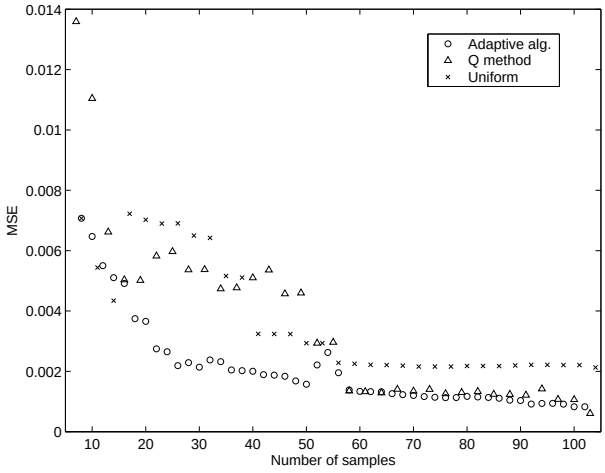


Figure 3.17: The mean square error in successive steps.

Although our scheme is designed for individual mobile sensors, it can be implemented in a distributed fashion if local reconstruction methods are used. For example, a Delaunay triangulation can be easily constructed out of DT(S). We can then treat each Delaunay triangle as a local cluster, and fit a two-dimensional piecewise linear function in each cell.

As we discussed in section 3.3.4, the routing cost of mobile sensors is not considered in this chapter, but it can enter the scheme in various ways. Although a rectangular sampling domain is considered in this chapter, it is not difficult to extend our algorithm to arbitrary boundaries. If measurement error has to be taken into account, approximation techniques such as minimum mean square error estimation can be used instead of interpolation.

CHAPTER 4

Source Coding in Wireless Sensor Networks

4.1 Introduction

Wireless sensor networks often operate under tight energy budgets, and communication power accounts for a substantial portion of sensor's energy consumption. While the processing power scales with more advanced IC technology, wireless communication power is fundamentally limited by the propagation loss and information theory. Hence, the data rate should be aggressively reduced to achieve conservation [PK00].

Sensor networks differ from traditional communication networks in that data generated at different sensors, especially proximate ones, have high correlation since they are observations of closely related physical phenomena. Source coding techniques can be employed to remove such redundancy among data streams from different sensors. In the first part of this chapter, we give a quick overview of distributed source coding, in which sensors conduct data compression independently without interacting with one another. Our brief discussion on the subject by no means belittles its importance. This is a dynamic research area, and efficient distributed codes are being actively pursued. In the rest of the chapter, our attention turns to source coding with explicit side information.

A lot of research has been focusing on judiciously routing packets through sensors with highly correlated data such that the overall transmission is mini-

mized [IGE03, CBV04]. As a simple illustration, consider Fig. 4.1 where three sensors transmit their observations to the fusion center. Sensor s_1 may route

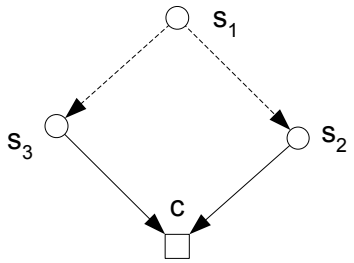


Figure 4.1: A simple joint compression/routing problem.

its packets through s_2 or s_3. The relay can read s_1's packets to further compress its own data. To determine the better routing strategy, we need to know how much *additional* rate reduction, which may vary with time, s_1's data can produce for s_2 and s_3. Most data-centric routing algorithms assumes this rate-reduction information. In the second part of the chapter, we propose a two-stage DPCM (differential pulse coded modulation) coding scheme that processes first local side information, which is available without cost, then samples from other sensors. Additional coding gain provided by distant helping samples can be continuously monitored such that spatial side information is used only when the gain outweighs the cost. (This information is generally not available in traditional coding schemes.) Our method can be combined with data-centric routing strategies for use in joint compression/routing optimization.

We use closed loop backward adaptation, which does not require coefficient transmission, and tracks the changing statistics. In contrast, a forward adaptive scheme computes prediction coefficients from a block of samples in advance. It offers slightly higher gain for a slowly evolving field, but requires data buffering and additional bandwidth for coefficient transmission. Along with the adaptive prediction, adaptive quantization is used to make the most of the coding gain.

DPCM has been widely used in speech and video coding [JN84]. Two-stage DPCM schemes in speech coding base their predictions on previous samples and samples that are about one pitch period away. In video coding, the two-stage scheme is applied when both adjacent and inter-frame samples are used. However, in these methods, the distant side information is as readily available as adjacent samples. Hence, their coder design has more flexibilities.

The chapter is organized as follows. We give an overview of distributed source coding in section 4.2. Then the two-stage DPCM scheme is presented in section 4.3, Some simulation results are presented in section 4.4 to evaluate the performance of our DPCM scheme. Section 4.5 concludes the chapter.

4.2 Distributed Source Coding

Distributed source coding, as depicted in Fig. 1.6(a), draws on the fundamental result of Slepian and Wolf [SW73], which states:

Theorem 4.2.1 (Slepian-Wolf) *When two correlated sources that have discrete alphabets are drawn i.i.d.*$\sim p(x, y)$*, the achievable rate region for distributed source coding is given by:*

$$\begin{aligned} R_1 &\geq H(X|Y), \\ R_2 &\geq H(Y|X), \\ R_1 + R_2 &\geq H(X, Y) \end{aligned}$$

This theorem shows, surprisingly, that there is no loss of efficiency in independently encoding two sources. This result has been partially extended to sources with continuous alphabets by [WZ76], in which one of the sources is assumed to be completely known at the encoder. Although the complete answer to the

distributed rate distortion coding is not yet known, [ZB99] observes that high resolution source coding resembles the Slepian-Wolf. Nevertheless, none of these papers proposed any constructive codes, and the proofs proceed using a coding procedure called random bins. Only recently have practical distributed source coding schemes [PKR02, XLC04, PR03, GZ03, ZSE02] begun to emerge in the literature. In this section, we give a quick overview of the principles of distributed source coding.

To illustrate the basic idea, we borrow a simple example from [PKR02]. Supposing X and Y are three bit binary words, and each bit has equal probability of being 0 or 1. Furthermore, X and Y are correlated such that the Hamming distance between two words is at most 1. Now assuming that Y is known at the decoder but not the encoder for X, how can we take advantage of this side information to transmit X at a rate smaller than 3 bits. This is achieved by observing that since Y is available at the decoder, there is no need to distinguish any words that have Hamming distance 3 for X. Hence, we can divide the space of X into the following sets: $\{000, 111\}, \{001, 110\}, \{010, 101\}, \{100, 011\}$, and only send the set index. Then the decoder can determine which of the two words in the set is the true word, but computing the Hamming distance to Y. The one with smaller Hamming distance ought to be X.

The coder design for distributed source coding involves dividing X's codewords into cosets. Various ways (e.g. a trellis) can be used to partition the codeword space. In transmitting X, only the index of the set that the codeword belongs to is sent. The decoding makes use of the side information Y by computing the distance from all codewords in the set to Y. The one with the minimum distance is declared to be the codeword for X. This procedure is illustrated in Fig. 4.2. For more complete overviews on the topic, readers are referred

to [PKR02, XLC04].

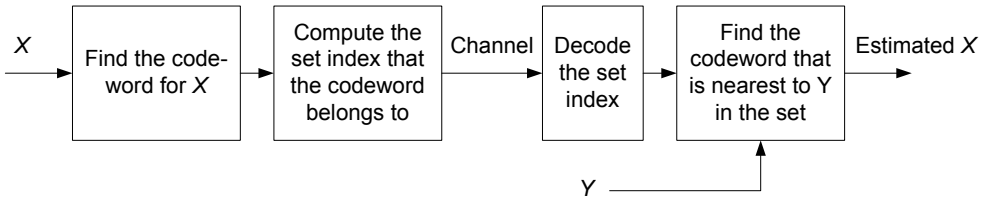


Figure 4.2: Encoder and decoder for source coding with side information. X is to be coded, and Y to act as side information

4.3 A Two-Stage DPCM Scheme for Sensor Networks

In this section, we propose a 2-stage DPCM scheme. In contrast to distributed source coding in the previous section, side information is assumed to be available at both the encoder and decoder in this coding method.

4.3.1 Two-stage suboptimal approach

The two-stage suboptimal approach is described in Fig. 4.3. Sequences $x_j(n),\ j =$

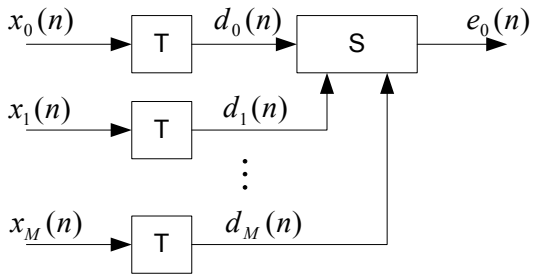


Figure 4.3: The block diagram of a two-stage DPCM encoder

$0, 1, \cdots, M$ are the measurements of sensor j at time n. Assuming that all the sequences have had their mean removed, a temporal DPCM stage is first run at

all sensors, then further compression is achieved using other sensors' temporal DPCM output as the side information. Define $\tau_j,\ j = 1, \cdots, M$ as the delay at $d_j(n)$ that yields the highest correlation with $d_0(n)$. It can be estimated using cross-correlation methods. Denote by a^* the complex conjugate of a, and A^H the complex conjugate and transpose of A. We have the following:

$$\begin{aligned} d_0(n) &= x_0(n) - \sum_{i=1}^{N} a_i^* \tilde{x}_0(n-i) \\ &= x_0(n) - \mathbf{w}_t^H \mathbf{y}_t(n) \end{aligned}$$

$$\begin{aligned} e_0(n) &= d_0(n) - \sum_{j=1}^{M} \sum_{k=-K}^{K} b_{j,k}^* \tilde{d}_j(n+\tau_j+k) \\ &= d_0(n) - \mathbf{w}_s^H \mathbf{y}_s(n) \end{aligned}$$

where

$$\mathbf{w}_t = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix},\ \mathbf{y}_t(n) = \begin{bmatrix} \tilde{x}_0(n-1) \\ \vdots \\ \tilde{x}_0(n-N) \end{bmatrix},$$

$$\mathbf{w}_s = \begin{bmatrix} b_{1,K} \\ \vdots \\ b_{1,-K} \\ \vdots \\ b_{M,K} \\ \vdots \\ b_{M,-K} \end{bmatrix},\ \mathbf{y}_s(n) = \begin{bmatrix} \tilde{d}_1(n+\tau_1+K) \\ \vdots \\ \tilde{d}_1(n+\tau_1-K) \\ \vdots \\ \tilde{d}_M(n+\tau_M+K) \\ \vdots \\ \tilde{d}_M(n+\tau_M-K) \end{bmatrix}$$

In a closed loop implementation, $\tilde{x}_0(n)$ and $\tilde{d}_j(n)$ are samples that are available at the decoder. Here, we assume they are the same as $x_0(n)$ and $d_j(n)$ with sufficient quantization bits. MMSE criteria on separate stages yield:

$$\mathbf{w}_t^{\text{opt}} = \mathbf{R}_{tt}^{-1}\mathbf{r}_t, \quad \mathbf{w}_s^{\text{opt}} = \mathbf{R}_{ss}^{-1}\mathbf{r}_s,$$

and the minimum mean square error

$$J_{\min}^{s} = \sigma_x^2 - \mathbf{r}_t^H \mathbf{R}_{tt}^{-1} \mathbf{r}_t - \mathbf{r}_s^H \mathbf{R}_{ss}^{-1} \mathbf{r}_s.$$

in which

$$\mathbf{R}_{tt} = \mathrm{E}\mathbf{y}_t \mathbf{y}_t^H,\ \mathbf{R}_{ss} = \mathrm{E}\mathbf{y}_s \mathbf{y}_s^H,\ \mathbf{r}_t = \mathrm{E}\mathbf{y}_t x_0^*,\ \mathbf{r}_s = \mathrm{E}\mathbf{y}_s d_0^*$$

Define the overall and spatial coding gains

$$G = \mathrm{E}x_0^2(n)/\mathrm{E}e_0^2(n) = \sigma_x^2/J_{\min}^{s}$$

$$G_s = \mathrm{E}d_0^2(n)/\mathrm{E}e_0^2(n) = (\sigma_x^2 - \mathbf{r}_t^H \mathbf{R}_{tt}^{-1} \mathbf{r}_t)/J_{\min}^{s}$$

In contrast, a one-stage scheme using the same set of side information would yield:

$$J_{\min} = \sigma_x^2 - \mathbf{r}_y^H \mathbf{R}_{yy}^{-1} \mathbf{r}_y$$

in which

$$\mathbf{R}_{ts} = \mathrm{E}\mathbf{y}_t \mathbf{y}_s^H, \mathbf{R}_{yy} = \begin{bmatrix} \mathbf{R}_{tt} & \mathbf{R}_{ts} \\ \mathbf{R}_{ts}^H & \mathbf{R}_{ss} \end{bmatrix}, \mathbf{r}_y = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{r}_s + \mathbf{R}_{ts}^H \mathbf{w}_t^{\mathrm{opt}} \end{bmatrix}$$

In general, $J_{\min}^{s} > J_{\min}$, but a two-stage implementation offers several other advantages over a one-stage approach. It improves stability. For highly correlated $x_0(n)$ and $x_j(n)$, matrix $\mathbf{R}_{yy}$ becomes near-singular. Separately designing temporal and spatial stages can help ensure that the temporal stage is minimum phase, thus stable. The spatial coding gain G_s sheds light on how much additional gain is provided by distant samples. At the spatial stage, $d_j(n)$ instead of $x_j(n)$ is used, so less decoding effort is required. In addition to compression, the temporal stage serves as a pre-whitening process, and the resulting $\mathbf{R}_{ss}$ has better eigenvalue structures. This enhances the adaptive performance of the second stage [Say03].

4.3.2 ϵ-NLMS adaptation

The detailed block diagram of the closed loop DPCM encoder is given in Fig. 4.4. Switch k_1 controls whether the spatial stage is used. The decoder has a similar structure, and hence is not shown here.

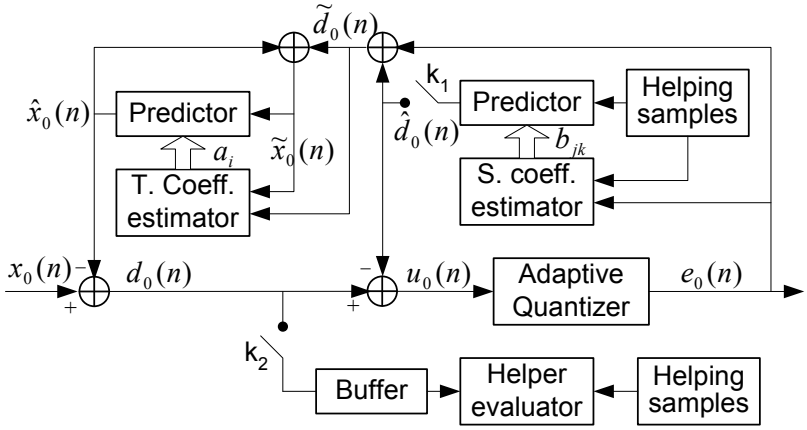


Figure 4.4: The detailed block diagram of the encoder

The weight iteration uses ϵ-NLMS with power update. The algorithm starts with $\mathbf{w}_t(-1)$, $p_t(-1)$, $\mathbf{w}_s(-1)$, and $p_s(-1)$, iterate for $n = 0, 1, 2, \cdots$

$$\hat{x}_0(n) = \mathbf{w}_t^H(n-1)\mathbf{y}_t(n), \quad d_0(n) = x_0(n) - \hat{x}_0(n)$$

$$\hat{d}_0(n) = \mathbf{w}_s^H(n-1)\mathbf{y}_s(n), \quad u_0(n) = d_0(n) - \hat{d}_0(n)$$

$$e_0(n) = \mathrm{Q}\left[u_0(n)\right]$$

$$\tilde{d}_0(n) = \hat{d}_0(n) + e_0(n), \quad \tilde{x}_0(n) = \hat{x}_0(n) + \tilde{d}_0(n)$$

$$p_s(n) = \beta_s p_s(n-1) + (1-\beta_s)|\tilde{d}_0(n)|^2$$

$$\mathbf{w}_s(n) = \mathbf{w}_s(n-1) + \frac{\mu_s}{\epsilon_s + p_s(n)} e_0^*(n)\mathbf{y}_s(n)$$

$$p_t(n) = \beta_t p_t(n-1) + (1-\beta_t)|\tilde{x}_0(n)|^2$$

$$\mathbf{w}_t(n) = \mathbf{w}_t(n-1) + \frac{\mu_t}{\epsilon_t + p_t(n)} \tilde{d}_0^*(n)\mathbf{y}_t(n)$$

4.3.3 Helper evaluator

The helper evaluator, controlled by k_2 has two functions: delay estimation and helper selection. Delay estimator determines the $\tau_j,\ j = 1, \cdots, M$ resulting in the highest cross-correlation

$$\phi_{0j}(\tau_j) = \frac{\sum_{i=n}^{n+L-1-\tau_j} d_0(i) d_j^*(i+\tau_j)}{\sqrt{\sum_{i=n}^{n+L-1} |d_0(n)|^2 \sum_{i=n}^{n+L-1} |d_j(n)|^2}}$$

Directly computing $\phi_{0j}(\tau_j)$ requires $\mathrm{O}(L)$ operations (L is the block size). The cost is reduced by using coarsely quantized samples [DSR76]. The helper selector comes into play when a decision needs to be made on using which set of sensors' data as side information. The autocorrelation method [LO88] is applied to applicable sets of sensors and the one with the best gain/cost tradeoff is selected assuming the cost of distant side information is known. Note that the correlation matrix has Toeplitz-like structures, and efficient algorithms exist for solving such systems [SK95]. The overall cost of the evaluator can be kept within $\mathrm{O}(L)$.

As the prediction error $u_0(n)$ varies, an adaptive quantizer is essential to maximize the coding gain and limit quantization error. Readers are directed to [JN84] for more details.

4.4 Simulations

In this section, we present three sets of simulations to show the effectiveness of our 2-stage DPCM scheme. First, an autoregressive source is considered. Then, a set of acoustic wave streams that is recorded by a sensor array while a tank moves by is used. Lastly, the simulation is conducted on some weather data obtained from [ncd].

4.4.1 Autoregressive source

In the first simulation, we consider an autoregressive source observed by two sensors.

$$s(n) = s(n-1) - .5s(n-2) + z(n)$$

$$x_j(n) = s(n) + u_j(n), \quad j = 1, 2$$

in which $z(n)$ and $u_j(n)$ are white Gaussian noise. Using one sensor's data as helping information, we plot the coding gains G and G_s against the variance ratio σ_z^2/σ_u^2 for different schemes. 'osf' indicates one-stage forward DPCM, 'tsd' means two-stage ϵ-NLMS with different step sizes on temporal and spatial stages, and 'tss' denotes two-stage ϵ-NLMS with the same step sizes. (The spatial coding gain of one-stage forward method is evaluated by comparing its result to the output of a single forward temporal stage.) It is observed that the spatial coding increases with the observation SNR, while temporal gain quickly saturates as it is circumscribed by the source statistics. In addition, our experience reveals that appropriately choosing the relative step sizes of spatial and temporal stages $\kappa = \mu_s/\mu_t$ can yield up to 2 dB gain improvement over simply setting $\kappa = 1$. This is explained as follows. Since the magnitude of $e_0(n)$ is smaller than that of $d_0(n)$, using the same step sizes discourages the update of spatial weights $\mathbf{w}_s$. Choosing κ according to the magnitudes of e_0 and d_0 results in an adaptation that resembles the one-stage DPCM. It is cautioned, however, that setting κ too big undermines the temporal stage and tends to exaggerate the spatial coding gain.

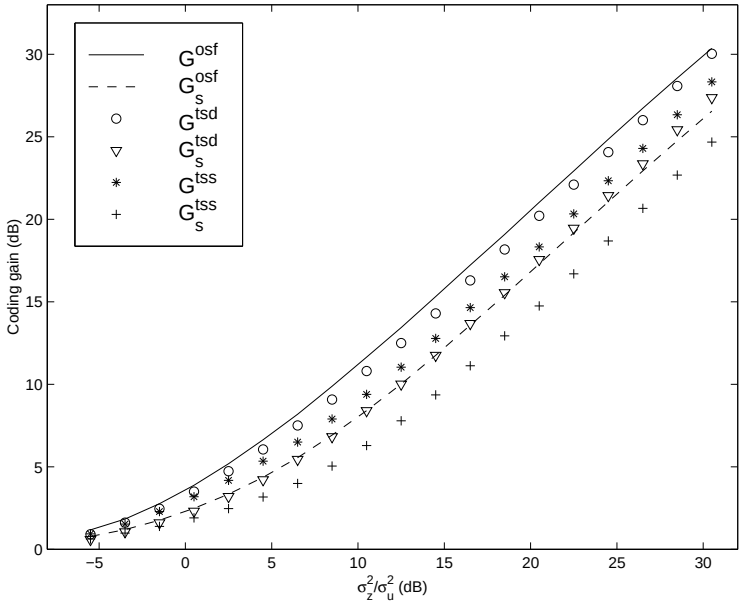


Figure 4.5: Coding gain for an autoregressive source

Table 4.1: Coding gains (dB) of different schemes.

sensor	G^{tsd}	G_s^{tsd}	G^{tss}	G_s^{tss}	G^{osf}	G_s^{osf}
s_1	22.27	9.08	21.60	8.41	21.13	7.95
s_2	21.93	8.64	21.25	7.97	18.70	5.76
s_3	21.55	8.35	20.69	7.49	19.13	6.32

4.4.2 Acoustic source

In the second simulation, we consider the acoustic data generated by a moving tank in a near field sensor array setup depicted in Fig. 4.6 [Yao]. 2000 samples are collected during the period. Observations from sensor s_0 are used as helping information to compress the data at sensor s_1, s_2, and s_3. Since relative delay τ_j varies when the tank passes by the array, ϵ-NLMS adaptation performs better than the forward scheme that estimates the prediction coefficients for blocks of samples. This is displayed in Table 4.1, where we compare the coding gains (in

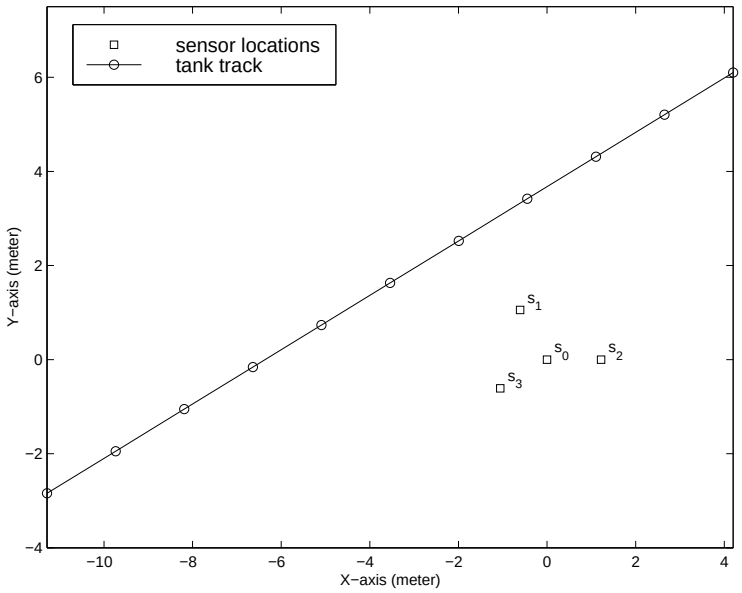


Figure 4.6: Near field sensor array configuration

dB) of ϵ-NLMS adaptation and the forward scheme with block size 200. Notice 'tsd' has slightly higher gain than 'tss'. We also observe that when the tank is closest to the sensor array, the forward method fares worst as τ_j varies the most. On the other hand, the ϵ-NLMS adaptation yields consistent results once it converges.

4.4.3 Weather data

So far, we have considered point sources. In the last simulation, we look at some correlated weather data obtained from NCDC [ncd]. We consider the daily mean temperature measured at three stations located at Hongkong, Macao, and Datong in 2003. We compress the set of data at Hongkong using those from Macao and Datong as side information. The coding gains (in dB) are given in Table 4.2. It shows that the spatial coding gain by Macao is much higher than that provided by Datong. This is expected because Hongkong and Macao are two cities near

Table 4.2: Coding gains (dB) by different cities.

city	G^{tsd}	G_s^{tsd}
Macao	19.27	10.08
Datong	9.81	.09

to one another, while Datong is at northern China, thousands miles away. In Fig. 4.7, we plot the input $x(n)$ (with mean removed) and outputs of temporal and spatial stages at the encoder when samples from Macao are used as side information. Note that the relatively large error at the beginning is caused by the initial weight convergence.

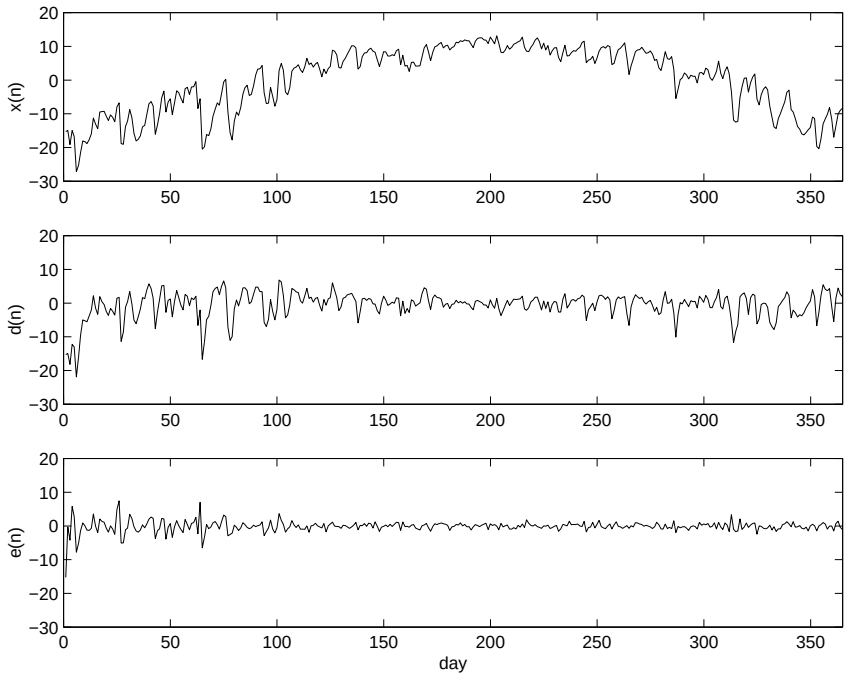


Figure 4.7: Input and outputs of the encoder at Hongkong

4.5 Conclusion

In this chapter, we considered source coding to remove redundancy among data streams from different sensors in sensor networks. First, we gave a quick overview on the distributed source coding. This is an active research area, and efficient codes with low complexity are still under study. In the second part of the chapter, we presented a two-stage DPCM scheme. Its ability to track the additional coding gain provided by distant side information makes it useful for joint compression/routing optimization in sensor networks. The ϵ-NLMS adaptation reasonably adjusts to the changes on sample correlation. Simulations demonstrate that the algorithm provides results close to optima when the step sizes are appropriately set.

CHAPTER 5

Combined Routing and Source Coding

5.1 Introduction

The need to lower the communication cost in wireless sensor networks has prompted many researchers to propose data-centric routing schemes that utilize in-network data fusion to reduce the transmission rate. There are two major difficulties in designing such routes. First, the lack of reasonably practical data aggregation models has led researchers to use overly simplified ones [KEW02, CBV04, GE03, IGE03]. For example, these models generally assume that sensors perform the same aggregation function regardless of the origin of the fused data. As a remedy, [GE03] suggests looking into models in which data aggregation is not only a function of the number of sources but also the identity of the sources. Second, the resulting optimization problem is often NP-hard due to the coupling of routing and in-network data fusion [KEW02, CBV04]. Hence, algorithms that find exact solutions in polynomial time are unlikely to exist. In this chapter, we attempt to build network models that are computationally useful yet reasonably approximate reality.

Source coding in sensor networks is generally lossy. Although high resolution lossy coding resembles Slepian-Wolf coding [ZB99], general network distortion coding remains an open problem. Also, distributed source coding schemes with performance near information theoretic bounds often employ long blocks of data,

which results in high complexity and long delays. In this chapter, we consider source coding with explicit side information. In other words, only when the side information is available at both the encoder and decoder, can it be used to reduce the data rate. In practice, a lossy encoder (such as the DPCM encoder in [LTP05]) can be employed at each sensor to compress its data using incoming flows as explicit side information. Alternatively, we can quantize the analog signal locally. Then joint entropy coding is conducted on merged data flows using, for example, a Lempel-Ziv encoder.

In many situations, data aggregation is possible because the fusion center (end user) is interested only in some fused function. For example, in [CYE03], only the direction of arrival estimation needs to be transmitted from each sensor sub-array to the fusion center to locate an acoustic source. In these cases, the way that data aggregation and communication is carried out is highly dependent on the specific application. This problem under the broad title of distributed data fusion is by itself an area under active research.

There has been much recent research activity on data-centric routing. In [SS02b], the interdependence of routing and data compression is addressed from the viewpoint of information theory. Clustering methods have been used by some researchers to aggregate data at the cluster head before transmitting them to the fusion center [BC03, HCB00]. Since the cluster head is responsible for data aggregation and relaying, it consumes the most energy. Hence, dynamically electing nodes with more residual power to be cluster heads and evenly distributing energy consumption in the network is a major issue in these schemes. In [IGE03], a diffusion type routing paradigm that attaches attribute-value pairs to data packets is proposed to facilitate the in-network data fusion. The correlated data routing problems studied in [KEW02, CBV04] are closely related to our work.

In [KEW02], the authors give a thorough comparison of data-centric and address-centric methods and a overview of recent effort in the field. [CBV04] casts the data-centric routing problem as an optimization problem and seek solutions to it when different source coding schemes are applied. A similar optimization problem is also the subject of [GE03], where a simplified data model is assumed.

The rest of the chapter is organized as follows. In section 5.2, we present our network flow and data rate models. Based on these models, an optimization problem CRSC, is formulated, and subsequently shown to be NP-hard in section 5.3. In section 5.4, a mixed integer program is formulated for one of CRSC's sub-instances. Section 5.5 concludes the chapter. In the next chapter, heuristic algorithms will be proposed for CRSC.

5.2 Network Models

5.2.1 Network flows

The topology of a sensor network is abstracted as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The node set $\mathcal{N}$ consists of a set $\mathcal{N}_s$ of n sensors and a special node t representing the fusion center. Denote by $\mathcal{N}_a$ the set of active sensors that produce data. $\mathcal{N}_a \subseteq \mathcal{N}_s$. Both active and non-active sensors can relay and process data. The edge set $\mathcal{E}$ includes m communication links. Here, we assume all the links are bi-directional and symmetric. If they are not, the network can be modelled as a directed graph, and the derivation in this chapter will apply similarly. We also assume that the network is connected so that messages from any sensor can reach t through direct transmission or relaying. A weight c_e is associated with each edge $e \in \mathcal{E}$ to indicate the cost (e.g. power) of transmitting data at unit rate across e. These weights are given a priori. The flow f_e is defined as the rate at

which data is transmitted across edge $e \in \mathcal{E}$. Data flow generated by node i and terminating at node j is denoted by f^{ij}. In particular, we define $f^i = f^{it}$. Clearly, $f_e = \sum_{i,j \in \mathcal{N}} f_e^{ij}$. Supposing $i, j \in \mathcal{N}$, denote by d_{ij} the minimum distance from i to j (i.e. the sum of edge weights along the shortest path from i to j). In particular, $d_i = d_{it}$. Our objective can be, for example, to minimize the total cost C of routing all the data from active sensors to the fusion center.

$$C = \sum_{e \in \mathcal{E}} c_e f_e \tag{5.1}$$

5.2.2 Source coding with explicit side information

Denote by X_i the data stream produced by sensor i. Assume X_i, $i \in \mathcal{N}_s$ satisfies the ergodic condition so that the results of statistical probability theory can be applied. In this chapter, we consider source coding with explicit side information as depicted in Fig. 5.1. In other words, only when the side information is

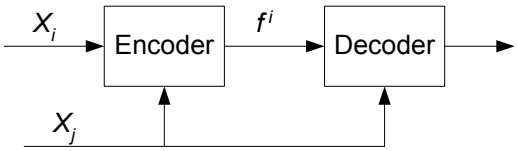


Figure 5.1: Encoder and decoder for X_i with explicit side information X_j.

available at both the encoder and decoder can it be used to compress the data. We decompose the data transmission into individual data flows f^i and consider the minimum rate required such that X_i can be recovered at the fusion center. Under lossy coding, suppose the side information for coding data stream X_i is $\hat{X}_{k_1}, \cdots, \hat{X}_{k_j}$, where $k_1, \cdots, k_j \in \mathcal{H}_i$. ($\mathcal{H}_i$ is the set of sensors whose data are correlated with X_i, and $\hat{X}_k$ denotes the coded version of X_k.) The minimum rate

required subject to some distortion constraint, $d(X_i, \hat{X}_i) \leq D$, is [CT91]:

$$f^i = \min_{d(X_i, \hat{X}_i) \leq D} I(X_i, \hat{X}_i | \hat{X}_{k_1}, \cdots, \hat{X}_{k_j}) \tag{5.2}$$

where $I(\cdot)$ denotes the mutual information. When data streams have discrete values, entropy coding can be used,

$$f^i = H(X_i | X_{k_1}, \cdots, X_{k_j}) \tag{5.3}$$

In either case, the data rate depends on what type of side information is available, hence is a function of $M_i = |\mathcal{H}_i|$ binary variables. ($|\mathcal{S}|$ denotes the number of elements in the finite set $\mathcal{S}$.) For a network of n sensors, M_i can be as large as $(n-1)$. Thus, this description alone has exponential complexity. To simplify the problem, we assume that M_i is relatively small and side information from at most k_s sensors is used. Our study in this chapter will concentrate on the simple case when $k_s = 1$:

$$f^i = \begin{cases} b_0^i & \text{no side information;} \\ b_1^{ij} & X_j, j \in \mathcal{H}_i, \text{ is used as side information.} \end{cases} \tag{5.4}$$

It is noted that data rates $b_0^i \geq b_1^{ij} \geq 0$, and $b_0^i = 0$ when i producing no data. When data streams X_i and X_j are correlated, we have $i \in \mathcal{H}_j$ and $j \in \mathcal{H}_i$. In addition, since $H(X_i) - H(X_i|X_j) = H(X_j) - H(X_j|X_i)$, we assume $b_0^i - b_1^{ij} = b_0^j - b_1^{ji}$.

The data rate model in Eq. (5.4) appears rather simple. However, there are practical reasons to assume such small values for M_i and k_s. First, in many physical situations, high correlation occurs only in a small neighborhood. In others, reconstruction fidelity constraints may permit thinning the number of active sensors, so again only a small number of sensors has high correlation. Second, due to the correlation among side information, coding gain often saturates quickly as

the number of helpers increases. In addition, determining coding gain information and processing side information incurs cost, and the gain of using additional helpers is usually not enough to be worth it. Third, using more helpers increases the complexity of the model and optimization. On the other hand, our study on this simpler case can serve as a first step in dealing with more complicated problems where multiple helpers are allowed.

To obtain the rate function in Eq. (5.4), a training process must take place before the route design. In our model, it is assumed that sensors in $\mathcal{H}_i$ are in the neighborhood of sensor i. Thus each active sensor only needs to transmit a small set of data to the sensors in its neighborhood. Due to the small values of M_i and k_s, the cost of this training process can be considered moderate. Alternatively, such information can be fed back from sensors or the fusion center that perform data aggregation. If neither are available, then simple indicators such as the attribute-value pairs used in [IGE03] may be used to indicate the level of data correlation.

5.2.3 Cost functions

Various cost functions can be formulated for sensor networks. One simplest cost function based upon the aggregate data rate is:

$$C = \sum_{e \in \mathcal{E}} f_e \tag{5.5}$$

This cost function assumes that all the links have equal weights. This is however often not the case. For example, as the capacity of the wireless link is affected by distance, fading, node transmission power etc, a normalized cost function can be formed to take this into account:

$$C = \sum_{e \in \mathcal{E}} f_e / u_e \tag{5.6}$$

where u_e is the capacity of link e. Another way of placing weights on communication links is to define c_e as energy per bit on link e and try to minimize the cost function in Eq. (5.1). Here it is assumed that on each link consumed energy is a linear function of data rate. This is a reasonable approximation to reality if increased data rate is due to increased radio operation time. However, if the data rate increase is done by switching modulation schemes, energy consumption on each link cannot then be thought of as a pure linear function of data rate. Moreover, state of the art radios can possess multiple antennas, and operate on different sets of sub-channels. Therefore, a better characterization of energy consumption is a piece-wise linear function of the data rate:

$$C = \sum_{e \in \mathcal{E}} \left(c_{e,1} f_{e,1} + c_{e,2} f_{e,2} + \cdots + c_{e,n_e} f_{e,n_e} \right) \tag{5.7}$$

The data rate across edge e is given by $f_e = \sum_{i=1}^{n_e} f_{e,i}$, and each portion of the flow $f_{e,i}$ has a capacity $u_{e,i}$, $i = 1, \cdots, n_e$.

More insight can be gained by looking at the energy consumption at individual sensors rather than links:

$$E_i = c_i^t \sum_{j \in \mathcal{N}} f_{ij} + c_i^r \sum_{j \in \mathcal{N}} f_{ji}$$

where c_i^t and c_i^r are the energy per bit consumed by sensor i when it is acting as the transmitter and receiver. In practice, additional energy may be required to power up the circuit. This represents a non-zero initial cost when sensors switch from the stand-by mode to a transmission mode. Denote this cost by c_i^0, we define a binary variable:

$$g_i = \begin{cases} 1 & \text{Transmission mode} \\ 0 & \text{Standby mode} \end{cases}$$

The energy consumption at node i is:

$$E_i = c_i^t \sum_{j \in \mathcal{N}} f_{ij} + c_i^r \sum_{j \in \mathcal{N}} f_{ji} + c_i^0 g_i \tag{5.8}$$

The objective can be minimizing the maximum E_i in the network.

Another metric of interest is delay. If the capacity of link e is u_e, the queueing delay at this link can be approximately modeled as:

$$\tau_e = \frac{\alpha f_e}{u_e - f_e}$$

Assuming τ_e dominates propagation and processing delays, we can propose to minimize the aggregate delay of the network:

$$C = \sum_{e \in \mathcal{E}} \tau_e \tag{5.9}$$

There has been research work that tries to build models for estimating the battery time-to-failure for a given load [RVW03]. If the communication power is the dominant power consumption in the sensor system, the sensor's battery life is a function of the transmission rate:

$$t_i = t_i(f_{ij}, f_{ki}), \quad j, k \in \mathcal{N} \tag{5.10}$$

To lengthen the overall network lifetime, we should route messages through sensors with more remaining battery power. Hence, it is reasonable to maximize the minimum t_i, $i \in \mathcal{N}_s$.

Among these possible cost functions, in this dissertation, we focus on minimizing Eq. (5.1) in our optimization problem.

5.2.4 Discussions

Data stream X_i can be compressed at sensor i or jointly coded with other data streams en route to the fusion center. Moreover, only when flows are jointly

coded, do they need to be bundled in transmission. For instance, in Fig. 5.2, if flows f_{21}^2 and f_{31}^3 are not jointly coded, they can split to take different paths f_{15}^2 and f_{14}^3 in ensuing transmissions. As a result, the overall routing structure is not necessarily a tree, and we point out that in data-centric routing, trees are not necessarily optimal. (Refer to Fig. 6.1 for one such example.) However, the splitting of an individual flow f^{ij} is prohibited.

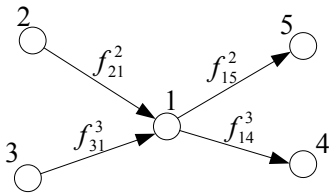


Figure 5.2: Data flows split in the network.

When applying source coding with explicit side information to sensor networks, we must avoid helping loops. In other words, if X_j's recovery relies on X_i, then X_j cannot be used as the side information for coding X_i. To formalize this idea, define a directed network $\mathcal{G}_h$ that consists of all the active sensor nodes. In addition, if X_i is used as side information for coding X_j, a directed edge (i, j) is formed from sensor i to sensor j. Then we have the following theorem:

Theorem 5.2.1 (Helping Loops) *No helping loop will be formed when using source coding with explicit side information if and only if the directed network $\mathcal{G}_h$ contains no directed cycles.*

The proof of this theorem is straightforward, and hence omitted. Special attention must be directed to this rule of no helping loops when designing data-centric routes.

We adopt an example from [LTP05] to solidify our data rate model. In Fig. 4.6, a near-field sensor array records the sound of a tank as it moves by.

Table 5.1: Data rate with different side information.

Side info	none	s_0	s_1	s_2	s_0, s_1	s_0, s_2	s_1, s_2
σ_3^2	5.33	.443	.973	3.43	.424	.431	.943
f^3(bits)	6.19	4.40	4.96	5.87	4.36	4.38	4.94

This array is part of a wireless sensor network (not shown in the figure), and their data are to be transmitted to a fusion center. The variance of observations at s_3 is listed in Table 5.1 when they are quantized alone or with side information using an adaptive DPCM encoder. The data rate is estimated by $0.5\log(\sigma_3^2/D)$, where $D = 0.001$. Notice that the coding gain varies significantly with sensor locations, and saturates as the number of helpers exceeds one. In practice, the cost of processing side information may lead to $\mathcal{H}_3 = \{s_0, s_1\}$ and using at most one helper.

5.3 Combined Routing and Source Coding

5.3.1 Problem formulation

Given the network models in section 5.2, the question becomes how to construct data transmission routes such that the total communication cost is minimized. We formulate this as an optimization problem, which is stated as follows.

Combined Routing and Source Coding (CRSC)

GIVEN: A network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with weight $c_e > 0$ defined on each edge $e \in \mathcal{E}$, a special node $t \in \mathcal{N}$ acting as the fusion center, a set $\mathcal{H}_i$ of helping sensors and data rate function f^i as in Eq. (5.4) defined for each sensor $i \in \mathcal{N}_s = \mathcal{N} \setminus \{t\}$.

FIND: Routes for transmitting side information and sensing data such that the total cost $C = \sum_{e\in\mathcal{E}} c_e f_e$ of routing data to the fusion center is minimized.

5.3.2 NP-hardness

We prove the following theorem in this section.

Theorem 5.3.1 (Complexity of CRSR) *Solving CRSC is NP-hard.*

Proof: We prove this by showing that the following special instance of CRSC is the minimum Steiner tree problem. Assume $|\mathcal{N}_a| < |\mathcal{N}_s|$ and $\mathcal{H}_i = \mathcal{N}_a$ for $i \in \mathcal{N}_a$. Define the rate function $f^i = 1$ without side information information, and $f^i = 0$ with side information for $i \in \mathcal{N}_a$.

We first show that the optimal route for this problem must be a tree. Since the data rate is an integer (0 or 1), the data flow on each edge f_e must also be an integer since the splitting of a individual flow is forbidden. Thus, the optimal route consists of a set of edges that carry flows $f_e \geq 1$. Suppose that the optimal solution is not a tree. In any solution route, there is at least one path from each $i \in \mathcal{N}_a$ to t. Otherwise X_i cannot be recovered by t. Therefore, we can find at least one tree that is embedded in the optimal solution and connects all the active sensors to t. If there is more than one such tree, we pick the one with the minimum number of edges, and use this tree as the transmission route. Since $\mathcal{H}_i = \mathcal{N}_a$ and $f^i = 0$ with side information, the flow rate remains 1 whenever flows merge. Hence, the data rate f_e on any edge of the tree is 1. Thus, the total cost is simply the weight sum of the edges on the tree, which is less than that of the optimal solution, in which the tree is embedded. This contradiction proves that the set of optimal routes must constitute a tree.

As $f_e = 1$ on every edge of the routing tree, finding the optimal routing tree is equivalent to constructing the minimum Steiner tree that connects t and all the sensors in $\mathcal{N}_a$, which is a well-known NP-hard problem. Therefore, our problem is also NP-hard. Q.E.D.

5.4 Mixed Integer Programming

We obtain one important sub-instance of the CRSC problem if the route is required to be a directed tree pointing toward the fusion center. The shortest path tree is one of the feasible solutions of this problem. In addition, for simplicity, we assume that $b_k^{ij} = b_k^i$ (i.e. rate reduction is independent of side information source) and compression on data stream X_i can only occur at sensor i. We call this subproblem the combined tree routing and source coding (CTRSC). It is easy to prove that CTRSC is also NP hard, so there is unlikely any efficient algorithm exactly solving the problem. In this section, we develop the mixed integer program [Wol98] for CTRSC, to which standard techniques, such as branch and bound, can be applied.

For CTRSC, it is more convenient to model the network as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{A}$ is the set of directed edges. If a directed edge (i, j) exists, direct transmission from i to j is allowed. In addition, we define:

$$\mathcal{O}(i) = \{j \in \mathcal{N} : \ (i, j) \in \mathcal{A}\}$$

$$\mathcal{I}(i) = \{j \in \mathcal{N} : \ (j, i) \in \mathcal{A}\}$$

Assume that the edge weights and data rate functions are properly defined. The objective is to minimize the cost of routing all the data to the fusion center.

$$\min \sum_{a \in A} f_a c_a \tag{5.11}$$

To ensure that the underlying routing structure is a tree, we use the multi-commodity flow formulation [MW95].

$$\sum_{j \in \mathcal{O}(t)} g_{tj}^k - \sum_{j \in \mathcal{I}(t)} g_{jt}^k \ = \ -1, \tag{5.12}$$

$$\sum_{j \in \mathcal{O}(k)} g_{kj}^k - \sum_{j \in \mathcal{I}(k)} g_{jk}^k \ = \ 1, \quad k \in \mathcal{N} \setminus \{t\} \tag{5.13}$$

$$\sum_{j\in\mathcal{O}(i)} g_{ij}^k - \sum_{j\in\mathcal{I}(i)} g_{ji}^k = 0, \quad i \in \mathcal{N}\setminus\{k,t\} \tag{5.14}$$

$$\sum_{a\in A} y_a = n \tag{5.15}$$

$$y_a \geq g_a^k \geq 0 \tag{5.16}$$

$$y_a = \{0,1\}, \quad a \in A \tag{5.17}$$

where t denotes the fusion center node and y_a is the binary variable indicating whether edge a is used to construct the routing tree. In this formulation, one unit of flow is generated at each sensor and consumed at the fusion center. The resulting route is a connected graph with exactly n edges, so it must be a tree. It is apparent that g_a^k can only be 1 or 0, which indicates whether edge a carries the flow generated by sensor k. With these in mind, the data flow and source coding with explicit side information is formulated as follows:

$$\sum_{j\in\mathcal{O}(i)} f_{ij} - \sum_{j\in\mathcal{I}(i)} f_{ji} = \sum_{m=0}^{k_s+1} b_m^i \lambda_m^i \tag{5.18}$$

$$\sum_{k\in\mathcal{H}_i}\sum_{j\in\mathcal{I}(i)} g_{ji}^k = \sum_{m=0}^{k_s+1} h_m^i \lambda_m^i \tag{5.19}$$

$$\sum_{m=0}^{k_s+1} \lambda_m^i = 1 \tag{5.20}$$

$$u_a y_a \geq f_a \geq 0, \quad a \in \mathcal{A} \tag{5.21}$$

$$\lambda_m^i \geq 0, \quad i \in \mathcal{N}\setminus\{t\} \tag{5.22}$$

where u_a is a large constant, and Inequality (5.21) is used to ensure that data only flows across the edges belonging to the prescribed spanning tree. k_s is the maximum number of sensors that provide side information for compressing any data stream. The data rate of sensor i is given by $f^i = \sum_{m=0}^{k_s+1} b_m^i \lambda_m^i$, and the number of helpers $h^i = \sum_{m=0}^{k_s+1} h_m^i \lambda_m^i$. If b^i is a convex function of h^i as shown in Fig. 5.3, in virtue of the cost minimization, exactly one λ_m^i will be 1 for each

Table 5.2: The coding model in Fig. 5.3.

m	0	1	2	3	4
h_m^i	0	1	2	3	M
b_m^i	b_0^i	b_1^i	b_2^i	b_3^i	b_3^i

sensor i. If b^i is an arbitrary function of h^i, an additional constraint called the special ordered set needs to be placed on λ_m^i [Wil99]. It simply stipulates that for each i at most two λ_m^i with adjacent m values be non-zero. This complicates the formulation somewhat, but is one of the standard modeling techniques in integer programming. For the data rate function in Fig. 5.3, we have $k_s = 3$, and the values of h_m^i and b_m^i are given in Table 5.2.

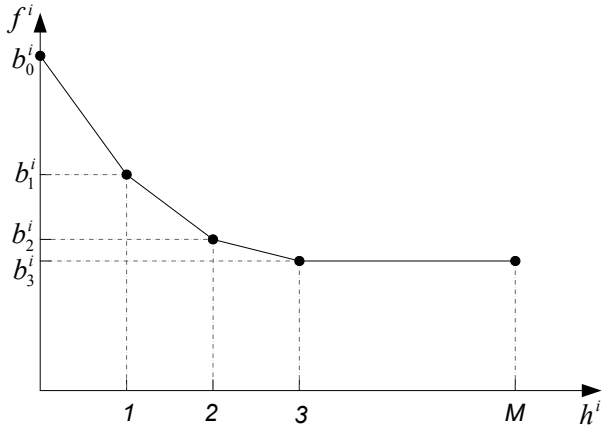


Figure 5.3: A convex rate reduction model. $k_s = 3$. The coding gain saturates when number of helping sensors exceeds 3.

5.5 Conclusion

Our study continues the recent development of data-centric routing. The data transmissions are decomposed into individual flows originated at different sensors

to build a simplified first order rate model. It is assumed that side information from only a small number of sensors can be used to effectively compress the data. Based on this model, an optimization problem CRSC is formulated and shown to be NP-hard. Mixed integer program is developed for a sub-instance of the CRSC problem. In the next chapter, we will discuss heuristic algorithms for constructing routes that result in small communication cost.

CHAPTER 6

Heuristic Algorithms for CRSC

6.1 Introduction

Since finding the exact solution of the CRSC problem in polynomial time is unlikely, we turn to heuristics in this chapter. We first present the SPT (shortest path tree) and Clustering methods that have been extensively studied in the routing literature. Then, two methods, which are called balanced aggregation scheme (BAS) and designated side information transmission (DSIT) are proposed.

Most previous work has considered using trees as the underlying routing structure [KEW02, CBV04, GE03] probably because trees are the optimal solution to the shortest path problem and have been pervasive in network routing. However, in data-centric routing, trees are not necessarily optimal. In this chapter, two strategies are proposed. One is called balanced aggregation scheme (BAS), and the other designated side information transmission (DSIT). Both methods can result in non-tree routing structures. To motivate the idea and give a preview of the chapter, consider the example depicted in Fig. 6.1. The edges between adjacent sensors (circles) have the weight $c_e = d$, and the edges connecting a sensor to the fusion center (square) have the weight $c_e = D$. The rate at which each sensor needs to transmit to the fusion center is R without explicit side information and r if explicit side information from an adjacent sensor is available. Assume $r \ll R$ and $d \ll D$. The objective is to minimize the cost $C = \sum_e c_e f_e$ of routing all

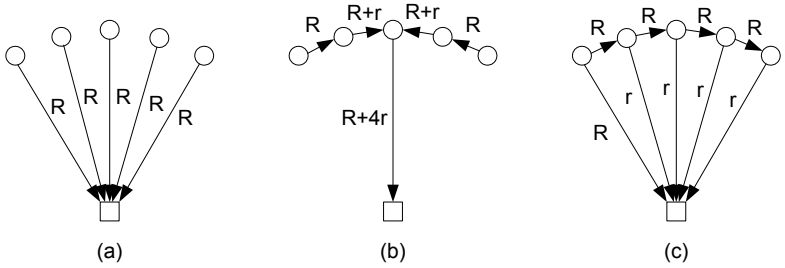


Figure 6.1: Three routing strategies: (a) SPT; (b) BAS; (c) DSIT.

the data to the fusion center, where f_e is the rate at which data are transmitted across edge e. Consider the three strategies described in Fig. 6.1. In (a), the shortest path tree (SPT) is used. In (b), before being routed to the fusion center, sensing data are aggregated at relaying nodes to reduce the communication cost. In (c), each sensor except for the rightmost one transmits its data to the sensor at its right. This transmission to the adjacent sensor provides explicit side information for data compression at the recipient and needs not to be relayed to the fusion center. Note that at least one sensor has to transmit at rate R to the fusion center so that all the data can be correctly recovered. The costs of the three strategies are: (a) $C = 5RD$; (b) $C = RD + 4rD + 4Rd + 2rd$; (c) $C = RD + 4rD + 4Rd$. The performances of strategies in (b) and (c) are about the same, and both are superior to that of (a). It is also evident that the scheme in (c) results in more evenly distributed traffic than that in (b). This is because in (c), the communication to the fusion center is separated from the explicit side information transmission, and can be routed through any path.

The rest of the chapter is organized as follows. In section 6.2, the SPT and clustering method are presented. Then, we discuss BAS and DSIT methods in section 6.3 and 6.4 respectively. The average performances of these two algorithms are studied and compared to the SPT and clustering method through

simulations in section 6.5. The chapter is concluded by section 6.6.

6.2 SPT and Clusters

6.2.1 SPT

A shortest path tree is used to route data to the fusion center t, and data compression is performed whenever explicit side information is available due to the merging of flows in the network. We establish a result regarding SPT's worst case performance when the rate model is given by:

$$f^i = \begin{cases} b_0 & \text{without side information} \\ \beta b_0 & \text{with side information} \end{cases}, \quad i \in \mathcal{N}_a \tag{6.1}$$

where $0 \leq \beta \leq 1$.

Theorem 6.2.1 (Performance Bound of SPT) *The costs of the SPT and optimal solution satisfy the following relation ($n_a = |\mathcal{N}_a|$):*

$$C_{OPT}/C_{SPT} \geq \beta + (1-\beta)/n_a \tag{6.2}$$

This bound is tight in that there are cases in which the equality in Eq. (6.2) holds.

Proof: Denote by $\mathcal{E}_{\text{ST}}$ the set of edges in the Steiner tree (ST) that connects all active sensors $\mathcal{N}_a$ to t, and d_i the distance from i to t on the SPT.

$$C_{\text{SPT}} \leq b_0 \sum_{i \in \mathcal{N}_a} d_i \leq b_0 n_a \sum_{e \in \mathcal{E}_{\text{ST}}} c_e \tag{6.3}$$

The first inequality is straightforward, and the second is due to the following relation [HRW92]:

$$\sum_{e \in \mathcal{E}_{\text{ST}}} c_e \geq \max_{i \in \mathcal{N}_a} d_i$$

On the other hand, we have the following for a optimal route.

$$C_{\text{OPT}} \geq \beta b_0 \sum_{i \in \mathcal{N}_a} d_i + b_0(1-\beta) \sum_{e \in \mathcal{E}_{\text{ST}}} c_e \tag{6.4}$$

where the first term is the minimum cost of routing data to the fusion center, and the second term represents the minimum cost of routing side information. Eq.s (6.3) and (6.4) lead to (6.2).

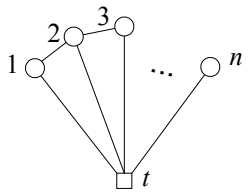


Figure 6.2: An instance achieves the bound in Eq. (6.2): $\beta = 0$; $\mathcal{N}_a = \{1, \cdots, n\}$; $\mathcal{H}_i = \mathcal{N}_a$, $c_{it} = 1, i \in \mathcal{N}_a$; $c_{k,k+1} = \epsilon$, $1 \leq k \leq (n-1)$.

To show that the bound on the performance ratio is tight, consider the instance depicted in Fig. 6.2. The worst case scenario in Eq. (6.2) is attained when $\epsilon \to 0$. Q.E.D.

It is no surprise that the worst case performance ratio is a strong function of β, which indicates the level of the coding gain. After all, the performance ratio represents the penalty that a SPT receives for ignoring the data correlation in designing transmission routes.

6.2.2 Clusters

Various clustering methods have been proposed [BC03, HCB00]. In this chapter, we use a clustering method based on geographical proximity. The sensing field is divided into rectangular cells, and sensors that fall in the same cell are grouped into a cluster. The sensor with the smallest distance to the fusion center is picked

as the cluster head. In each group, sensors transmit to the cluster head to have their data fused, and the fused data are subsequently sent to the fusion center.

To enforce the rule in Theorem 5.2.1 for the SPT and clustering methods, we label the set of sensors $\mathcal{N}_a$ according to some order such that (1) there is a unique relation $i < j$ defined for any pair $i, j \in \mathcal{N}_a$; (2) if $i < j$ and $j < k$, then $i < k$ for $i, j, k \in \mathcal{N}_a$. It is postulated that i can be in set $\mathcal{H}_j$ only when $i < j$.

6.3 Balanced Aggregation Scheme

6.3.1 Motivation

In this section, we propose an approximation algorithm that is inspired by the idea of balancing shortest path trees and trees with small total weights [KRY95]. In this algorithm, all transmissions terminate at the fusion center. Data aggregations are performed when messages are relayed toward t.

To motivate the algorithm, we assume $b_1^{ij} = b_1^i$ in Eq. (5.4), and decompose the flow f_e^i into two parts.

$$f_e^i = q_e^i + r_e^i$$

When $f_e^i = 0$, $q_e^i = r_e^i = 0$. Otherwise, we define

$$q_e^i = b_1^i$$

$$r_e^i = \begin{cases} b_0^i - b_1^i & \text{no side information} \\ 0 & \text{with side information} \end{cases}$$

Thus, q_e^i represents the portion of f_e^i that is independent of the side information, and r_e^i is the part that is compressible by the helper's data. Accordingly, the

total cost of routing data to t can be decomposed into $C = C_q + C_r$, where

$$C_q = \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{N}_a} c_e q_e^i, \quad C_r = \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{N}_a} c_e r_e^i$$

Consider, for a moment, minimizing the costs C_q and C_r separately. C_q is minimized when the route is a set of shortest paths. On the other hand, to obtain a small C_r, we should try to jointly code X_i and $X_j, j \in \mathcal{H}_i$, and merges flows using routes that have small weights. This resembles a Steiner tree problem, but each aggregation involves only a subset of active sensors. We apply this to the two extreme cases of minimizing C. When coding gain is small ($b_1^i \gg b_0^i - b_1^i$ for $i \in \mathcal{N}_a$), the route is expected to be close to a sub-tree of SPT. Whereas, when there is substantial coding gain ($b_1^i \ll b_0^i - b_1^i$), the focus is on achieving aggregation with small routing cost. For the general case of varying coding gains, we speculate that an approximation to the optimal solution can be obtained by constructing balanced aggregation routes that have small total weights and reasonable distance from each sensor to the fusion center, and the appropriate balance is struck based on the relative values of b_0^i and b_1^i, $i \in \mathcal{N}_a$.

6.3.2 Constructing balanced paths

We first examine how to route a sensor's data to t using an existing path while taking into account the data compression. In Fig. 6.3, there is a path connecting the active sensor k to t. Define $\mathcal{P}_k = \{k, (k, v_1), v_1, \cdots, v_p, (v_p, t), t\}$ the sequence of nodes and edges on the path. Denote by $d_{uv}^{\mathcal{P}_k}$ $(u, v \in \mathcal{P}_k)$ the distance from u to v along path $\mathcal{P}_k$. Set $d_u^{\mathcal{P}_k} = d_{ut}^{\mathcal{P}_k}$. We want to find a path to route the data of active sensor i to t such that the resulting cost is minimized. This amounts to determining an aggregation node $j \in \mathcal{P}_k$ where the two flows f^k and f^i joins one another. There are two possible situations.

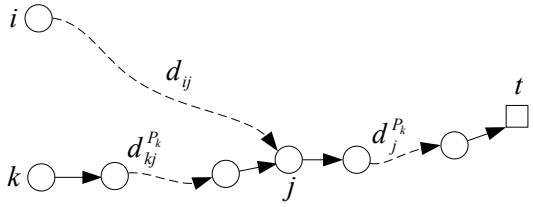


Figure 6.3: Construct a balanced path for i.

(1) X_i and X_k are uncorrelated. The optimal route is the shortest path from i to t, and the cost is:

$$C_{ik} = b_0^i d_i$$

(2) X_i and X_k are correlated. If the two flows merge at node $j \in \mathcal{P}_k$, the cost of routing f^i to t is:

$$C_{ij} = b_0^i d_{ij} + b_1^{ik} d_j^{\mathcal{P}_k}$$

For (2), we choose $\arg\{\min_{j\in\mathcal{P}_k} C_{ij}\}$ as the aggregation node and define $C_{ik} = \min_{j\in\mathcal{P}_k} C_{ij}$.

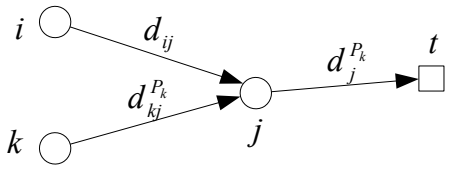


Figure 6.4: An aggregation tree constructed from Fig. 6.3.

As we discussed in section 5.2.4, the resulting route may not be a tree. For example, the path from i to j may have used some nodes in $\mathcal{P}_k$ as relays. However, the routes can be transformed into a tree as follows. Remove all the nodes except for i, j, k, and t. Directed edges are formed from i to j, from k to j, and from j to t. Assign the corresponding distances on the paths as weights to these edges. The resulting tree is shown in Fig. 6.4. Note that each edge carries a constant

data flow, and this aggregation tree captures the essential picture of how the data aggregation takes place in the network.

6.3.3 Balanced aggregation scheme

The complete algorithm involves successive steps of adding the routes of active sensors to the aggregation tree. Each time, the newly added sensor has the smallest additional cost among all the remaining sensors. We state our algorithm as follows.

Balanced Aggregation Scheme (BAS)

Given a graph $\mathcal{G}$ with edge weights and data rate functions properly set, define $\mathcal{U} = \mathcal{N}_a$ and $C = 0$. Carry out the following steps.

(1) Find the shortest path from each active sensor to t. Denote by I the sensor that has the minimum routing cost, $C_i = b_0^i d_i$, $i \in \mathcal{N}_a$.

(2) $C = C + C_I$. Remove I from $\mathcal{U}$, and add I's path to the solution route. If $\mathcal{U}$ is empty, stop the algorithm.

(3) For each sensor $i \in \mathcal{U}$, find C_{ik} resulting from merging f^i with f^k, $k \in \mathcal{N}_a \backslash \mathcal{U}$. Compute

$$C_i = \min_{k \in \mathcal{N}_a \backslash \mathcal{U}} C_{ik}$$

(4) Find $I = \arg\{\min_{i \in \mathcal{U}} C_i\}$. Return to step (2).

Since exactly one active sensor is removed from $\mathcal{U}$ during each iteration, we can number active sensors according to the order that their routes are added to the solution route. If we construct the graph $\mathcal{G}_h$ in Theorem 5.2.1 for the route built from BAS, a directed edge (i, j) is possible only when sensor i has been

marked with a smaller number than j. It is easy to show that such a $\mathcal{G}_h$ contains no directed loops. Thus, no helping loop is formed using BAS.

BAS completes in n_a iterations. During each round, there is no need to compute C_i, $i \in \mathcal{U}$ in step (3) all over again. It suffices to update C_i such that the newly added path in the previous iteration is accounted for. The bottleneck of BAS is on constructing shortest paths to each active sensor. Using Dijkstra's algorithm, it runs in $O(n_a m \log n)$ time. The distributed implementation of BAS also relies on efficient parallel algorithms for constructing shortest paths between pairs of sensors, which is a well-researched topic. We refer readers to, for example, [Hal97] for more discussions on the subject.

For a set of sensors with uncorrelated data, BAS builds the shortest path from each active sensor to the fusion center. For the special instance that results in the minimum Steiner tree problem in section 5.3.2, BAS collapses to the shortest path heuristic [TM80], which has a worst case performance ratio of $C_{\text{BAS}}/C_{\text{OPT}} = [2 - 2/(n_a + 1)]$. Although not proven, we suspect that this is also the worst case performance ratio of BAS for the general CRSC problem.

6.4 Designated Side Information Transmission

6.4.1 Motivation

In this section, we take up the approach in Fig. 6.1 (c). Data flows from active sensors to t are routed independently and do not merge with one another. Side information transmissions are carried out on designated routes, and these transmissions need not to be relayed to the fusion center. Accordingly, the total routing cost can be decomposed into the cost of routing explicit side information

C_s and the cost of transmitting data to the fusion center C_t:

$$C = C_s + C_t \tag{6.5}$$

where

$$C_s = \sum_{i,j \in \mathcal{N}_s} \sum_{e \in \mathcal{E}} c_e f_e^{ij}, \quad C_t = \sum_{i \in \mathcal{N}_s} \sum_{e \in \mathcal{E}} c_e f_e^{i} \tag{6.6}$$

In addition to achieving good performance when there is high data correlation in the network, DSIT offers greater flexibilities than strategies that bundle the flows to the fusion center and carrying side information. For instance, data flow f^i can virtually be routed to t through any path. As a result, traditional address-centric routing schemes that are designed to evenly distribute the traffic load in the network and maximize node lifetime [CT00, SWR98] can be readily applied.

6.4.2 Designated side information transmission

We first consider constructing routes for transmissions to the fusion center. These routes affect only C_t. In addition, as $f^i, i \in \mathcal{N}_a$ does not provide any side information, its routing is decoupled from the data aggregation process. Hence, the shortest path should be used to achieve the minimum C_t:

$$C_t = \sum_{i \in \mathcal{N}_a} d_i f^i \tag{6.7}$$

where d_i is the minimum distance from sensor i to t, and f^i is a function of the side information transmission.

Designing routes for side information transmission is more complicated. First, it has the minimum Steiner tree as a subproblem. This is illustrated by the following problem instance. Given network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, we have a subset of the active sensors $\mathcal{S} \subset \mathcal{N}_a$, and there is a sensor $u \in \mathcal{N}_a \setminus \mathcal{S}$. Assume $\mathcal{H}_i = u$ if $i \in \mathcal{S}$, and $\emptyset$ if $i \in \mathcal{N}_a \setminus \mathcal{S}$. In addition, we assume that the rate function

and edge weights are defined such that the cost of transmitting side information from u to any sensor in $\mathcal{S}$ using appropriately chosen routes is less than the cost reduction resulting from the coding gain of side information. The optimization problem becomes constructing a subtree that connects u and the sensors in $\mathcal{S}$, which is a minimum Steiner tree. Therefore, the overall optimization problem is NP-hard. Second, we need to ensure that no helping loop is formed while routing the side information. This amounts to avoiding directed cycles in G_h according to Theorem 5.2.1.

For a moment, we ignore the Steiner tree part, and use shortest paths to route all the side information. This leads to constructing a network $\mathcal{G}_a$ as follows. $\mathcal{G}_a$ consists of the set of active sensors $\mathcal{N}_a$. In addition, for each ordered pair of nodes $i, j \in \mathcal{N}_a$, create a directed edge (i, j) from sensor i to j and assign the weight w_{ij} to represent the net coding gain resulting from routing side information from i to j.

$$w_{ij} = \begin{cases} (b_0^j - b_1^{ji})d_j - d_{ij}b_0^i & i \in \mathcal{H}_j \\ -d_{ij}b_0^i & \text{otherwise} \end{cases} \tag{6.8}$$

Denote by $\mathcal{A}_a$ the set of directed edges with $w_{ij} > 0$, and define $\mathcal{G}_a = (\mathcal{N}_a, \mathcal{A}_a)$. A branching on the directed graph $\mathcal{G}_a$ is a set of directed edges $\mathcal{B} \subseteq \mathcal{A}_a$ satisfying the conditions that no two edges in $\mathcal{B}$ enter the same node, and $\mathcal{B}$ has no directed cycle. It is evident that a branching on $\mathcal{G}_a$ represents a feasible set of routes for side information transmission. No two directed edges in $\mathcal{B}$ entering the same node ensures that a sensor uses side information from at most one helper, and no directed cycle avoids the helping loop. The problem of minimizing the total cost is equivalent to maximizing the weight sum of the branching $\mathcal{B}$, which is the so called maximum weight branching problem.

Maximum Weight Branching (MWB)

GIVEN: A directed graph $\mathcal{G}_a = (\mathcal{N}_a, \mathcal{A}_a)$ with a weight w_e defined on each directed edge $e \in \mathcal{A}_a$.

FIND: A branching $\mathcal{B} \subseteq \mathcal{A}_a$ that maximizes $\sum_{e \in \mathcal{B}} w_e$.

An algorithm that solves this problem in polynomial time has been independently proposed by [CL65] and [Edm67]. The combinatorial proof of this algorithm's optimality was provided by [Kar71], and efficient implementations were described in [Tar77] and [GT88]. Once the optimal branching $\mathcal{B}$ is determined, we revert to using Steiner trees. Define $\mathcal{S}_i$ as the set of sensors that receive side information from $i \in \mathcal{N}_a$ based on the optimal branching $\mathcal{B}$. We use the shortest path heuristic [TM80] to construct the subtree that connects k and $\mathcal{S}_k$. Our heuristic algorithm is a combination of the maximum weight branching and the Steiner tree approximation. We state it as follows:

Designated Side Information Transmission (DSIT Heuristic)

Given a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with edge weights and rate functions properly defined, carry out the following steps.

(1) Find the shortest path from each active sensor to the fusion center. These are the routes for transmitting data to the fusion center.

(2) Construct a directed graph $\mathcal{G}_a = (\mathcal{N}_a, \mathcal{A}_a)$. $\mathcal{N}_a$ are the set of active sensors. Construct directed edges from i to j ($i, j \in \mathcal{N}_a$ and $i \neq j$), and assign weights w_{ij} according to Eq. (6.8). $\mathcal{A}_a$ consists of the set of edges whose weights are greater than zero.

(3) Find the maximum weight branching $\mathcal{B}$ on $\mathcal{G}_a$. Based on $\mathcal{B}$, determine the set of sensors $\mathcal{S}_i$ that each active sensor $i \in \mathcal{N}_a$ transmits side information

to.

(4) Run a shortest path heuristic for the Steiner tree problem to find the subtree for transmitting side information from $i \in \mathcal{N}_a$ to the sensors in $\mathcal{S}_i$.

6.4.3 Performance analysis

Finding the maximum weight branching takes $O(m_a \log n_a)$ time, where $m_a = |\mathcal{A}_a|$. The shortest path heuristic for a Steiner tree requires $O(n_a m \log n)$ time for a sparse network. The actual running time of the shortest path heuristic is in general much less because the number of nodes involved in constructing the shortest path is $|S_i|$, and thus often a lot smaller than n. Therefore, the computational cost of DSIT and BAS are on the same order. The distributed implementation of DSIT requires efficient parallel algorithms for constructing not only shortest paths but also optimum branchings. For the latter, we refer readers to [Hum83], which discusses a distributed version of the Edmonds' method [Edm67].

Regarding the performance of our heuristic algorithm comparing to that of the optimal solution attainable using a DSIT strategy, we prove the following theorem.

Theorem 6.4.1 (Performance Bound of DSIT Heuristic) *The ratio of the cost C_{DSIT} resulting from our DSIT heuristic algorithm and the minimum cost C_{MIN} using the DSIT strategy is bounded by:*

$$C_{DSIT}/C_{MIN} \leq N \tag{6.9}$$

where $N = \max\{1, \max_{i \in \mathcal{N}_a} |\mathcal{S}_i^{opt}|\}$, the greater of one and the maximum number of sensors that one sensor needs to transmit side information to in the optimal solution. The bound is tight in the sense that there is a problem instance that attains the worst case performance ratio.

Proof: It is first noted that $\mathcal{S}_i^{\text{opt}}$ is in general not the same as the $\mathcal{S}_i$ in our heuristic algorithm, and C_{MIN} is the optimal result achievable by the DSIT approach, which may be greater than the minimum cost of the CRSC problem.

In the optimal DSIT solution, the side information is routed from i to $\mathcal{S}_i^{\text{opt}}$ using the minimum Steiner tree, and data is transmitted from active sensors to the fusion center through shortest paths. Denote by $\mathcal{E}_i^{\text{ST}}$ the set of edges of the Steiner tree for circulating side information supplied by sensor $i \in \mathcal{N}_a$. Define $C_i^{\text{ST}} = \sum_{e \in \mathcal{E}_i^{\text{ST}}} c_e$, the sum of edge weights of the Steiner tree, and $C_i^{\text{ST}} = 0$ if $\mathcal{S}_i^{\text{opt}} = \emptyset$. We can write the minimum cost as follows:

$$C_{\text{MIN}} = \sum_{i \in \mathcal{N}_a} f^i d_i + \sum_{i \in \mathcal{N}_a} b_0^i C_i^{\text{ST}} \tag{6.10}$$

Instead of the Steiner tree, consider relying on a shortest path tree to route the side information from i to the sensors in $\mathcal{S}_i^{\text{opt}}$. Denote by C_i^{SPT} the sum of edge weights of such shortest path trees. The corresponding cost C' will be:

$$C' = \sum_{i \in \mathcal{N}_a} f^i d_i + \sum_{i \in \mathcal{N}_a} b_0^i C_i^{\text{SPT}} \tag{6.11}$$

Since $N_i C_i^{\text{ST}} \geq C_i^{\text{SPT}}$ [TM80], where $N_i = |\mathcal{S}_i^{\text{opt}}|$, we have

$$C'/C_{\text{MIN}} \leq N = \max\{1, \max_{i \in \mathcal{N}_a} N_i\}$$

On the other hand, C_{DSIT} is at least as good as the optimal result of using the shortest path tree to route the side information. Therefore, $C_{\text{DSIT}} \leq C'$. Together with above inequality, this gives rise to the bound in Eq. (6.9).

To show the bound is tight, we consider the instance in Fig. 6.5. The network setup is given in (a). The edge weights between sensors v_k and u_k $(k = 1, 2, 3)$ is 1. Other edges have weight $\delta \ll 1$. All the sensors are active with data rate R without side information and 0 when side information is available. Denote

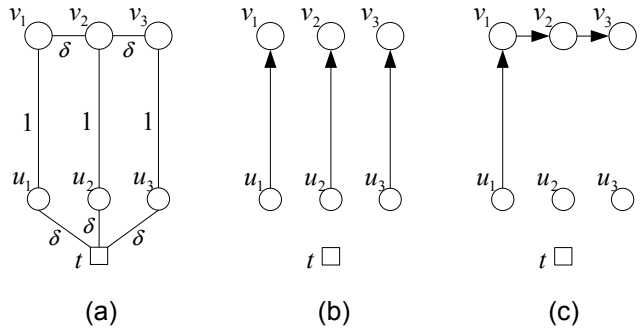


Figure 6.5: A problem instance that attains the worst performance ratio: (a) sensor network setup; (b) routes of side information transmission using DSIT heuristic; (c) routes of side information transmission in the optimal solution.

$\mathcal{U} = \{u_1, u_2, u_3\}$, and $\mathcal{V} = \{v_1, v_2, v_3\}$. We assume $\mathcal{H}_i$ is $\mathcal{U}$ when $i \in \mathcal{V}$, and $\mathcal{V}$ when $i \in \mathcal{U}$. In Fig. 6.5, (b) and (c) illustrate how side information is transmitted in the DSIT heuristic and optimal DSIT solution. Accordingly, $C_{\text{DSIT}} = 3R+3R\delta$ and $C_{\text{MIN}} = R + 5R\delta$. When $\delta \to 0$, the ratio $C_{\text{DSIT}}/C_{\text{MIN}}$ approaches $N = 3$ asymptotically. In a similar fashion, problem instances with arbitrary values of N can be devised. Q.E.D.

The worst case scenario in the proof can be avoided by running multiple maximum weight branching and shortest path heuristic iterations in the DSIT heuristic. At each iteration, only one sensor is added to $\mathcal{S}_i, i \in \mathcal{N}_a$. However, this greatly increases the computational cost. Moreover, the pathological case in Fig. 6.5, where high correlation exists between sensors that are far away from one another, rarely occurs in our assumed data rate model. The value of N is expected to be small as one's data helps mostly nearby sensors. Also since side information is often circulated within one's neighborhood, using shortest paths to approximate a Steiner tree introduces a moderate amount of error. What we are more interested in is the average behavior of the algorithm, which is examined through simulations in the next section.

6.5 Simulations

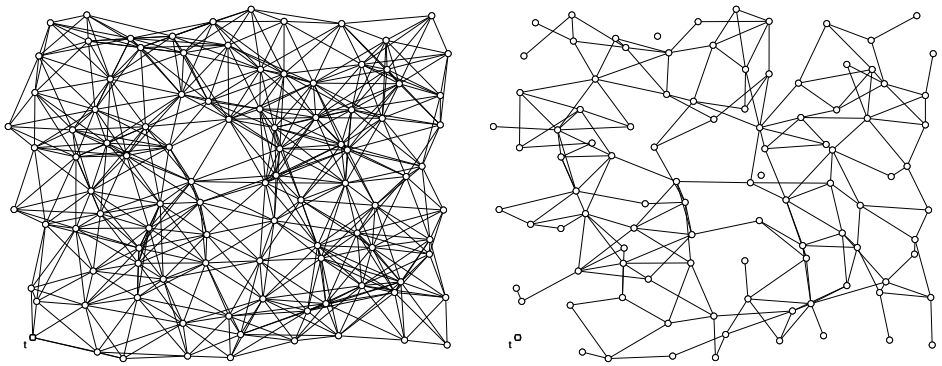
6.5.1 Simulation setup

In our simulations, we place $(n+1)$ nodes including the fusion center and n sensors in an $n_d \times n_d$ square, where $n_d = \lceil \sqrt{n+1} \rceil$. (Denote by $\lceil z \rceil$ the smallest integer such that $\lceil z \rceil \leq z$, and $\lfloor z \rfloor$ the largest integer such that $\lfloor z \rfloor \geq z$.) Supposing $\tilde{x}_i$ and $\tilde{y}_i, i = 1, \cdots, n+1$, are random variables that are uniformly distributed in $[0, 1]$, the coordinates of node i is given by:

$$\begin{aligned} x_i &= [(i \mod n_d) - 1] + \tilde{x}_i \\ y_i &= \lfloor (i-1)/n_d \rfloor + \tilde{y}_i \end{aligned}$$

We define a transmission radius r_c. If two nodes are no more than r_c away from each other, direct communication between the two nodes is allowed. Otherwise, a relay has to be used. Denote by d_e the Euclidean length of edge e. When $d_e \leq r_c$, the edge weight c_e is proportional to d_e^α, where $\alpha = 2$ is the path loss factor. When the number of sensors increases, the network covers a larger area while maintaining the communication range and sensor to sensor spacing. A typical 100 node network constructed in this manner is depicted in Fig. 6.6(a). The node in the lower left corner is the fusion center.

In our simulation, we assume that all the sensors are active. The helping set H_i of sensor i is defined as follows. Any pair of sensors that are no more than r_d away from one another has a probability of p_h to be in the helping sets of one another. Fig. 6.6(b) shows the resulting data correlation in the network. For simplicity, the data rate function in Eq. (6.1) is used.



(a) (b)

Figure 6.6: Simulation setup: $r_c = \sqrt{5}$, $r_d = 1.8$, $p_h = 0.5$. Two nodes are connected if (a) direct transmission is allowed; (b) their data are correlated.

6.5.2 Simulation results

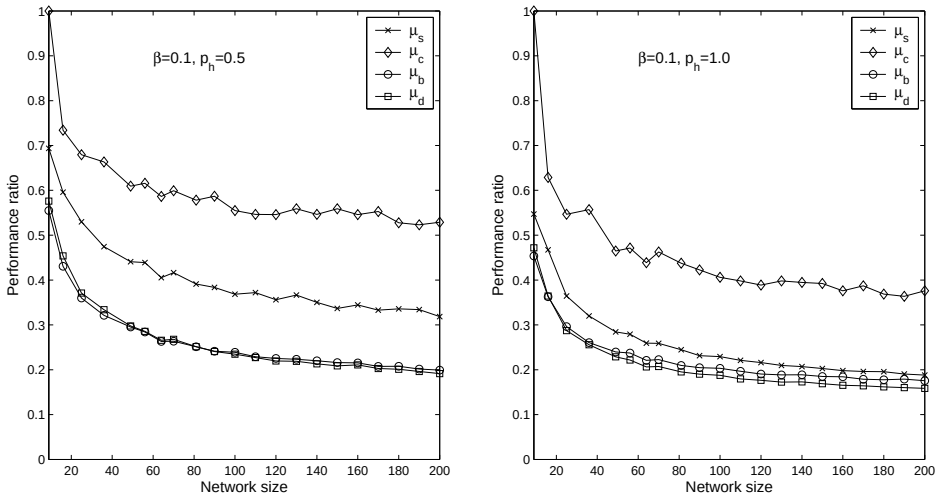
Denote by C_{SPT}, C_{CLU}, C_{BAS}, and C_{DSIT} the routing cost of SPT, Cluster, BAS and DSIT heuristics. Define C_{ASP} to be the cost of routing data when an address centric SPT, in which no data compression is performed, is used. The performance ratios of heuristic algorithms to address centric SPT are computed:

$$\mu_s = \frac{C_{\mathrm{SPT}}}{C_{\mathrm{ASP}}},\ \mu_c = \frac{C_{\mathrm{CLU}}}{C_{\mathrm{ASP}}},\ \mu_b = \frac{C_{\mathrm{BAS}}}{C_{\mathrm{ASP}}},\ \mu_d = \frac{C_{\mathrm{DSIT}}}{C_{\mathrm{ASP}}}$$

We simulate for different network sizes and vary the values of β and p_h. The performance ratios, averaged over 20 randomized network setups, are plotted under different conditions.

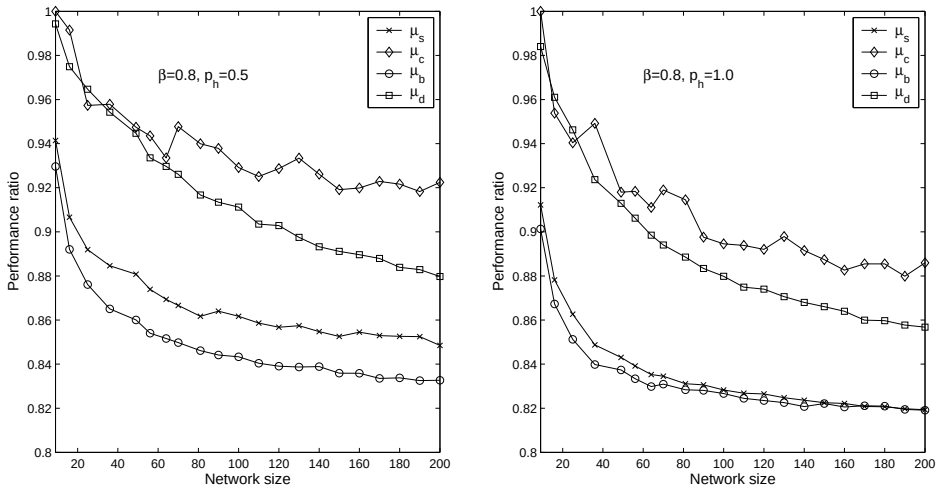
The simulation results under high coding gain, $\beta = 0.1$, are plotted in Fig. 6.7, where $p_h = 0.5$ in (a) and 1 in (b). The wiggling of μ curves is mainly due to the irregular sensor distribution when $\sqrt{n+1}$ is not a integer. It has a more pronounced effect on the clustering method as it results in uneven number of sensors in different clusters. It is apparent that the all data-centric algorithms result in significant gains over the address-centric SPT, and the performance ratio

improves as network size increases. The latter is because the data rate reduction affects the total cost more as the average distance to the fusion center increases. The ratios will eventually saturate before reaching β. Both BAS and DSIT are superior to SPT. However, we notice that the Clustering method has the worst performance among data-centric schemes. This is not surprising considering that we group sensors based solely on geographical proximity. Possible improvements include taking into account data correlation in forming clusters, varying cluster size, and using data-centric routes for intra-cluster data transmission etc. The SPT fares fairly well for our network topology and source correlation. Sensors in neighborhoods have good chances of quickly merging their flows, which leads to data compression and cost reduction. When p_h is raised from 0.5 to 1, significant performance improvement is observed for the Clustering method only. It appears that the benefit of data correlation has been exploited fully in other methods. Thus, increasing the size of $\mathcal{H}_i$ produces little additional gain.



(a) (b)

Figure 6.7: Performance ratios plotted against network size when coding gain is high, $\beta = 0.1$: (a) $p_h = 0.5$; (b) $p_h = 1.0$.



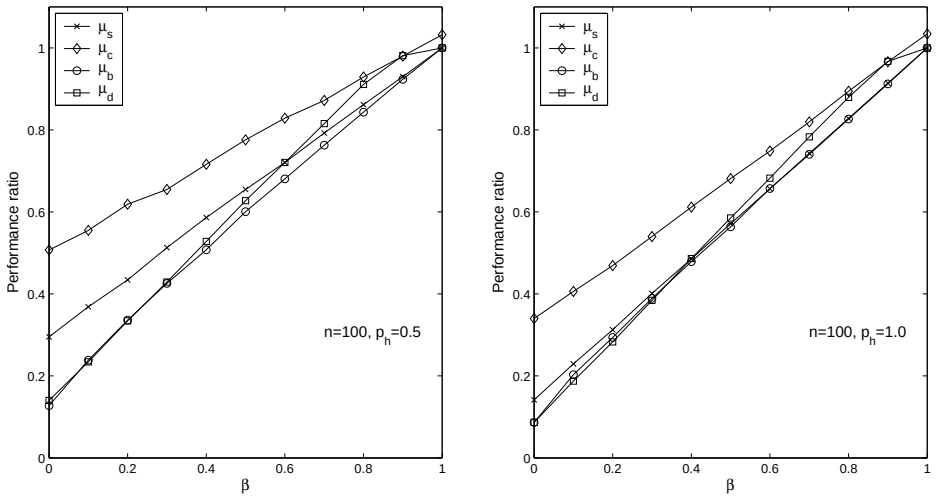
(a) (b)

Figure 6.8: Performance ratios plotted against network size when coding gain is low, $\beta = 0.8$: (a) $p_h = 0.5$; (b) $p_h = 1.0$.

The performance ratios are plotted under small coding gain, $\beta = 0.8$, in Fig. 6.8. The DSIT method suffers the most from the decrease of data correlation. It performs worse than the SPT algorithm. However, BAS still produces the best result.

To better illustrate the effect of coding gain on different schemes, we simulated with a network of 100 nodes, and plot μ as a function of β in Fig. 6.9. It is observed that all performance ratios increase monotonically with β, which is expected. BAS has the smallest ratio under almost all coding gain conditions. DSIT has about the same performance as BAS when β is small, but μ_d surpasses μ_s at moderate coding gain and eventually converges to 1 together with μ_s and μ_b. We also observe that $\mu_d, \mu_s, \mu_b \leq 1$. This is because these three algorithms never perform worse than the address-centric SPT under any coding gain. In contrast, μ_c becomes greater than 1 when β is close to 1. The gaps between different algorithms narrow when the increasing of p_h leads to more potential

helpers.



(a) (b)

Figure 6.9: Performance ratios plotted against β: (a) $p_h = 0.5$; (b) $p_h = 1.0$.

6.5.3 Discussion

Our simulations highlights the importance of coding gain information in designing data-centric routes. The clustering method based on geographical proximity illustrates that serious performance loss may occur when routing is ill-advised.

The SPT fares fairly well when $p_h = 1$. In a SPT, sensors in neighborhoods have good chances of quickly merging their flows, which leads to data compression and cost reduction. However, in practice, sensors in a network may carry out different tasks and collect various types of data, so it is more common that p_h is less than 1. In addition, our simulations are conducted on networks where sensors are evenly distributed, and cases similar to the worst scenario in Fig. 6.2 rarely occur.

BAS and DSIT methods yield good results under high coding gain and con-

verge to SPT when coding gain diminishes. The two schemes' ability to take advantage of data correlation is evident when p_h drops from 1 to 0.5. While decreasing the number of potential helpers gives rise to significant performance loss for SPT and clustering methods, μ_b and μ_d only dip slightly. Although our simulations show that the performance of DSIT deteriorates when β becomes greater than 0.5, the performance loss is relatively benign due to the low coding gain and the convergence of μ_d to 1. Although we only present centralized algorithms for BAS and DSIT, distributed implementations are not difficult to devise given the extensive research that have been conducted on SPT and MWB.

6.6 Conclusion

Based on the model in the previous chapter, we proposed a balanced aggregation scheme and an algorithm that separately routes side information for cost minimization. Simulations show that both methods work effectively in high coding gain situations. In particular, the balanced aggregation scheme achieves good results under all levels of data correlation. Our study also highlights the importance of coding gain information in designing data-centric routes. Serious performance loss may occur when routing is ill-advised.

CHAPTER 7

Concluding Remarks and Future Directions

In this dissertation, we studied the problem of efficiently utilizing resources in wireless sensor networks to observe and estimate physical phenomena. Specifically, we discussed estimation fidelities in successive steps of source estimation, adaptively sampling a distributed field, conducting data aggregation through local processing, and constructing data-centric routes for transmitting data to the fusion center. Although a sub-optimal approach that consider sampling, coding, and routing separately is taken due to the theoretical difficulty and computational complexity of the unified approach, we observe that these three steps are interwoven with one another during the design process.

We started with the study on various types of distortion associated with sensing, source coding, and field reconstruction in wireless sensor networks in chapter 2. The bounds on sensing and coding error were determined by the limited resources such as network density, node energy, and communication capacity etc. In turn, these constraints circumscribe the mesh size in approximation and set the bound of estimation error. Many topics for future research in this general area suggest themselves. Different sensing models can be proposed, which will lead to different behaviors for the sensing error. Practical local fusion algorithms can be designed and compared to the rate bounds. In addition, we considered only source coding in the chapter, but channel coding can enter the picture either by setting the limits on quantization rate or joint source-channel coding. Lastly,

the convergence of distortion in other interpolation schemes (besides the cubic spline) in the presence of sensing and quantization noise can also be studied.

Then, in chapter 3, an algorithm based upon the Bayesian framework was proposed to adaptively sample the sunlight field using mobile sensors. During each step of the algorithm, the most desirable set of sampling sites are picked from a candidate pool based on the probability of convergence in the Voronoi cells centered at candidate sits. Various extensions of this scheme can be made. For example, it can be easily modified to be used in a static wireless sensor network where sensor nodes are woken up from sleep to take measurements. In this chapter, a rectangular sampling domain is considered. However, it is not difficult to extend our algorithm to arbitrary boundaries. If measurement error has to be taken into account, the adaptive algorithm can be developed using approximation techniques such as minimum mean square error estimation instead of interpolation. As we mentioned, although the routing cost of mobile sensors is not considered in this chapter, it can be incorporated into the scheme in various ways. More studies are required to better approximate the probability updating process and account for field heterogeneity. For instance, sophisticated source statistical models that consider the smoothing effect of wind on the mean sunlight field are currently under investigation. Additionally, in a multi-scale sensing approach, we often have a rough overview of the field, which may reveal valuable information on field heterogeneity, and should be incorporated into our Bayesian framework.

In a sensor network, data streams at different sensors often have significant redundancy in them since they are the results of observing correlated physical phenomena. In chapter 4, we considered how to efficiently conduct data compression such that the communication rate could be kept at a minimum. We first

gave an overview on distributed source coding, in which sensors independently encode data streams. The distributed source coding is by itself an active research area. Efficient and low complexity codes are actively sought after. Then, a 2-stage DPCM coding scheme that utilized explicit side information to quantize the data was proposed. This method monitors the coding gain of side information continuously, and thus is suitable for use in joint routing/compression optimization.

The data-centric routing problem is complicated due to the coupling of data processing and route design. In chapter 5, an optimization problem was formulated for the combined routing and source coding with explicit side information. This problem is shown, without surprise, to be NP hard, and a mixed integer program was considered for one sub-instance of CRSC. We mentioned in section 5.2.2 that the number of sensors with highly correlated data can be brought down in a process of thinning the number of active sensors based on the reconstruction requirement. This pre-routing step makes in practice our procedure a two-phase operation. First determine the set of sensors that will participate in the fusion, then design the routes for transmitting the data to the fusion center. Currently, the first step is generally approached from a sampling point of view [WMN04] trying to meet the distortion constraint, while route design attempts to minimize the energy consumption. It is of interest to ask whether a combined approach will yield better results. [GCB04] is an interesting preliminary effort on that direction.

As a continuation of the previous chapter, chapter 6 studied the heuristic algorithms for the CRSC. Out of the four schemes, SPT, Clusters, BAS, and DSIT, that we considered, BAS and DSIT were both shown to achieve good performance under high data correlation and converge to SPT when the coding gain

diminished. However, these two algorithms relied heavily on our assumed network model. In particular, we assumed that data streams were highly correlated only when they were from a small group of sensors close to one another, and the coding gain saturated when the number of helpers exceeded one. Therefore, these methods may not be as effective in cases where the network topology and data correlation deviate from these assumptions. Nonetheless, as we discussed in section 5.2.2, there are practical reasons to consider this simplified case. Moreover, if more than one helper has to be considered, we speculate that the problem can be approached in a multi-step procedure. At each step, the number of helpers is restricted to at most one, and an algorithm similar to our heuristic scheme is used. This is an area that needs further research.

APPENDIX A

$R_{\min}$ for uniformly distributed source and sensors

First consider dividing the unit area into n identical square cells, and placing one sensor at the center of each cell. This is equivalent to fixing the sensor at $(0, 0)$ and placing the source in the area $[0, \frac{1}{2\sqrt{n}}] \times [0, \frac{1}{2\sqrt{n}}]$ uniformly. The probability density function of $R_{textmin}$ is as follows:

$$f_{R_{\min}}(r) = \begin{cases} 2\pi nr & \text{when } 0 \leq r \leq \frac{1}{2\sqrt{n}} \\ 4nr\left[\frac{\pi}{2} - 2\arccos\left(\frac{1}{2\sqrt{n}r}\right)\right] & \text{when } \frac{1}{2\sqrt{n}} \leq r \leq \frac{1}{\sqrt{2n}} \end{cases} \tag{A.1}$$

The density function is 0 for any r not included in the definition, which also applies to other probability density functions in this section. The average $R_{\min}$ can be easily evaluated:

$$\mathrm{E}(R_{\min}) \approx \frac{0.3826}{\sqrt{n}} \tag{A.2}$$

Next, we consider a point source that appears in a unit disc ($R_0 = \frac{1}{\sqrt{\pi}}$) according to the uniform distribution. Denote by R_s the distance from the source to the center of the disc.

$$f_{R_s}(r_s) = 2\pi r_s \quad \text{when } 0 \leq r \leq R_0$$

Denote by R, the distance between the source and a randomly placed sensor according to the uniform distribution. Given that the source is r_s away from the

disc center, the probability density function for R, is:

$$f_R(r \mid r_s) = \begin{cases} 2\pi r & \text{when } 0 \le r \le R_0 - r_s \\ 2\theta r & \text{when } R_0 - r_s \le r \le R_0 + r_s \end{cases} \tag{A.3}$$

where

$$\theta = \arccos\left(\frac{r_s^2 + r^2 - R_0^2}{2 r_s r}\right)$$

Therefore,

$$P(R \le r \mid r_s) =$$

$$\begin{cases} \pi r^2 & \text{when } 0 \le r \le R_0 - r_s \\ \phi R_0^2 + \theta r^2 - R_0^2 \sin\phi\cos\phi - r^2 \sin\theta\cos\theta & \text{when } R_0 - r_s \le r \le R_0 + r_s \end{cases}$$

where

$$\phi = \begin{cases} \arcsin\left(\frac{r\sin\theta}{R_0}\right) & \text{when } R_0 - r_s \le r \le \sqrt{R_0^2 + r_s^2} \\ \pi - \arcsin\left(\frac{r\sin\theta}{R_0}\right) & \text{when } \sqrt{R_0^2 + r_s^2} \le r \le R_0 + r_s \end{cases}$$

Now, consider n sensors are randomly placed in the disc in the same fashion. That the distance between the source and the closest sensor is r is to say that there is one sensor r away from the source and the other $(n-1)$ sensors are at least r away from the source.

$$f_{R_{\min}}(r \mid r_s) = n f_R(r \mid r_s)[1 - P(R \le r \mid r_s)]^{n-1}$$

Noticing that R_s is itself uniformly distributed, take the expectation on R_s.

$$f_{R_{\min}}(r) = \mathrm{E}_{R_s}[f_{R_{\min}}(r \mid r_s)]$$

This formula is numerically evaluated for different n, and the mean value is computed. It is found that

$$R_{\min} \approx \frac{0.5101}{\sqrt{n}}$$

APPENDIX B

$\sum_{i=0}^{N} \left| \frac{\partial S_\Delta}{\partial y_i} \right|$ is bounded

Rearrange the cubic spline function for $x_{j-1} \leq x \leq x_j$.

$$S_\Delta(x, y_0, y_1, \ldots, y_N) = \alpha_j M_{j-1} + \beta_j M_j + \gamma_j y_{j-1} + (1 - \gamma_j) y_j$$

where

$$\alpha_j = \left[\frac{(x_j - x)^3}{6h_j} - \frac{h_j(x_j - x)}{6} \right], \quad |\alpha_j| \leq \frac{h_j^2}{9\sqrt{3}}$$

$$\beta_j = \left[\frac{(x - x_j)^3}{6h_j} - \frac{h_j(x - x_{j-1})}{6} \right], \quad |\beta_j| \leq \frac{h_j^2}{9\sqrt{3}}$$

$$0 \leq \gamma_j = \frac{(x_j - x)}{h_j} \leq 1$$

Differentiate the spline function about y_i.

$$\frac{\partial S_\Delta}{\partial y_i} = \alpha_j \frac{\partial M_{j-1}}{\partial y_i} + \beta_j \frac{\partial M_j}{\partial y_i} + \gamma_j \delta_{i,j-1} + (1 - \gamma_j)\delta_{i,j}$$

where

$$\delta_{i,j} = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{otherwise} \end{cases}$$

Take the absolute values on both sides and sum all the equations over $i = 0, 1, \ldots, N$.

$$\sum_{i=0}^{N} \left| \frac{\partial S_\Delta}{\partial y_i} \right| \leq \frac{h_j^2}{9\sqrt{3}} \sum_{i=0}^{N} \left[\left| \frac{\partial M_{j-1}}{\partial y_i} \right| + \left| \frac{\partial M_j}{\partial y_i} \right| \right] + 1. \tag{B.1}$$

Now, differentiate both sides of Equation (2.17) about y_i.

$$\begin{bmatrix} \frac{\partial M_0}{\partial y_i} \\ \frac{\partial M_1}{\partial y_i} \\ \vdots \\ \frac{\partial M_N}{\partial y_i} \end{bmatrix} = \begin{bmatrix} 2 & \lambda_0 & \dots & 0 \\ \mu_1 & 2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 2 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial d_0}{\partial y_i} \\ \frac{\partial d_1}{\partial y_i} \\ \vdots \\ \frac{\partial d_N}{\partial y_i} \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} \frac{\partial d_0}{\partial y_i} \\ \frac{\partial d_1}{\partial y_i} \\ \vdots \\ \frac{\partial d_N}{\partial y_i} \end{bmatrix} \tag{B.2}$$

Notice

$$\frac{\partial d_j}{\partial y_i} = \frac{6\delta_{i,j+1}}{h_{j+1}(h_j + h_{j+1})} - \frac{6\delta_{i,j}}{h_j h_{j+1}} + \frac{6\delta_{i,j-1}}{h_j(h_j + h_{j+1})}$$

Take the absolute values on both sides of Equations (B.2), and sum over $i = 0, 1, \dots, N$. The following is obtained.

$$\sum_{i=0}^{N} \begin{bmatrix} |\frac{\partial M_0}{\partial y_i}| \\ |\frac{\partial M_1}{\partial y_i}| \\ \vdots \\ |\frac{\partial M_N}{\partial y_i}| \end{bmatrix} \leq \| \mathbf{B}^{-1} \| \sum_{i=0}^{N} \begin{bmatrix} |\frac{\partial d_0}{\partial y_i}| \\ |\frac{\partial d_1}{\partial y_i}| \\ \vdots \\ |\frac{\partial d_N}{\partial y_i}| \end{bmatrix} \leq \frac{12}{h_j h_{j+1}} \| \mathbf{B}^{-1} \|$$

in which $\| \mathbf{B}^{-1} \|$ is the row-max norm of matrix $\mathbf{B}^{-1}$ (page 20 [Ahl67]). For proper end conditions ($\lambda_0, \mu_N < 2$), [Ahl67] showed the following is true.

$$\| \mathbf{B}^{-1} \| \leq \max\left[(2-\lambda_0)^{-1}, (2-\mu_N)^{-1}, 1\right]$$

Therefore,

$$\sum_{i=0}^{N} \left| \frac{\partial S_\Delta}{\partial y_i} \right| \leq 1 + \frac{8\sqrt{3}\eta}{9},$$

$$\eta = \max\left[(2-\lambda_0)^{-1}, (2-\mu_N)^{-1}, 1\right] \left(\frac{h_j}{h_{j+1}} \right)$$

Thus, for meshes with bounded $(\frac{h_j}{h_{j+1}})$, the quantity $\sum_{i=0}^{N} \left| \frac{\partial S_\Delta}{\partial y_i} \right|$ is bounded.

References

[Ahl67] J. H. Ahlberg. *The Theory of Splines and Their Applications.* Academic Press, 1967.

[BC03] S. Bandyopadhyay and E. J. Coyle. "An energy efficient hierarchical clustering algorithm for wireless sensor networks." In *Proc. IEEE Infocom*, 2003.

[BMK02] J. Burke, E. Mendelowitz, J. Kim, and R. Lorenzo. "Networking with knobs and knats: towards ubiquitous computing for artists." In *Ubiquitous Computing: Concepts and Models Workshop*, Gothenburg, Sweden, USA, 2002.

[Boo89] F. L. Bookstein. "Principal warps: thin-plate splines and the decomposition of deformations." *IEEE Trans. Pattern Analysis and Machnine Intelligence*, **11**(6):567–585, June 1989.

[BRY04] M. A. Batalin, M. H. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. Kaiser, M. Hansen, G. Pottie, M. Srivastava, and D. Estrin. "Call and response: experiments in sampling the environment." In *Proceedings of ACM Sensys*, Baltimore, USA, Nov 2004.

[BY89] T. Berger and R. W. Yeung. "Multiterminal source coding with one distortion criterion." *IEEE Trans. Information Theory*, **35**:228–236, 1989.

[CBV04] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. "On network correlated data gathering." In *Proc. IEEE Infocom*, Hongkong, China, March 2004.

[CEE01] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. "Habitat monitoring: application driver for wireless communications technology." In *Proc. ACM Sigcomm Workshop on Data Comm. in Latin America and the Caribbean*, April 2001.

[CHB03] S. Capkun, J. P. Hubaux, and L. Buttyn. "Mobility helps security in ad hoc networks." In *Proceedings of Fourth ACM Symposium on Mobile Ad Hoc Networking and Computing*, pp. 46–56, Annapolis, USA, 2003. ACM Press.

[CL65] Y. J. Chu and T. H. Liu. "On the shortest arborescence of a directed graph." *Science Sinica*, **14**:1396–1400, 1965.

[Cov98] T. M. Cover. "Comments on broadcast channels." *IEEE Trans. Information Theory*, **44**:2524–2530, 1998.

[CP94] M. M. Caldwell and R. W. Pearcy. *Exploitation of Environmental Heterogeneity by Plants: Ecophysiological Processes Above- and Belowground*. Academic Press, San Diego, USA, 1994.

[Cre93] N. A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, New York, USA, 1993.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[CT00] J. H. Chang and L. Tassiulas. "Energy conserving routing in wireless ad-hoc networks." In *Proc. IEEE Infocom*, 2000.

[CYE03] J. Chen, L. Yip, J. Elson, H. Wang, D. Maniezzo, R. E. Hudson, K. Yao, and D. Estrin. "Coherent acoustic array processing and localization on wireless sesnor networks." *Proc. IEEE*, **91**(8):1154–1162, Aug 2003.

[Dav72] L. D. Davisson. "Rate-distortion theory and applicaiton." *IEEE Trans. Information Theory*, **60**(7), July 1972.

[DSR76] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner. "Real-time digital hardware pitch detector." *IEEE Trans. ASSP*, **assp-24**(1):2–8, Feb 1976.

[Edm67] J. Edmonds. "Optimum branchings." *J. Research of the National Bureau of Standards*, **71B**:233–240, 1967.

[EGH00] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the internet: introduction." *Comm. of ACM*, **43**(5):38–41, 2000.

[EGP01] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. "Instrumenting the world with wireless sensor networks." In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, volume 4, p. 2033, Salt Lake City, USA, 2001.

[Fed72] V. V. Fedorov. *Thory of Optimal Experiments*. Academic Press, London, UK, 1972.

[FJ89] P. J. Flynn and A. K. Jain. "On reliable curvature estimation." In *Proceedings of Conference on Computer Vision and Pattern Recognition*, San Diego, USA, June 1989.

[Gal68] R. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, 1968.

[GC80] A. El Gamal and T. M. Cover. "Multiple user information theory." *Proc. IEEE*, **68**:1466–1483, 1980.

[GCB04] D. Ganesan, R. Cristescu, and B. Beferull-Lozano. "Power-efficient sensor placement and transmission structure for data gathering under distortion constraints." In *Proc. IPSN*, Berkeley, CA, 2004.

[GE03] A. Goel and D. Estrin. "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk." In *ACM/SIMA Symposium on Discrete Algorithms*, 2003.

[GK00] P. Gupta and P. R. Kumar. "The capacity of wireless networks." *IEEE Trans. Information Theory*, **46**, 2000.

[GT88] H. N. Gabow and R. E. Tarjan. "Algorithms for two bottleneck optimization problems." *Journal of Algorithms*, **9**:411–417, 1988.

[GT02] M. Grossglauser and D. N. Tse. "Mobility increases the capacity of ad hoc wireless networks." *IEEE/ACM Trans. Networking*, **10**(4):477–486, 2002.

[GZ03] J. Garcia-Frias and W. Zhong. "LDPC codes for compression of multi-terminal sources with hidden Markov correlation." *IEEE Comm. Letter*, **7**(3):115–117, 2003.

[Hal97] S. Haldar. "An 'all pairs shortest paths' distributed algorithm using $2n^2$ messages." *Journal of Algorithms*, **24**:20–36, 1997.

[HCB00] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. "Energy-efficient communication procotol for wireless microsensor networks." In *Proc. Hawaii International Conference on System Sciences*, Jan 2000.

[HD72] R. L. Harder and R. N. Desmarais. "Interpolation using surface splines." *J. Aircraft*, **9**(2):189–191, February 1972.

[HM03] P. Hall and I. Molchanov. "Sequential methods for design-adaptive estimation of discontinuities in regression curves and surfaces." *Annals of Statistics*, **31**(3):921–941, 2003.

[HRW92] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. North-Holland, 1992.

[Hum83] P. A. Humblet. "A distributed algorithm for minimum weight directed spanning trees." *IEEE Trans. Comm.*, **31**:756–762, June 1983.

[IGE03] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. "Directed diffusion for wireless sensor networking." *IEEE/ACM Trans. Networking*, **11**(1):2–16, Feb 2003.

[Jai89] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[JMY90] M. Johnson, L. Moor, and D. Ylvisaker. "Minimax and maximin distance designs." *J. Stat. Plan. and Infer.*, **26**(2):131–148, October 1990.

[JN84] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice Hall, Englewood Cliffs, NJ, USA, 1984.

[Kar71] R. M. Karp. "A simple derivation of Edmonds' algorithm for optimum branching." *Networks*, **1**:265–272, 1971.

[KEW02] B. Krishnamachari, D. Estrin, and S. Wicker. "Modelling data-centric routing in wireless sensor network." Technical Report CENG 02-14, University of Southern California, 2002.

[KKP05] A. Kansal, W. Kaiser, G. Pottie, and M. B Srivastava. "Actuation Techniques for Sensing Uncertainty Reduction." Technical Report 51, University of California, Los Angeles, March 2005.

[KPS04a] W. Kaiser, G. Pottie, M. Srivastava, G. S. Sukhatme, J. Villasenor, and D. Estrin. "Networked infomechanical systems (NIMS) for ambient intelligence." Technical Report 31, University of California, Los Angeles, December 2004.

[KPS04b] W. Kaiser, G. Pottie, M. Srivastava, G. S. Sukhatme, J. Villasenor, and D. Estrin. "Networked Infomechanical Systems (NIMS) for Ambient Intelligence." In *Invited Contribution to Ambient Intelligence.* Springer-Verlag, 2004.

[KRY95] S. Khuller, B. Raghavachari, and N. Young. "Balancing minimum spanning trees and shortest-path trees." *Algorithmica*, **14**:305–321, 1995.

[LL86] L. D. Landau and E. M. Lifshitz. *Theory of Elasticity.* Butterworth-Heinemann, Boston, USA, 1986.

[LLK85] E. L. Lawler, Jan Karel Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, New York, USA, 1985.

[LO88] L. S. Lim and A. V. Oppenheim, editors. *Advanced Topics in Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[LTP05] H. Luo, Y. Tong, and G. Pottie. "A two-stage DPCM scheme for wireless sensor networks." In *Proc. IEEE International Conference on Accoustic, Speech, and Signal Processing*, Philadelphia, USA, 2005.

[Max60] J. Max. "Quantization for minimum distortion." *IEEE Trans. Information Theory*, **6**, 1960.

[mic] *http://www.tinyos.net/scoop/special/hardware/*, TinyOS website.

[MM02] R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, New York, USA, 2002.

[MPS02] A. Mainwaring, J. Polastre, R. Szewezyk, D. Culler, and J. Anderson. "Wireless sensor networks for habitat monitoring." In *Proc. 1st ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, USA, Sept 2002.

[MRA89] R. B. Myneni, J. Ross, and G. Asrar. "A review on the theory of photon transport in leaf canopies." *Agriculture and Forest Meteorology*, **45**(1):1–153, 1989.

[MW95] T. L. Magnanti and L. A. Wolsey. "Optimal Trees." In C. L. Monma M. O. Ball, T. L. Magnanti and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, chapter 9. Elsevier, 1995.

[ncd] *http://www.ncdc.noaa.gov/oa/ncdc.html*, national climatic data center.

[NMW04] R. Nowak, U. Mitra, and R. Willett. "Estimating Inhomogeneous Fields Using Wireless Sensor Networks." *IEEE J. Selected Area in Comm.*, **22**(6):999–1006, 2004.

[OBK00] A. Okabe, B. Boots, and S. N. Chiu K. Sugihara, and K. Sugihara. *Spatial Tessellations : Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, New York, NY, USA, 2000.

[Ooh97] Y. Oohama. "Gaussian multiternimal source coding." *IEEE Trans. Information Theory*, **43**:1912–1923, Nov 1997.

[Oza80] L. H. Ozarow. "On the source coding problem with two channels and three receivers." *Bell Syst. Tech. J.*, **59**:1909–1922, 1980.

[PK00] G. Pottie and W. Kaiser. "Wireless sensor networks." *Communications of ACM*, **43**(5):51–58, May 2000.

[PKR02] S. S. Pradhan, J. Kusuma, and K. Ramchandran. "Distributed compression in a dense microsensor network." *IEEE Signal Processing Magzine*, **19**:51–60, March 2002.

[Pow94] M. J. D. Powell. "The uniform convergence of thin plate spline interpolation in two dimensions." *Numer. Math.*, **68**(1):107–128, June 1994.

[PP03] A. Pandya and G. Pottie. "On scalability and source/channel coding decoupling in large scale sensor networks." Technical Report 15, University of California, Los Angeles, 2003.

[PR03] S. S. Pradhan and K. Ramchandran. "Distributed source coding using syndromes (DISCUS): design and construction." *IEEE Trans. Information Theory*, **49**:626–643, March 2003.

[Raf86] E. Rafajlowicz. "Optimum choice of moving sensor trajectories fordistributed-parameter system identification." *Int. J. Control*, **43**(5):1441–1451, 1986.

[RHS04] M. Rahimi, M. Hansen, G. Sukhatme, W. Kaiser, and D. Estrin. "Adaptive sampling for environmental field estimation using robotic sensors." In *Proceedings of Information Processing in Sensor Networks*, New York, January 2004. Springer.

[RPK04] M. H. Rahimi, R. Pon, W. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava. "Adaptive Sampling for Environmental Robotics." In *IEEE International Conference on Robotics and Automation*, pp. 3537–3544, New Orleans, USA, 2004.

[RSS98] J. Ross, M. Sulev, and P. Saarelaid. "Statistical treatment of the PAR variability and its application to willow coppice." *Agriculture and Forest Meteorology*, **91**(1):1–21, May 1998.

[RVW03] D. Rakhmatov, S. Vrudhula, and A. Wallach. "A model for battery lifetime analysis for organizing applications on a pocket computer." *IEEE Trans. VLSI Systems*, **11**(6):1019–1030, December 2003.

[Sak70] D. Sakrison. "The rate of a class of random processes." *IEEE Trans. Information Theory*, **16**, Jan 1970.

[Sak71] D. Sakrison. "Comparison of line-by-line and two-dimensional encoding of random images." *IEEE Trans. Information Theory*, **17**, July 1971.

[Say03] A. H. Sayed. *Fundamentals of Adaptive Filtering.* John Wiley & Sons Inc., Hoboken, NJ, USA, 2003.

[Ser02] S. D. Servetto. "On the feasibility of large scale wireless sensor networks." In *Proc. 40th Annual Allerton Conf. on Commun., Contr. and Comput.*, 2002.

[SH03] J. Scott and M. Hazas. "User-friendly surveying techniques for location-aware systems." In *Proceedings of Fifth International Conference on Ubiquitous Computing*, pp. 45–54, Seatle, USA, 2003. Springer-Verlag.

[Sha48] C. E. Shannon. "A mathematical theory of communication." *Bell Sys. Tech. Journal*, **27**:379–423, 623–656, 1948.

[SK95] A. H. Sayed and T. Kailath. "A look-ahead block Schur algorithm for Toeplitz-like matrices." *SIMAX*, **16**(2):388–413, Apr 1995.

[SMP01] M. Srivastav, R. Muntz, and M. Potkonjak. "Smart kindergarden: sensor-based wireless network for smart developmental problem-solving environments." In *Proc. ACM Mobicom*, pp. 132–138, July 2001.

[SS02a] A. Savvides and M. Srivastav. "A distributed computation platform for wireless embedded sensing." In *Proc. IEEE Inter. Conf. Computer Design: VLSI in Computers and Processors*, pp. 220–225, Sept 2002.

[SS02b] A. Scaglione and S. Servetto. "On the interdependence of routing and data compression in multi-hop sensor networks." In *Proc. ACM/IEEE Mobicom*, 2002.

[SU00] J. R. Sack and J. Urrutia. *Handbook of Computational Geometry.* Elsevier, Amsterdam, Netherlands, 2000.

[SW73] D. Slepian and J. K. Wolf. "Noiseless coding of correlated information sources." *IEEE Trans. Information Theory*, **19**:471–480, 1973.

[SWR98] S. Singh, M. Woo, and C. S. Raghavendra. "Power-aware routing in mobile ad-hoc networks." In *Proc. ACM/IEEE Mobicom*, Dallas, Texas, 1998.

[Tar77] R. E. Tarjan. "Finding optimum branchings." *Networks*, **7**:25–35, 1977.

[Tel99] I. E. Telatar. "Capacity of multiple antenna Gaussian channel." *European Transactions on Telecommunications*, **10**:585–595, 1999.

[Ten00] D. Tennenhouse. "Proactive computing." *Communications of ACM*, **43**(5), May 2000.

[TM80] H. Takahashi and A. Matsuyama. "An approximate solution for the steiner problem in graphs." *Math. Japonica*, **24**(6):573–577, 1980.

[VP99] F. Valladares and R. W. Pearcy. "The geometry of light interception by shoots of Heteromeles arbutifolia: morphological and physiological consequences for individual leaves." *Oecologia*, **121**(2):171–182, 1999.

[Wei91] M. Weiser. "The computer for the 21st century." *Sci. Am.*, **265**(3):94–104, September 1991.

[Wil99] H. P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, 1999.

[WMN04] R. Willett, A. Martin, and R. Nowak. "Backcasting: adaptive sampling for sensor networks." In *Proc. IPSN*, Berkeley, CA, 2004.

[Wol98] L. A. Wolsey. *Integer Programming.* John Wiley & Sons, 1998.

[WZ76] A. D. Wyner and J. Ziv. "The rate-distortion function for source coding with side information at the decoder." *IEEE Trans. Information Theory*, **22**:1–11, 1976.

[XLC04] Z. Xiong, A. D. Liveris, and S. Cheng. "Distributed source coding for sensor networks." *IEEE Signal Processing Magazine*, **21**:80–94, Sept 2004.

[Yao] K. Yao. Private communication.

[ZB99] R. Zamir and T. Berger. "Multiterminal source coding with high resolution." *IEEE Trans. Inform. Theory*, **45**(1):106–117, 1999.

[ZSE02] R. Zamir, S. Shamai, and U. Erez. "Nested linear/lattice codes for structured multiterminal binning." *IEEE Trans. Information Theory*, **48**(6):1250–1276, 2002.