

UNIVERSITY OF CALIFORNIA  
Los Angeles

## **Sampling Strategies in Sensor Networks**

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

by

Xiangming Kong

2008

UMI Number: 3357366

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3357366

Copyright 2009 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



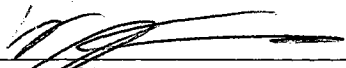
ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

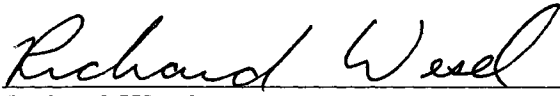
© Copyright by

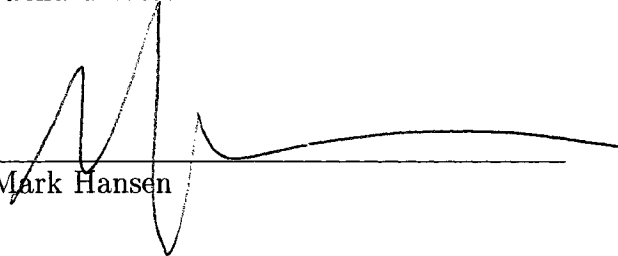
Xiangming Kong

2008

The dissertation of Xiangming Kong is approved.

  
\_\_\_\_\_  
William Kaiser

  
\_\_\_\_\_  
Richard Wesel

  
\_\_\_\_\_  
Mark Hansen

  
\_\_\_\_\_  
Gregory Pottie, Committee Chair

University of California, Los Angeles

2008

*To my parents and my dear husband*

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Organization of the Dissertation . . . . .	5
<b>2</b>	<b>Light Field Models</b>	<b>7</b>
2.1	Sunlight Field Characterization . . . . .	7
2.2	Understory Sunlight Decomposition . . . . .	9
2.3	Sunlight Models . . . . .	10
2.3.1	Sunflecks . . . . .	12
2.3.2	Umbras . . . . .	13
2.3.3	Penumbras . . . . .	14
2.4	Direct Beam Radiation Simulator . . . . .	18
2.4.1	Simulation Model Generation . . . . .	18
2.4.2	Simulation Model Validation . . . . .	19
2.5	Conclusion . . . . .	23
<b>3</b>	<b>Overview of Existing and Proposed Sampling Methods</b>	<b>24</b>
3.1	Methods for Sampling Random Variables . . . . .	25
3.1.1	Random Sampling . . . . .	26
3.1.2	Deterministic Sampling . . . . .	27
3.2	Methods for Sampling a Signal . . . . .	29
3.2.1	Uniform Sampling . . . . .	30
3.2.2	Static Design . . . . .	31
3.2.3	Adaptive Sampling . . . . .	33
3.2.4	Compressed Sensing . . . . .	34
3.3	Our Approaches . . . . .	34
3.3.1	Adaptive Sampling . . . . .	35
3.3.2	Multiscale Sampling . . . . .	36
3.4	Conclusion . . . . .	37
<b>4</b>	<b>Reconstruction Techniques</b>	<b>38</b>
4.1	Polynomial Regression . . . . .	41
4.2	Spline Regression . . . . .	43

4.3	Kriging . . . . .	45
4.4	Error Analysis . . . . .	46
4.5	Conclusion . . . . .	47
<b>5</b>	<b>An Adaptive Sampling Algorithm in Sensor Networks</b>	<b>48</b>
5.1	Algorithm Overview . . . . .	50
5.2	Candidate Pool . . . . .	51
5.3	Sample Selection . . . . .	53
5.3.1	The Probability of Meeting Sampling Objective . . . . .	53
5.3.2	MAP Sample Selection . . . . .	54
5.3.3	Probability Update . . . . .	55
5.3.4	Incorporating Field Models . . . . .	58
5.4	Field Reconstruction . . . . .	60
5.5	Simulation . . . . .	62
5.5.1	Algorithm Implementation . . . . .	62
5.5.2	Simulation Results . . . . .	63
5.6	Conclusion . . . . .	65
<b>6</b>	<b>Multiscale Sensing</b>	<b>71</b>
6.1	Introduction . . . . .	71
6.2	Multiscale Formulation of Sunlight Field Estimation . . . . .	76
6.3	Image Processing . . . . .	79
6.3.1	Existing Segmentation Methods . . . . .	79
6.3.2	Image Segmentation Procedures . . . . .	82
6.3.3	Image Calibration . . . . .	85
6.4	Model Fitting and Selection . . . . .	87
6.4.1	Probability Density Function Estimation . . . . .	87
6.4.2	Model Selection . . . . .	94
6.4.3	Field Model Library . . . . .	102
6.5	Simulations . . . . .	107
6.5.1	Algorithms Implementation . . . . .	107
6.5.2	Simulation Results . . . . .	109
6.6	Conclusion . . . . .	113
<b>7</b>	<b>Discussions and Future Directions</b>	<b>115</b>
	<b>Bibliography</b>	<b>118</b>

## List of Tables

6.1	Parameters for simulating Gaussian mixture data . . . . .	98
6.2	Comparison between criteria for selecting mixture models. No. of samples = 100. . . . .	98
6.3	Comparison between criteria for selecting mixture models. No. of samples = 1000. . . . .	98
6.4	Parameters for simulating data. $w$ is the weight of the distribution, $\alpha$ and $\beta$ are shape parameters for Beta distribution, $\lambda$ is the component weight in the Gaussian mixture, $\mu$ and $\sigma$ are mean and standard deviation of each Gaussian component respectively. . . . .	101
6.5	Average likelihood under different models for three umbras . . . . .	107



## List of Figures

2.1	Typical light field and its intensity . . . . .	8
2.2	Light intensity over time at two spots. . . . .	9
2.3	Typical light intensity distribution under canopy . . . . .	12
2.4	Light intensity distribution in umbras. (a) site 1; (b) site 2 . . . . .	13
2.5	Light correlation model in umbrae . . . . .	14
2.6	Penumbra effect . . . . .	15
2.7	Intensity of a penumbra generated by a disc at 3 meters high. . . . .	16
2.8	Typical light intensity correlation . . . . .	17
2.9	Apparent sun image to an observer on the ground . . . . .	20
2.10	Simulated direct beam intensity on the ground . . . . .	20
2.11	Gap distribution comparison . . . . .	21
2.12	Direct beam intensity distribution comparison . . . . .	22
2.13	Light correlation model in penumbras. . . . .	23
3.1	The multiscale sensing scheme. . . . .	37
5.1	The block diagram of the adaptive sampling algorithm . . . . .	51
5.2	Delaunay triangulation and Voronoi tessellation . . . . .	54
5.3	The boundary of the Voronoi cells before a new sample is added is shown in solid lines. After a sample is added, a new cell is generated, whose boundary is shown in dashed lines. Outside this cell, the old boundaries remain unchanged. . . . .	56

5.4	Variable definition for determining weights. Dashed lines are boundaries of Delaunay triangles and solid lines are boundaries of Voronoi cells. . . . .	56
5.5	Sunlight field variation model . . . . .	60
5.6	Adding new sampling sites in the Q-Method (a) Original cells. The dots represents sampling sites and the shaded area represents the cell with the largest estimation error. (b) Cells and sampling sites after new sites are added. . . . .	64
5.7	Ground truth of the light field . . . . .	65
5.8	Reconstruction result for real sunlight field. . . . .	66
5.9	The square root of MSE in successive steps. . . . .	67
5.10	The field bending energy in successive steps. . . . .	67
5.11	The pattern of the sampling sites for the light field . . . . .	68
5.12	Ground truth of the piecewise constant field . . . . .	69
5.13	Reconstruction result for the piecewise constant field. . . . .	70
6.1	The spectral response of LI-90 PAR sensor . . . . .	76
6.2	Typical camera spectral response . . . . .	78
6.3	Image series and its corresponding partition . . . . .	85
6.4	Kodak gray scale . . . . .	86
6.5	Camera characteristic curve . . . . .	87
6.6	Effect of calibration . . . . .	88
6.7	Effect of bandwidth of density estimation . . . . .	91
6.8	Extra discrepancy of using nonparametric method than using parametric method. Values above zero mean the nonparametric density has a larger discrepancy. . . . .	100
6.9	Estimated density functions . . . . .	102
6.10	Light intensity over time at two spots. . . . .	103

6.11 Light intensity over time at two spots . . . . .	105
6.12 Umbra reflectivity classification represented in pseudo-color. Each color corresponds to one type of reflectivity. . . . .	106
6.13 Ground truth . . . . .	110
6.14 Reconstructed field with different sampling schemes . . . . .	111
6.15 Multiscale sensing sampling pattern . . . . .	112
6.16 Reconstructed field with multiscale sensing scheme under low fidelity requirement . . . . .	112
6.17 MSE convergence rate . . . . .	113

## Acknowledgements

I am greatly indebted to my advisor, Prof. Gregory Pottie, who took so much effort in guiding me in my research. His suggestions and constant support lead me to the final success of my Ph.D study. Gregory always provides a comfortable and stimulating research environment. In this environment, I was able to gain knowledge in broad areas. I'm also grateful to other members in my dissertation committee: to Prof. William Kaiser for his indispensable guidance; to Prof. Mark Hansen for his help that aids me in overcoming many obstacles; to Prof. Richard Wesel for teaching me so much knowledge. I am also thankful all my teachers at UCLA. In particular, I would like to express my gratitude to Prof. Kung Yao for helping me get the NSF fellowship.

I feel lucky to have the wonderful opportunity of working with my colleagues and friends in UCLA. I benefited a lot from the stimulating discussions with my colleagues in Prof. Pottie's group: Huiyu Luo, Yu-Ching Tong, Yue Zhao, Nabil Hajj Chehade, Kevin Ni, Roja Bandari and Dorna Bandari. The pleasant time I spent with them makes my experience in UCLA memorable and rewarding. I also would like to thank Jun Shi, Huijun Wu, Yang Zou and Scott Enserink for their warm helps in my early years in UCLA.

Finally I want to thank my family for their encouragement and support through all these years. I am especially grateful to my husband Pengcheng Jia. Without his constant love and care this work would never have come into existence.

## VITA

1976	Born, Shijiazhuang, Hebei, China
July, 1998	B.S., Electronics Engineering, Tsinghua University, Beijing, China
May, 2000	M.S., Electrical Engineering, Iowa State University, Ames, Iowa, USA
2000-2003	Ericsson Inc, Santa Barbara, California
2004-2007	National Science Foundation Fellowship
November, 2008	Ph.D., Electrical Engineering, University of California at Los Angeles, California, USA

## PUBLICATIONS

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*.

Huiyu Luo, Xiangming Kong, and Gregory Pottie, "An adaptive algorithm for sampling two-dimensional fields using mobile sensors", *IEEE International Conference on Acoustic, Speech and Signal Processing*, Honolulu, HI, USA, 2007.

Xiangming Kong, Richard Pon, William Kaiser and Gregory Pottie, "Environmental sampling with multiscale sensing", *IEEE International Conference on Acoustic,*

*Speech and Signal Processing*, Toulouse, France, 2006.

Leilei Lock, Xiangming Kong and Richard Barton, "Simulation of time-varying, frequency-selective multipath fading channels for spread-spectrum waveforms", *IEEE Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, USA, 1999.

ABSTRACT OF THE DISSERTATION

## **Sampling Strategies in Sensor Networks**

by

Xiangming Kong

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2008

Professor Gregory Pottie, chair

The widespread use of network sensors have greatly improved the sensing capability of many systems. However, the resources for obtaining high fidelity data are still very limited. Therefore, resource allocation in sensor networks remains an important issue. In many applications, especially in field reconstruction problems, the data acquisition process takes up a large portion of resources. Efficiently deploying sensors for data acquisition will achieve a large resource saving in these applications. The way to accomplish this objective is discussed in this dissertation.

The approaches we take are model based. Since models are typically specific to certain types of fields, we concentrate on the incident sunlight field under forest canopies. The characteristics of this field are studied first. Existing models are carefully examined before they are applied in our algorithm design. A new simulator is created to simulate the field. New models are extracted from experimental and simulated data.

Based on the field models, two algorithms are designed to efficiently take samples for the reconstruction of the field. The first algorithm is an adaptive algorithm which adjusts the sampling density to the field roughness. It maintains an estimate of the reconstruction quality based on a Bayesian framework. This estimate provides an indication on the sampling sites that will improve the reconstruction result most.

The second algorithm, a multiscale sensing scheme, fuses information from different levels. It coordinates the operation of two types of sensor. One type of sensor provides the high-level information, including the field partition and field models. The field models from the real data, complemented by the models from the simulated data, direct the other type of sensor to gather high fidelity measurements of the field. The overview of the field from the first type of sensor enables the optimal allocation of the second type of sensor, thereby greatly improving the utilization efficiency of sensor resources. Both algorithms were tested with experimental data and proved to be effective in resource saving.



# CHAPTER 1

## Introduction

In our society, sensors are widely used. Sometimes, they work as guards around us, sending alarms to us whenever undesired events occur. On other occasions, they are our assistants, helping us learn about the world we live in. Depending on the phenomena a sensor measures, its fabrication may involve several scientific subjects, such as physics, chemistry, biology, etc. Due to the importance of sensors, they have undergone a long history of development. In the last several decades, the advancement in solid state integrated circuit design and its widespread usage have greatly pushed forward the sensor development process.

The availability of a large amount of low cost, low capability sensors motivated the deployment of integrated network sensors [12][86]. The network sensors combine microsensor technology, signal processing, computation and communication in one system. These sensors are typically distributed. They communicate their data between each other and/or to a fusion center. By combining the measurements from many sensors, the network sensors provide a new monitoring and controlling capability that was never seen before.

One important motivation in deploying network sensors is to acquire high-quality data. Accordingly, the primary concern in a distributed sensor network is the sensing fidelity. Many factors can degrade measurement fidelity. These factors can be grouped into three categories:

- Manner of sensors deployment

When data from a sensor is used to determine the location of an event, it is

not surprising that the sensor needs to be aware of its own location. Even if its measurements are not directly related to locations, its position is still important information because this information limits the fidelity of the sensor data. When sensors are assumed to be deployed randomly or when robotic sensors are involved, knowledge about sensor locations is assumed to be very vague. A considerable amount of work has been devoted to the localization of sensors. The general practice is to determine the location of a sensor node by finding its distances to seed nodes, which are aware of their own locations[39]. A better approach is adding infrastructures to sensor nodes such that sensors are deployed with known locations. In addition, mobility of mobile sensors with infrastructure is predictable and precise. The Networked Informechanical Systems (NIMS) is an example of such a system[46].

- Limitation in sensor technology

Sensors are not ideal. They can be broken and give erroneous data. Their measurements can have bias. Even the best sensor can only provide noisy data. Choosing appropriate sensors alleviates the problem. Some of the errors, like broken sensors and bias, can be removed through calibration. Others are not completely removable. But fusion of data from multiple sensors can reduce such errors.

- Characteristics of the measured phenomena

These factors are typically out of the realm we have control over, because they are inherent in the phenomena. But this does not mean we can do nothing to reduce the uncertainty brought by these factors. Instead, algorithms are designed to combat the problem. As an illustration, suppose we are interested in determining the properties of an event. A group of ten sensors are required to reside at the location where the event happens in order to obtain the necessary data. However, the time and location at which the event will occur are

unpredictable. One way to fulfil the task is to densely deploy groups of sensors and allow them to operate in the field all the time until we get all necessary data. A smarter way is detecting the occurrence of the event with one sensor. Other sensors are moved over the spot right after the event occurs. In this way, a small fraction of the sensors dwell in the field at any time.

There are numerous applications for sensor networks. These applications cover all aspects of our everyday life, such as traffic control, health care and security, to name a few. Among these applications, environmental monitoring has drawn more and more attention from people because of the deterioration of the natural environment over these years. Studies for this application are of practical importance. In this study, we focus on the incident light distribution under forest canopies. There are several reasons for the selection of this physical phenomenon:

- Light patterns are of interest both for the generation of energy in solar harvesting systems to power sensor nodes and for the growth of plants.
- It is relatively easy to identify the underlying physical causes of the resultant patterns. Simple geometrical models suffice to closely approximate reality in some situations.
- In more complicated situations which call for statistical approaches, light patterns are often modelled as a realization of a two-dimensional random field. Experience gained in studying this phenomenon can be easily applied to other random field studies.
- We have access to data from both laboratory and field experiments.
- There are many theoretical and simulation models that can be drawn upon when experiments are impractical.

The essence of designing algorithms is concentrating available resources in the places where they are needed most. Because so many procedures are involved in

a sensing process, resource saving algorithms can be carried out in any procedure. Historically, sensors in a distributed sensor network are assumed to be wireless and not very powerful. Therefore, sensor network studies commonly emphasize improvements in communication and local data processing. One important step in the sensing process, the data acquisition, has not received sufficient attention. However, in problems like the field reconstruction, this step often absorbs a large portion of the total resources. Resource saving in this step is essential in some systems.

The data acquisition process often involves sample selection. This is because it is either impossible or unnecessary to acquire all the data. The simplest approach is taking samples randomly. There has been a long history of people using random samples to estimate the whole population. But the subject of random sampling was not put on a rigorous scientific basis until Gossett developed the theory of small-sample statistics[101]. Since then, various sampling algorithms have been designed for different data sets. Some algorithms aim at reaching optimal performance asymptotically, while others aim at saving resources and meeting a certain objective at the same time.

A good design with one objective in mind may turn out to be a very bad design for a different objective. Therefore, it is important to clarify the objective of a design before it is discussed. Common objectives for sampling algorithms design include: 1) detect and/or track the occurrence of an event; 2) extract model parameters for a phenomenon; 3) predict the values at unmeasurable sites; 4) reconstruct a signal. The scope of this dissertation will be restricted to the problem of designing sampling algorithms for signal reconstruction within a given fidelity range.

The enrichment of resources over these years also results in more ambitious goals. Accompanying these ambitious goals are higher fidelity requirements. To meet these requirements, the existing algorithms demand a large number of sensors to be deployed densely in many practical situations, which frequently is an infeasible

solution. Such situations motivate the design of new sampling algorithms discussed in this dissertation.

## 1.1 Organization of the Dissertation

The rest of the dissertation is organized as follows.

Chapter 2 examines the characteristics of the incident sunlight field. Because sunlight is an important energy source to plants, its distribution has intrigued biologists over a long period of time. There have been numerous attempts to model and simulate this distribution. However, available resources and existing sampling algorithms largely limit the resolution and accuracy of these models. A new simulator with higher resolution is created. Simulated data can supplement real data to generate better models for the field. Based on the field characteristics and statistical analysis of available data, challenges in the sampling algorithm design specific to the sunlight field are analyzed.

To aid readers in understanding the topic of this dissertation, background knowledge on sampling is given in Chapter 3. In addition to explaining the basic sampling methods, some of the more advanced methods are also reviewed. Among these methods, some are generic and applicable to all types of fields. Others are designed for specific fields. The methods suitable for sampling the sunlight field are detailed. The differences between the new algorithms and the existing algorithms are explained at the concept level.

Because of their close relationship to sampling algorithms, signal reconstruction techniques are reviewed in the following chapter. Three popular reconstruction methods, the polynomial, spline and Kriging methods, are elaborated. Two of the three methods will be applied in our sampling algorithm.

Starting from Chapter 5, two new algorithms designed for reconstructing the sunlight field are described in detail. In particular, Chapter 5 depicts the design process

of an adaptive sampling algorithm. In this algorithm, sampling and reconstruction processes are carried out iteratively. In each iteration, the most informative points are selected as the new sampling points based on the reconstruction result from the previous iteration. A second sampling algorithm, a multiscale sampling algorithm, is described in Chapter 6. Just as its name implies, information used in the field reconstruction does not come from one source only, which is the case in the adaptive sampling algorithm. The results in Chapter 6 demonstrate that higher efficiency in exploiting resources is achievable when information from multiple levels are fused together. Both algorithms are compared with existing schemes and shown to work effectively.

We make concluding remarks and carry out some discussion on future research directions in Chapter 7.

## CHAPTER 2

### Light Field Models

Knowledge about the physical formation of a phenomenon is important and useful in many aspects. When we want to get a faithful reconstruction of the phenomenon from samples, this knowledge gives us hints on sampling density. When we want to learn some statistical properties of the phenomenon, this knowledge helps reduce uncertainty in the estimation. We can also apply this knowledge in simulating the phenomenon when a large amount of high quality real data is not readily available. The formation of the sunlight field is studied in this chapter.

This chapter is organized as follows. The sunlight field is characterized in section 2.1. In the following section, the causes of these characteristics are analyzed. In section 2.3, probabilistic distribution models are fitted to measured data in simple fields. No simple distribution function is suitable for fitting the data in more complex fields. Therefore a simulator is employed to model such fields. Details about the generation and validation of the simulator are presented in section 2.4. Section 2.5 concludes this chapter.

#### 2.1 Sunlight Field Characterization

The light distribution under forest canopies has attracted attention from many biologists since this knowledge is very important for studying the ecosystem in the forest. In particular, it is important for calculations such as energy balance and in answering questions regarding distribution of plant species. A lot of research has been done to better understand this field. But until now, no study has been able

to provide very high resolution reconstruction of this field. The primary reason is it exhibits high spatial and temporal variation. Fig. 2.1 plots a snapshot of a light field. This figure clearly demonstrates the quick spatial variation of the field.

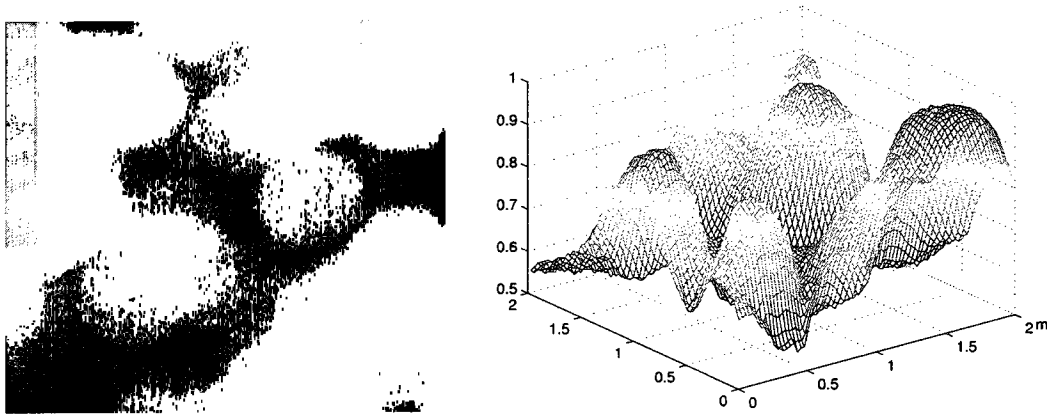


Figure 2.1: Typical light field and its intensity

The temporal variation of light fields originates from the movement of the sun and the wind. When the wind is strong, the forest canopy and hence the shadows cast by it keep on changing. Even in very calm days when there is almost no wind, the movement of the sun still leads to the movement of shadows. Fig. 2.2 illustrates the temporal variation of light. To put this variation on a quantitative ground, we can quickly make a rough estimate of the movement speed of shadows. The sun moves  $0.00436$  degrees per minute. As a result, the shadow cast by a leaf at 3 meters height moves at  $1.3\text{cm}/\text{min}$ . That means the shadow is  $13\text{cm}$  away from the place where it was ten minutes ago. The change in the angle of the sun also causes shadow shapes to change.

The complexity of light distribution patterns makes the study of light fields a very challenging task. Therefore, it is highly desirable to simplify it before a field is studied. The light decomposition process described in the next section fulfils this work.



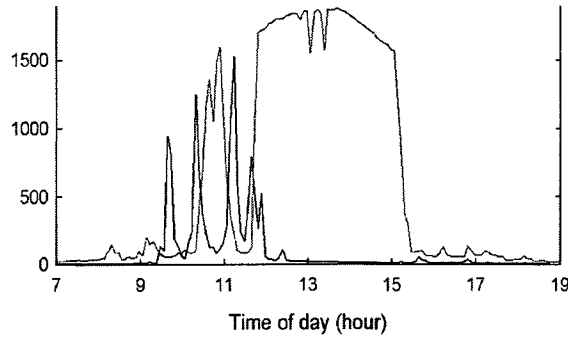


Figure 2.2: Light intensity over time at two spots.

## 2.2 Understory Sunlight Decomposition

The sun radiates energy to the earth in the form of light beams. These beams span a large frequency band. The only band of interest to us is the visible band, which is what we mean by sunlight from now on. When these light beams reach the atmosphere of the earth, some beams reach the ground directly while others go through diffusion, reflection and refraction processes. The strength of these activities differs according to the obstacles these beams encountered in their trips to the ground. In an open area, the only obstacles are the air molecules. The light reaching the ground in an open area is thus composed of the direct beam  $S$  and the sky diffused light  $D_s$ . The intensity of each component is closely related to the air pressure. Weiss et al.[109] made an approximation on the intensity of these two components based on Bouguer's (Beer's) Law as:

$$\begin{aligned}
 I_S &= 600 \exp[-0.185(P/P_0)m] \cos \theta \\
 I_{D_s} &= 0.4(600 - I_S) \cos \theta
 \end{aligned}
 \tag{2.1}$$

where  $P/P_0$  is the ratio of actual to sea level (101.325kPa) pressure,  $m$  is the optical air mass and  $\theta$  is the zenith angle. Under a forest canopy, these light components are further diffused by the forest foliage. Then two more components are added as the light reaches the ground level: a foliage scattered direct beam  $H_S$ , and a

foliage scattered sky diffused light  $H_D$ . Since  $D_s$  is much weaker than  $S$ ,  $H_D$  can be ignored. Therefore, the light under a forest canopy is composed of three possible components: direct beam  $S$ , sky diffused light  $D_s$  and foliage scattered direct beam  $H_s$ .

The range in which the foliage scattered light can reach is limited. When there are holes in a forest canopy, the foliage scattered light reaching the areas under these holes is very small and can be ignored. Such areas are called sunflecks. They are the brightest among all areas. The light in such areas is similar to that in an open area, which is composed of direct light beams and sky diffused light. Areas where all the direct beams and sky diffused light are blocked by the foliage are called umbras. In umbras, the only light available is the foliage scattered light. These are the darkest areas. The third type of areas are those where light intensity lies in between sunflecks and umbras. Such areas are denominated penumbras. These three types of areas are mingled together, with penumbras lying in between sunflecks and umbras as transition areas. Only a portion of the direct beams and sky diffused light reaches penumbras. Besides, the vicinity of penumbras to the foliage also results in stronger foliage scattered light than in open areas, which cannot be ignored. Hence the light reaching penumbras is a mixture of direct beam, sky diffused light and foliage scattered light.

## 2.3 Sunlight Models

Light models are useful in studying the change of the radiation regime. With the help of proper light models, one can decide on ideal plant patterns that maximize the accessible light of each plant or improve the water-use efficiency.

Many models have been developed to describe the radiation regime inside plant canopies. These models are divided into two classes, depending on whether a geometrical approach or a statistical approach is applied. A geometric model typically

deals with an individual type of plants. It simulates the tree crown through a geometric shape such as a cone, a sphere or a cylinder[4][76]. In statistical models, probability density functions are classically derived by considering the plant canopy as a horizontally homogeneous medium. The canopy is divided into  $N$  independent layers. Depending on the leaf dispersion patterns, the probability of a light beam being intercepted by  $n$  layers can be modelled as one of four distributions[13][58]: 1) a Poisson distribution, corresponding to random leaf dispersion ( $N$  is infinity in this case); 2) a positive binomial distribution, corresponding to regular leaf dispersion; 3) a negative binomial distribution, corresponding to clumped leaf dispersion; 4) a markov chain, corresponding to variable leaf dispersion. All these statistical treatments deal with light penetration within a canopy and cannot be directly applied to the study of light distribution under the canopy.

Other authors tried to model the light distribution[78][70] and correlation[9] directly from the data measured over a transect under a canopy. In these studies, the spatial resolution of the data is low, limiting the model accuracy. In addition, their light models are obtained through the data measured over the whole area. The models are applicable to a mixture of all three lights components. As illustrated in Fig. 2.3, the typical light distribution under a forest canopy is very complex. A different empirical density function is necessary for each field. Therefore, modelling the light distribution over the whole field unnecessarily complicates the models.

Since different light components go through different paths, their statistical distributions also differ from each other. Consequently, light distributions at different areas are not all the same. Instead of looking at the light distribution in a whole, Ross[91] modelled the three types of areas separately. This approach is also followed in this dissertation.

First, the density distribution function of each light component is analyzed one by one here. The direct beam intensity is constant. The sky diffused light is the direct beams being diffused by a lot of air molecules. Based on the law of large numbers,

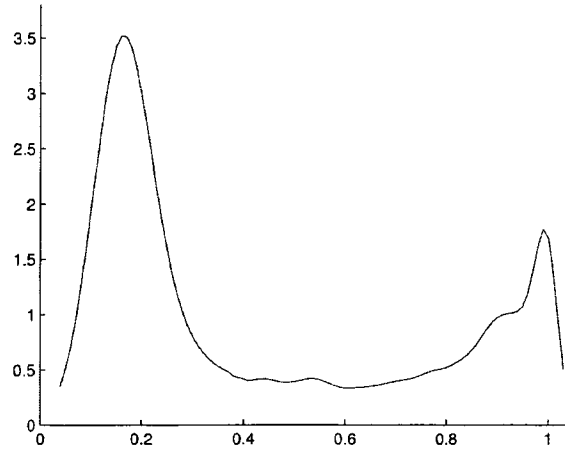


Figure 2.3: Typical light intensity distribution under canopy

it can be modelled as a Gaussian distribution, as Ross[91] has done. But in our measurements, we found its variance is very small and can be safely approximated by a constant too. The foliage diffused light has a much higher variance than the sky diffused light. Ross[91] also used a Gaussian model with a larger variance to model its density function. One problem with the Gaussian model is that it allows negative values, which is impossible for light intensity. Our measurements indicate that a log-normal distribution provides a better fit on the data, as demonstrated in Fig. 2.4. Intuitively, since this process results from many reflections (a multiplicative process), on physical grounds a log-normal distribution is also expected. Based on the light components contained in an area, the light distribution of each type of area is analyzed next.

### 2.3.1 Sunflecks

Light in sunflecks is composed of direct beam and sky diffused light. Since the intensity of both components is approximately constant, the light intensity in sunflecks is a constant

$$I_F = I_S + I_{D_s} \sim C_F \quad (2.2)$$

Its spatial correlation is also a constant.

### 2.3.2 Umbras

The only light component in an umbra is the foliage diffused light, which follows a log-normal distribution

$$\log I_U = \log I_{H_s} \sim \mathcal{N}(\mu, \sigma) \quad (2.3)$$

Depending on plant species and canopy thickness, the mean and variance of this distribution varies in a large range. When the canopy is thick, the light is reflected and refracted more frequently. Hence more energy is absorbed. In this case, the mean light intensity in umbras is low and the variance is high. Fig. 2.4 shows the light intensity distribution in umbras under two different types of canopies. The variances of the log-normal distributions fitted to them are 0.1887 and 0.3523 respectively.

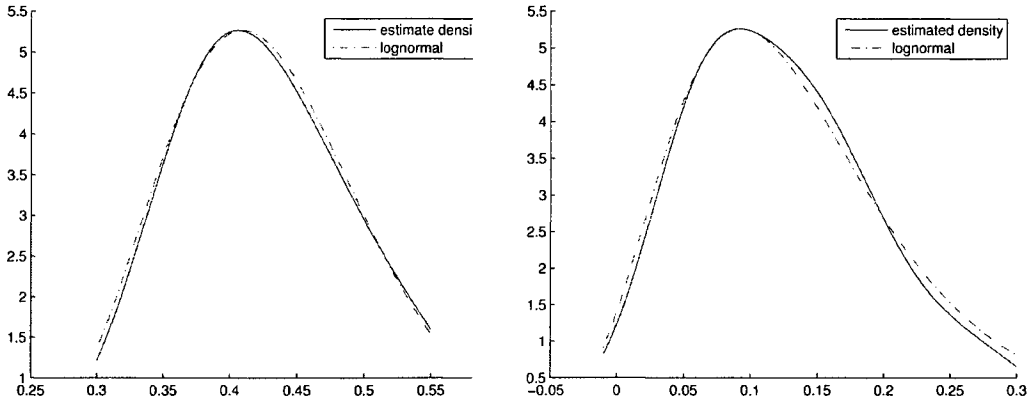


Figure 2.4: Light intensity distribution in umbras. (a) site 1; (b) site 2

The spatial correlation of the light in umbras is also affected by the canopy thickness. The thicker the canopy is, the broader the area the same light beam affects, resulting in a larger correlation value and longer correlation distance. This is demonstrated in Fig. 2.5.

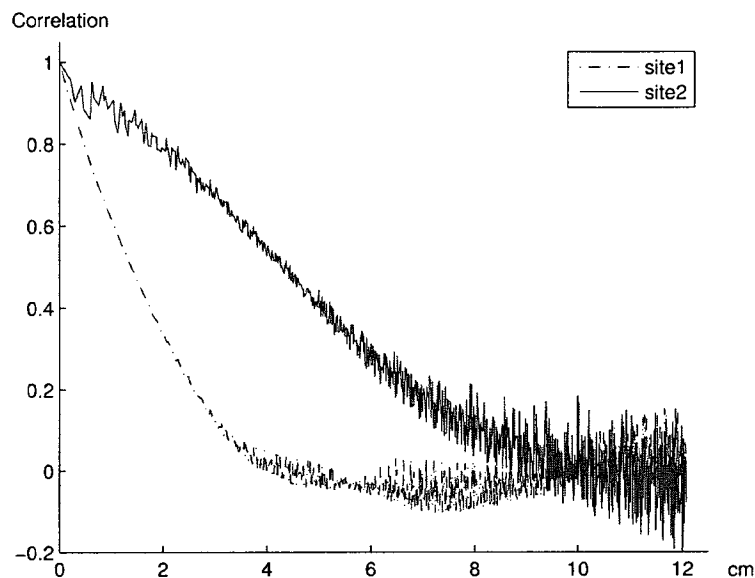


Figure 2.5: Light correlation model in umbrae

### 2.3.3 Penumbras

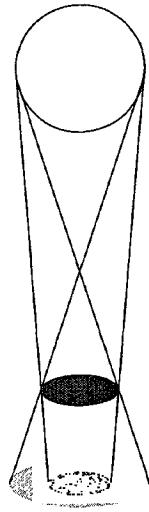
The fact that the sun is not a point source creates the so-called penumbral effect. In penumbras, the solar disk is neither fully visible nor completely blocked. This effect is illustrated in Fig. 2.6. Direct beam intensity due to the simple penumbral effect can be computed through geometry. Let the maximum direct beam intensity be 1. Suppose there is an disc with radius  $r$  at  $d$  meters high. The shade casted by this disc is a circle. Let  $x$  be the distance of a point to the center of the circle. Inside the ring with outer radius  $r + x_1$  and inner radius  $r - x_2$ , the direct beam intensity  $I_x$  decrease continuously from 1 to 0 and can be expressed as

$$I_x = 1 + \frac{u\sqrt{1-u^2} - \cos^{-1}u}{\pi} \quad (2.4)$$

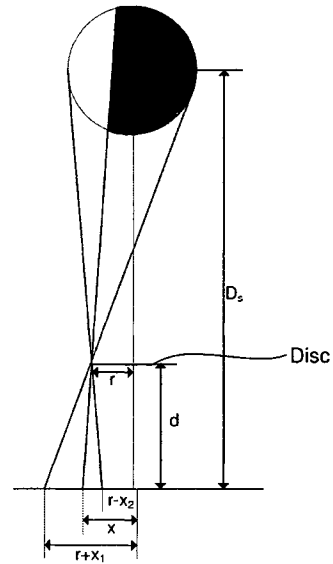
where

$$\begin{aligned}
 u &= h/R_s \\
 h &= \frac{xd - D_s(x - r)}{\sqrt{d^2 - (x - r)^2}} \\
 x_1 &= \frac{b + \sqrt{b^2 + ac}}{a} \\
 x_2 &= \frac{-b + \sqrt{b^2 + ac}}{a} \\
 a &= (D_s - d)^2 - R_s^2, \quad b = (D_s - d)rd, \quad c = (R_s^2 - r^2)d^2 \quad (2.5)
 \end{aligned}$$

$D_s$  is the distance from the sun to the earth and  $R_s$  is the radius of the sun.



(a) 3-dimensional view



(b) 2-dimensional view

Figure 2.6: Penumbra effect

When the ratio between the radius and the height of the disc  $r/d$  is smaller than the angular radius of the sun  $R_s/D_s$ , there are always some direct beams reaching the shade cast by the disc. The direct beam intensity never reaches 0 in the shade in this case. Both the normal shade intensity and small shade intensity are illustrated in Fig. 2.7.

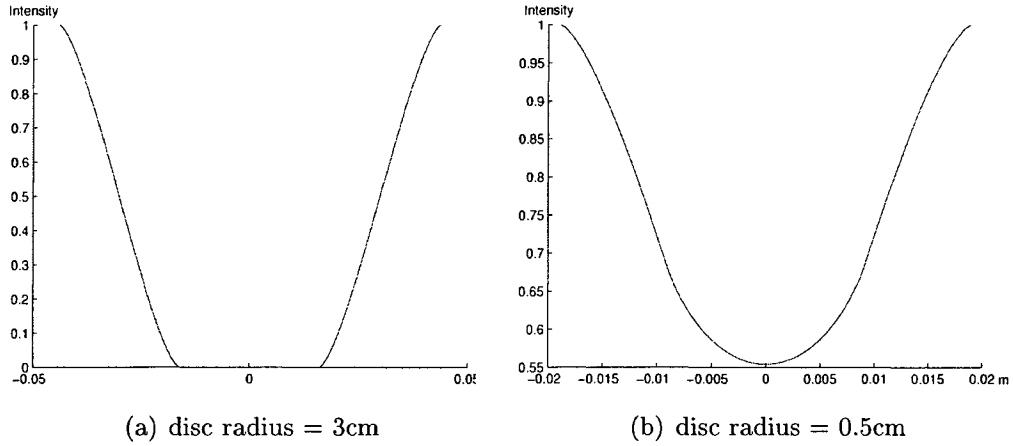


Figure 2.7: Intensity of a penumbra generated by a disc at 3 meters high.

In penumbras, the direct beam is partially blocked. Similarly, the sky diffused light is also partially blocked. Because of its vicinity to the canopy, more foliage diffused light reaches penumbras than in open areas. The contribution of diffused light to the total light in penumbras cannot be ignored. Thus, the light intensity in the penumbras is the summation of all three light components:

$$I_P = \alpha_1 I_s + \alpha_2 I_{D_s} + \alpha_3 I_{H_s} \quad (2.6)$$

In a simple penumbra as the one shown in Fig. 2.6,  $\alpha_1$  and  $\alpha_2$  are approximately equal, as indicated in Eq. (2.4). But under a real canopy, penumbras are much more complex and  $\alpha_i$ 's should be modelled as random variables. Therefore, the light intensity in a penumbra is a summation of three random variables. Its distribution is very complicated. Ross[91] used a Beta distribution to approximate the ensemble light distribution of all penumbras. Even if we assume the light distribution within an individual penumbra can be approximated by a Beta distribution, the parameters of the Beta distributions vary over penumbras.

The path each component of light goes through is independent from each other.



So the three light components are independent. Let  $C(i, j)$  denote the spatial correlation between two points  $i$  and  $j$ . In penumbras, the light correlation model can be expressed as

$$C_P(i, j) = C_{I_s}(i, j) + C_{D_s}(i, j) + C_{H_s}(i, j) \quad (2.7)$$

Note that although  $I_s$  and  $D_s$  are constant, their weights  $\alpha_1$  and  $\alpha_2$  are random.

Traditionally the light intensity field is assumed to be a spatially stationary process[9]. Hence the correlation between two points only depends on the distance between the two points, i.e  $C(i, j) = C(|i - j|)$ . Different areas have different correlation curves, as demonstrated in Fig. 2.8. Usually the correlation in umbras is stronger than that in penumbras, but both exhibit an exponential decrease over distance. In sunflecks, since the light intensity is constant, the correlation is 0. Thus we omit them in Fig. 2.8.

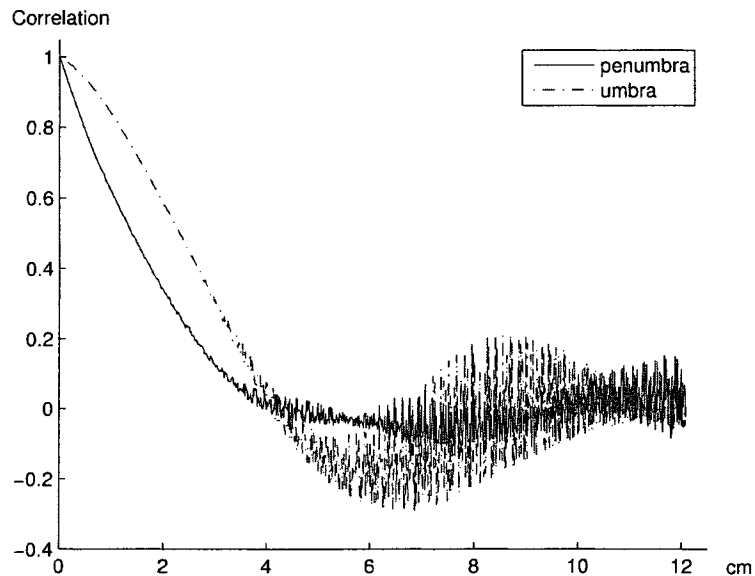


Figure 2.8: Typical light intensity correlation

## 2.4 Direct Beam Radiation Simulator

### 2.4.1 Simulation Model Generation

The complexity of the light distribution in penumbras and the difficulty in obtaining data with high resolution make simulation an attractive approach to obtain light models in penumbras. Oker-Blom[81][82] assumed a model of randomly dispersed leaves of equal size. For each point, she located a volume that a leaf needs to reside in to cast (at least partial) shade at that point. She then randomly generated leaves within the volume and determined the solar radiation intensity at that point. The assumption that leaves are of equal size and shape and are uniformly distributed within the canopy is not very realistic. This assumption restricts the model's application to few types of trees. Wang et al.[107] generated a light radiation model for an array of trees. But the resolution of their result is much coarser than what we need. Hence, a new simulation model is created to analyze the direct beam radiation.

The tree crown is simulated as a cone or a box, depending on users' input. The top and bottom of the box/cone corresponds to the maximum and minimum height of the canopy provided by users. Leaves are assumed to be either randomly scattered in the canopy or clumped. If the leaves are clumped, the clump model should be given as an input. A detailed leaf distribution model is also acceptable as the input. Leaves' azimuth angles are assumed to be uniformly distributed between  $[0 - 360^\circ]$ . Leaves' zenith angle are also assumed to be uniformly distributed between  $[\beta_{min} - \beta_{max}]$ , where  $\beta_{min}$  and  $\beta_{max}$  are specified by users. A leaf image is read in as the prototype for leaf generation. The true leaf size should also be provided. If the user specifies that leaves are not all of equal size, maximum and minimum leaf size should be provided. The leaf size follows the distribution specified by the user. If this input is absent, the leaf size is assumed to be uniformly distributed between the minimum and maximum size. The inputs of the simulator are summarized below:

1. Crown shape: box or cone.
2. Leaf distribution model. Default is uniform distribution within the crown.
3. Maximum and minimum leaf zenith angle.
4. Leaf image.
5. Maximum and minimum leaf size.
6. Leaf size distribution. Default is a uniform distribution between minimum and maximum size.

The simulator is implemented as follows. Let  $r$  be a point on the ground. Assume an observer looks up at the sun from  $r$ . Also assume his view is partially obstructed by a leaf. The sun disc is projected at the leaf's height level to determine the portion of the sun that is blocked. We dub this projected sun disc "apparent sun image". The apparent sun image is illustrated in Fig. 2.9. One sun image is generated at each leaf's height level. Each leaf image is convolved with its corresponding sun image to compute the shade the leaf left on the ground. Generally a sun beam does not encounter only one leaf on its path to the ground. When multiple leaves are encountered, the intensity value of the shade generated by each leaf at the same point is multiplied together. This approach essentially reaches the same statistical result as counting the number of leaves a beam would encounter when the leaves are uniformly distributed within the crown. One realization of the simulation result is shown in Fig. 2.10.

#### **2.4.2 Simulation Model Validation**

Validation is the process of determining whether a simulation model is a faithful representation of the system. Once the simulation program is written, an important procedure is to check the validity of the model. Since a simulation model is always generated for a special purpose, the first validation step should be checking whether

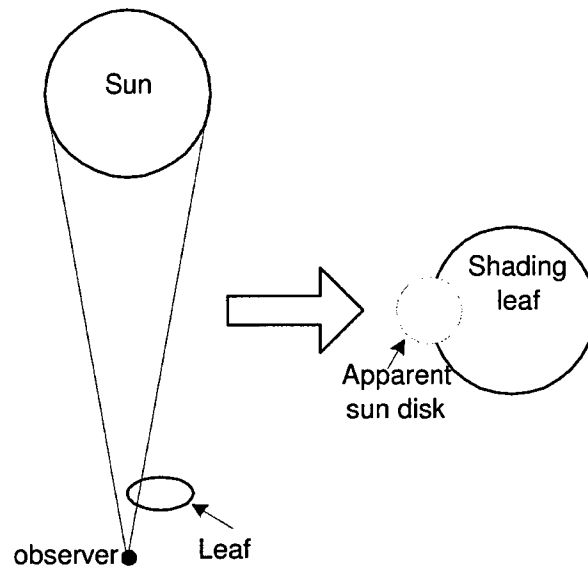


Figure 2.9: Apparent sun image to an observer on the ground

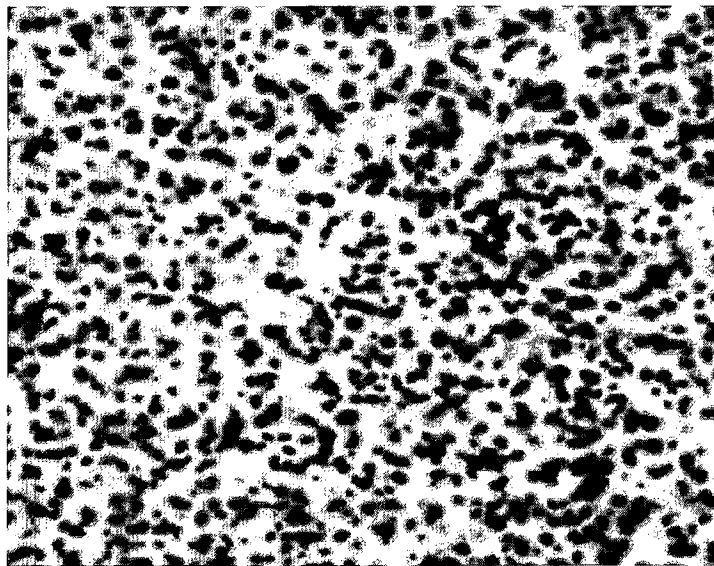


Figure 2.10: Simulated direct beam intensity on the ground

the simulation program serves that specific purpose. It should also be checked to see if it matches the measured data. There are also many other validation techniques. In our study, we validate our simulation model by comparing it with other available models and collected data.

Norman et al.[69][79] analyzed the gap distribution in a tree canopy and the penumbral effect. They derived theoretical models for sunfleck size distribution and direct beam radiation intensity distribution. The same two distribution models are empirically obtained from the simulated data and compared with those theoretical models. Fig. 2.11 verifies that the gap distribution from the simulation data is similar to the theoretical distribution. The curves from the two different sources also match very well for light intensity distribution, as can be seen in Fig. 2.12.

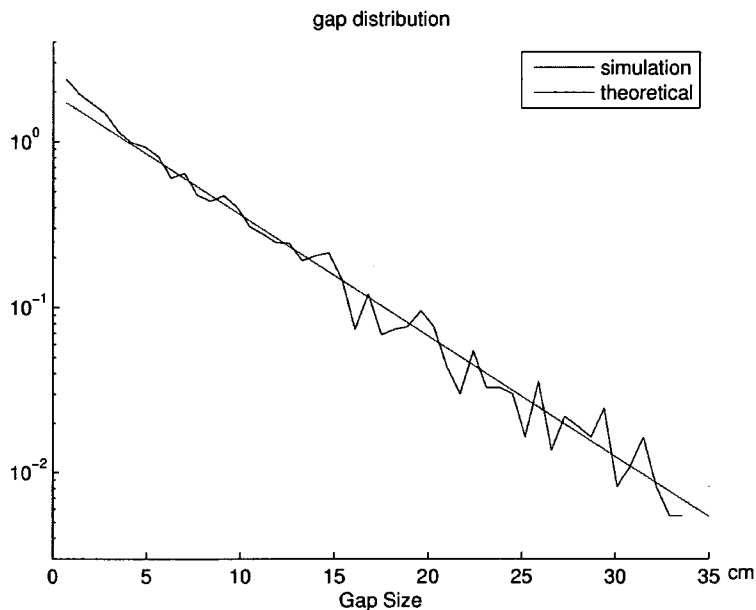


Figure 2.11: Gap distribution comparison

One important contribution of this simulation model is to determine the spatial correlation model in penumbras. As explained before, the light distribution in

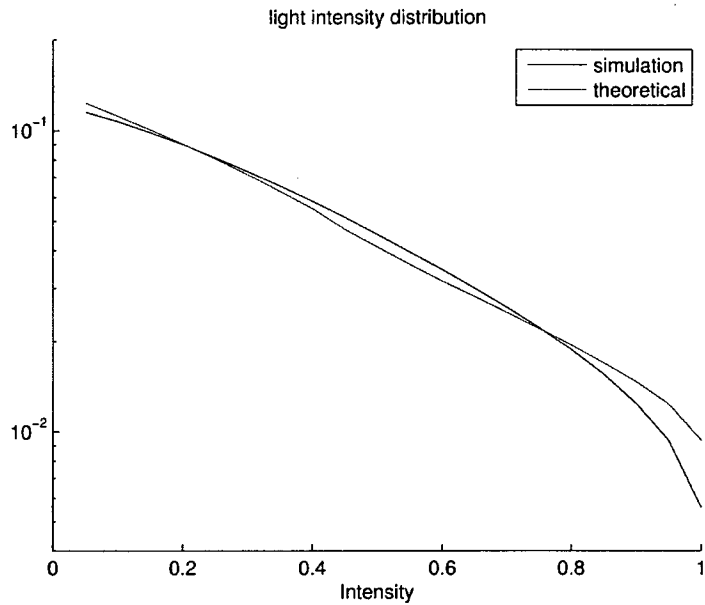


Figure 2.12: Direct beam intensity distribution comparison

penumbras is complex and differs from area to area. Without high-resolution measurements, we can use the simulated data to evaluate the correlation structure in penumbras. Since the weight of the direct beam and sky diffused light is similar in a penumbra while the foliage scattered light is much weaker than the other two components, the correlation structure of the direct beam should be a good indication of the total correlation structure. To prove this, the correlation model from simulated data is compared with the one from measured data and is shown in Fig. 2.13. It shows that the direct beam correlation has a similar shape as the total correlation and plays an important role in the total correlation. Hence, without measurements, total correlation level can be roughly inferred from the simulated direct beam correlation. This information is important for determining the minimum sampling density, which defines an upper limit on the total number of samples.

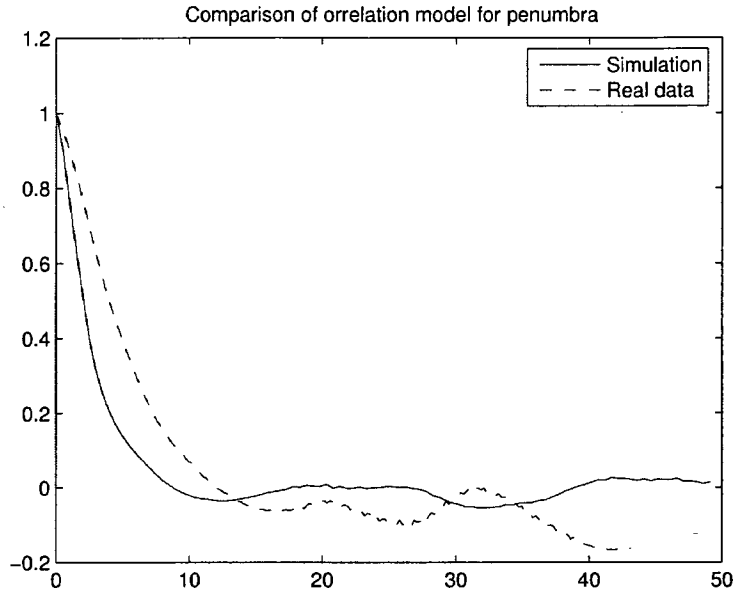


Figure 2.13: Light correlation model in penumbras.

## 2.5 Conclusion

Light beams radiated from the sun undergo refraction, reflection and diffraction before they reach the ground level on earth. The obstacles they meet on their path determine the light intensity distribution on the ground. Decomposing light into components and modelling each component separately greatly simplifies the analysis of light distribution. In this chapter, we examine the density distribution and correlation models of light components from measured data. Based on the components in the light reaching an area, the area can be categorized into one of three types: a sunfleck, a penumbra or an umbra. The light in sunflecks and umbras is modelled with simple density functions. However, the light in penumbras is too complex to be modelled in this way. To overcome the difficulty, a simulator simulating the distribution of direct beams, the light component that has the most important effects on the light distribution in penumbras, is created and validated. This simulator will complement real data to obtain the correlation models of penumbras.

## CHAPTER 3

# Overview of Existing and Proposed Sampling Methods

The concept of a sampling process has two meanings: 1) to get a representation from a large population; 2) to convert a continuous signal into a discrete stream. Sampling of a random variable refers to the former and is well studied by statisticians. In the signal processing community, we usually refer to the latter when we talk about sampling processes. Sampling a random signal involves both concepts. On one hand, the signal is discretized in the process. On the other hand, each sampled signal is only one realization of the signal space.

Sampling a single or a small number of random variables is much easier than sampling a random signal because of the smaller space involved. However, methods for sampling random variables can be extended to sampling random signals.

There are two basic sampling schemes: deterministic and random. In deterministic sampling, the sampling pattern follows a deterministic rule. In random sampling, sampling points are chosen according to a randomized rule. More complex sampling schemes are usually a combination of the basic sampling rules.

In this dissertation, the primary objective in sampling design is to reconstruct a signal from samples, although other statistics will also be briefly discussed in later sections. For a random variable, this corresponds to finding the mean of the variable. The reconstruction error is the Mean Square Error (MSE). It includes both bias and variance:

$$MSE = Var + (Bias)^2 \quad (3.1)$$



A set of samples is only a subset of the whole population. Each time samples are taken, a different subset is drawn from the whole population. Therefore, sampling processing itself is also random. Thus the  $MSE$  in Eq. (3.1) is actually the expected  $MSE$ . When the whole population is not available, which is true in almost all practical problems, the expected  $MSE$  also has to be estimated from the samples.

This chapter is organized as follows. The methods for sampling random variables are briefly reviewed in the first section. This is followed by a review of the existing methods for sampling signals. An overview of the two major sampling methods discussed in this dissertation is presented in section 3.3. Section 3.4 concludes this chapter.

The notation used in this chapter is listed below.

$\mathcal{X}$	a space of random variables
$ \mathcal{X} $	the cardinality of $\mathcal{X}$
$X$	a random variable in $\mathcal{X}$
$x$	a realization of $X$
$X(s), s \in \mathbb{R}^d$	a continuous random signal
$X(i), i \in \mathbb{N}^d$	a discrete random signal
$N =  \mathcal{X} $	the cardinality of a discrete random variable space
$R(\tau)$	the correlation structure of a wide-sense stationary (WSS) random signal
$\sigma$	the standard deviation of a random variable. $\sigma^2 = R(0)$ for a WSS random signal.

### 3.1 Methods for Sampling Random Variables

In this section, most of the sampling methods and statistical results are given for the discrete random variables. These results are based on Cochran's book[21]. They

can be easily extended to continuous random variables. Since all these methods are unbiased, only the variance term exists in Eq. (3.1).

### 3.1.1 Random Sampling

The easiest way to take samples is making random draws from a space. There are two approaches to treat these samples: the samples drawn from the whole space are treated equally or samples from each subspace are treated differently. This difference results in two random sampling methods: simple random sampling and stratified random sampling.

#### 3.1.1.1 Simple Random Sampling

This is the method that all samples are drawn from the whole space with equal probability. The estimated mean and MSE are:

$$\text{Estimated mean} \quad E\hat{X} = \frac{1}{n} \sum_i^n x_i$$

$$\text{Mean Square Error} \quad MSE = \frac{\sigma^2}{n}$$

If  $N$  is finite, MSE should be corrected with a multiplicative factor of  $1 - \frac{n}{N}$ .

#### 3.1.1.2 Stratified Random Sampling

In stratified random sampling, the whole space  $\mathcal{X}$  is first divided into  $L$  nonoverlapping subspaces  $\mathcal{X}_i$ . These subspaces are called strata. In each stratum,  $n_i$  samples are taken. Some of the advantages gained from stratified sampling include: 1) precision in each stratum can be controlled separately; 2) stratification may be required for convenience; 3) stratification may improve overall estimate precision. Let  $w_i = \frac{|\mathcal{X}_i|}{|\mathcal{X}|}$  represents the stratum weight. The estimated mean and MSE are:

$$\text{Estimated mean} \quad \hat{EX} = \sum_i^L w_i \frac{1}{n_i} \sum_j^{n_i} x_{ij}$$

$$\text{Mean Square Error} \quad MSE = \sum_i^L \frac{w_i^2 \sigma_i^2}{n_i} - \sum_i^L \frac{w_i \sigma_i^2}{N}$$

If each sample generates a cost of 1, the optimal allocation of samples in each stratum  $n_i$  that minimizes the total cost is proportional to  $w_i \sigma_i$ ,  $\sigma_i$  being the standard deviation of  $i^{th}$  stratum. With the total number of samples fixed, the MSE of different allocations of samples has the following relationship:  $MSE_{opt} \leq MSE_{prop} \leq MSE_{ran}$ .  $MSE_{opt}$  and  $MSE_{ran}$  are the MSE's of optimal and random allocation respectively.  $MSE_{prop}$  corresponds to the allocation when number of samples in each stratum is proportional to its weight, i.e.  $n_i = w_i n$ .

### 3.1.2 Deterministic Sampling

#### 3.1.2.1 Systematic Sampling

Suppose the whole population is numbered in some order. Systematic sampling is the method of taking samples at every  $l^{th}$  index. The index of the first sample is randomly picked. For example, assume the index of the first sample is  $k \in [1, l]$ . Then the indices of samples are  $k, k + l, k + 2l \dots$ . This method is commonly known as uniform sampling in signal sampling process. The estimated mean and MSE are:

$$\text{Estimated mean} \quad \hat{EX} = \frac{1}{n} \sum_i^n x_i$$

$$\text{Mean Square Error} \quad MSE = \frac{\sigma^2}{n} \frac{N-1}{N} [1 + (n-1)\rho]$$

where  $\rho$  is the correlation between samples. If the population is randomly numbered or there is no correlation between samples, and  $N \gg n$ , the systematic sampling has the same expected MSE as random sampling. When samples are correlated, they effectively form a signal. This situation is covered in more detail in section 3.2.

### 3.1.2.2 Cluster Sampling

Sometimes the characteristics of the whole population can be inferred from several groups of samples. A group of samples forms a cluster. Thereby comes the name "cluster sampling". Previously explained sampling methods can be employed to select clusters. That is, clusters are chosen randomly or systematically, with or without stratification. In random cluster selection, the probability with which a cluster is selected is either equal among all clusters or proportional to clusters' size.

Once clusters are fixed, depending on the way the selected clusters are treated, this sampling method can be further classified as one-stage cluster sampling or two-stage cluster sampling. In one-stage cluster sampling, each selected cluster is thoroughly sampled. That is, all elements within the cluster are measured. If instead only some samples are taken from each cluster, the method is called two-stage cluster sampling. Within each cluster, random or systematic sampling methods can be applied.

Each combination of sampling methods for cluster selection and element sampling is associated with an different MSE. For the simplest two-stage cluster sampling, random cluster selection with equal probability combined with random sampling of  $m$  elements within each cluster is

$$\begin{aligned} \text{Estimated mean} \quad \hat{EX} &= \sum_i^n \sum_j^m \frac{x_{ij}}{mn} \\ \text{Mean Square Error} \quad MSE &= \frac{\sigma_1^2}{n} \frac{N-n}{N} + \frac{\sigma_2^2}{mn} \frac{M-m}{M} \end{aligned}$$

where  $M$  is the number of elements within each cluster,  $\sigma_1$  is the standard deviation among clusters, and  $\sigma_2$  is the standard deviation within cluster (assuming all clusters have the same intracluster standard deviation). Readers interested in other methods can refer to [21].

Another result that is worth noting is the probability of selecting a cluster. If the element mean and variance within clusters are not related to the cluster size,

the methods that select a cluster with probability proportional to its size gives the smallest overall MSE compared with other selection methods.

### 3.1.2.3 Double Sampling

All the sampling methods introduced up to now do not require any auxiliary variate. Sometimes such an auxiliary variate is necessary and its characteristics is not known in advance. Under such conditions, a number of samples need to be allocated first to learn the information about the auxiliary variate. This information is then applied in the chief sampling task in later phases. Such techniques are called double sampling. To determine the number of samples for allocation in each phase, a cost function is constructed:

$$C = nc_n + n'c_{n'}$$

where  $n$  is the number of samples allocated to the main task and  $n'$  is the number of samples used in characterizing the auxiliary variate.  $c_n$  and  $c_{n'}$  are the cost associated with these two tasks.

Although there are all kinds of auxiliary variates, the one that is most interesting to us is the weight of a stratum in stratified sampling. The expression for determining the best allocation of  $n$  and  $n'$  is very complex. A simplified approximation shows that the following relationship should hold:

$$\frac{n}{\sqrt{(\sum W_h \sigma_h)^2 c_{n'}}} = \frac{n'}{\sqrt{c_n \sum W_h (EX_h - EX)^2}}$$

where  $W_h$  is the true weight of a stratum,  $\sigma_h$  is the standard deviation within a stratum, and  $EX_h$  is the variable mean within a stratum.

## 3.2 Methods for Sampling a Signal

Sampling a signal is much more complicated than sampling a variable. Error in reconstructing a signal from samples originates from two sources: 1) measurement

error; 2) aliasing. Measurement error adds noise to the measured signal and is the source of the variance item in Eq. (3.1). Even if there is no measurement error, when  $X(t)$  is discretized, there can still be bias in the reconstructed signal due to aliasing. Various sampling methods have been designed to minimize the reconstruction error and are briefly reviewed here. Since some methods are specifically designed for sampling a two-dimensional signal, we sometimes use the word "field" instead of "signal" to emphasize that the signal is two dimensional.

### 3.2.1 Uniform Sampling

The most popular scheme for sampling a signal is uniform sampling, where sampling points are taken on a uniform grid. For a one-dimensional signal, the samples are  $X(kT)$ ,  $k = 1, 2, \dots$ , where  $T$  is a fixed period. For a multi-dimensional signal, the periods for each dimension are not necessarily the same. For example, the samples of a two-dimensional signal are  $X(mS_1, nS_2)$ .

Shannon, in his classic paper, pointed out that a signal can be faithfully reconstructed by sinc functions as long as the sampling rate is higher than the signal's Nyquist rate[95]. This basic sampling theorem is extended in several directions.

Vetterli et al.[105] noted that a signal is not necessarily bandlimited to reach errorless reconstruction. They defined the rate of innovation of a signal as the number of degrees of freedom per unit of time. They proved that signals with finite rate of innovation (not necessarily bandlimited) can be perfectly reconstructed if they are uniformly sampled at (or above) the rate of innovation. They also extended their work to allow the presence of measurement error[63]. Instead of getting perfect reconstruction, they tried to minimize the error and got an approximation of the signal.

Unser[103] also looked at signal approximation. He extended Shannon's model in basis function spaces from a single sinc function to a Riesz basis. A Riesz basis is a subset of  $L^2$  whose functions have strictly positive lower and upper bounds.

He presented a detailed approximation error analysis from the frequency domain. Specifically, he emphasized the importance of prefiltering, pointing out that without prefiltering, the out-of-band signal generates two sources of errors: 1) its information is not presented by the samples; 2) it introduces aliasing. He also looked at the rate of decay of approximation error and provided alternatives to the Nyquist rate for selecting the sampling step.

Balakrishnan[6] discussed Shannon's sampling theorem in the context of random signals and proved that the sampling theorem also held in the mean square sense. He relaxed the strict requirement on the signal bandwidth. Instead of requiring the frequency components outside the cutoff bandwidth to be strictly zero, he proved that as long as the following two conditions are satisfied, i.e. 1) the frequency on the negative side and on the positive side are complements to each other in the out-of-bandwidth region; 2) the frequency spectrum is continuous at the cutoff bandwidth, the signal can be fully reconstructed.

The uniform sampling method is also widely used in sensor networks. Due to the usual assumption of the limitation on the sensors' communication capabilities, study of sampling process is often combined with transmission rate[56][27]. Cristescu et al.[27] claimed that there exists an optimal sampling density when the information is relayed through sensors. When the sampling density is higher than that optimal density, the approximation error would increase due to the noise added by the relaying sensors.

Since the uniform sampling method is simplest and the most important, there are also many other discussions on this sampling method. These discussions are not detailed here. See [103] for a review and related references.

### **3.2.2 Static Design**

There are two disadvantages associated with uniform sampling: 1) when the whole signal is not uniformly distributed, the uniform sampling method doesn't make best

use of all samples; 2) uniform sampling pattern suffers from a lack of refinability when the number of samples increases. Müller[74] improved the refinability of uniform sampling by the solution which he denominated "coffee house" design. But this problem is usually solved by deploying samples adaptively and will be addressed in the next section.

The first problem is solved by designing the sampling points based on the variance matrix of the signal. Such an approach is known as an optimal design in the statisticians' community. Let  $n$  be the number of samples and  $t_i, 1 \leq i \leq n$  be the sampling points. A simple extension to the uniform sampling is to obtain each sample at  $(n - 1)^{-1}$  percentile of a density function  $h$ [14]:

$$\int_0^{t_i} h(x)dx = \frac{i - 1}{n - 1}$$

Müller[71] proved that the optimal density function for minimizing the integrated MSE is proportional to the square root of the autocorrelation function of the signal when constant bandwidth is used in nonparametric regression from the samples.

Johnson et al.[43] looked at the design problem from the perspective of maximizing the distance between sampling points. They provided two designs. Suppose all points are selected from a  $T \times T$  grid. The selected points are in a set  $S \subset T$ . Let  $d(t, s)$  be the distance between two points  $t$  and  $s$ . The *minimax distance design*  $S^*$  is chosen to be

$$S^* = \arg \min_S \max_{t \in T} d(t, S)$$

where  $d(t, S) = \min_{s \in S} d(t, s)$ . This design is a G-optimal design, which minimizes the maximum entry of the residual error matrix  $X'(X'X)^{-1}X$ . The second design, the *maximin distance design*

$$S^o = \arg \max_S \min_{s, s' \in S} d(s, s')$$

is a D-optimal design, which maximizes the determinant of the information matrix of the sampling points. Although both designs are straightforward in concept, their



computational complexity is very high. Besides, they restrain the sampling points to grid points, which may be a strong limitation in some applications.

The above mentioned design approaches require prior knowledge of the signals to be sampled. Sometimes we lack such knowledge. Then we turn to adaptive sampling (also known as *sequential design*), where we select sampling points based on information gained from previous samples.

### 3.2.3 Adaptive Sampling

Various adaptive sampling methods have been proposed. These algorithms mainly differ in the assumptions on the signal and cost functions to minimize.

Several authors tried to add new sampling points following the density of the existing points and minimize the difference between the measurements and the prediction values. Müller-Gronbach et al.[72] assumes the signal is continuously differentiable. They selected points with density proportional to the estimate of the local Hölder constant. Similarly, Faraway[33] assumed the signal was two-differentiable and chose design points following the density of the second derivative of the signal.

All the approaches mentioned above assume a certain order of smoothness in the signal. In reality, such a condition may be violated. Nowak et al.[80] dropped this assumption and tried to reconstruct inhomogeneous two-dimensional fields. They assumed a field was composed of two or more smooth regions separated by smooth boundaries. They estimated the boundary of the field from existing samples and allocated new samples at the boundary. Guestrin et al.[38] made a different assumption on the field. They modelled the field as a Gaussian process and looked at the sampling problem from the perspective of mutual information. They selected sampling points that provided the maximum increase in mutual information. This approach requires a lot of computation and is difficult to be extended to large fields and high resolution sampling. Both works also incorporate communication costs in their sampling consideration[55][111].

Sometimes, the field to be sampled exhibits both spatial and temporal variation. In such cases, we need sampling methods that can capture the temporal variation while reconstructing the spatial field. Rahimi et al.[88] considered this temporal variation in their algorithm. They gave lower weight to previous samples in making prediction at the new sampling points and minimized the difference between measurements and prediction values. They also took the sampling and movement speed of robot sensors into account in selecting the next sampling points. Wikle et al.[110] approached the similar problem from a different angle. Instead of using lower weight on previous samples, they updated the field through a Kalman filter.

### 3.2.4 Compressed Sensing

Although all the adaptive approaches exhibit improvement over static approaches when there is no prior information available, a recent work by Donoho[30][102] proved that adaptive sampling does not improve the error convergence rate. He showed that as long as the signal is compressible by transform coding with a known transform, the signal can be accurately reconstructed with random samples. When the transform coefficients belong to an  $l_p$  ball,  $0 < p \leq 1$ , it is possible to reach reconstruction accuracy with  $n$  nonadaptive measurements comparable to that attainable with direct knowledge of the  $N$  most important transform coefficient. The  $N$  coefficients can be extracted from the  $n$  measurements by solving a linear optimization problem.

## 3.3 Our Approaches

The work by Donoho is a milestone on the sampling algorithm design. It puts an upper bound on the improvement an adaptive algorithm can reach. But this method is not universally applicable. First of all, it is only optimal on the signals that are compressible in a known transform domain. The knowledge on the domain that

a signal is compressible has to be available in the reconstruction process. If this knowledge is not available or the signal is not compressible in any transform domain that is known at present, the signal cannot be optimally reconstructed. Second, although error convergence rate is important, when a large number of samples are involved, the constant in the MSE expression also plays an important role. Sampling an inhomogeneous field with high reconstruction fidelity is an example of such a situation. Therefore, adaptive sampling algorithms are still necessary in some practical applications. We designed two sampling algorithms that overcome some limitations imposed on the adaptive sampling methods reviewed in the previous section. A brief overview of these two algorithms is given here. Details will be presented in Chapter5 and Chapter6 respectively.

### 3.3.1 Adaptive Sampling

The adaptive sampling algorithm is an iterative algorithm that adapts sampling site to the field variation. A sampling step is followed by a reconstruction step in each iteration. At the sampling step, the most desirable sampling sites are picked out based on the reconstruction result from the previous iteration. These sites form a candidate pool. A certain number of samples are selected from this pool. Instead of looking at the derivative of the signal directly, we introduce the concept of *bending energy*, which is an integration of the signal's second derivative. The locations where the bending energy is high correspond to the locations the field varies quickly. Intuitively, more samples are necessary in these locations. Therefore, bending energy is a good indication of the sampling density. If the field is not strictly differentiable, we can set an upper limit in the numerically computed derivative and remove the indifferentiable points in the integration. In this way, we relax the strict differentiable constraint on the field.

Without ground truth, it is hard to determine the stopping point of the iteration. The common practice is to compare two consecutive reconstructions and set a

threshold on the difference. But this approach often results in stopping the algorithm prematurely. To avoid this problem, we setup a Bayesian framework to estimate the probability that the fidelity constraint is not met. In addition, our algorithm also allows source statistical models to be incorporated in the Bayesian framework when they are available. This probability is updated after each reconstruction step.

### 3.3.2 Multiscale Sampling

When we study the sunlight field in Chapter 2, we have shown that the light field is a heterogeneous field. In some areas the field varies quickly, while in other areas it varies slowly. This field, as a whole, is compressible neither in time/space domain, nor in frequency domain. However, if we are given extra information so that we can partition the whole field into homogeneous subfields, each subfield being compressible, we will be able to optimally sample the field. Since the extra information is not directly connected to the field reconstruction process, it resides at a higher level than measurements of the field. Therefore, we establish a hierarchy of information. The concept of fusing information from different levels in the sampling process is called multiscale sensing. This concept is illustrated in Fig. 3.1.

There are many possible sources for the high-level information. They can be the measurements of the hidden forces that drive the variation of the field, such as the time of a day, the wind speed, etc. They can be the measurements from sensors whose data are not directly used in the field reconstruction. Historical data can also provide useful information if the time correlation of the field is strong.

To effectively fuse the high-level information with the field measurements, we design a multiscale sensing algorithm that is similar to the two-phase stratified sampling scheme we explained in section 3.1. In this algorithm, two types of sensors with different resolution and accuracy are employed. One type of sensors provide high-level information such as field partition and field models. Sampling points are designed based on the models. Sensors of second type take samples at these points.

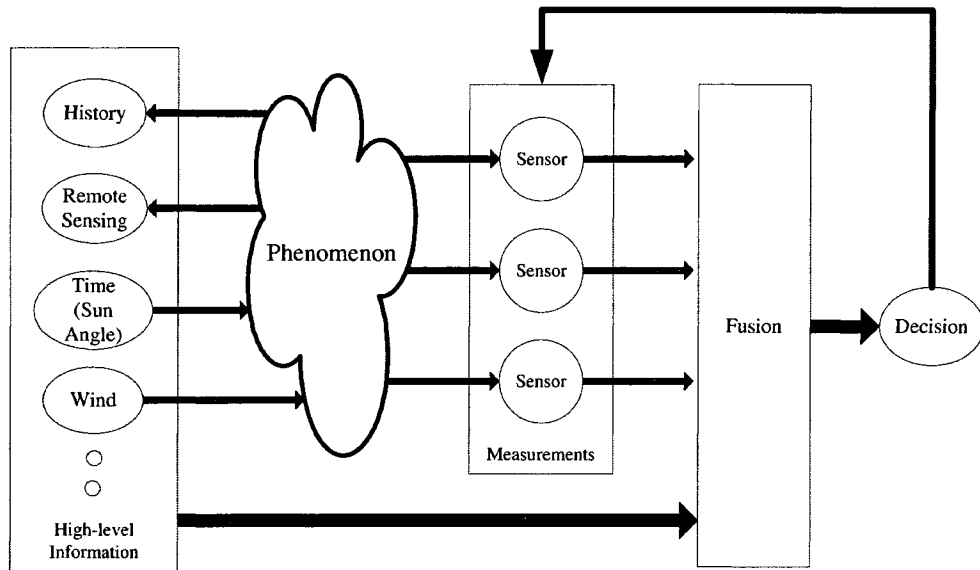


Figure 3.1: The multiscale sensing scheme.

The field is reconstructed through the samples and models.

### 3.4 Conclusion

In this chapter, we provide some background on the sampling process. Reconstruction errors are analyzed for simple schemes. The motivation for designing two new sampling algorithms was explained after cutting-edge sampling algorithms were reviewed. An overview of the two new algorithms was given at the concept level. Their details are left to later chapters.

## CHAPTER 4

### Reconstruction Techniques

The final objective of sampling a field in this dissertation is reconstructing the field within a certain fidelity constraint. In Chapter 3, the sampling techniques, which discretize the field, were reviewed. The reconstruction process is the reverse process to sampling. In the reconstruction process, the value of the field at locations without samples are estimated. The sampled discrete field is converted to a discrete field with much denser grid points than the sampling points or back to a continuous field. This process is sometimes called curve fitting or regression analysis. In Chapter 3, the reconstruction process was referred to without being elaborated on when some sampling methods were reviewed. In this chapter, reconstruction techniques are described in detail.

There are two types of reconstruction: interpolation and approximation. The distinction between them is whether the reconstructed field has exactly the same values as the measurements at the sampling points. Interpolation requires that the fitted curve passes precisely through the sampling points, while approximation does not impose such a restriction on the fitted curve. We will use the word "regression" to refer to both methods.

In this chapter, our measurement model will be

$$y(x_i) = f(x_i) + \epsilon_i, \quad i = 1, 2, \dots, n, \quad x_i \in [0, 1] \quad (4.1)$$

$\epsilon_i$  being Gaussian noise.  $f(x)$  will be called the underlying function of the field. The  $k^{\text{th}}$  derivative of  $f(x)$  is denoted by  $f^{(k)}$ . We say  $f \in L^k$  if  $f$  is differentiable at exactly  $k$  times. The reconstructed signal will be denoted by  $\hat{f}$ .

The curving fitting process can be either parametric or nonparametric. If the curve is believed to follow a certain simple function, such as a polynomial or an exponential function, the parameters of the function (the coefficients and maybe the order of the polynomial as well) are estimated from the measurements, thus getting the name "parametric". In contrast, nonparametric method is used when the function format is unknown or when the curve is so complex such that one set of parameters is not suitable everywhere. Since the fields we study fall in the latter class, we will only review the nonparametric method here. In this case, the curve fitting is typically convolution based. The fitted curve is expressed as

$$\hat{f}(x) = \sum_{i=1}^n f(x_i)k(x, x_i) \quad (4.2)$$

where  $k(x, x_i)$  is a basis function (also known as a kernel function).

In interpolation, the noise is assumed to be zero. The Eq. (4.2) is equivalent to

$$\hat{f}(x) = \sum_{i=1}^n y(x_i)k(x, x_i) \quad (4.3)$$

When the measurements contain noise, the approximation approach is more appropriate. In this approach, a fitted curve usually does not pass exactly through the sampling points. Hence, the weights for the kernel function are estimated values instead of sampled values. If there is not enough information to get an exact representation of the true curve, approximation is a better approach than interpolation.

The quality of kernel functions can be evaluated either in the frequency domain or in the time domain<sup>1</sup>. From the sampling theory explained in chapter 3, we know that when a signal is bandlimited, the signal can be fully reconstructed from a sinc function. The sinc function in the frequency domain is a rectangular function, which has flat passband and no sideband. Since the sinc function is not time-limited, it

---

<sup>1</sup>We usually refer to an one-dimensional signal as a time signal and a two-dimensional signal as a spatial signal. Both sampling and reconstruction methods reviewed in the previous chapter and in this chapter are not limited to one-dimensional signals. But to follow the convention, we still called the signal that is bounded in its dimensions "time-limited" and the dual to the frequency domain the "time domain". Our time signal here is not necessarily one-dimensional.

cannot be used in reality. When the kernel functions are evaluated in the frequency domain, their Fourier transforms are compared with the transform of the sinc function. Hence, the goal of designing a good kernel in frequency domain is to reduce the sideband while keeping the passband flat. Lehmann et al.[57] presents a comprehensive catalog of kernel functions with a comparison of their Fourier transforms.

In the time domain, important characteristics of the a kernel function include its smoothness, its bandwidth and whether the sum of all overlapped kernel functions at a point is one, i.e.  $\sum_i k(x_0, x_i) = 1$ ,  $k(x_0, x_i)$  being the kernel function  $k(x, x_i)$  evaluated at a point  $x_0$ .

A lot of regression methods have been proposed. A chronological review of interpolation methods by Meijering[68] is a good starting point to learn the evolution of the regression methods. The simplest method is the nearest neighbor interpolation, where the value at a location  $x$  takes the values of the sampling point closest to it. Its kernel is a rectangular function in the time domain. Various other kernels have been designed. A popular family of kernels is that of polynomial kernels. Within the polynomial family, spline functions are given a lot of attention because of their special characteristics. Among the unbiased linear estimators, the kriging estimator is the one that has minimum variance. It is not kernel based. Instead, it estimates the value at a point by the weighted sum of its neighbors.

Some regression methods are developed for special applications, such as image edge enhancement[42], unknown sampling points[65], distributed regression[37], etc. Although these methods are useful to some extent to our study, they are not closely related to the methods used in our algorithms and are not reviewed here.

This chapter is organized as follows. The basic polynomial regression methods are reviewed in section 4.1. Although spline regression is essentially polynomial regression, because of its special features, it is separately discussed in section 4.2. Kriging is introduced in section 4.3. Reconstruction errors are analyzed in section 4.4. Section 4.5 concludes this chapter.



## 4.1 Polynomial Regression

Polynomials are popular because they are smooth and their values are easy to compute. We usually assume the underlying function supporting the measured data is smooth. The family of polynomial kernels is a proper choice of regression functions under this assumption. Its property of low computational complexity makes it especially attractive for curve fitting in computer graphics, where a large amount of computation is involved. Chung[19] developed an algorithm for fast interpolation using a cubic polynomial kernel.

In approximation, when the parametric method is used, the order of the underlying polynomial has to be determined from the data, which is not an easy task. Higher order polynomials provide a better fit to the available samples, but they also incur more danger in overfitting, resulting in too much noise being incorporated. In such cases, the *Akaike Information Criterion*(AIC) or the *Bayesian Information Criterion*(BIC) is often used to determine this parameter.

The order of polynomial kernels is also an important parameter in nonparametric regression. For interpolation, the cubic polynomial is a frequent pick. Different cubic polynomial kernels have been designed by imposing different boundary conditions. The most popular form of the kernel is

$$k(x) = \begin{cases} A_1|x|^3 + B_1|x|^2 + C_1|x| + D_1, & \text{if } 0 \leq |x| < 1 \\ A_2|x|^3 + B_2|x|^2 + C_2|x| + D_2, & \text{if } 1 \leq |x| < 2 \\ 0 & \text{if } 2 \leq |x| \end{cases} \quad (4.4)$$

By enforcing continuity and continuous differentiability requirement, seven coefficients can be determined and the kernel reduces to

$$k(x) = \begin{cases} (a+2)|x|^3 + (a+3)|x|^2 + 1, & \text{if } 0 \leq |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & \text{if } 1 \leq |x| < 2 \\ 0 & \text{if } 2 \leq |x| \end{cases} \quad (4.5)$$

Such a kernel first appeared in [10]. Further conditions can be added to determine the value of  $\alpha$ , such as requesting the interpolation function and the Taylor series expansion of the underlying function agree for as many terms as possible[51]. Cubic polynomial kernels under this condition yield a fourth order approximation<sup>2</sup> of the underlying function. A piecewise cubic Lagrange polynomial also reaches a fourth-order approximation. But the interpolation function is not continuously differentiable and has sharp edges at the sample points. This effect is not desirable in many scenarios. Instead of four-point interpolation, the interpolation kernel size can also be increased to meet the need of some applications. Lehmann[57] gives some examples of larger kernel size.

Approximation theory stated that every continuous function on a closed interval can be approximated uniformly to any prescribed accuracy by a polynomial. This is not necessarily true for interpolation[68]. So polynomial kernels are often used in approximation. Again, cubic polynomial kernels are prevalently used. A special family of polynomial kernels, the splines, is particularly favored and explained in the next section.

Another important parameter in kernel regression is the kernel bandwidth. Most reconstruction methods use fixed bandwidth. Muller[73] proved that locally varying bandwidth leads to a reduction in the leading term of the asymptotic expression of the MSE. He also gave an expression for the optimal local bandwidth when samples were taken in equidistance. This optimal bandwidth is determined by the smoothness of the underlying function, the number of samples, the noise variance and the kernel function itself.

---

<sup>2</sup>We call an approximation "Lth-order approximation" if the approximation function space reproduces polynomials of order  $L - 1$ .

## 4.2 Spline Regression

Let  $(t_0 =)a < t_1 < t_2 < \dots < t_N < b(= t_{N+1})$  be a subdivision of the interval  $(a, b)$  by  $N$  distinct points. A spline function of order  $d$  is a function which

- (1) in each open interval  $(t_{i-1}, t_i)$  for  $i = 1, 2, \dots, N + 1$ , is a polynomial of degree  $\leq (d - 1)$
- (2) has  $(d - 2)$  continuous derivatives in the open interval  $(0, 1)$ .

$t_i$ 's are called knots of the spline function. For each fixed set of knots, the class of the spline functions defined above is a linear space of functions with  $(N + d)$  free parameters. The B-spline basis for this linear space is formed by  $(N + d)$  normalized B-splines

$$B_i(x) = (t_{i+d} - t_i)[t_i, \dots, t_{i+d}](t - x)_+^{d-1}, \quad i = 1, \dots, N + d \quad (4.6)$$

where  $[t_i, \dots, t_{i+d}]\phi$  denotes the  $d^{\text{th}}$  order divided difference on the  $(d + 1)$  points  $t_i, \dots, t_{i+d}$  of the function  $\phi$ .  $f_+$  equals  $f$  if  $f > 0$  and zero otherwise. The B-splines defined in Eq. (4.6) have the following properties[2]:

- (1)  $0 < B_i(x) \leq 1$  for  $x \in (t_i, t_{i+d})$  and  $B_i(x) = 0$  otherwise;
- (2)  $\{B_i\}_{i=j}^{j+l}$  is linearly independent over the interval  $[t_{j+d-1}, t_{j+l+1}]$  for any  $l \geq d - 1$  and any  $1 \leq j \leq N + d - l$

$$(3) \sum_{i=1}^{N+d} B_i(x) = 1 \text{ for all } x$$

$$(4) \int_0^1 B_i(x) dx = (t_{i+d} - t_i)/d, \quad i = 1, \dots, N + d$$

These properties make B-splines very attractive in interpolation. Since they are easy to construct, they have wide utilization in curve plotting. The book by Bartels et al.[8] gives a thorough explanation of B-splines.

Because of the smooth nature of splines, they are also often employed in function approximation. When the knots are fixed, the weights of the splines are usually determined by a least square error (LSE) estimator. Reinsch[90] proved that, for a given set of points  $(x_i, y_i), x_0 < x_1 < \dots < x_n$ , cubic splines are the solution to the problem

$$\begin{aligned} & \min_{g(x) \in L^2[x_0, x_n]} \int_{x_0}^{x_n} g''(x)^2 dx \\ \text{s.t. } & \sum_{i=0}^n \left( \frac{g(x_i) - y_i}{\Delta y_i} \right)^2 \leq S \end{aligned} \quad (4.7)$$

where  $\Delta y_i$  and  $S$  are given numbers. Since then, the  $m^{\text{th}}$  derivative of the fitted function has often been applied as the penalty in the LSE estimator. Assume the measurement model as expressed in Eq. (4.1). The fitted spline can be expressed as the minimizer of

$$n^{-1} \sum_{i=1}^n (g(t_i) - y_i)^2 + \lambda \int_0^1 (g^{(m)}(t))^2 dt \quad (4.8)$$

The parameter  $\lambda$  controls the amount of smoothing. To avoid incurring too much smoothing in the approximation, Craven and Wahba[25] proposed to use a generalized cross-validation estimate to determine  $\lambda$ . Wahba also discussed the relationship between spline smoothing and Bayesian estimation and derived confidence intervals based on the posterior covariance function of the estimate[53][106].

If knots are not fixed, the space of splines is no longer a linear space of functions. Determining both the knots location and the weights is a nonlinear least square problem. Sometimes it is referred to as the full functional problem. Optimal solution to such a problem requires the knowledge of the underlying function and complex computation[2]. Various fast algorithms have been developed to reach suboptimal solutions, such as knot insertion[61], knot selection[67], separating knot location determination and LSE estimation[45], etc.

Just as one-dimensional splines originate as the solution to Eq. (4.7), two-dimensional splines are the functions that minimize the quantity

$$\iint_{\mathbf{R}^2} \left( \frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 g}{\partial y^2} \right)^2 dx dy \quad (4.9)$$

One family of two-dimensional splines for fitting a surface to scattered data is the *thin-plate spline*. It is expressed as

$$g(x, y) = a_0 + a_1 + a_2 + \sum_{i=1}^n w_i r_i^2 \log(r_i^2) \quad (4.10)$$

where  $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ . The term  $U(r) = r^2 \log(r^2)$  satisfies the equation

$$\Delta^2 U = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)^2 U \quad (4.11)$$

and is called the fundamental solution of the biharmonic equation  $\Delta^2 U = 0$ [11].

### 4.3 Kriging

The kriging estimator is the best unbiased linear estimator. The estimated value at a point  $x_0$  is a linear combination of  $n$  surrounding data points

$$\hat{f}(x_0) = \sum_{i=1}^n w_i y(x_i) \quad (4.12)$$

The weights  $w_i$  are chosen as:

$$\begin{aligned} & \min \text{Var}[\hat{f}(x_0) - y(x_0)] \\ & \text{s.t. } E[\hat{f}(x_0) - y(x_0)] = 0 \end{aligned} \quad (4.13)$$

Since the kriging estimator is an unbiased estimator, the reconstruction MSE is the kriging variance

$$\text{Var}[\hat{f}(x_0) - y(x_0)] = \sum_{i=1}^n \sum_{j=1}^n w_i w_j R(x_i, x_j) + \text{Var}(y) - 2 \sum_{i=1}^n w_i R(x_i, x_0) \quad (4.14)$$

where  $R(x_i, x_j)$  is the correlation between  $x_i$  and  $x_j$ .

Several types of kriging exist, depending on the assumption on the trend of the underlying signal[26]: 1) simple kriging: the trend is constant and known; 2) ordinary kriging: the trend is constant and unknown; 3) universal kriging: the trend follows a linear model.

Kriging originates from the mining industry. Therefore, it is frequently used in predicting spatial data not on regular grids. Its origin also determines that it is most suitable for making estimation at scattered points. When the number of points to be estimated is large, the computational complexity will be very high. Kerwin et al.[50] extended the technique for space-time function estimation by updating the correlation model of the data in a way similar to the Kalman filter.

Kriging and spline interpolation are closely related. As pointed out in [66], spline interpolation is equivalent to kriging with fixed covariance and fixed degree of polynomial trend. Dubrule[31] compared the two interpolation methods and concluded that kriging is more accurate than the spline, although the latter is generally less intensive in computation.

#### 4.4 Error Analysis

Shannon's sampling theory tells us that there is no reconstruction error if the samples are taken above the Nyquist rate and the reconstruction is done through the sinc function. For sampling rates lower than the Nyquist rate or in approximation, we evaluate sampling and reconstruction methods through the mean reconstruction error. With an  $L^{th}$ -order approximation, the bound for the minimum reachable error in reconstructing a function  $f \in L^2$  from uniform samples is[100]

$$\min \|f - \hat{f}\| \leq Ch^L \|\hat{f}^{(L)}\|_2 \quad (4.15)$$

where  $h$  is the sampling step and  $C$  is a constant. Unser et al.[104] investigated this general bound in several specific problems. In particular, they noted that

when samples are sparse (below the Nyquist rate), the least squares approximation performs as well as the corresponding interpolator with twice the order.

For random signals, the error can be analyzed through the correlation function of the signals. Specifically, let  $\Phi(x)$  be the Fourier transform of the correlation function and  $\omega_s$  be the sampling frequency. The mean reconstruction error for a  $d$ -dimensional signal can be expressed as[84]

$$E[f(x) - \hat{f}(x)]^2 = \frac{1}{(2\pi)^d} \int \Phi(\omega) \left[ 1 - \frac{\Phi(\omega)}{\sum_m \Phi(\omega + m\omega_s)} \right] d\omega \quad (4.16)$$

The error convergence rate with the sample size  $n$  is another important criterion for evaluating sampling and regression methods. Under appropriate regularity conditions, the optimal convergence rate of an estimator for a function of a  $d$ -dimensional variable  $f \in L^p$  is  $r = 2p/(2p + d)$ . That is,  $\|f - \hat{f}\|_q$  decreases at a rate of  $n^{-r}$  for  $0 < q < \infty$ . If  $q = \infty$ , the optimal rate is  $(n^{-1} \log n)^r$ [99].

## 4.5 Conclusion

In this chapter, several reconstruction techniques are reviewed. A signal is usually reconstructed through the convolution of samples and a kernel function. The basic kernels are polynomials. Reconstruction with this family of kernels is reviewed in general. An important member of this kernel family, the set of splines, is given further attention. In linear regression, we focus on the best unbiased linear estimator, i.e. the kriging estimator. Reconstruction errors are briefly discussed at the end. The splines and the kriging estimator will be used in our sampling algorithms.

## CHAPTER 5

# An Adaptive Sampling Algorithm in Sensor Networks

In traditional sampling problems in the signal processing area, samples are taken on a known signal. The purpose of sampling is to discretize a signal for transmission. In sensor networks, this process is reversed. There is no access to the signal before sensors are deployed in the field. Because a dense sampling of the signal is often prohibitively expensive, we need to learn the signal while samples are taken. This process brings about one major disadvantage: we cannot prefilter the signal. We have noted in Chapter 3 that out-of-band signals generate aliasing without prefiltering. To alleviate the aliasing problem, we employ a nonuniform sampling approach and try to add denser samples in the high-variation (high-frequency) regions. But when no or only partial knowledge about the signal is available, we cannot determine in advance where such regions are. Adaptive algorithms are especially helpful under this condition.

Although adaptive sampling has the same asymptotic behavior as random sampling according to the compressed sensing theory[30], adaptive sampling still exhibits improvement over random sampling and uniform sampling in term of MSE in sparse sampling scenarios. Many adaptive algorithms have proved this improvement as reviewed in Chapter 3. In this dissertation, a new adaptive algorithm is designed. This algorithm has also been briefly introduced in Chapter 3. The details of this algorithm will be explained here.

In this chapter, we will assume a similar measurement model as in Eq. (4.1),



except that we replace  $x_i \in [0, 1]$  by  $s \in \mathbb{R}^2$  to emphasize that we concentrate on a two dimensional field:

$$y(s) = f(s) + \epsilon(s), \quad s \in A \quad (5.1)$$

where  $\epsilon$  is a Gaussian noise with variance  $\sigma^2$ .  $A$  is a subset of  $\mathbb{R}^2$ . The reconstructed signal is represented by  $\hat{f}$ . When the noise is small, bias is the dominant term in the MSE. We will assume this is the case and focus only on the bias of the reconstruction error. Let  $D(s) = |\hat{f}(s) - f(s)|^2$  represent the reconstruction distortion at a point  $s$ .  $D_{max}$  and  $D_{ave}$  represent maximum distortion among all points and average distortion over all points respectively. The area of  $A$  is denoted by  $|A|$ . The design objective is

$$\begin{aligned} \|\hat{f} - f\|_2 &\leq D_{ave} \cdot |A| \\ \|\hat{f} - f\|_\infty &\leq D_{max} \end{aligned} \quad (5.2)$$

In sampling of the incident sunlight field, the application we focus on, even under mild wind conditions, the variation in the field is too fast to adapt to. So we only try to reconstruct the average field over a short period of time. We assume the field is static over a short period of time and do not incorporate time variation in the sampling design. We also assume the samples are taken by mobile sensors with an infrastructure, such as the Network InfoMechanical Systems (NIMS)[46], so that the locations of the sensors are not restricted to predefined grid points and are accurately known.

This chapter is organized in the following way. An overview is given in section 5.1. Key function blocks of the algorithm are elaborated in the next few sections. Simulation results of this algorithm are given in section 5.5 and are compared with some other algorithms. Section 5.6 concludes this chapter.

Some important symbols used in this chapter are listed below:

- $S$ : a set of points
- $S^k$ : the set of all sampling points up to iteration  $k$
- $V(S)$ : Voronoi tessellation of  $S$
- $V_i^k$ : the Voronoi cell surrounding a sampling point  $i$  at iteration  $k$
- $P_i^k$ : the probability that the fidelity constraint is not satisfied in  $V_i^k$
- $v$ : average fidelity constraint in a Voronoi cell

## 5.1 Algorithm Overview

Some of the adaptive sampling algorithms introduced in Chapter 3 have limitations that make them not suitable for our field. In particular, the algorithm in [38] is computationally intensive. It assumes that the field exhibits high spatial correlation to limit the number of samples. This assumption does not hold in our applications. The algorithm in [80] originates from image compression applications. Just like normal image compression techniques which begin with a panorama of the field, the algorithm exhibits good performance only in relatively dense sampling scenarios. The constraint on the number of samples in our application limited the usefulness of this algorithm.

Our algorithm follows a similar train of thoughts as [33] and [72] in that sampling density is proportionate to the roughness of a field. But we relax the smoothness assumption. Although we still use a function of the second derivative of the field as a criterion for selecting sampling points, we replace the second derivative by a large constant in regions where the field is not differentiable. In this way, we avoid clustering all samples in any such region. Thereby our method is extended to be applicable in reconstructing a field with discontinuities as used in [80]. Besides, using a Maximum A Posterior (MAP) sample selection criterion also takes into account the reconstruction fidelity constraint. Following the roughness of the field

instead of the estimated reconstruction error alone as done in [89] not only reduces the influence of measurement errors, but also improves the efficiency in sampling inhomogeneous fields.

The block diagram of our algorithm is depicted in Fig. 5.1. The algorithm is initiated with a small number of samples selected based on the "coffee house" designing pattern[74] and a field is reconstructed from these samples. Then the algorithm runs iteratively. A pool of sampling candidates, which closely follow the current sampling pattern, is maintained and updated after each set of samples are taken. At the beginning of each iteration, a new set of sampling sites are selected from the pool based on the MAP criterion. After samples are taken, the field is reconstructed through thin plate splines. The probability that the design objective in Eq. (5.2) has not been reached is evaluated. A maximum number of samples can be fixed in advance. The iterations stop after the design objective is achieved or the maximum number of samples are acquired, whichever happens first.

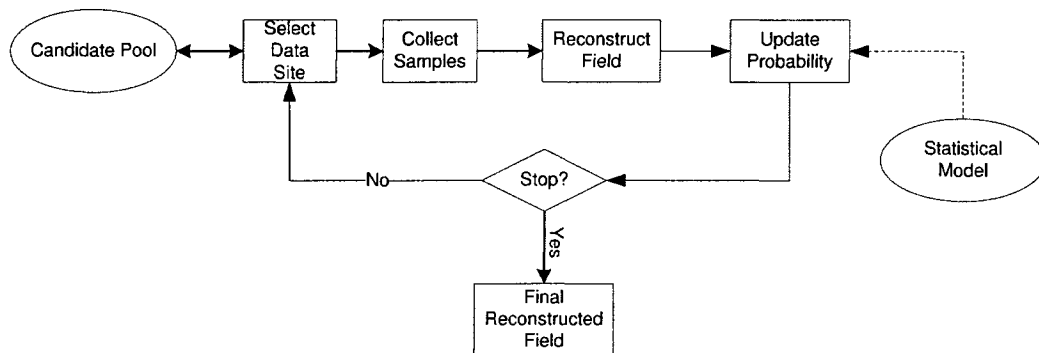


Figure 5.1: The block diagram of the adaptive sampling algorithm

## 5.2 Candidate Pool

Without enough knowledge of the whole field, we cannot design the globally optimal sampling points. But results from optimal design still provide useful insight.

Specifically, the *maximin distance design* in [43] is optimal in the sense of maximizing the differential information. This design maximizes the minimum distance between sampling points. So the candidates for new sampling sites are chosen to be the points that are farthest away from all existing samples. These sites are the centers of the triangles formed by existing samples. A tool for locating these centers is Delaunay triangulation.

A Delaunay triangulation for a set  $S$  of points in a plane is a triangulation  $DT(S)$  such that no point in  $S$  is inside the circumcircle of any triangle in  $DT(S)$ [92]. Delaunay triangulation maximizes the minimum angle of all the angles of all triangles in  $DT(S)$ . Edges of the triangles in the Delaunay triangulation are called Delaunay edges. For each Delaunay edge, there exists a circle that passes through its vertices and contains no other points in  $S$ . Delaunay triangulation can be obtained by finding line segments that satisfy such a condition. The closest pair of points are the vertices of Delaunay edges. Hence Delaunay triangulation locates the minimum distance between samples. An example of the Delaunay triangulation is illustrated in Fig. 5.2. Readers can refer to [92] for further details on Delaunay triangulation.

In our algorithm, Delaunay triangulation is performed on the existing samples. The centers of the resulting Delaunay triangles are the candidates for the next round of sampling, following the principle of the *maximin distance design*. It is possible that four or more points are on a circle such that the centers of multiple triangles coincide. A minor problem with this choice is that some samples may fall outside the boundary of the field. A correction to this problem is to use the centroid instead of the center of the triangles as the candidates when this problem happens. This correction violates the maximin design principle a little bit. But since the problem usually happens in the triangles near the boundary, only a small portion of candidates in the whole set is affected. These candidates will not influence the whole design a lot.

Let  $S^k$  denote the set of samples at iteration  $k$  and  $n_k = |S^k|$  be the number

of points in  $S^k$ . One useful property of Delaunay triangulation is that the total number of Delaunay triangles in  $DT(S^k)$  is no more than  $(2n_k - 5)$ . This means the number of candidates in the pool increases only linearly with the number of existing samples. There exists a fast algorithm for Delaunay triangulation with running time  $O(n_k \log n_k)$ . The running time can be further shortened by incremental schemes, which only update the pool in the neighborhood of new sampling sites. Therefore, computational complexity will not be a big concern in locating the candidates.

Another advantage of choosing candidates in this way is that the density of potential sites follows that of the existing samples. This means the density follows the local heterogeneity of the field. As proved by Muller-Gronbach et al.[72], a design following local heterogeneity is an optimal design under smooth conditions. Even for a field with abrupt boundaries, this property is also desirable since it helps locate the boundaries.

## 5.3 Sample Selection

### 5.3.1 The Probability of Meeting Sampling Objective

Before taking a new set of samples, we need to determine whether the fidelity constraint in Eq. (5.2) has been met. When the true field is not available, it is always a challenge to determine how well the fitted curve matches the true field. This is particularly true when we also need to determine the necessary number of samples. To avoid being overconfident on the fitted curve, we employ a probabilistic approach.

The probability is defined on the Voronoi cells of the existing samples. Voronoi cells are formed through Voronoi tessellation. A Voronoi tessellation  $V(S)$  of a set of points  $S$  is the dual graph of a Delaunay triangulation  $DT(S)$ . It is a kind of decomposition of a metric space determined by distances to the points in  $S$ . If the Voronoi regions of two points in  $S$  are edge-adjacent, they are connected by a Delaunay edge. Moreover, the centers of Delaunay triangles are the vertices of

Voronoi cells. An example of the Voronoi tessellation of a set of points is depicted in Fig. 5.2.

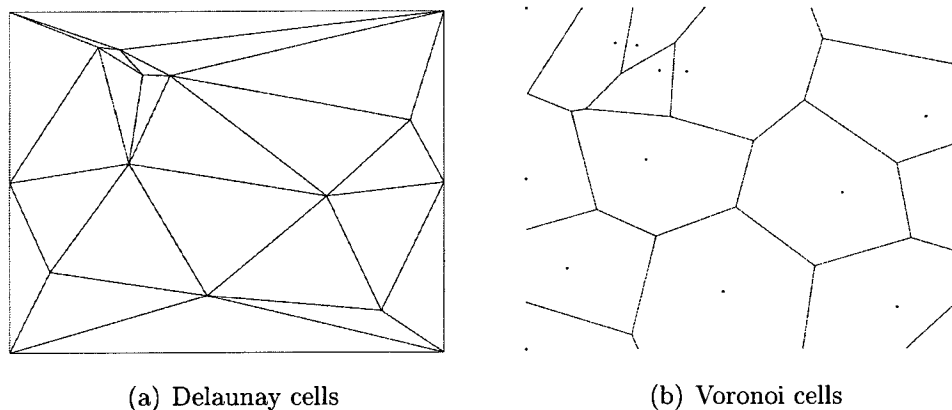


Figure 5.2: Delaunay triangulation and Voronoi tessellation

Let  $V(S^k)$  be the the Voronoi tessellation of the samples at iteration  $k$  and  $V_i^k$  denote the Voronoi cell corresponding to the  $i^{th}$  sampling site. The same fidelity constraint defined in Eq. (5.2) is imposed on all Voronoi cells. When the fidelity constraint is met in each cell, it is also satisfied in the whole field. But the reverse is not true. This means that requiring the fidelity constraint to be met in each Voronoi cell is a more stringent condition. Hence, we can loosen the constraint on each Voronoi cell a little bit by choosing a value  $v \geq D_{ave}$ .

Even after we shrank our fidelity test region from the whole area to each Voronoi cell, we still cannot determine whether the constraint is satisfied within the cell because there is only one sample within each cell. Therefore we define  $P_i^k$  the probability that the fidelity constraint is not met in the cell  $V_i^k$ .

### 5.3.2 MAP Sample Selection

To achieve the sampling objective with the minimum number of samples, we should first take samples at the site where the probability that the fidelity constraint is not

met is largest. Hence, we follow the Maximum A Posterior (MAP) criterion when sampling sites are selected from the candidates.

Each new sample generates a new Voronoi cell as well as causing some old cells to shrink. Recall that a candidate is the center (or centroid) of a Delaunay triangle as well as the common vertex of the Voronoi cells surrounding it. Denoting a candidate point for the  $k+1$  iteration by  $m$  and the set of vertices of its corresponding Delaunay triangle by  $\mathcal{O}_m$ , we can see clearly from Fig. 5.3 that this new Voronoi cell is mainly composed of portions of the  $V_j^k$ 's,  $j \in \mathcal{O}_m$ . Thus the probability  $\hat{P}_m^{k+1}$  that the fidelity constraint is not satisfied in the cell  $V_m^{k+1}$  can be estimated by a weighted sum of  $P_j^k, j \in \mathcal{O}_m$ :

$$\hat{P}_m^{k+1} = \sum_{j \in \mathcal{O}_m} \mu_j P_j^k \quad (5.3)$$

Intuitively, the weight  $\mu_j$  should be proportional to the area of the intersection between  $V_m^{k+1}$  and  $V_j^k$ . But to simplify the computation, we approximate the weight by the quantity

$$\mu_j \propto (d_{j1} + d_{j2})/r_j, \text{ and } \sum_{j \in \mathcal{O}_m} = 1 \quad (5.4)$$

in which  $d_{j1}, d_{j2}$  and  $r_j$  are defined as in Fig. 5.4. Note that  $r_j$  is the same for all  $\mu_j$  unless the centroid instead of the center of a Delaunay triangle is used.

The quantity  $\hat{P}_m^{k+1}$  of all candidates is computed and compared. The candidates with the largest probabilities are chosen to be the sampling sites in the next iteration. It is not necessary that we only take one sample at each iteration. Hence, this set can contain multiple sampling sites. The whole set of samples is then updated by including new sampling sites.

### 5.3.3 Probability Update

Once samples are collected and Voronoi tessellation is performed on the updated set of samples, we should update the probability  $P_i^k$  in all Voronoi cells to  $P_i^{k+1}$ . The

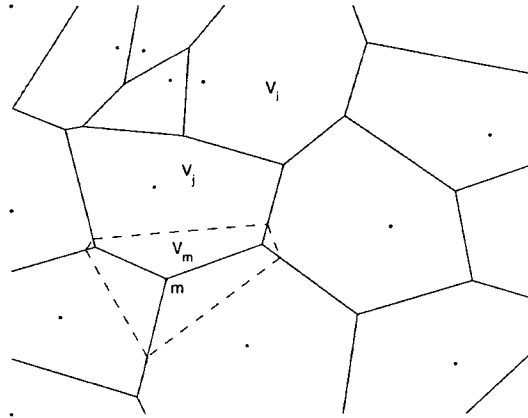
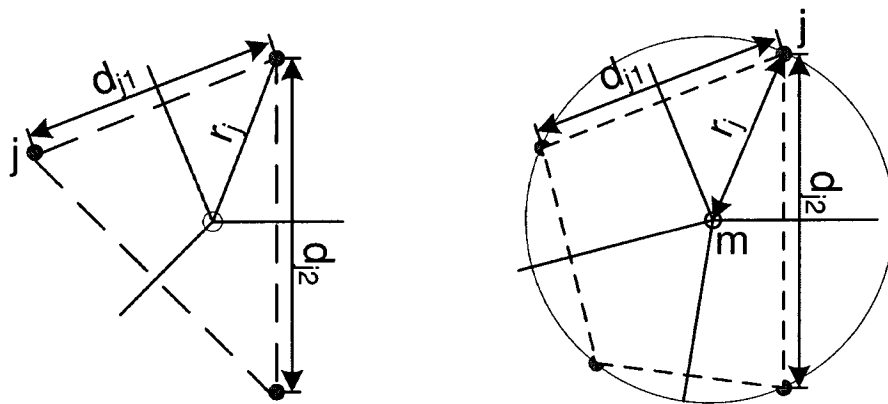


Figure 5.3: The boundary of the Voronoi cells before a new sample is added is shown in solid lines. After a sample is added, a new cell is generated, whose boundary is shown in dashed lines. Outside this cell, the old boundaries remain unchanged.



(a) A normal Delaunay triangle

(b) Two triangles coincide on a circle

Figure 5.4: Variable definition for determining weights. Dashed lines are boundaries of Delaunay triangles and solid lines are boundaries of Voronoi cells.



estimated error at a sampling point  $m$ , being denoted by

$$\Delta = \hat{f}(x_m, y_m) - f(x_m, y_m)$$

is tested against the fidelity constraint. The way the probabilities are updated depends on the testing result. If  $\Delta > \nu$ , we know that the fidelity constraint is more likely to be unsatisfied in the Voronoi cell containing  $m$  and its neighbors. Their probabilities of unsatisfying the fidelity constraint should be increased. Otherwise, they should be decreased. There are three different scenarios. Under each scenario,  $P_i^k$  is updated differently (Readers can refer to Fig. 5.3 for an example of each type of cell):

1.  $V_m^{k+1}$  contains a new sampling point  $m$

$$P_m^{k+1} = \begin{cases} 1 & \text{if } \Delta > D_{max} \\ \alpha \sum_{j \in \mathcal{O}_m} \mu_j P_j^k & \text{if } \nu < \Delta \leq D_{max} \\ \beta \sum_{j \in \mathcal{O}_m} \mu_j P_j^k & \text{if } \Delta \leq \nu \end{cases} \quad (5.5)$$

2.  $V_j^{k+1}$ ,  $j \in \mathcal{O}_m$  is one of the cells surrounding  $m$

$$P_i^{k+1} = \begin{cases} \alpha^{\mu_j} P_j^k & \text{if } \Delta > \nu \\ \beta^{\mu_j} P_j^k & \text{if } \Delta \leq \nu \end{cases} \quad (5.6)$$

3.  $V_i$  is far away from  $m$

The testing result at  $m$  does not provide much knowledge on  $V_i$  and its probability of not satisfying the constraint should remain unchanged:

$$P_i^{k+1} = P_i^k \quad (5.7)$$

In these equations,  $\alpha > 1$  and  $\beta < 1$  are two constants set according to  $\nu$ .

### 5.3.4 Incorporating Field Models

In Chapter 2, we emphasized the importance of a field model. A field model provides us some prior knowledge about the field to be sampled. It usually helps improve the efficiency of an algorithm. Although we often assume that the field is unknown when we design an adaptive algorithm, if a certain field model does exist, it will be helpful to incorporate this field model in the algorithm. As analyzed in Chapter 2, the incident light field is a heterogeneous field. In different subfields, spatial correlation is different. Algorithms will be more efficient if we account for this heterogeneity.

Since the field heterogeneity directly influences the sampling density, it is most convenient to absorb a field model in the estimated probability  $\hat{P}_m^{k+1}$  of candidates. Hence, we apply a compensating factor  $h_i$  to Eq. (5.3) and define

$$C_m = h_m \hat{P}_m^{k+1} \quad (5.8)$$

We then select the candidates with largest  $C_m$  instead of largest  $\hat{P}_m^{k+1}$  as the new sampling sites.

There are several statistical models we can consider. The most direct one is the correlation model. But to use this model requires knowledge on what subfield a sample is in. This knowledge cannot be obtained from one sample value.

The second candidate model is the field roughness, which is typically measured by the second derivative of the field. Considering that the size of a Voronoi cell is an indication of the sampling density in the area around it, we want to combine the two to avoid following the roughness of the field blindly. The following compensation factor is applied:

$$h_m^r = \left[ \left( \frac{\partial^2 \hat{f}(x_m, y_m)}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 \hat{f}(x_m, y_m)}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 \hat{f}(x_m, y_m)}{\partial y^2} \right)^2 \right] A_m \quad (5.9)$$

where  $A_m$  is the size of the Voronoi cell  $V_m^{k+1}$ . The roughness of the field is readily

available when the thin plate spline is the reconstruction kernel, as shown in the next section.

The third model we consider is the field variation model, which characterizes the uncertainty in the gaps between sampling points. This model measures the difference in value between two points  $d$  units away from each other. Let  $p, q$  be two points and the Euclidean distance between the two points be  $d$ . The field variation over distance is calculated as

$$\gamma(d) = E [f(x_p, y_p) - f(x_q, y_q)]^2, \quad \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} = d \quad (5.10)$$

Since the field is heterogeneous, the field variation is not uniform over the whole field. As shown in Chapter 2, the field exhibits fast variation in penumbras and small variation in umbras and sunflecks. Due to this characteristic, the field variation model is different in different areas. Since we cannot determine what area a point is in, we construct the model based on the value at a point  $p$ , assuming sunflecks are in general small. If  $f(x_p, y_p)$  is above a threshold  $I_t$ , it is more likely that the variation around  $p$  is high. Otherwise, it is low. Thus, two models are extracted from experiment data and are shown in Fig. 5.5. In both models,  $\gamma(d)$  is fitted with a rational quadratic model:

$$\gamma_i(d, f(x, y)) = \frac{a_i d^2 + b_i d}{1 + c_i d^2}, \quad i = \begin{cases} 1 & \text{if } f(x, y) > I_t \\ 2 & \text{otherwise} \end{cases} \quad (5.11)$$

These models are applied in the compensation factor as

$$h_m^v = \sum_{j \in \mathcal{O}_m^k} \mu_j \gamma_j(d_{mj}, f(x_j, y_j)), \quad d_{mj} = \sqrt{(x_m - x_j)^2 + (y_m - y_j)^2} \quad (5.12)$$

where  $m$  is a candidate point. Other parameters have the same definition as in Eq. (5.3).

Since the latter two models are rough and do not correctly characterize the field all the time, both models are used in our algorithm so that one model may still

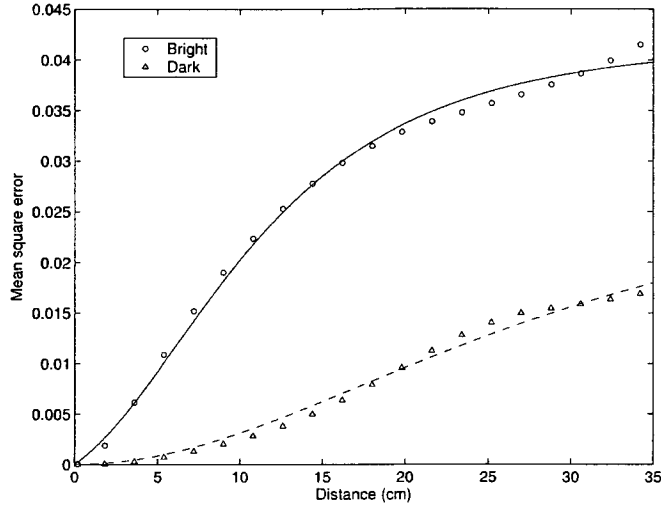


Figure 5.5: Sunlight field variation model

work should the other model fail. Hence, in the algorithm, two sets of samples are selected in each iteration: one is compensated by the field roughness model and the other by the field variation model.

## 5.4 Field Reconstruction

The field is reconstructed with the thin plate spline. The thin plate spline(TPS) has been briefly introduced in Chapter 4. Here, we will focus on some aspects that are particularly useful in our algorithm.

The kernel function of the thin plate spline is

$$g(x, y) = a_0 + a_1x + a_2y + \sum_{i=1}^n w_i r_i^2 \log(r_i^2) \quad (5.13)$$

where  $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ . This function, as a solution to the biharmonic equation (4.11), minimizes the bending energy defined as

$$I(g) = \iint_{\mathbf{R}^2} \left( \frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 g}{\partial y^2} \right)^2 dx dy \quad (5.14)$$

Hence,  $g(x, y)$  is the the physical shape a weightless elastic plate would take when

point forces are applied on it. This physical interpretation gives rise to some useful properties:

- The bending energy is non-decreasing over the number of point sources applied. Therefore, in our algorithm, the bending energy is non-decreasing over iterations.
- The thin plate spline converges to the true field. The reconstruction error is bounded by the difference in the bending energy of the true field and that of the reconstructed field[87]:

$$|f(x, y) - \hat{f}(x, y)| \leq h\sqrt{[I(f) - I(\hat{f})](\log 3)/(24\pi)} \quad (5.15)$$

where  $h$  is the longest side of the triangle formed by interpolation points that encircle  $(x, y)$ .

Since the bending energy indicates the roughness of the field, Eq. (5.15) suggests that we should sample more densely in the area that is not smooth, as being done in our algorithm. This matches our intuition and also provides a theoretical basis for the effectiveness of our algorithm.

If the measurement noise can not be ignored, the smoothing TPS should be used instead of the interpolation TPS. The weight for  $a_i$  in Eq. (5.13) is different. But the property of the spline does not change much. Therefore our algorithm remains the same.

If there is no field model available, the algorithm follows the roughness of the field. One problem is that it tends to cluster all samples in a small region when there is a discontinuity in the field. To overcome this problem, a small modification is made in our algorithm. We set an upper bound on the second derivative of the field. In places where the absolute value of the numerically computed second derivative of the reconstructed field is larger than the upper bound, the second derivative is set to equal the upper bound. The distance between samples is also properly chosen to

avoid too much clustering. A contour detection algorithm such as Snakes[49] can also be applied to track the edges in the field based on the estimated roughness. If such a step is executed, the field can be divided into two separate regions. Each region is then reconstructed separately to further improve the result.

## 5.5 Simulation

### 5.5.1 Algorithm Implementation

At the initialization, a number of sampling sites are selected based on the "coffee house" design. The probabilities of not satisfying the fidelity constraint  $P_i^1$  are set to equal values. All samples are collected in a set  $S$ .

During each iteration, the number of chosen candidates can be larger than one. This number is represented by  $N_v$  and  $N_r$  for candidates compensated by field variation model and field roughness model respectively. At the beginning, the field roughness model is very inaccurate due to lack of samples. At this phase, we rely more on the field variation model. At later iterations, the field is estimated more and more accurately and the field roughness model improves over time. So later on, more confidence is put on field roughness model. Therefore,  $N_v$  and  $N_r$  vary over the iterations to reflect this change.

At each iteration, whenever a new candidate is chosen,  $C_i$  is updated to avoid picking two candidates that are too close to each other.

The algorithm implementation is summarized below:

1. Select initial set of  $n$  sampling points.  $k=1$ . Candidate pool  $S_c = \emptyset$ .
2. Collect samples and reconstruct the field with the thin plate spline. Set  $P_i^k$ ,  $i \in S$ .
3. Determine the new set of sampling candidates  $S_c^k$ . Update the candidate pool  $S_c = S_c \cup S_c^k$ .

4. Compute  $\hat{P}_m^{k+1}$ ,  $m \in S_c$  according to Eq. (5.3).
5. Compute the compensation factor  $h_m^v$  based on the field variation model in Eq. (5.12). Let  $C_m^v = h_m^v \hat{P}_m^{k+1}$ .
  - (a) Pick the candidate  $m_v$  with largest  $C_m^v$ .  $N_v = N_v - 1$ .
  - (b)  $S_c = S_c - m_v$ .  $S = S + m_v$ .  $n = n + 1$ . Update  $C_m^v$  by treating  $m_v$  as a sampled site. If  $N_v > 0$ , go back to 5a.
6. Compute the compensation factor  $h_m^r$  based on the field variation model in Eq. (5.9). Let  $C_m^r = h_m^r \hat{P}_m^{k+1}$ .
  - (a) Pick the candidate  $m_r$  with largest  $C_m^r$ .  $N_r = N_r - 1$ .
  - (b)  $S_c = S_c - m_r$ .  $S = S + m_r$ .  $n = n + 1$ . Update  $C_m^r$  by treating  $m_r$  as a sampled site. If  $N_r > 0$  go back to 6a.
7. Collect samples and reconstruct the field.
8. Update the probability  $P_i^{k+1}$ ,  $i \in S$  according to Eq. (5.5)–(5.7).
9. If  $P_i^{k+1}$ ,  $\forall i \in S$  is below a certain threshold or if  $n$  is larger than a given number, exit. Otherwise,  $k = k + 1$ ; go back to step 3.

Note that  $h_m^v$  and  $h_m^r$  are computed only if there is a model available. If no model is available,  $C_m^v = C_m^r = \hat{P}_m^{k+1}$ . Step 5 and step 6 merges into one step.

### 5.5.2 Simulation Results

Our sampling algorithm was tested on a variety of fields. For comparison, results from uniform sampling and a modification of the adaptive algorithm given in [89] are also presented. The adaptive algorithm in [89] is an iterative stratified sampling algorithm. Our modification to it is to reduce the number of samples added in each iteration without deteriorating its effectiveness a lot so that it is comparable

with our algorithm. We call this modified algorithm the Q-method. Its basic idea is: 1) divide the field into four cells and take a sample in the center of each cell; 2) reconstruct the field and locate the cell with the biggest reconstruction error; 3) remove the sample in the located cell, further divide it into four cells and take samples in each of the new cells. Steps (2) and (3) are repeated until the fidelity constraint is met or the preallocated number of samples is reached. The way new sampling sites are added in the Q-method is illustrated in Fig. 5.6.

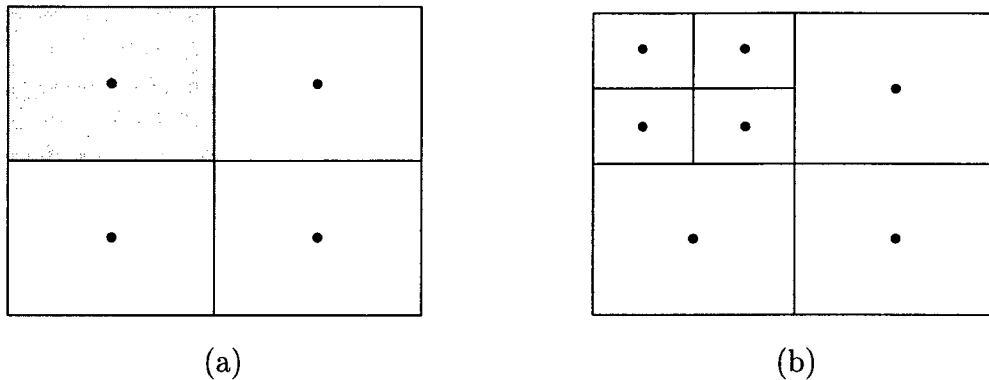


Figure 5.6: Adding new sampling sites in the Q-Method (a) Original cells. The dots represents sampling sites and the shaded area represents the cell with the largest estimation error. (b) Cells and sampling sites after new sites are added.

Our algorithm was first tested on a smooth field. The purpose of this test is to complete the proof that our algorithm works on both homogeneous and heterogeneous fields. Our algorithm samples such a field with almost uniform density as we have expected.

The next test was done on a real incident sunlight field shown in Fig. 5.7, which is inhomogeneous and varies quickly in some regions. The reconstruction results shown in Fig. 5.8 demonstrate that our algorithm largely outperforms the uniformly sampling and is also better than the other adaptive algorithm. The plots of the MSE and bending energy vs. the number of samples in Fig. 5.9 and Fig. 5.10 further prove that our algorithm converges to the truth field faster than the



other two algorithms. To better understand our algorithm, the sampling patterns of the three algorithms are plotted in Fig. 5.11. From this figure, we can see that our algorithm closely follows the heterogeneity of the field and samples are more densely deployed in the fast varying region. The other adaptive algorithm also puts more samples in the high spatial frequency region. But it tends to overlook the regions with low spatial frequency.

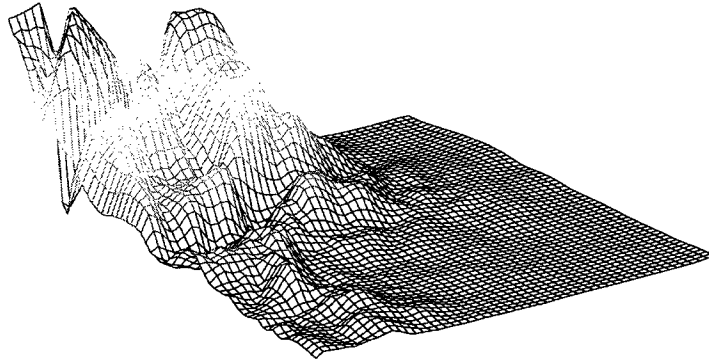
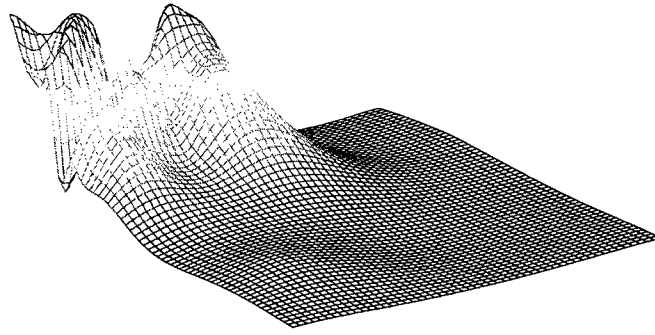


Figure 5.7: Ground truth of the light field

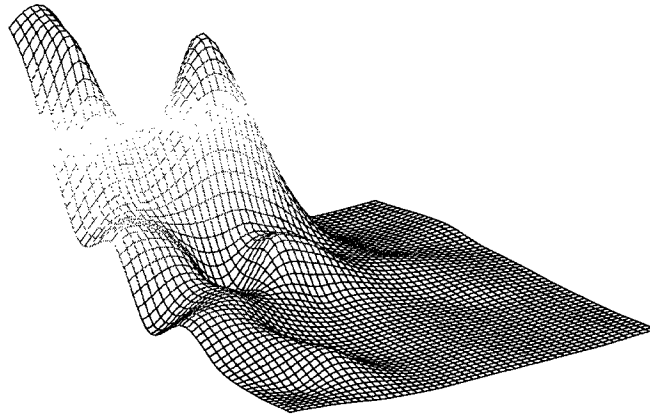
The most challenging test is reconstructing a field with discontinuities. We performed this test on a simulated field, which is composed of two flat regions with a big jump between the two regions, as shown in Fig. 5.12. The modification mentioned at the end of section 5.4 helps locate the discontinuities and avoid clustering. Fig. 5.13 presents the reconstruction results. Since our algorithm follows the roughness of the field closely, it also outperforms the Q-Method and the uniform sampling in such a field.

## 5.6 Conclusion

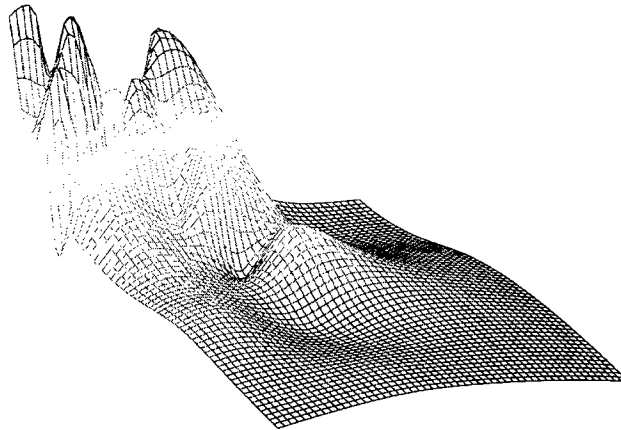
In this chapter, a newly designed adaptive sampling algorithm for reconstructing a static (or time-average of a) field is elaborated. This algorithm iteratively selects new sampling points where a predefined fidelity constraint is most probably not satisfied



(a) Adaptive sampling algorithm. Number of samples=88. MSE=0.0717



(b) Uniform sampling. Number of samples=89. MSE=0.11



(c) Q-Method. Number of samples=87. MSE=0.0881

Figure 5.8: Reconstruction result for real sunlight field.

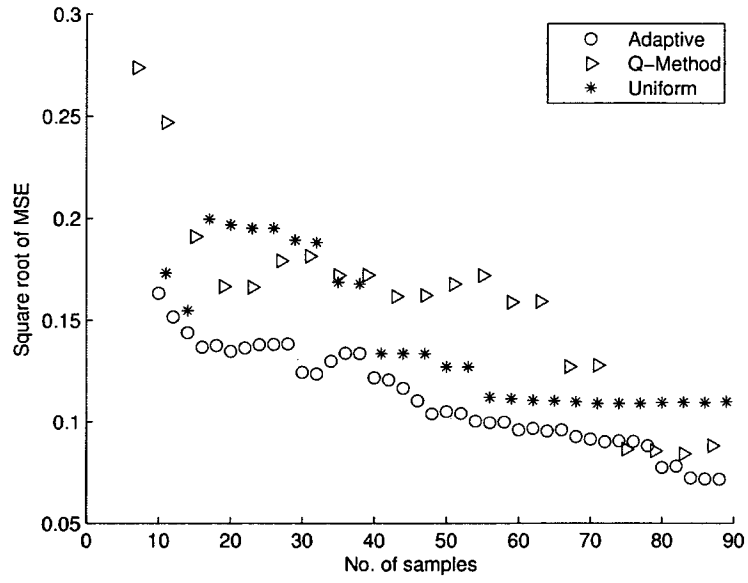


Figure 5.9: The square root of MSE in successive steps.

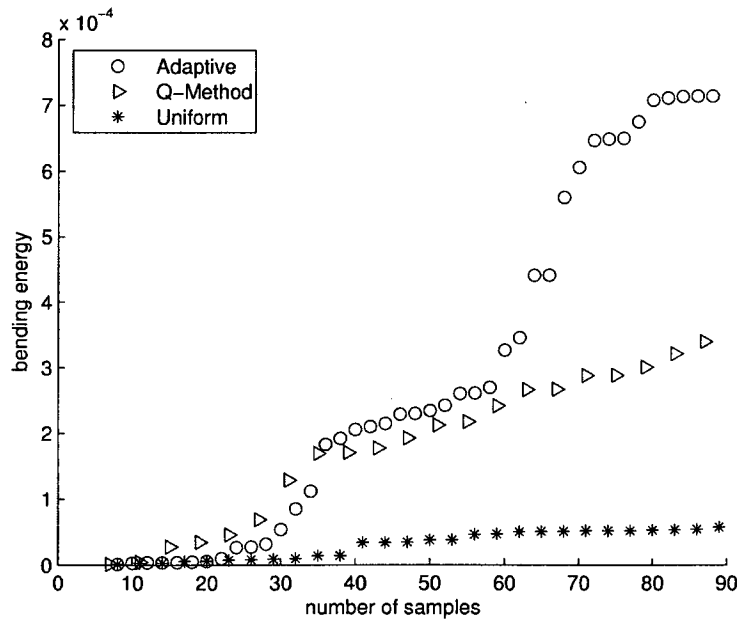
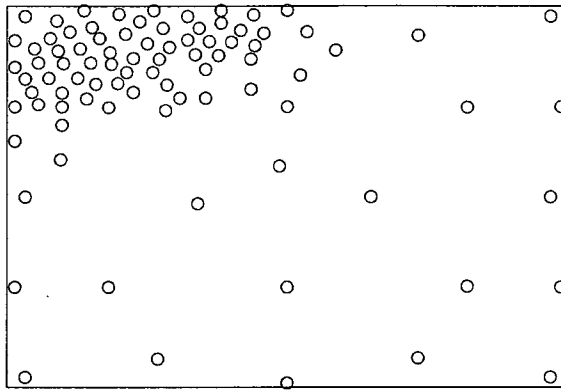
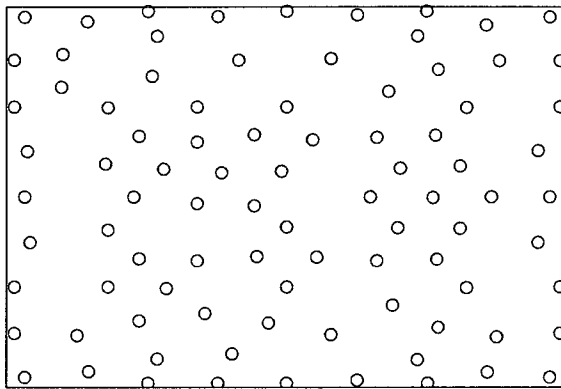


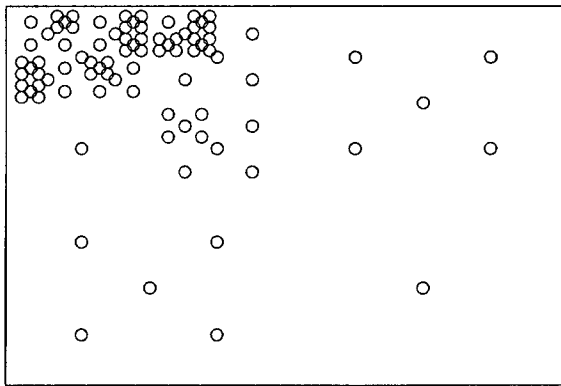
Figure 5.10: The field bending energy in successive steps.



(a) Adaptive sampling algorithm



(b) Uniform sampling



(c) Q-method

Figure 5.11: The pattern of the sampling sites for the light field

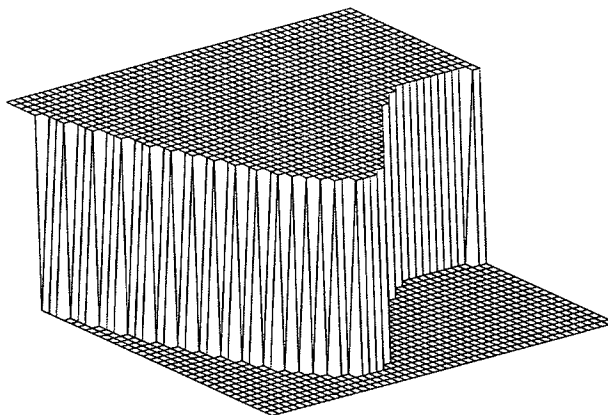
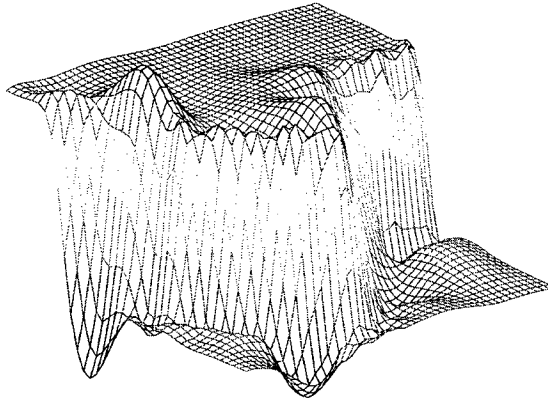


Figure 5.12: Ground truth of the piecewise constant field

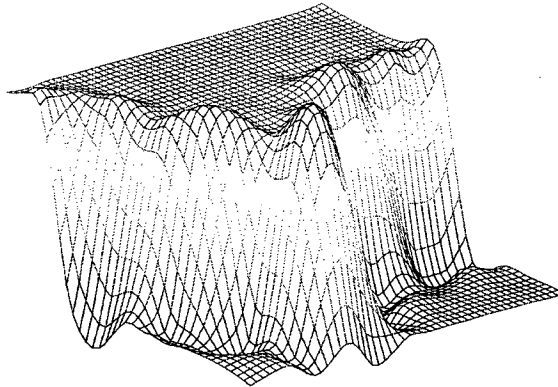
based on previous reconstructions. Simulation results prove the effectiveness of this algorithm.

In designing the algorithm, a mobile sensor is assumed. The employment of a mobile sensor removes the constraints on the possible sampling sites. Hence, the algorithm can always choose the sites that maximize the utilization function. Since no cost associated with the mobile sensor is considered, routing schemes for moving the mobile sensor are not discussed here.

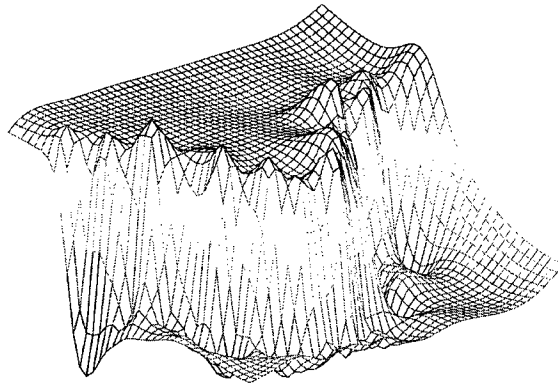
In the next chapter, another sampling algorithm, the multiscale sensing, is described. In that algorithm, the sampling density is also adapted to the field variation. The difference from the adaptive sampling algorithm is that it gets this information from a different type of sensor. The algorithm's efficiency is greatly increased because of the second sensor type, which also provides field models at the sensing time without relying on prior information. However, its dependence on the second sensor type also limits its application to the problems where multiple types of sensors are able to measure the field. This limitation does not exist in the adaptive sampling algorithm.



(a) Adaptive sampling algorithm. Number of samples=104. MSE=0.071



(b) Uniform sampling. Number of samples=104. MSE=0.0897



(c) Q-Method. Number of samples=103. MSE=0.078

Figure 5.13: Reconstruction result for the piecewise constant field.

## CHAPTER 6

# Multiscale Sensing

### 6.1 Introduction

With the advancement of sensor technology, sensors for observing all kinds of phenomena have been developed. For a certain phenomenon, there may exist several different types of sensors. These sensors possess different capabilities and their costs may vary over a wide range. Hence, they are usually deployed in different situations. As more and more sensors are deployed in the field, fusing information from sensors becomes increasingly important. Multisensor data fusion thus becomes a rapidly evolving research area.

Multisensor fusion refers to the combination of data from multiple sensors to provide more reliable and accurate information. Sensor data fusion can happen at different levels[60]:

- signal-level fusion: fusion at this level is popular in real-time applications where signals are directly combined.
- Pixel-level fusion: this is similar to the signal-level fusion. It is regularly used in image processing to improve the performance of image processing techniques, such as image enhancement, image segmentation, etc.
- Feature-level fusion: features are extracted and form a feature space from each set of data. Feature spaces from different sets of data are then combined into one larger feature space.

- Symbol-level fusion: data are abstracted to the symbol level. Symbols are then combined.

Sometimes fusion happens at a single level, like the fusion in [15] where features (segments and landmarks) are extracted from the data and combined to cross-validate the data and enhance map accuracy. More often, data are fused at several different levels. For example, in [112], features (activity levels) are first fused to form a new feature space. Based on this new feature space, data are combined to finish the fusion process.

In recent years, more and more studies confirm that fusing redundant and complementary data from multiple sensors provides advantages over single sensor data processing. Various algorithms have been developed to fuse the data. These algorithms can be broadly classified into four categories[60], which are loosely correlated with the four fusion levels:

- Estimation methods

These methods are suitable for real-time processing of low-level data and are successfully applied in signal-level data fusion. Popular algorithms include weighted average, least squares and Kalman filtering. Among these algorithms, Kalman filtering is the most preferred one because it reaches optimal performance in a statistical sense while maintaining the same level of computational complexity as other algorithms. Data from multiple sensors can be either grouped and updated together or updated sequentially[32]. Because of the linear nature of the Kalman filter, sometimes the extended Kalman filter has to be resorted to for handling nonlinear models.

- Classification methods

A feature space extracted from data can be partitioned into distinct regions, each representing an identity class. Many classification algorithms exist, including parametric template classification, cluster analysis, learning vector



quantization, Kohonen feature map, etc. When the final objective is beyond classification, data are assigned to one of the regions based on similarity criteria before they are further processed. Classification methods are commonly used at the pixel and feature level data fusion, such as the data fusion in remote sensing[98].

- Inference methods

A higher level of information than features can be extracted from the data. Generally this information is expressed as the probability distribution. Such extraction allows information to be combined according to the rules of the probability theory. Fusion methods at this level include Bayesian inference, Dempster-Shafer method and generalized evidence processing. Among these methods, Bayesian inference is the basis while the other two methods are the extensions of the Bayesian method.

The Bayesian method combines likelihood ratios obtained from the data with prior information on the parameters to make decisions. Depending on the confidence we have on prior information, we can set the prior distribution to be concentrated if we want to put more emphasis on the prior information, or flat if we have little confidence. The more observations that are available, the more that decisions are dependent on them. Therefore the Bayesian method is essentially objective.

In sensor networks, the Bayesian method is prevalently applied in detection and tracking problems, such as in [24][113]. Since the Bayesian update procedure can be interpreted as[62]

$$\textit{posterior information} = \textit{prior information} + \textit{observation information}$$

the Bayesian method is very intuitive from an information-theoretic point of view. This observation leads to the application of information utility in the data fusion process[113].

- Artificial intelligence methods

High-level inference requires high level intelligence and a learning process. It may also involve human reasoning. It is an advanced version of the other three methods. Expert systems, neural networks and fuzzy logic are popular forms of artificial intelligence.

A fifth fusion method that is related but does not strictly belong to the four categories is a cuing type of sensory processing. In this situation, one sensor guide the operation of other sensors. It is typically used for fusing information from different types of sensor. Usually robotic sensors are involved so that they can react to the guidance, as done in [96].

Multiscale sensing refers to the process when sensory data are obtained and processed at different levels. There are two possible scenarios: 1) data come from only one type of sensors; 2) data come from at least two different types of sensors. We denominate the first scenario as multi-resolution analysis and the second scenario as multi-modal analysis. These two scenarios often convolve together. Data of different resolution usually comes from different types of sensors, while different types of sensors can hardly provide data at exactly the same resolution. Hence, as long as the data from different sensors are of the same nature and can be directly fused at the signal or pixel level, we will term such fusion as multi-resolution analysis. If higher level data fusion is involved, the process will be named multi-modal analysis.

Wavelet decomposition is a very popular method for multi-resolution analysis. Because it is local in both the frequency domain and the time domain, it has been successfully applied in many signal and image processing problems and attracted a lot of attention. Kolaczyk et al. even applied the concept in likelihood analysis[54].

Similar to the Wavelet method, Chou et al.[18] also generated a pyramid-like representation of the signals from sensors. The signal processing in their approach involves a fine-to-coarse filtering sweep followed by a coarse-to-fine smoothing sweep.

The fine-to-coarse sweep corresponds to a generalization of the Kalman filter to multiscale models on trees. It consists of a three-step recursion of a measurement update, two parallel fine-to-coarse predictions and fusion of these predicted estimates. The coarse-to-fine smoothing sweep has the same form as a standard Rauch-Tung-Striebel (RTS) algorithm, which involves a forward Kalman filtering sweep followed by a backward sweep to compute the smooth estimate. This multiscale framework not only achieves large computational savings, especially in 2-D data processing, but also allows the fusion of multi-resolution data. The application of this algorithm in [34] exhibits a substantial reduction in computational complexity in estimating ocean surface data from a satellite altimeter.

The advantage of fusing data at multiresolution enabled by this framework is presented in [29]. In that study, the multiscale framework is applied to airborne microwave radiometer data to estimate the snow-water-equivalent distribution. Processing the data in the multiscale framework makes it compatible with in situ measurement for evaluation and variation determination. Spaceborne Special Sensor Microwave/Imager data is also processed through the multiscale framework and is used to aid the interpretation of the microwave data.

The multiscale framework can also be applied in multi-modal analysis, as long as data of different natures are transformed into a common set of parameters, as done in [97], where interferometric radar and laser altimeter data are fused to estimate surface topology and vegetation height. Because of the underlying Kalman formulation in the multiscale framework, it can accommodate irregularly spaced and sparse observations. This feature is also exploited in [97]. But the linearity of the Kalman filter also brings a limitation to the framework, which necessitate the decoupling of nonlinearity of the model from the multiscale estimation, an extra step in the data processing.

Instead of partitioning data into different scales, the multiscale concept can also be applied to the formulation of data processing. Clark[20], in his Hierarchical

Bayesian (HB) structure, represented the processing in multiple levels: data, process, parameter and possibly latent variant and/or hyperparameters. The level partition is realized through the decomposition of the probability density function. It is similar to the work in [54] in this sense.

The usefulness of multiscale sensing prompt us to apply it to our sunlight field estimation problem. Due to the lack of an existing field model suitable for the Kalman filter, it is difficult to apply the multiscale framework reviewed in this section directly. Instead, we developed a new scheme for solving our problem, which is explained in the next few sections.

## 6.2 Multiscale Formulation of Sunlight Field Estimation

During photosynthesis, plants use energy in the region of the electromagnetic spectrum from 400-700 nm. The radiation in this range is referred to as Photosynthetically Active Radiation (PAR). Biologists and environmental scientists usually use special PAR sensors to measure this radiation. The spectral response of a popular PAR sensor LI-COR LI-90 is shown in Fig. 6.1. Since it is specially designed for measuring PAR, PAR sensors are highly sensitive to this radiation and provide accurate measurements under natural light environments.

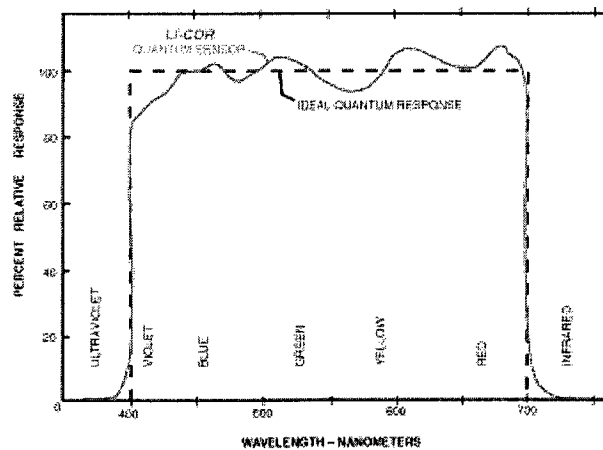


Figure 6.1: The spectral response of LI-90 PAR sensor

The spectrum of PAR falls in the range of the observable light. Hence, it can also be measured with a second type of light sensor: a camera. Unlike the PAR sensors which measure the light intensity one point at a time, the camera can measure the light intensity over many points simultaneously. This capability is an important complement to the PAR sensors in sampling and reconstructing a field with high spatial and temporal frequency. With PAR sensors alone, the reconstruction of such a field with high fidelity requires a large amount of PAR sensors, which is prohibitively expensive. But shortcomings of the camera also limit its usage as a light sensor by itself. The camera is not specially designed for measuring PAR. Thus its response is not flat within the range of the PAR's spectrum, as illustrated by a typical camera's spectral response in Fig. 6.2. Besides, the camera's typical characteristic curve shown in Fig. 6.5 exhibits linear response only over a very limited intensity range. While intensity can vary ten-folds between sunflecks and umbras, both may end up in the nonlinear response range of the camera. Most importantly, if the camera is placed at the ground level and faces up to the canopy, it behaves just like a PAR sensor with less accuracy. To explore its capability of measuring the light intensity at ground level over a large area, the camera needs to face down from above. In this way, it measures the reflected light intensity instead of incident light intensity. The ground reflectivity would strongly affect the measurement results, rendering the incident light measurement completely unreliable.

Although either type of sensor, the PAR sensor or the camera, can hardly fulfil the requirement of reconstructing the incident light field with high fidelity by itself, the combination of the two may achieve the goal, since their data are good complements to each other. The major challenge is finding an effective data fusion approach to combine the two data sets.

Even though both sensors measure the light intensity, their data are of different natures. The camera measures the reflected light while the PAR sensor measures the incident light. This rules out the possibility of fusing data at the signal or pixel

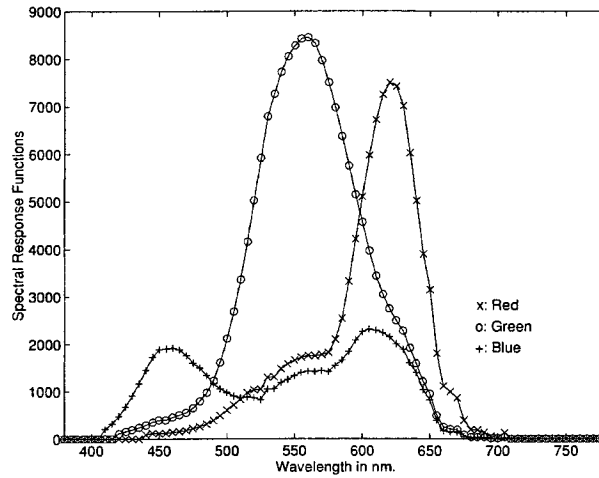


Figure 6.2: Typical camera spectral response

level. A higher level of data fusion is necessary.

The camera provides measurements over a large area with high resolution. These measurements give an overview of the field and are useful in obtaining global information. From these measurements, features such as field partition and field models can be obtained. Hence, feature level data fusion is a proper choice. Based on these features, various sampling schemes suitable for each segment of the field, such as uniform sampling or random sampling, can be applied to get local information through the PAR sensors. With the assistance of the field models extracted from the global information, the whole field can be reconstructed from the PAR data.

The remainder of this chapter is organized as follows. The processing of the camera data involves a lot of image processing techniques. These techniques are discussed in the next section. In section 6.4, model fitting and selection process is explained in detail. The following section elaborates on the sampling and reconstruction process adopted in our multiscale sensing scheme. The experimental results in this section demonstrate the effectiveness of the whole scheme.

## 6.3 Image Processing

The camera data fulfill two tasks: field partition and field model extraction. Through field partition, ground reflectivity can be classified and modelled. As the types of ground reflectivity in an area are very limited, a model library is setup to collect all the models. After enough training, most of the ground reflectivity types in the neighborhood of the studied area would be covered. The biases in the camera data resulting from ground reflectivity can then be removed so that the camera data can be used in estimating the incident light models.

The field partition task is implemented through image segmentation techniques. The objective of general image segmentation is to partition an image into subregions that are homogeneous with respect to color and texture variation. These subregions are considered as objects, each following a type of intensity distribution, which are known as features of the image. The intensity distribution of the whole image is a mixture of the distributions of these objects.

### 6.3.1 Existing Segmentation Methods

Due to its complexity, image segmentation is still an active research area. There exist many segmentation techniques. In general, image segmentation can be implemented either as a bottom-up procedure whose result depends only on intrinsic image intensity, or a top-down procedure guided by the knowledge on the particular problem. Most techniques reviewed here are bottom-up procedures since we do not assume any prior knowledge on ground reflectivity.

The simplest and often adopted approach for image segmentation is setting a threshold in the histogram. In this sense, each object is assumed to correspond to a mode in the histogram. These modes are the cluster centers in the feature space. Hence, a popular clustering scheme, k-means clustering, is often applied in the image segmentation problem[35]. The basic solution for the k-means problem is

an iterative algorithm that finds a local minimum of a penalty function. There are a number of variants to this algorithm, such as gradient descent or ascent methods and mean-shift methods. Cheng[17] generalized the mean-shift methods and proved that it is more effective than other methods in terms of adapting to the right step size. To reduce the computational complexity, Kanungo et al.[48] presented an efficient algorithm which was shown to be much faster than other algorithms.

The simplicity of the k-means scheme comes at the price of low accuracy. With randomly chosen initial centers, the algorithm generates arbitrarily bad clustering results with high possibility. Arthur et al.[5] augmented the k-means algorithm with a simple seeding technique and effectively reduced the probability of obtaining bad clusters. Another problem associated with the k-means scheme in its application in image segmentation is that it requires prior knowledge on the number of segments, which is often unavailable. Comaniciu et al.[22] tried to solve this problem by combining the analysis in the feature space and in the image domain. They applied the mean-shift algorithm in the feature space analysis. Connectivity of pixels in each cluster were then checked in the image domain to remove unnecessary features and to separate objects. Similarly, Kam et al.[47] also obtained feature clusters through the mean-shift algorithm. Instead of assigning a feature label to a pixel by only looking at its distance to the feature center, they used MAP estimation based on all the neighbors of the pixel to select a proper feature label. The added complexity enables this algorithm's application in texture segmentation.

Instead of obtaining features from the intensity domain, some authors tried to extract features in the frequency domain. Liu and Pok[59] transformed the intensity image into frequency domain through a Gabor filter. They predicted the feature of a pixel from its neighbors and got a map of prediction variance. The edges in the prediction variance map were used as the boundary of the features. Segmentation was then done based on the pixels' feature values.

Another thread of segmentation techniques segments images by locating edges



between objects. The edge map is obtained from first-order differentiation maxima or second-order differentiation zero-crossings. Various filters for obtaining the differentiation have been designed[41]. In practice, the edge map is often noisy with many broken edge lines. Therefore, edge tracking and linking is frequently required to obtain closed boundaries of objects. Among the edge tracking methods, the "snake" method is a very popular one[49]. Schneider et al.[93] also segmented images by finding the object boundaries. But their edge map did not come from direct filtering. Instead, they defined energy functions for both the image and the edge strength. By minimizing both energy functions, they accurately located the object boundaries.

The method in [93] is more region based than boundary based. Region based methods look at much larger neighborhoods than boundary based methods in doing hypothesis testing. Although the former's complexity is higher, their results are usually better too. Thus they have attracted a lot of attention. These methods grow a region around a seed pixel inside an object until the whole object is filled up[1], or perform global optimization based on energy functions[16][75]. A common drawback of these methods is that they cannot distinguish between two distributions with the same mean but different variances. Zhu and Jullie [114] attacked this problem by applying MDL criterion in the optimization step. They showed that region growing algorithms can be directly derived from their method.

Recently, there has been a considerable amount of work using the curve evolution technique. It is also a region based method. It is inspired by the minimum surface problem in physics. A closed curve can be either explicitly represented as a one-dimensional function  $y = f(x)$  or implicitly represented by the zero-level set of a two-dimensional surface  $z = f(x, y)$ . When the surface is diffused over time due to forces on it, its zero-level set also evolves. Correspondingly, the region enclosed in the curve grows or shrinks depending on the forces on its boundary. This is similar to the region competition concept in [114]. A smooth surface such as a signed distance

function is often adopted in the curve evolution. There are three basic types of forces: a force on the normal direction to the curve  $V_N$ , an external vector field  $S$  and a force based on the curvature of the curve  $\alpha$ . The speed of the curve evolution is determined by all three forces as:

$$\frac{\partial f}{\partial t} + \underbrace{\vec{S} \cdot \nabla f}_{\text{Vector field based}} + \underbrace{V_N |\nabla f|}_{\text{In normal direction}} = \underbrace{\alpha |\nabla f|}_{\text{Curvature based}} \quad (6.1)$$

The basic usage of the curve evolution method in image segmentation is defining an edge function, pulling the curve towards the edges by a vector field force, and stopping the curve around the edges. The forces are often defined as a function of the probability density of the intensity values inside or outside a region, as done in [3]. Fisher et al.[40] argued that minimizing the mutual information between the features and the classifiers lowers the lower bound of the classification error. Inspired by this idea, Kim et al.[52] proposed to use the mutual information between the pixel intensity value and its segmentation label as the force in the normal direction. They proved that minimizing this force resulted in correct segmentation. The use of nonparametric density estimation in their work also broadened the horizon of the covered density functions than that in [114].

### 6.3.2 Image Segmentation Procedures

Although field partition from camera data can essentially be fulfilled by image segmentation techniques, our objective in segmentation is slightly different from typical image segmentation tasks. Separating objects in the image is still important because different objects correspond to different reflectivities. But our primary interest in field partition is locating the three type of fields: sunflecks, umbras and penumbras. This information is not available from normal segmentation results since they tend to focus on objects and ignore the slight lighting difference on different regions of an object. Therefore, existing segmentation techniques cannot satisfy our needs. Instead of designing a brand new segmentation technique, we obtain the necessary

information through controlling the camera and combining several different segmentation techniques.

### 6.3.2.1 Image Series

One important difference in our procedure from ordinary image segmentation tasks is that we have some degree of control over the source data. This added flexibility facilitates us in obtaining desired results in image segmentation.

As shown in Fig. 6.5, the camera saturates at both high and low intensity. Although this phenomenon is not welcomed in general as it adds nonlinearity to the camera data, it can actually be exploited in controlling the image data. In umbras, except when the reflection of an object is very strong, pixel intensities are low and often in the saturated region in the camera response curve. Due to the compressed value range, the variance among these pixel values is reduced. The deeper the saturation, the smaller the variance. As a result, these pixels have a very different density distribution from the pixels in the linear intensity range. Since the curve evolution segmentation technique in [52] is very good at partitioning pixels with different variance, it can easily separate out the umbras. To enhance the result and reduce classification error, we make the image darker than usual by reducing the camera aperture and shutter speed. Such a maneuver pushes the intensity of pixels in umbras and penumbras into the saturation region. It also pulls the intensity of the pixels in sunflecks into the linear region by reducing their intensity value. Thus sunflecks are distinguished from a dark image. Pixels with low intensity values are in the same situation when we increase the brightness of the image by increasing the camera aperture and shutter speed.

Another advantage of controlling pixel intensity is that we get more pixels in the linear range. For example, intensity of the pixels in a sunfleck may be in the saturated region in a normal image. But in a dark image, it would be in the linear region. Hence, with three different images for the same scene - one dark,

one normal and one bright - almost all pixels are in the linear range in at least one image. Although a calibration procedure can reduce the nonlinearity problem of the camera (explained in section 6.3.3), its effectiveness is limited and renders this intensity control necessary for image data to be used in model extraction.

One problem with the curve evolution segmentation technique is that it sometimes puts very bright and very dark areas into one class since both have small variance. Such a scenario occurs when an area is too bright or too dark such that even the intensity control cannot shift its intensity into the linear region. But because of their big intensity difference, the two classes of pixels can be very easily separated. The k-means algorithm is chosen to solve this problem.

Objects are further separated by segmenting the normal image. At this time, segmentation in the three types of subareas, the umbras, the sunflecks and the penumbras, is performed separately. The reduced intensity range enables us to assume a different set of parameters from those used in segmenting the bright and dark images. Strong reflectivity sometimes causes misclassification of a subarea. This problem is alleviated by the normal image segmentation because subareas with saturation intensity in the other two images are more separable in a normal image.

Because the curve evolution segmentation technique only works on intensity images, we apply it to the three color components of each image to get better object separation.

In summary, a series of three images are taken at each training/reconstruction step with different apertures and shutter speeds. In the field partition step, the image taken with a small aperture and a low shutter speed, i.e. the dark image, is used to locate the sunflecks. The image taken with a large aperture and a high shutter speed, i.e. the bright image, is used to locate the umbras. The normal image, which is taken with best balance between the high and low intensity values, is used to identify objects. All three images will be used in model extraction step too. Fig. 6.3 illustrates a series of images and their respective segmentation results.

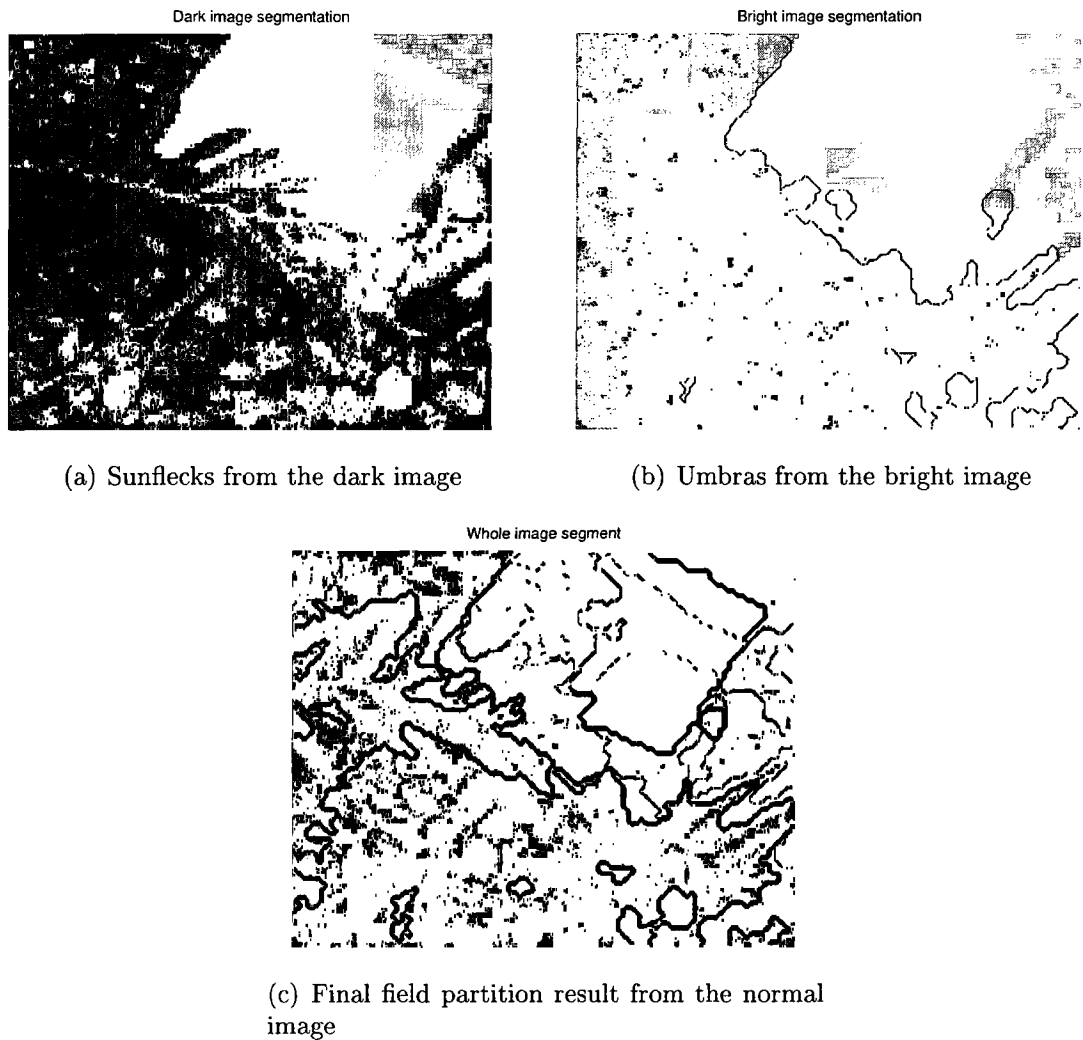


Figure 6.3: Image series and its corresponding partition

### 6.3.3 Image Calibration

Although saturation in the camera intensity response is useful in field partition, it is harmful to field model extraction. Since the saturation effect increases the spatial correlation between pixels, it not only leads to erroneous sampling density

estimation, but also deteriorates the reconstruction result. Hence, it is desirable to remove this saturation effect as much as possible before the image data is used for model extraction. This task can be completed through image calibration.

A simple calibration tool used in this research is a Kodak gray scale, as shown in Fig. 6.4. The gray scale is a quality-control device of stepped, neutral values. It

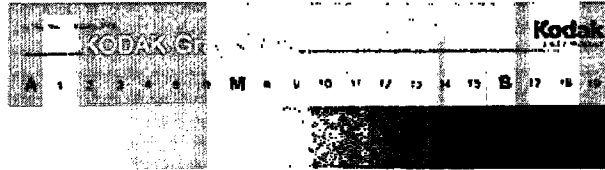


Figure 6.4: Kodak gray scale

helps photographers to compare the tone values of reflection originals with the tone values of their reproduced images. It is composed of 20 density steps. The relative density values range from a nominal "white" of approximately 0.05 to 1.95. The density increment of each step is 0.1. These increments are tightly controlled and vary only slightly from the nominal density value. The relationship between the density value and the reflectance, which is proportional to the reflective intensity under same incident light conditions, is

$$Density = -\log_{10} Reflectance \quad (6.2)$$

A typical density/reflectance vs. intensity curve is given in Fig. 6.5.

One calibration image is captured for each aperture and shutter speed used in taking image data. From the calibration image, the intensity value of an array of pixels of each density is read out and the median is taken as the representative value. A third order polynomial fits well to the intensity-density curve, as can be seen in Fig. 6.5(a). When the image using the corresponding aperture and shutter speed is calibrated, pixel reflectance is computed from its intensity value using the fitted polynomial and Eq. (6.2). From Fig. 6.6, we can clearly see that the pixel values in the calibrated image stretch out in the saturated regions.

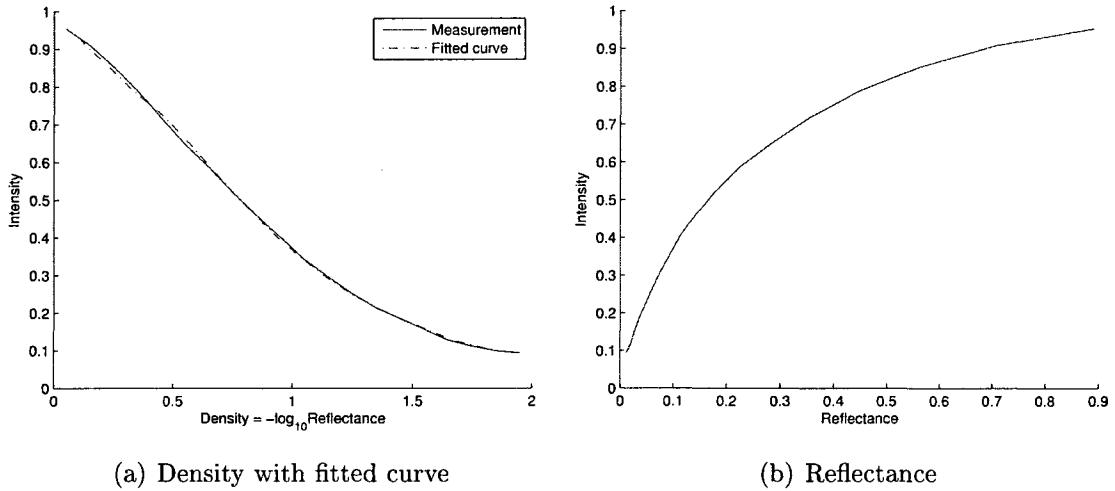


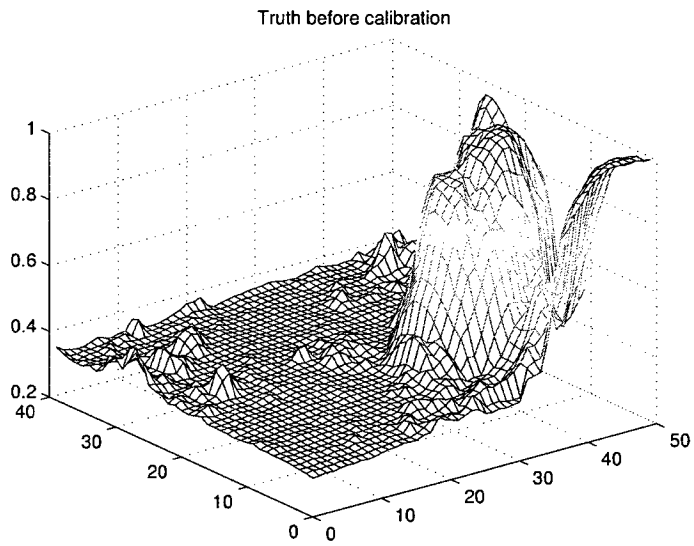
Figure 6.5: Camera characteristic curve

## 6.4 Model Fitting and Selection

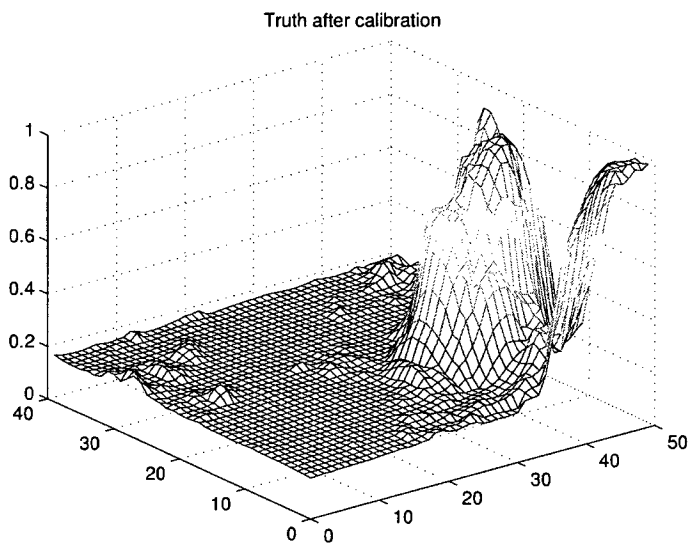
A model is a description of an object. Described from different aspects, an object can have many different models. For example, a function describing a surface is often referred to as a model in literatures. The spline fitting for the field given in Chapter 5 is in line with such a model. In reconstructing the field through the multiscale framework explained in this chapter, we also need to find the fitted surface for the field. In the context of this dissertation, we will use the term "field" to refer to the surface. When we talk about models, we refer to the statistical models, including probability density functions and correlation functions. The probability density functions will be used to classify the ground reflectivity, while the correlation functions will be applied in determining the sampling density.

### 6.4.1 Probability Density Function Estimation

Probability density function estimation is the process of fitting a curve to the (*value, probability*) pair. In this sense, it is similar to the function regression explained in



(a) Intensity before calibration



(b) Reflectance after calibration

Figure 6.6: Effect of calibration



Chapter 4. Therefore, some of the concepts in the function regression are also applicable here. But there are two major differences between them: 1) there is no noise in density fitting. Hence, when a nonparametric fitting method is used, interpolation instead of approximation is more applicable. Basic density functions instead of polynomials or spline functions typically serve as kernels. Among the kernels, the Gaussian kernel is the most popular one because of many nice properties associated with the Gaussian distribution. 2) In parametric estimation, the underlying curve belongs to a known family of probability distributions. Statistical measures such as mean, mode, etc are estimated for the fitted distribution.

#### 6.4.1.1 Nonparametric Method

Given a sequence of i.i.d random variables  $X_1, \dots, X_N$ , to estimate their probability density function, we can start from estimating the cumulative distribution function (CDF), which completely describes the probability distribution of a random variable. The most straight forward estimation of the CDF is

$$F_N(x) = 1/N \{ \# \text{ of observations } \leq x \text{ among } X_1, \dots, X_N \} \quad (6.3)$$

The density function, which is the first derivative of the CDF, can be estimated as the finite difference of the CDF:

$$\begin{aligned} f_N(x) &= F_N(x+h) - F_N(x-h)/2h \\ &= \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x-X_i}{h}\right) \end{aligned} \quad (6.4)$$

Eq. (6.4) naturally leads to the nonparametric density estimation with the kernel being

$$\begin{aligned} K(x) &= \frac{1}{2}, \quad |x| \leq 1 \\ &= 0, \quad |x| > 1 \end{aligned} \quad (6.5)$$

Parzen[83] proved that with a bounded kernel satisfying

$$\lim_{x \rightarrow \infty} |xK(x)| = 0$$

the nonparametric density estimation in Eq. (6.4) is a consistent estimation of the true density function.

The density estimation with the kernel in Eq. (6.5), which is a rectangular function, provides the same result as a histogram does. Histogram estimation is another popular nonparametric method. Histograms are usually very bumpy. Hence, smoother kernels are often resorted to for density estimation. Parzen[83] gave a list of kernels that satisfy the required condition. In univariate density estimation, the Gaussian kernel is a frequent choice. Multivariate density estimation usually calls for more complex kernels. Although there have been many efforts in designing good kernels, it is now widely recognized that the estimation quality is primarily determined by the kernel bandwidth and much less by the choice of the kernel [94]. Fig. 6.7 illustrates the effect of an improper bandwidth. When the bandwidth is too large, it smears the structures in the density function; when it is too small, it reveals fake modes in the density.

Jones et al.[44] surveyed the bandwidth selection methods for density estimation. They divided the existing bandwidth selection methods into "first generation", and "second generation". Most "first generation methods" were developed before 1990, while "second generation methods" were developed more recently. The principal distinction between the two generations is in the rate of convergence. Most first generation methods are surveyed by Marron[64]. In [44], the best known "first generation methods", including the rule of thumb, least square cross-validation and biased crossed validation method, and some "second generation" methods, including plug-in approach and smoothed bootstrap, are compared. The authors concluded that the "second generation" methods outperform the first generation ones. They also

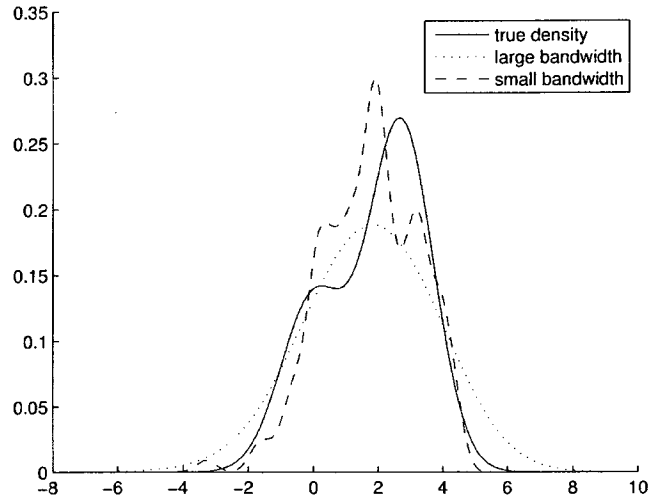


Figure 6.7: Effect of bandwidth of density estimation

showed through simulation that the rule of thumb and biased cross validation methods tend to oversmooth the density function while the least square cross-validation method undersmooths the density function. In general, the smoothed bootstrap method has the best performance.

When the number of variables increases, a big problem that plagues density estimation is the so-called "curse of dimension". It describes the phenomenon that almost all neighborhoods of given samples are empty in high dimension such that the samples are not sufficient for meaningful density estimation. Clustering the samples usually can effectively solve the problem. When the data are clustered, local estimates are made near the cluster centers. Comaniciu and Meer[23] modified the mean shift algorithm given in [17] for multivariate data cluster and density estimation and showed their approach is more robust to noise.

#### 6.4.1.2 Parametric Method

Parametric methods assume the data are random realizations of a probability distribution from a known family. The parameters for the distribution are estimated

from the data. Knowledge of the true family of distributions is important for good estimation. But this knowledge is often absent. To avoid relying too much on this knowledge, Gaussian mixture density estimation is often used.

A probability mixture model is a probability distribution which is a convex combination of other probability distributions. Suppose a random variable  $X$  is a mixture of  $k$  component random variables  $Y_i, i = 1, \dots, k$ , each having a distribution  $f_{Y_i}$ . This means that with probability  $\lambda_i$ ,  $X$  follows the distribution  $f_{Y_i}$ . Then the distribution of  $X$  is a weighted sum of its component distributions

$$f_X(x) = \sum_{i=1}^k \lambda_i f_{Y_i}(x) \quad (6.6)$$

Although  $f_{Y_i}(x)$  can be any distribution, without any knowledge on what these component distributions are, the Gaussian distribution is a popular choice. The widespread use of Gaussian mixture density lies in the fact that a univariate Gaussian distribution has a simple representation with only two parameters. It assumes the least prior knowledge in terms of entropy of the distribution in estimating a density function.

The Gaussian mixture model can be thought as the interim between the non-parametric model and the simple parametric model. If there is only one component in the mixture, it reduces to a simple parametric model. If each data is taken as a component in the mixture, with the mean and variance of the component being the data value and a given kernel bandwidth respectively, the mixture model in Eq. (6.6) is equivalent to the nonparametric model in Eq. (6.4).

The maximum-likelihood method has been most widely adopted for estimating the parameters of a Gaussian mixture model. An expectation-maximization(EM) algorithm originated from [28] is often used for finding the maximum likelihood estimation. The EM algorithm alternates between an expectation(E) step, which computes an expectation of the likelihood, and a maximization(M) step, which computes the maximum likelihood estimates of the parameters by maximizing the

expected likelihood obtained from the E step. The EM algorithm is prevalent because the EM iteration does not decrease the likelihood function of the observed data. But it also encounters the following problems:

- The number of components in the mixture distribution has to be known a priori.
- The solution for maximizing the log-likelihood function is sensitive to the initial value used in the numerical procedure.
- The estimated parameters often come from the local maxima of the log-likelihood function instead of the global maxima.
- Existence of outliers will lead to erroneous estimates.

To solve the problems, Popat et al.[85] proposed to cluster the data before estimating the model parameters. This is essentially the same as the dimension reduction approach in the nonparametric method. Their clustering algorithm does not require prior knowledge on the number of clusters. Besides, parameters can be easily estimated from each cluster since the number of modes within each cluster is very small. The parameters are then optimized using the EM algorithm. Because the original estimates from clusters are already pretty good, the optimized parameters after the EM algorithm tend to reach the global maxima of the likelihood function. Another advantage of clustering is that it reduces the effect of outliers since they are treated as coming from a separate component in the model.

Instead of clustering the data, Zhuang et al.[115] modelled one component explicitly and modelled all other components as the uncertainty such that the density function is expressed as

$$f(x) = (1 - \epsilon)G(x) + \epsilon H(x) \tag{6.7}$$

where  $G(x)$  is the Gaussian distribution and  $H(x)$  is the density of all other components. They denoted such a model "the contaminated Gaussian density". They

identified the data belonging to the Gaussian model through the criterion

$$G = \{x^i : G(x^i) > \frac{\epsilon H(x^i)}{1 - \epsilon}\} \quad (6.8)$$

There exist an infinite number of models that satisfy the classification criterion, among which  $H(x^i) = \delta$  assumes the least configurational information and can be used to estimate the parameters for the Gaussian component. The Gaussian component is then subtracted from the mixture model and the previous procedures are repeated until all components are identified. This iterative algorithm is independent of initial conditions and robust to outliers.

#### 6.4.2 Model Selection

Models obtained through different estimation methods are different. Each model has its individual strength and weakness. Under a certain circumstance, some models will be closer to the truth than the other models are. But under another circumstance, it may be farther away from the truth. Therefore, it is important to select the proper model for one's specific objective and circumstance.

A measure of lack of fit between the estimated model and the true model is called discrepancy. It is composed of two parts: discrepancy due to approximation and discrepancy due to estimation. Discrepancy due to approximation measures the distance between the selected model family and the true model. It does not depend on the sample values. It is the discrepancy between the true model and the best approximating model from the family. How good an estimated model is depends on both the model family it belongs to and the sample values because the parameters for the model are estimated from the samples. The estimation error results in the second component of the discrepancy.

Similar to curve fitting, the integrated mean square error (IMSE) is a common criterion to judge a model. This criterion is also known as the Gauss discrepancy[116]. Let  $f(x)$  represent the true model and  $\hat{f}(x)$  represent the estimated model. The

Gauss discrepancy is defined as

$$\Delta_G(f, \hat{f}) = \int_x \left( f(x) - \hat{f}(x) \right)^2 dx \quad (6.9)$$

Using IMSE as the density estimation criterion is often questioned because the purpose of estimating density is often not to obtain the exact same curve as the true density, but to gain more information about the data. This critique leads to another popular discrepancy criterion, the Kullback-Leibler discrepancy, which is also known as relative entropy in information theory:

$$\Delta_{KL}(f, \hat{f}) = -E_f \log \hat{f}(x) = - \int \log \hat{f}(x) f(x) dx \quad (6.10)$$

A third general-purpose discrepancy criterion is the Pearson chi-square discrepancy:

$$\Delta_P(f, \hat{f}) = \sum_x \left( f(x) - \hat{f}(x) \right)^2 / \hat{f}(x), \quad \hat{f}(x) \neq 0 \quad (6.11)$$

This discrepancy is useful for grouped data and is consistent with the use of minimum chi-squared estimation as the parameter estimation method.

Expected discrepancy is closely related to the complexity of a model family. The discrepancy due to approximation favors flexible complex model families, while the discrepancy due to estimation favors rigid simple families. Consequently, the proper model complexity needs a tradeoff between the two discrepancies. Besides the model complexity, the expected discrepancy also depends on the sample size. Small sample size leads to large estimation discrepancy.

Since the true model is not available in reality, an estimator is needed for the expected discrepancy. Sometimes, an analysis of variance and covariance of a fitted model exists for finite samples. When such an analysis is not available, there are three prevalent approaches to do the estimation: asymptotic methods, bootstrap methods and cross-validation methods[116]. Asymptotic methods only analyze the model under the condition that the sample size goes to infinity. Hence, their practical usefulness is determined by how well they estimate the expected discrepancy with

finite sample size. They can also simplify the discrepancy function. In particular, if the true model belongs to the approximating model family, the Kullback-Leibler discrepancy is equivalent to the well-known Akaike information criterion (AIC)

$$C_{KL} = AIC/2n, \quad \text{where } AIC = -2 \log(L) + 2p \quad (6.12)$$

where  $L$  refers to the likelihood and  $p$  the model order. Another criterion which is similar to AIC in form, namely the Bayesian information criterion (BIC), is also a frequently used simplified discrepancy criterion

$$BIC = -2 \log(L) + p \log(n) \quad (6.13)$$

where  $n$  is the number of samples. Clearly, as the number of samples increases, BIC favors simple approximating families (corresponding to smaller  $p$ ) because it puts less emphasis on the prior distribution.

Besides being discrepancy estimators, the above mentioned criteria are also used to determine the proper model complexity. These criteria are reviewed in [44] as explained in the previous section. When applying these criteria, we make an implicit assumption that there is a true model, from which the data are sampled. The goal of model selection is finding the model that best approximates the true model. But this assumption may be incorrect. This concern kindles the Minimum Description Length (MDL) principle. The goal of the MDL principle is not approximating the assumed true model, but finding a good probability model that helps separate useful information in the data from noise. The MDL principle is closely related to Bayesian inference, but it avoids some of the interpretation difficulty of the Bayesian approach. One important advantage of the MDL principle is it automatically protects against overfitting. [36] provides a good starting point for learning the MDL principle. Many algorithms have been developed to realize the MDL principle. The latest development dubbed Normalized Maximum Likelihood (NML) provides a theoretically elegant solution. Without assuming any true model, it minimizes the



worse-case discrepancy as[77]

$$p^* = \arg \min_p \max_q E_q \left[ \log \frac{f(x|\hat{\theta}_x)}{p(x)} \right] \quad (6.14)$$

where  $q$  ranges over the set of all probability distributions satisfying  $E_q \left[ \log \frac{q(x)}{f(x|\hat{\theta}_x)} \right] < \infty$ , and  $f$  is the best approximating model for the given family. The resulting solution is[7]

$$p^*(x) = \frac{f(x|\hat{\theta}_x)}{\int f(y|\hat{\theta}_y) dy}$$

The quantity  $-\ln p^*(x)$  indicates the model complexity. Asymptotically it can be simplified to the form

$$MDL = -2 \log f(x|\hat{\theta}_x) + p \log \left( \frac{n}{2\pi} \right) \quad (6.15)$$

In reality, with only samples available, the best approximating model  $f(x|\hat{\theta}_x)$  is the fitted model. Eq. (6.15) is then reduced to a similar form as AIC and BIC. Abusing notation a little bit, we refer to the resulting quantity as the MDL criterion:

$$MDL = -2 \log(L) + p \log \left( \frac{n}{2\pi} \right) \quad (6.16)$$

According to Wax and Kailath[108], the MDL criterion yields a consistent estimate of the number of signals in the signal detection problem, while the AIC criterion yields an inconsistent estimate that tends to overestimate the number of signals. In a mixture model, each mixture component can also be thought of as a "signal". Therefore, a similar conclusion is expected for determining the order of a Gaussian mixture model. But this conclusion may not be true for a small number of samples. The performance of the three criteria are compared on simulated data from three different Gaussian mixture distributions. The parameters of these distributions are listed in Table 6.1.

Given a set of data, five hypotheses, namely 1, 2, 3, 4 or 5 components in the mixture, are assumed. Under each hypothesis, a Gaussian mixture density function

	Model 1	Model 2	Model 3
No. of components	1	2	3
Component weights	1	0.435, 0.565	0.333, 0.333, 0.333
Component means	0	1, 5	1, 8, 16
Component variances	1	3, 2	2, 2, 2

Table 6.1: Parameters for simulating Gaussian mixture data

is fitted to the data. The mixture parameters are optimized with the EM algorithm. These hypotheses are tested and the one giving the minimum criterion value is picked under each criterion. Two scenarios are considered: one with a large number of data samples and the other with a small number of samples. The hypothesis test is repeated 200 times under each scenario. In each run a new set of data is generated using the parameters in Table 6.1. The number of times a hypothesis is selected is counted and shown in Table 6.2 for a small number of samples and Table 6.3 for a large number of samples respectively.

	Model 1					Model 2					Model 3				
	<b>1</b>	2	3	4	5	1	<b>2</b>	3	4	5	1	2	<b>3</b>	4	5
AIC	177	15	5	1	2	27	152	18	3	0	0	0	135	56	9
BIC	200	0	0	0	0	135	64	1	0	0	0	2	158	38	2
MDL	192	6	1	1	0	50	140	1	0	0	0	0	149	45	6

Table 6.2: Comparison between criteria for selecting mixture models. No. of samples = 100.

	Model 1					Model 2					Model 3				
	<b>1</b>	2	3	4	5	1	<b>2</b>	3	4	5	1	2	<b>3</b>	4	5
AIC	186	9	4	1	2	0	191	9	0	0	0	0	179	21	0
BIC	200	0	0	0	0	0	195	5	0	0	0	0	184	16	0
MDL	200	0	0	0	0	0	195	5	0	0	0	0	184	16	0

Table 6.3: Comparison between criteria for selecting mixture models. No. of samples = 1000.

From these tables, we have the following observations: 1) BIC favors simple models, especially when the confidence is low. Hence, when the modes of some components are close, it tends to underestimate the number of mixture components. This can be clearly seen in Table 6.2. The two modes in Model 2 are close together. Then BIC chooses the one-component hypothesis most of the time when only 100 samples are available. When there are 1000 samples available, it makes the correct judgement most of the time. On the contrary, when the modes of all components are far from each other as in Model 3, BIC has a lower tendency to overestimate the number of mixture components than the other two criteria. 2) AIC tends to overestimate the number of mixture components. This is true even for large data sets. 3) Overall, MDL makes the best tradeoff between overestimation and underestimation.

Parametric and nonparametric methods for density estimation were also compared on simulated data of two different sample sizes. Data were generated using the previous three mixture models (Model 1-3) and four other distributions: 1) Model 4: a uniform distribution; 2) Model 5: a Beta distribution; 3) Model 6: a mixture of a Beta distribution and two Gaussian distributions with mean 0 and 1 respectively. This distribution is simulated because it resembles an incident sunlight field modelled by Ross[91]; 4) Model 7: a mixture of a uniform distribution, a Beta distribution and a Gaussian mixture. The parameters of Models 4-7 are listed in Table 6.4. The algorithm in [115] is followed to estimate the Gaussian mixture density and the MDL criterion is used to determine the number of components in the Gaussian mixture. The KL discrepancy between the estimated distribution and the true distribution was computed for both the parametric and nonparametric methods. The extra discrepancy of using the nonparametric method vs. the parametric method (i.e.  $KL_{NP} - KL_P$ ) is shown in Fig. 6.8. Some of the estimated densities are plotted in Fig. 6.9.

As expected, the parametric method outperforms the nonparametric method

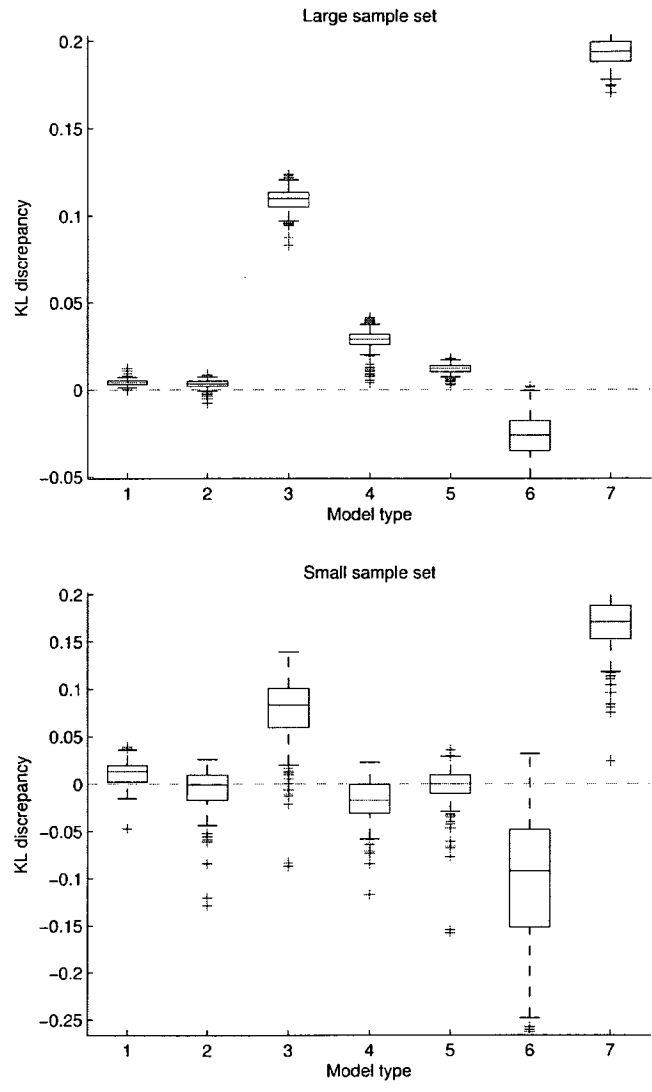


Figure 6.8: Extra discrepancy of using nonparametric method than using parametric method. Values above zero mean the nonparametric density has a larger discrepancy.

	Uniform		Beta			Gaussian Mixture			
	$w$	range	$w$	$\alpha$	$\beta$	$w$	$\lambda$	$\mu$	$\sigma$
Model 4	1	$[\sqrt{3}, \sqrt{3}]$							
Model 5			1	1.7	4				
Model 6			0.45	1.5	1.8	0.55	0.7,0.3	0,1	0.13,0.06
Model 7	1/3	$[-5,9]$	1/3	1.7	4	1/3	0.45,0.55	1,5	3,2

Table 6.4: Parameters for simulating data.  $w$  is the weight of the distribution,  $\alpha$  and  $\beta$  are shape parameters for Beta distribution,  $\lambda$  is the component weight in the Gaussian mixture,  $\mu$  and  $\sigma$  are mean and standard deviation of each Gaussian component respectively.

when the true distribution is a Gaussian mixture, especially when all components are far apart. If some components are close together, the parametric method may not perform as well since it makes the wrong decision in the number of components when the number of samples is small, as in the case of Model 2. It is clear that the parametric method is better at tracing modes in the density function than the nonparametric method. This is partly because a global bandwidth is used in the nonparametric method. When relatively large bandwidth is used to avoid overfitting, the nonparametric method misses some local structure, as in the case of Model 6 and 7. When the true distribution is relatively flat, the nonparametric method often beats the parametric method because the parametric method sometimes generates fake modes in the estimated density function, as seen in Model 5.

When the true model is far from a Gaussian mixture, such as the uniform distribution of Model 4 and the Beta distribution of Model 5, one would expect that the nonparametric method should outperform the parametric method. This is true for small data sets. But surprisingly, the reverse is observed for large data sets. It seems that the parametric method improves faster with the number of samples than the nonparametric method with respect to the KL discrepancy. Of course, this observation may not be true for other distributions. Further study is needed to draw a conclusion on this point.

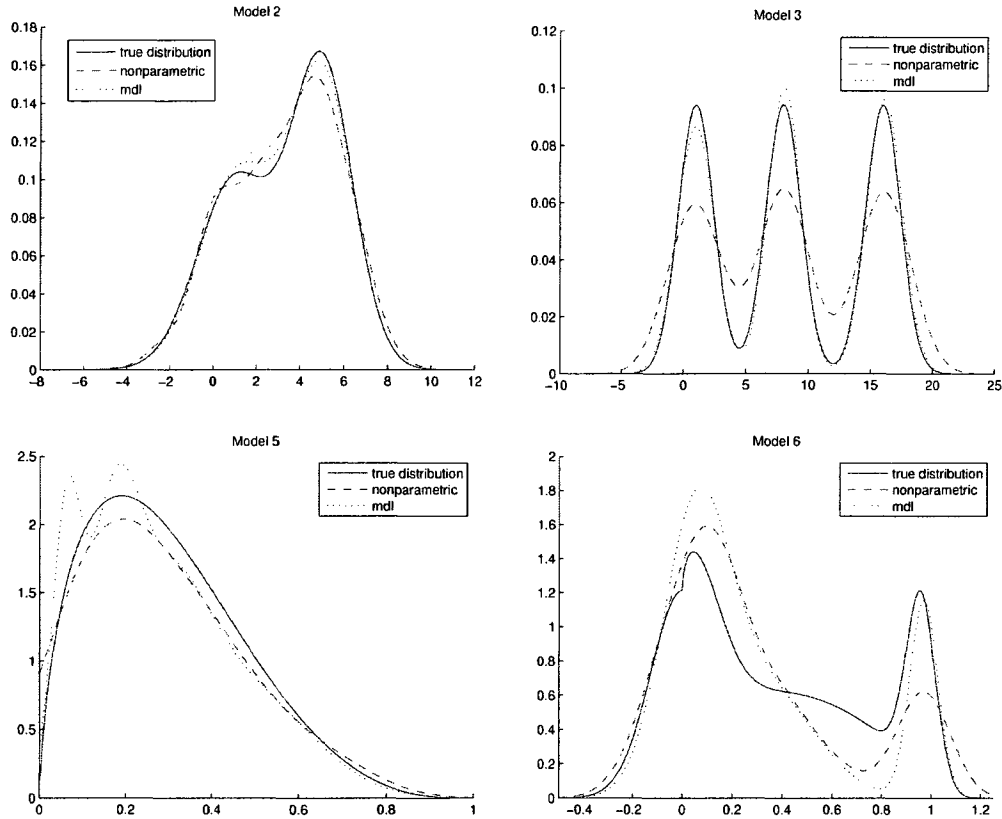


Figure 6.9: Estimated density functions

### 6.4.3 Field Model Library

The purpose of estimating density functions from the image data is to classify the ground reflectivity. As we have noted, the incident light in sunflecks has very small variance and can be taken as constant. Hence, except for a shift in the mean value, the reflected light in sunflecks has the same distribution as that of ground reflectivity. If every spot in the area we study is in a sunfleck for a while over the whole day period, we can save the trouble of classifying ground reflectivity and learn its distribution directly. But this is often not the case, especially when the canopy is thick such that most of the area is shaded. As illustrated in Fig. 6.10, the incident

light intensity is measured with static PAR sensors at two spots and recorded over time. One spot enjoys a long period of sunshine during day time while the other spot hardly has a view of the sun. Under this circumstance, another choice is still available, i.e. learning the ground reflectivity through close-by areas.

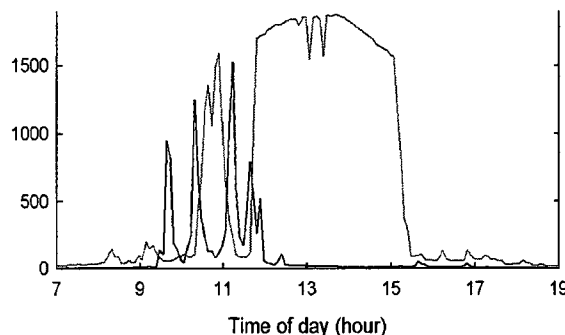


Figure 6.10: Light intensity over time at two spots.

The types of ground reflectivity are limited and often repeated within a contained area. This means if we learn the reflectivity in a subarea, it is very possible that we will find other subareas that have the same reflectivity. Therefore, we can learn the reflectivities in all sunflecks and match them to those not in sunflecks. Here we encounter another problem. Since the reflected light is the summation of incident light and reflectivity (in the log-transformed domain), the density of the reflected light, which is the only density we can obtain from the camera data, is the convolution of the density of incident light and that of reflectivity. In this situation, the incident light behaves like a noise and degrades the classification results. In umbras, the noise distribution is lognormal. In penumbras, the noise distribution is unknown and complex. This complex distribution prevents us from obtaining meaningful classification results from penumbras. Hence, we only classify the reflectivity in umbras. As the sun moves in the sky over a day, sunflecks and umbras shift in locations. Ignoring the penumbras should be good enough to cover most portions of the whole area.

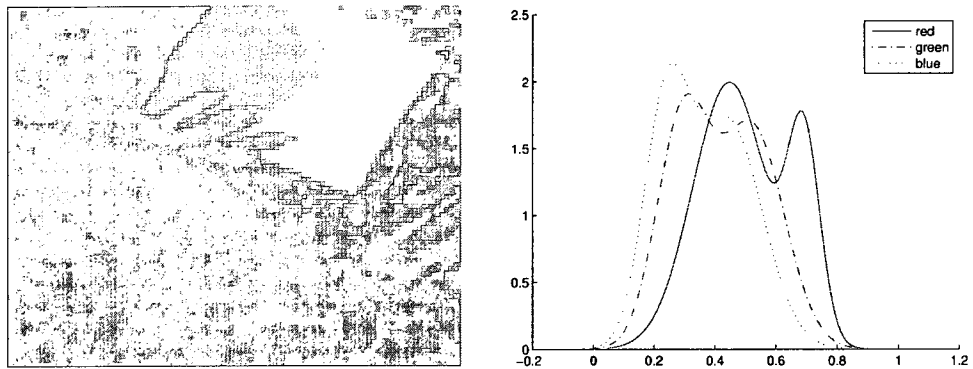
Bases on our previous analysis, we choose the parametric method to estimate the

density functions of reflectivity. The reason is two-fold. First, the density functions usually contain multiple modes. The parametric method is better in estimating such densities. In the field partition step, the field has been partitioned into small subareas such that the reflectivity in each subarea is more homogeneous and its density function contains a smaller number of modes. But in that step, we want to focus on the variance of each subarea and set the bandwidth for nonparametric density estimation used in the image segmentation algorithm to a relatively large value. Hence, many local features may be overlooked. These features need to be identified in the density estimation step.

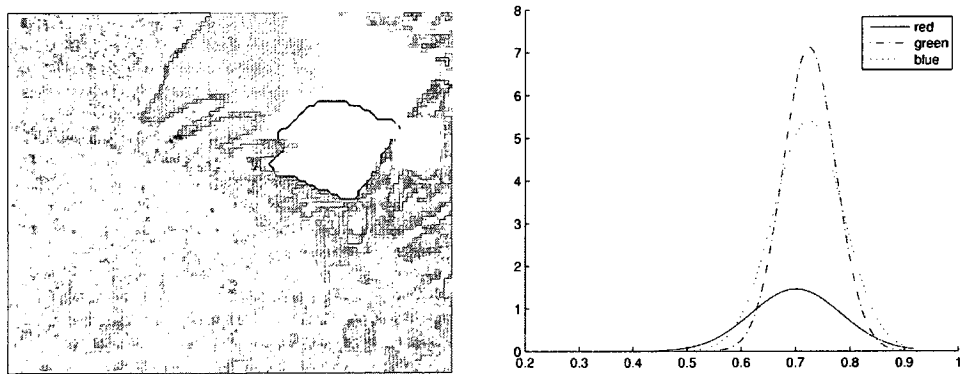
The second reason is that the convolution of a Gaussian distribution and a Gaussian mixture distribution is still a Gaussian mixture distribution. Hence the convolution of the reflectivity distribution with the incident light distribution in umbras, which is Gaussian in the log-transformed domain, is still a Gaussian mixture. This property facilitates obtaining good classification results.

All the reflectivity models obtained from sunflecks are collected in a model library. There are two sources for the reflectivity model library: the models from the sunflecks in the same area to be studied, and the models from sunflecks in a nearby area or even a nearby open area if one is available. When a new model from a sunfleck is available, it is first compared with all the models in the library. If it is believed to cover the same reflectivity type as an existing model, the old model is updated with the new data set. Otherwise, it is added to the library. After training over time, the library will contain the models of most reflectivity types in the area. Fig. 6.11 presents a couple of estimated density functions in the library. It is also possible that some types of reflectivity never appear in the library. For example, some plants are light averse and only grow in the shades. In this case, human involvement may be needed to recognize these types of reflectivity and their models have to be obtained through other ways. Further study is needed to handle this case.





(a) Dead leaves and twigs



(b) Paper

Figure 6.11: Light intensity over time at two spots

Reflectivity models are classified by the average likelihood value

$$i = \arg \max_{f_i} \sum_x \frac{E_{f_i} [\log L_{f_i}(x)]}{n} \quad (6.17)$$

where  $L_{f_i}(x)$  is the likelihood value of  $x$  under model  $f_i$  and  $n$  is the size of the data set. At the beginning, the model library is incomplete and the classification error is high. But as more and more models are available, the classification results will improve. This process is demonstrated in Fig. 6.12. Comparing Fig. 6.12(a) and Fig. 6.12(b), we can easily see the difference made by the addition of the model for a green branch.

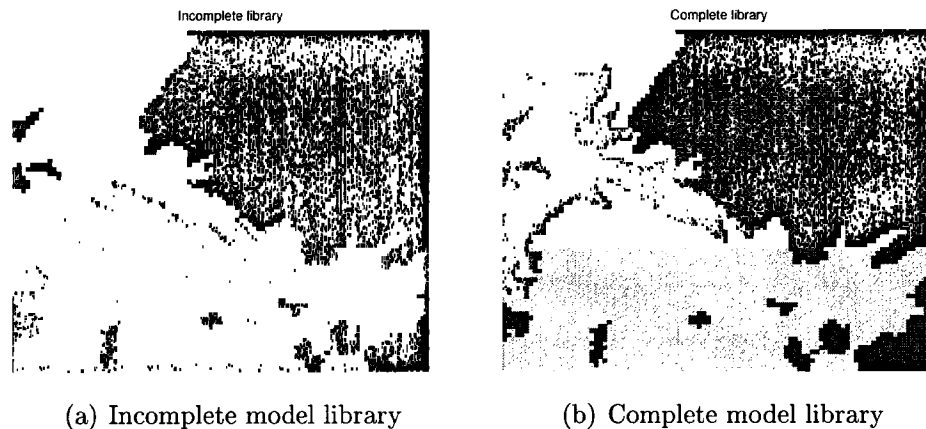


Figure 6.12: Umbra reflectivity classification represented in pseudo-color. Each color corresponds to one type of reflectivity.

Misclassification due to an incomplete library is very undesirable. As can be seen from Table 6.5, the average likelihood values under different models are very different. Therefore, it is possible to set a threshold to determine whether a subarea should be assigned a model. But the thresholds among models should be different too. The initial threshold for a model lies in between the average likelihood value of its own data set and the maximum average likelihood value of all other data sets. The threshold is updated dynamically as the training process goes on.

An umbra that has been a sunfleck is not reclassified in general. But its reflectivity value is still closely monitored by checking its likelihood under its assigned

	Model 1	Model 2	Model 3	Model 4
Umbra 1	0.211	<b>0.656</b>	-0.269	-0.933
Umbra 2	<b>0.558</b>	0.016	-4.811	-0.0284
Umbra 3	1.029	0.297	-3.99	<b>1.254</b>

Table 6.5: Average likelihood under different models for three umbras

reflectivity model. If there is a sudden jump in the likelihood value, the subarea is believed to have undergone a change in its reflectivity due to events such as new objects flying in this subarea. Then the subarea is back to the classification process again.

In addition to the density model, a spatial correlation model is also associated with each reflectivity type. The spatial correlation of the incident light field determines the bandwidth of its power spectrum, which consequently determines the proper sampling density. Without enough incident light data, the correlation of the incident light, which is the difference between the correlation of the reflected light and that of the reflectivity, can be obtained from the reflected light for the subareas with known reflectivity.

## 6.5 Simulations

### 6.5.1 Algorithms Implementation

Once the field is partitioned and modelled, PAR samples are taken and the whole field is reconstructed. The sampling density in each subarea is determined by its correlation model. To minimize the reconstruction error over the entire field, the water-filling principle is adopted. All the sampling methods reviewed in Chapter 3 are applicable in subareas. The good choices are those that allow a small increment in the number of samples to best execute the water-filling principle. At this stage, the uniform sampling method is used for simplicity.

Ideally we should be able to reconstruct a sunfleck without taking any samples

(assuming the light intensity above canopies is available). In reality, the image segmentation result is not perfect. A portion of penumbras surrounding a sunfleck may be included in the sunfleck. Therefore, if the fidelity requirement is high, it is also necessary to take several samples in a sunfleck and reconstruct it with linear interpolation.

The kriging method is chosen as the reconstruction method for two reasons. First, it is able to predict values on ungridded data. This capability is necessary because of the irregular shape of each subarea. More importantly, the expected prediction error is readily available. This error is useful in determining the subarea to be sampled next based on the water-filling principle. The disadvantage of the kriging method is that it is computationally expensive. If the computation power or computation time is limited, the thin plate spline can serve as an alternative reconstruction method.

The entire algorithm is summarized below:

1. Take a series of three images, a bright one, a dark one and a normal one.
  - (a) Segment the bright image to locate umbras.
  - (b) Segment the dark image to locate sunflecks.
  - (c) Partition each type of areas into subareas through the normal image.
2. Calibrate the images to remove the nonlinearity in the camera data.
3. Build the model library.
  - (a) Take a PAR sample at the center of a sunfleck. Verify that it is a true sunfleck instead of an subarea with strong reflectivity.
  - (b) Estimate the density distribution model of the reflectivity in the sunfleck.
  - (c) Compare the model with those in the model library. If it is similar to an existing model, update the existing model with the new set of data. If no similar model exists, add the model in the library.

- (d) Repeat the above steps until all sunflecks are processed.
  - (e) Obtain the correlation model for each type of reflectivity.
4. Assign models in the library to the reflectivity in umbras.
  5. Reconstruct the field
    - (a) Pick a subarea. If its reflectivity is known, estimate the spatial correlation of the incident light through the reflected light and the reflectivity model. If its reflectivity is unknown, apply the correlation model from simulated data as explained in Chapter 2.
    - (b) For a given fidelity requirement, estimate the sampling density for the subarea based on the correlation model.
    - (c) Take samples uniformly and reconstruct the subarea with the kriging method. Estimate the reconstruction error in the subarea.
    - (d) Refine the sampling density in each subarea based on the water-filling principle and its estimated reconstruction error.

### 6.5.2 Simulation Results

This algorithm was tested on a set of data taken at the edge of a canopy over a period of an hour. 13 sets of images were taken at an interval of every five minutes. At each interval, the ground truth of the field was also measured. Because there is no easy way of getting a real ground truth without a very large number of PAR sensors, the ground truth is also obtained from the camera data. To get the ground truth, a white board with uniform reflectivity is placed in the field. The reflected light is measured with the camera. The camera data is then transformed into PAR data through a transfer function obtained from densely deployed PAR sensors in a small area. The calibrated camera data should closely resemble the ground truth and is good enough for our testing purpose.

During the testing period, the sun moved substantially such that everywhere in the field had been covered by sunflecks for a while. To test the classification results, only several images were selected for establishing the model library. After training, the field is reconstructed and compared with the ground truth. One of the reconstruction results for the ground truth in Fig. 6.13 is presented in Fig. 6.14. For comparison, the reconstruction result with uniform sampling and the wavelet reconstruction are also shown. The wavelet reconstruction is presented here because the wavelet domain is the domain we are aware of that can best compress the field without the multiscale scheme. Therefore, it is the best possible result we can get with the compressed sensing approach. For the wavelet reconstruction, a threshold is set based on the fidelity requirement. All coefficients below the threshold are set to 0.

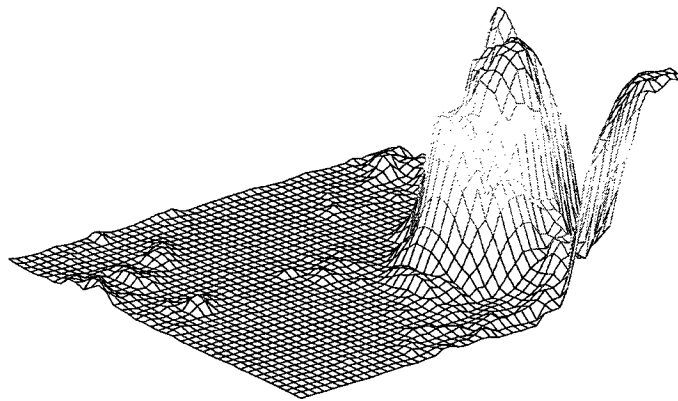
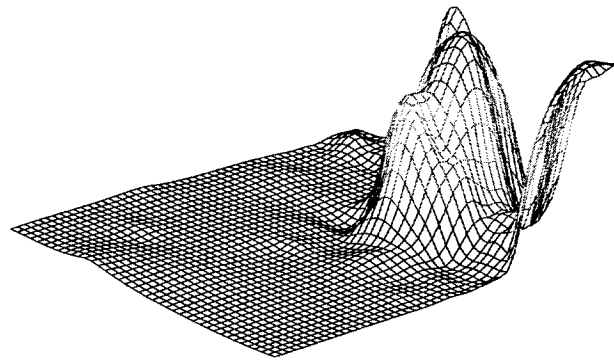
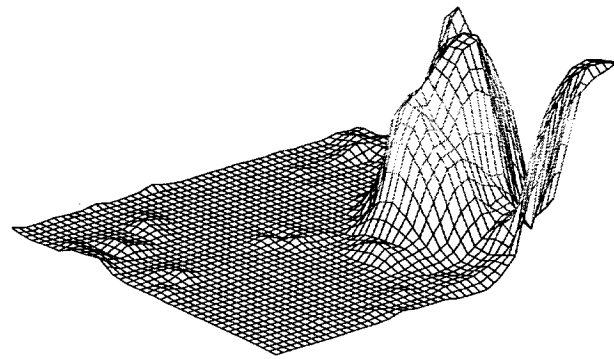


Figure 6.13: Ground truth

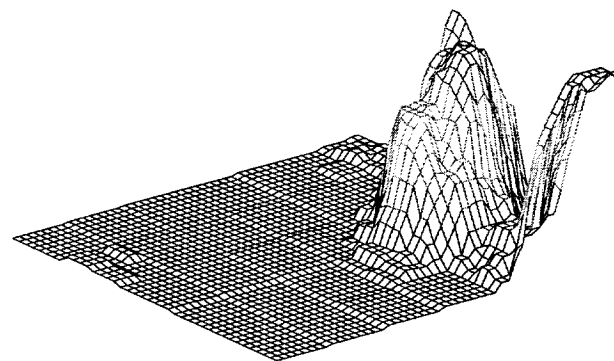
The sampling pattern of the multiscale sensing scheme is shown in Fig. 6.15. Because the spatial correlation in penumbras is smaller than that in umbras, the sampling density in penumbras is much higher. Since the final MSE is relatively low, we take more than one sample in sunflecks. When we lower the fidelity requirement such that no samples are needed in sunflecks, the reconstructed field in the sunflecks is completely flat, as shown in Fig. 6.16.



(a) Multiscale sampling. Number of samples=248. MSE=0.0488



(b) Uniform sampling. Number of samples=546. MSE=0.0452



(c) Wavelet reconstruction. Number of samples=338. MSE=0.0467

Figure 6.14: Reconstructed field with different sampling schemes

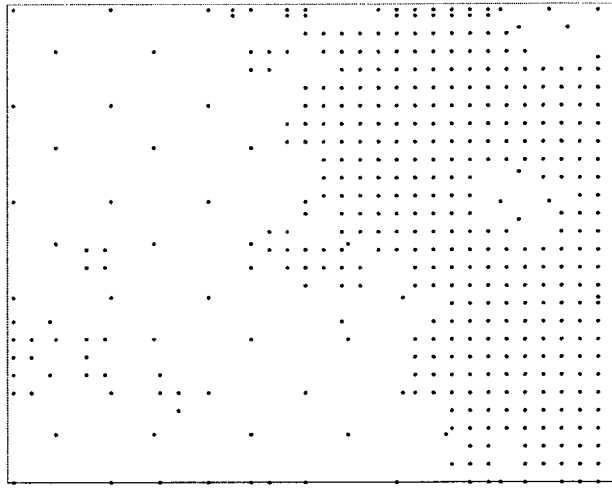


Figure 6.15: Multiscale sensing sampling pattern

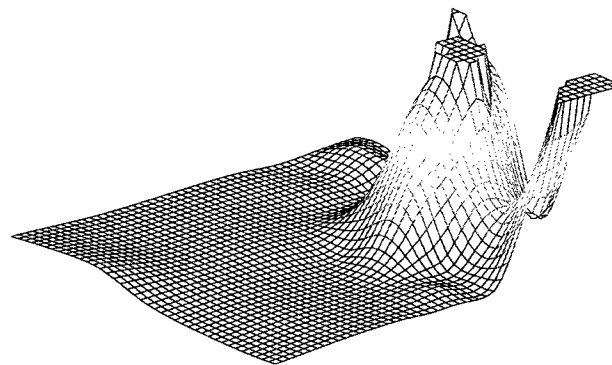


Figure 6.16: Reconstructed field with multiscale sensing scheme under low fidelity requirement



The reduction of MSE with the number of samples for all three methods is given in Fig. 6.17. Obviously, the uniform sampling method has the lowest convergence rate. The multiscale sensing scheme exhibits a better convergence rate than the wavelet reconstruction. As explained in [30], the convergence rate depends on the compressibility of a signal in a domain. In a sense, the field partition provides a new domain in which the signal is more compressible than in the wavelet domain. Although the domain in which the signal is compressible does not have to be known at the sampling time, this knowledge is necessary at the reconstruction time. Multiscale sensing enables us to access this new domain.

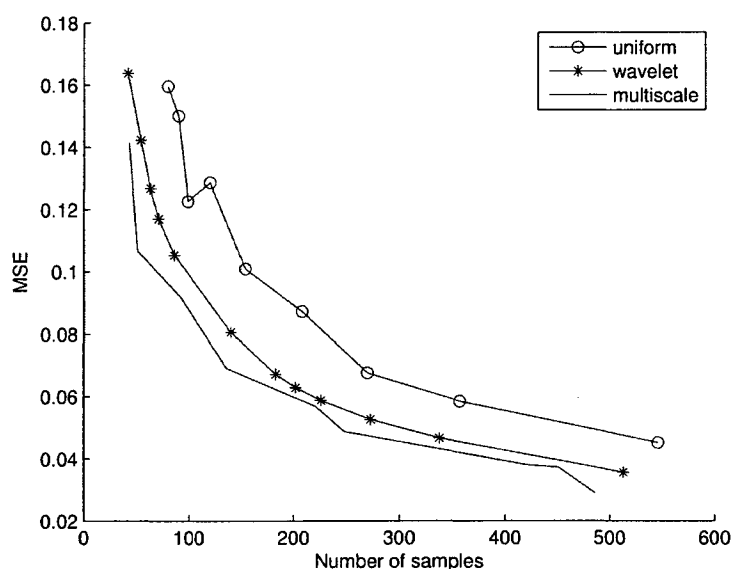


Figure 6.17: MSE convergence rate

## 6.6 Conclusion

The details of a new sampling scheme, the multiscale sensing scheme, are explained in this chapter. In this scheme, information on the field comes from more than one type of sensor. The information from different sources is combined at feature and higher levels. In particular, field partition and field models are provided by one

type of sensor. The field partition provides us with access to a new domain in which the signal is highly compressible. A second type of sensor takes samples for the reconstruction of the field in this new domain. The scheme was tested on the real data and proved to sample the field more efficiently than existing methods. This result demonstrates that this scheme can efficiently fuse information from multiple scales. This scheme can be further improved by combining it with other sampling methods, such as the adaptive sampling methods in the subfield sampling process.

## CHAPTER 7

### Discussions and Future Directions

New sampling strategies in sensor networks were introduced in this dissertation. The characteristics of the incident light field under a forest canopy, the field we concentrate our discussion on, was first analyzed in Chapter 2. The light was decomposed into three elements according to the path it travels through. Based on the components contained in the light reaching an area, the light field was classified into three types. All three types of fields were characterized with statistical models. A simulator was implemented to study the most complex field type. Existing sampling and reconstruction techniques suitable for the light field were reviewed in Chapter 3 and Chapter 4 respectively. In the next two chapters, the design of two new sampling algorithms was elaborated. The first algorithm, an adaptive algorithm, adapts the sampling density to the roughness of the field. It evaluates potential sampling sites by their probability of meeting a predefined fidelity requirement. The most informative sites, i.e., the sites corresponding to the highest probability of not meeting the fidelity requirements, were chosen to be the final sampling sites. The second algorithm, a multiscale sensing scheme, combined information at different levels. Information from a higher level direct the movement of sensors collecting information at a lower level. Both schemes are able to efficiently sample the incident light field.

Although adaptive algorithms have been proved to have no effect on the error convergence rate, they are still useful in the applications where a relatively small number of samples are permitted. Tsaig and Donoho[102] demonstrated that the constant term in the MSE of the reconstructed field is not small for many signals.

Therefore, in the case of a small number of samples, the constant instead of the error rate dominates the final error. If an adaptive algorithm can reduce the constant term, it still provides a big advantage over random sampling, as proved by our simulation results in Chapter 5.

The multiscale sensing scheme makes improvements over existing sampling algorithms not only on the constant term, but also on the error convergence rate. This is because with this scheme the field can be reconstructed in a new domain where the signal is more compressible. Unlike the adaptive algorithm, which is applicable to any field, the utilization of the multiscale sensing scheme is restricted to the fields which can be measured with different types of sensors. At least one type of sensor has to be able to get a global view of the field. At the current stage, it is specially designed for the incident sunlight field. More generalization is needed for this sampling concept. There are several directions in which this generalization can happen:

- The improvement in the multiscale sensing scheme comes from the access to the domain information in which the signal is more compressible. The information stems from a global view of the field through one type of sensor. The current design relies on the image segmentation to obtain the information. However, in many applications, a global view is available in forms other than images. Therefore, a general scheme for determining the sparse domain of a signal from a global view of the signal will be very helpful in sampling algorithms design. Moreover, the high level information is assumed to be free in our study. When this is not the case, a cost function should be associated with both the high level information collection process and the low level information collection process. A resource allocation decision should be made on obtaining these two levels of information as in the double sampling.
- The field modelling process can be extended and generalized. Currently, no

prior knowledge of the field is assumed. Therefore, the field models are extracted from the data at the time of sensing. In the applications where field models do exist, a general way of incorporating the existing field models can be developed. Data fusion schemes that based on field models, such as the Kalman filtering scheme, can then be incorporated in the design.

- A simulator was implemented to simulate the light distribution in penumbras. The models extracted from the simulated data are only used in subareas where empirical models from the real data are not available. The utilization of the simulator can be expanded to provides priors in the subareas where an empirical model is also available. More generally, when models from simulated data and real data are both available, effective combination of the models and the interaction between the simulator and the real measurements is worth a further look.
- In the present development, only the spatial correlation of the field is taken into account. But the field varies both spatially and temporally. True temporal-spatial models should be applied to catch both variations.

## Bibliography

- [1] R. Adams and L. Bistrof, *Seeded region growing*, IEEE Trans. Pattern Analysis and Machine Intelligence **16** (1994), 641–647.
- [2] G. G. Agarwal and W. J. Studden, *Asymptotic integrated mean square error using least squares and bias minimizing splines*, The Annals of Statistics **8** (1980), no. 6, 1307–1325.
- [3] M. S. Allili and D. Ziou, *An automatic segmentation combining mixture analysis and adaptive region information: A level set approach*, Proc. 2nd Canadian Conf. on Computer and Robot Vision, May 2005, pp. 73–80.
- [4] J. G. Annandale, N. Z. Jovanovic, G. S. Campbell, N. Du. Sautoy, and P. Lobit, *Two-dimensional solar radiation interception model for hedgerow fruit trees*, Agricultural and Forest Meteorology, vol. 121, 2004, pp. 207–225.
- [5] D. Arthur and S. Vassilvitskii, *k-means++: The advantages of careful seeding*, Symposium on Discrete Algorithms (New Orleans, LA), Jan. 2007, pp. 1027–1035.
- [6] A.V. Balakrishnan, A Note on the Sampling Principle for Continuous Signals, vol. 3, June 1957, pp. 143–146.
- [7] A. Barron, J. Rissanen, and B. Yu, *The minimum description length principle in coding and modeling*, IEEE Trans. Information Theory **44** (1998), 2743–2760.
- [8] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An introduction to splines for use in computer graphics & geometric modeling*, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987.

- [9] P. Becker and A. P. Smith, *Spatial autocorrelation of solar radiation in a tropical moist forest understory*, *Agricultural and Forest Meteorology* **52** (1990), 373–379.
- [10] R. Bernstein, *Digital image processing of earth observation sensor data*, *IBM Journal of Research and Development* **20** (1976), no. 1, 40–57.
- [11] F. L. Bookstein, *Principal warps: Thin-plate splines and the decomposition of deformations*, *IEEE Trans. Pattern Analysis and Machine Intelligence* **11** (1989), no. 6, 567–585.
- [12] K. Bult, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T.H. Lin, W.J. Kaiser, H. Marcy, R. Mukai, P. Nelson, F.L. Newburg, K.S.J. Pister, G. Pottie, H. Sanchez, K. Sohrabi, O.M. Stafsudd, K.B. Tan, G. Yung, S. Xue, and J. Yao, *Low power systems for wireless microsensors*, *Intl. Symp. Low Power Electronics and Design* (Monterey, CA, USA), Aug. 1996, pp. 17–21.
- [13] M. M. Caldwell and R. W. Pearcy, *Exploitation of environmental heterogeneity by plants: Ecophysiological processes above- and belowground*, ch. 2: *The physical Nature of Solar Radiation in Heterogeneous Canopies: Spatial and Temporal Attribute*, pp. 21–71, Academic Press, 1994.
- [14] S. Cambanis, *Sampling designs for time series*, *Handbook of Statistics* (E. J. Hannan, P. R. Krishnaiah, and M. M. Rao, eds.), vol. 5, Elsevier Science Publishers, 1985, pp. 337–362.
- [15] J. A. Castellanos, J. Neira, and J. D. Tardós, *Multisensor fusion for simultaneous localization and map building*, *IEEE Trans. Robot and Automation* **17** (2001), 908–914.

- [16] T. F. Chan and L. A. Vese, *An efficient variational multiphase motion for the mumford-shah segmentation model*, Thirty-Fourth Asilomar Conf. on Signals, Systems and Computers (Pacific Grove, CA), vol. 1, Oct. 2000, pp. 490–494.
- [17] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE Trans. Pattern Anal. Machine Intell. **17** (1995), 790–799.
- [18] K. C. Chou, A. S. Willsky, and A. Benveniste, *Multiscale recursive estimation, data fusion, and regularization*, IEEE Trans. Automatic Control **39** (1994), 464–478.
- [19] W. L. Chung, *Automatic curve fitting using an adaptive local algorithm*, ACM Trans. Mathematical Software **6** (1980), no. 1, 45–57.
- [20] J. S. Clark, *Why environmental scientists are becoming bayesian*, Ecology Letters (2005), no. 8, 2–14.
- [21] W. G. Cochran, *Sampling techniques*, 2nd ed., John Wiley & Sons, 1963.
- [22] D. Comaniciu and P. Meer, *Robust analysis of feature spaces: Color image segmentation*, 1997 Proc. Computer Vision and Pattern Recognition, IEEE, Jun. 1997, pp. 750–755.
- [23] ———, *Distribution free decomposition of multivariate data*, Pattern Analysis & Applications **1** (1999), no. 1, 22–30.
- [24] C. Coué, Th. Fraichard, P. Bessière, and E. Mazer, *Multi-sensor data fusion using bayesian programming : an automotive application*, IEEE Intl. Conf. on Intelligent Robots and System, vol. 1, 2002, pp. 141–146.
- [25] P. Craven and G. Wahba, *Smoothing noise data with spline functions*, Numerische Mathematik **31** (1979), 377–403.
- [26] N. Cressie, *Statistics for spatial data*, Wiley, 1993.



- [27] R. Cristescu and M. Vetterli, *On the optimal density for real-time data gathering of spatial temporal processes in sensor networks*, IEEE Int. Conf. on Information Processing in Sensor Networks, Apr. 2005, pp. 159–164.
- [28] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society **39** (1977), no. 1, 1–38.
- [29] C. Derksen, A. E. Walker, B. E. Goodison, and J. W. Strapp, *Integrating in situ and multiscale passive microwave data for estimation of subgrid scale snow water equivalent distribution and variability*, IEEE Trans. Geoscience and Remote Sensing **43** (2005), 960–972.
- [30] D. L. Donoho, *Compressed sensing*, IEEE Trans. Information Theory **52** (2006), 1289–1306.
- [31] O. Dubrule, *Comparing splines and kriging*, Computers & Geosciences **10** (1984), no. 2–3, 327–338.
- [32] Hugh Durrant-Whyte, *Multi sensor data fusion (course notes)*, 2001.
- [33] J. J. Faraway, *Sequential design for the nonparametric regression of curves and surfaces*, Proc. 22nd Symposium on the Interface between Computing Science and Statistics **22** (1990), 104–110.
- [34] P. W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch, *Multiresolution optimal interpolation and statistical analysis of topex/poseidon satellite altimetry*, IEEE Trans. Geoscience and Remote Sensing **33** (1995), 280–292.
- [35] K. Fukunaga, *Introduction to statistical pattern recognition*, 2<sup>nd</sup> ed., Academic Press, Boston, 1990.
- [36] P. Grünwald, *A tutorial introduction to the minimum description length principle*, <http://homepages.cwi.nl/~pdg/ftp/mdlintro.pdf>.

- [37] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, *Distributed regression: an efficient framework for modeling sensor network data*, IEEE Intl. Conf. on Information Processing in Sensor Network (Berkeley, CA), Apr 2004.
- [38] C. Guestrin, A. Krause, and A. P. Singh, *Near-optimal sensor placements in gaussian processes*, Proc. 22nd Intl. Conf. on Machine Learning (Bonn, Germany), 2005, pp. 265–272.
- [39] J. Hightower and G. Borriello, *Location systems for ubiquitous computing*, IEEE Computer (2001), 57–66.
- [40] J. W. Fisher III and J. C. Principe, *A methodology for information theoretic feature extraction*, Neural Networks Proc. (Anchorage, AK, USA), vol. 3, May 1998, pp. 1712–1716.
- [41] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, New Jersey, 1999.
- [42] K. Jensen and D. Anastassiou, *Subpixel edge localization and interpolation of still image*, IEEE Trans. Image Processing 4 (1995), no. 3, 285–295.
- [43] M. E. Johnson, L. M. Moore, and D. Ylvisaker, *Minimax and maximin distance designs*, Journal of Statistical Planning and Inference 26 (1990), 131–148.
- [44] M. C. Jones, J. S. Marron, and S. J. Sheather, *A brief survey of bandwidth selection for density estimation*, Journal of the American Statistical Association 91 (1996), 401–407.
- [45] D. L. B. Jupp, *Approximation to data by splines with free knots*, SIAM Journal on Numerical Analysis 15 (1978), no. 2, 328–343.
- [46] W. Kaiser, G. Pottie, M. Srivastava, G. Sukhatme, J. Villasenor, and D. Estrin, *Networked infomechanical systems (nims) for ambient intelligence*, Tech. Report #31, Center for Embedded Networked Sensing, 2003.

- [47] A. H. Kam and W. J. Fitzgerald, *Unsupervised multiscale image segmentation*, Intl. Conf. on Image Analysis and Processing (Venice, Italy), 1999, pp. 316–321.
- [48] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, *An efficient k-means clustering algorithm: Analysis and implementation*, IEEE Trans. Pattern Analysis and Machine Intelligence **24** (2002), 881–892.
- [49] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, Intl. Conf. on Computer Vision (London), vol. 4, 1987, pp. 7–38.
- [50] W. S. Kerwin and J. L. Prince, *The kriging update model and recursive space-time function estimation*, IEEE Trans. Signal Processing **47** (1999), no. 11, 2942–2952.
- [51] R. G. Keys, *Cubic convolution interpolation for digital image processing*, IEEE Trans. Acoustics, Speech, and Signal Processing **29** (1981), no. 6, 1153–1160.
- [52] J. Kim, J.W. Fisher, A. Yezzi, M. Cetin, and A.S. Willsky, *A nonparametric statistical method for image segmentation using information theory and curve evolution*, IEEE Trans. Image Processing **14** (2005), no. 10, 1486–1502.
- [53] G. S. Kimeldorf and G. Wahba, *A correspondence between bayesian estimation on stochastic processes and smoothing by splines*, The Annals of Mathematical Statistics **4** (1970), no. 2, 495–502.
- [54] E. D. Kolaczyk and R. D. Nowak, *Multiscale likelihood analysis and complexity penalized estimation*, The Annals of Statistics **32** (2004), no. 2, 500–527.
- [55] A. Krause, A. Gupta, C. Guestrin, and J. Kleinberg, *Near-optimal sensor placements: Maximizing information while minimizing communication cost*, IEEE Intl. Conf. on Information Processing in Sensor Network (Nashville, TN), Apr 2006.

- [56] A. Kumar, P. Ishwar, and K. Ramchandran, *On distributed sampling of smooth non-bandlimited fields*, Intl. Conf. on Information Processing in Sensor Network (Berkeley), Apr. 2004, pp. 89–98.
- [57] T. M. Lehmann, C. Gönner, and K. Spitzer, *Survey: Interpolation methods in medical image processing*, IEEE. Trans. Medical Imaging **18** (1999), no. 11, 1049–1075.
- [58] R. L. Lemeur and B. L. Blad, *A critical review of light models for estimating the shortwave radiation regime of plant canopies*, Agric. Meteorol. **14** (1974), 255–268.
- [59] J. C. Liu and G. Pok, *Texture edge detection by feature encoding and predictive model*, Proc. on Acoustic, Speech and Signal Processing, vol. 2, IEEE, 1999, pp. 1105–1108.
- [60] R. C. Luo, C. Yih, and K. L. Su, *Multisensor fusion and integration: Approaches, applications, and future research directions*, IEEE Sensor Journal **2** (2002), 107–119.
- [61] T. Lyche and K. Morken, *Making the oslo algorithm more efficient*, SIAM Journal on Numerical Analysis **23** (1986), no. 3, 663–675.
- [62] J. Manyika and H. Durrant-Whyte, *Data fusion and sensor management: A decentralized information-theoretic approach*, Ellis Horwood Series in Electrical and Electronic Engineering, Ellis Horwood, West Sussex, England, 1994.
- [63] I. Maravic and M. Vetterli, *Sampling and reconstruction of signals with finite rate of innovation in the presense of noise*, IEEE Trans. Signal Processing, vol. 53, Aug. 2005, pp. 2788–2805.
- [64] J. S. Marron, *Automatic smoothing parameter selection: A survey*, Empirical Economics **13** (1988), 187–208.

- [65] P. Marziliano and M. Vetterli, *Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations*, IEEE Trans. Signal Processing **48** (2000), no. 12, 3462–3471.
- [66] G. Matheron, *Splines and krigins: Their formal equivalence*, Down-to-Earth Statistics, Solutions Looking for Geological Problem: Syracuse University Geology Contribution (D. F. Merriam, ed.), no. 8, 1981, pp. 77–95.
- [67] J. R. McMahon and R. Franke, *Knot selection for least squares thin plate splines*, SIAM Journal on Scientific and Statistical Computing **13** (1992), no. 2, 484–498.
- [68] E. Meijering, *A chronology of interpolation: From ancient astronomy to modern signal and image processing*, Proc. of the IEEE **90** (2002), 319–342.
- [69] E. E. Miller and J. M. Norman, *A sunfleck theory for plant canopies. part ii. penumbra effect: Intensity distributions along sunfleck segments*, Agronomy Journal **63** (1971), no. 5, 739–743.
- [70] M.J.Brown and G.G.Parker, *Canopy light transmittance in a chronosequence of mixed-species deciduous forests*, Canadian Journal of Forest Research **24** (1994), no. 8, 1694–1703.
- [71] H. Müller, *Optimal design for nonparametric kernel regression*, Statistics and Probability Letters (1984), 285–290.
- [72] T. Müller-Gronbach and K. Ritter, *Spatial adaptation for predicting random functions*, The Annals of Statistics **26** (1998), no. 6, 2264–2288.
- [73] H. Muller and U. Stadmuller, *Variable bandwidth kernel estimators of regression curves*, The Annals of Statistics **15** (1987), no. 1, 182–201.
- [74] W. G. Muller, *Collecting spatial data*, 2 ed., Verlag, Unknown, 2000.

- [75] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. on Pure and Applied Mathematics **42** (1989), no. 5, 577–685.
- [76] R. B. Myneni and I. Impens, *A procedural approach for studying the radiation regime of infinite and truncated foliage spaces. i: Theoretical considerations*, Agricultural and Forest Meteorology **33** (1985), no. 4, 323–337.
- [77] J. I. Myung, D. J. Navarro, and M. A. Pitt, *Model selection by normalized maximum likelihood*, Journal of Mathematical Psychology **50** (2006), 167–179.
- [78] A. B. Nicotra, R. L. Chazdon, and S. V. Iriarte, *Spatial heterogeneity of light and woody seedling regeneration in tropical wet forests*, Ecology **80** (1999), no. 6, 1908–1926.
- [79] J. M. Norman, E. E. Miller, and C. B. Tanner, *Light intensity and sunfleck-size distributions in plant canopies*, Agronomy Journal **63** (1971), 743–748.
- [80] R. Nowak, U. Mitra, and R. Willett, *Estimating inhomogeneous fields using wireless sensor networks*, IEEE Journal on Selected Areas in Communications **22** (2004), no. 6, 999–1006.
- [81] P. Oker-Blom, *Penumbra effects of within-plant and between-plant shading on radiation distribution and leaf photosynthesis: A monte-carlo simulation*, Photosynthetica **18** (1984), 522–528.
- [82] ———, *The influence of penumbra on the distribution of direct solar radiation in a canopy of scots pine*, Photosynthetica **19** (1985), 312–317.
- [83] E. Parzen, *On estimation of a probability density function and mode*, The Annals of Mathematical Statistics **33** (1962), no. 3, 1065–1076.
- [84] D. P. Petersen, *Sampling and reconstruction of wave-number-limited functions in  $n$ -dimensional euclidean spaces*, Information and Control **5** (1962), 279–323.

- [85] K. Popat and R. W. Picard, *Cluster-based probability model and its application to image and texture processing*, IEEE Trans. Image Processing **6** (1997), 268–284.
- [86] G.J. Pottie and W.J. Kaiser, *Wireless integrated network sensors*, Communications of the ACM **43** (2000), 51–58.
- [87] M.J.D. Powell, *The uniform convergence of thin plate spline interpolation in two dimensions*, Numerische Mathematik **68** (1994), 107–128.
- [88] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin, *Adaptive sampling for environmental field estimation using robotic sensors*, IEEE/RSJ Intl. Conf. on Intelligence Robots and Systems (Edmonton, Canada), 2005, pp. 3692–3698.
- [89] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava, *Adaptive sampling for environmental robotics*, IEEE Intl. Conf. on Robotics and Automation (New Orleans, LA), 2004, pp. 3537–3544.
- [90] C. H. Reinsch, *Smoothing by spline functions*, Numerische Mathematik **10** (1967), 177–183.
- [91] J. Ross, M. Sulev, and P. Saareleid, *Statistical treatment of the par variability and its application to willow coppice*, Agricultural and Forest Meteorology **91** (1998), 1–21.
- [92] J. R. Sack and J. Urrutia, *Handbook of computational geometry*, Elsevier, Amsterdam, Netherlands, 2000.
- [93] M. K. Schneider, P. W. Fieguth, W. C. Karl, and A. S. Willsky, *Multiscale methods for the segmentation and reconstruction of signals and images*, IEEE Trans. Image Processing **9** (2000), 456–468.

- [94] D. W. Scott, *Multivariate density estimation - theory, practice and visualization*, John Wiley & Son Inc, 1992.
- [95] C. E. Shannon, *Communication in the presense of noise*, Proceedings of the IEEE, vol. 86, Feb. 1998, pp. 447–457.
- [96] A. Singh, D. Budzik, W. Chen, M. A. Batalin, and W. J. Kaiser, *Multiscale sensing: A new paradigm for actuated sensing of high frequency dynamic phenomena*, IEEE Intl. Conf. Intelligent Robots and Systems, Oct. 2006, pp. 328–335.
- [97] K. C. Slatton, M. M. Crawford, and B. L. Evans, *Fusing interferometric radar and lasor altimeter data to estimate surface topography and vegetation heights*, IEEE Trans. Geoscience and Remote Sensing **39** (2001), 2470–2482.
- [98] B. Solaiman, L. E. Pierce, and F. T. Ulaby, *Multisensor data fusion using fuzzy concepts: Application to land-cover classification using ers-1/jers-1 sar composites*, IEEE Trans. Geoscience and Remote Sensing **37** (1999), 1316–1326.
- [99] C. J. Stone, *Optimal global rates of convergence for nonparametric regression*, The Annals of Statistics **10** (1982), no. 4, 1040–1053.
- [100] G. Strang and G. Fix, *A fourier analysis of the finite element variational method*, Constructive Aspect of Functional Analysis, 1971, pp. 796–830.
- [101] Student, *The probable error of a mean*, Biometrika **VI** (1908), 1–25.
- [102] Y. Tsaig and D. L. Donoho, *Extensions of compressed sensing*, Signal Processing **86** (2006), 549–571.
- [103] M. Uncer, *Sampling–50 years after shannon*, Proc. of the IEEE **88** (2000), no. 4, 569–587.



- [104] M. Unser and I. Daubechies, *On the approximation power of convolution-based least squares versus interpolation*, IEEE Trans. Signal Processing **45** (1997), 1697–1711.
- [105] M. Vetterli, P. Marziliano, and T. Blu, *Sampling signals with finite rate of innovation*, IEEE. Trans. Signal Processing **50** (2002), no. 6, 1417–1428.
- [106] G. Wahba, *Bayesian confidence intervals for the cross-validated smoothing spline*, Journal of Royal Statistical Society, Series B **45** (1983), no. 1, 133–150.
- [107] Y. P. Wang and P. G. Jarvis, *Description and validation of an array model - maestro*, Agricultural and Forest Meteorology **51** (1990), 257–280.
- [108] M. Wax and T. Kailath, *Detection of signals by information theoretic criteria*, IEEE Trans. Acoustics, Speech and Signal Processing **ASSP-33** (1985), 387–392.
- [109] A. Weiss and J. M. Norman, *Partitioning solar radiation into direct and diffuse, visible and near infra-red components*, Agricultural Forest Meteorology **34** (1985), 205–213.
- [110] C. K. Wikle, L. M. Berliner, and N. Cressie, *Hierarchical bayesian space-time models*, Environmental and Ecological Statistics **5** (1998), 117–154.
- [111] R. Willett, A. Martin, and R. Nowak, *Backcaster: Adaptive sampling for sensor networks*, Intl. Conf. on Information Processing in Sensor Networks (Berkeley, CA), 2004, pp. 124–133.
- [112] Z. Zhang and R. S. Blum, *A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application*, Proc. of the IEEE **87** (1999), 1315–1326.

- [113] F. Zhao, J. Shin, and J. Reich, *Information-driven dynamic sensor collaboration*, IEEE Signal Processing Magazine (2002), 61–72.
- [114] S. C. Zhu and A. Yullie, *Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation*, IEEE Trans. Pattern Analysis and Machine Intelligence **18** (1996), 884–900.
- [115] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, *Gaussian mixture density modeling, decomposition, and applications*, IEEE Trans. Image Processing **5** (1996), 1293–1302.
- [116] W. Zucchini, *An introduction to model selection*, Journal of Mathematical Psychology **44** (2000), 41–61.