INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

STATISTICAL ASPECTS OF DIGITAL IMAGE PROCESSING

By
Malgorzata Winiszewska
B.Sc., Carleton University

A Thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Science in Mathematics

Carleton University
Ottawa, Ontario
July 2001
©2001, Malgorzata Winiszewska

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-66920-3

Canada

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

**STATISTICAL ASPECTS
OF DIGITAL IMAGE PROCESSING**

Submitted by
Malgorzata Winiszewska, B.Sc.

in partial fulfilment of the requirements
for the degree of Master of Science

Director, School of Mathematics and Statistics

Thesis Supervisor

External Examiner

Carleton University

May 4, 2001

© 2001, Malgorzata Winiszewska

Abstract

In this thesis, we assume a Markov random field model for an image. The equivalence of a Markov random field with a Gibbs distribution is established, and we express the local interactions of pixel values in terms of an energy function. We present the Bayesian approach to image restoration and approximate the image estimators. The methods of approximation include Markov Chain Monte Carlo, Simulated Annealing and Iterated Conditional Modes. The simulation algorithms presented are Hasting, Metropolis and Gibbs.

Acknowledgments

I would like to thank my supervisor, Professor Amitava Bose, for his guidance and availability throughout the time of writing this thesis. I appreciate his advice, encouragement and expectations.

I wish to thank Dr. A. Dabrowski, Dr. S. Mills and Dr. J. Oommen whose comments and suggestions contributed to the improvement of this thesis.

I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for their trust in my research abilities. The financial support they provided allowed me to focus entirely on my studies.

Finally, I wish to thank my family, friends and acquaintances: Bartek, Darek, David, Gosia, Isabelle, Jan, Lal, Zahra, Vahid and Victor for many helpful conversations.

But most, I am grateful to my parents, my sister and Steven for their constant support and understanding.

Mojemu Dziadkowi
(To My Grandfather)

Contents

Abstract	iii
Acknowledgments	iv
Dedication	v
1 Introduction	1
2 Image Enhancement	4
2.1 Numerical Representation of Images	4
2.2 Point Operations	5
2.2.1 Grey Level Histogram	6
2.2.2 Histogram Equalization	6
2.3 Image Filtering	9
2.3.1 Spatial Domain Filters	9
2.3.2 Frequency Domain Filters	11
2.4 Edge Detection	14
2.4.1 Determining the Edge Points	15
2.4.2 Sobel Edge Detection	18
3 Image Restoration	21
3.1 Model for Degraded Image	21
3.2 Image as a Random Field - Power Spectrum	23
3.3 Inverse Filter	26

3.4	Wiener Filter	28
3.5	Power Spectrum Approximation	29
4	Markov Random Fields	31
4.1	Random Fields as Models for Images	31
4.2	Markov Random Fields and Gibbs Distributions	34
4.3	Standard Models for Images	42
4.3.1	Ising Model	42
4.3.2	Potts Model	45
4.4	Parameter Estimation for Gibbs Distribution	47
5	Simulation of Random Fields	52
5.1	Review of Markov Chain Theory	52
5.2	From Markov Chains to Random Fields	55
5.3	Simulation Algorithms	58
5.3.1	Hasting Algorithm	59
5.3.2	Metropolis Algorithm	61
5.3.3	Sitewise Hasting and Metropolis	62
5.3.4	Gibbs Sampler	63
5.3.5	Example: Ising Model	65
5.4	Markov Chain Monte Carlo (MCMC)	67
6	Bayesian Image Restoration	70
6.1	Bayesian Approach to Image Restoration	70
6.2	Prior and Posterior Distributions	71
6.2.1	Example: Ising model	74
6.3	Estimation of the True Image	74
6.4	Techniques of Approximating Image Estimators	76
6.4.1	MCMC Approximation	76
6.4.2	Simulated Annealing	78
6.4.3	Iterated Conditional Modes (ICM)	82
6.4.4	Example: Potts Model	83

7	Experimental Results	85
7.1	Numerical Methods of Image Processing	88
7.2	Statistical Methods of Image Restoration	93
8	Conclusion	96
A	Fundamentals of Digital Signal Processing	98
A.1	Signals	98
A.2	Spatial Domain Representation of 2-D Digital Signal	100
A.2.1	Linear Shift-Invariant Systems	100
A.2.2	The Impulse Response	101
A.3	Frequency Domain Representation of 2-D Signals	102
A.3.1	Two-Dimensional Discrete Fourier Transform	102
A.3.2	The Frequency Response	104
A.3.3	Fast Fourier Transform	104
	Bibliography	108

Chapter 1

Introduction

We store a digital monochrome image as matrix of integers. These integers, called pixel values, represent the observed intensity levels.

The methods of manipulating the image to produce enhanced, restored or compressed version of the input are collectively known as image processing. In this thesis, we place emphasis on the aspects of subjective and model-based image improvement and first present numerical, then statistical approaches.

In the statistical approach, the observed image is assumed to be a sample from a known probability distribution, which incorporates local interactions of pixel values. It is also assumed that the pixels located within small neighborhoods tend to have similar values. We quantify these statements by the means of a Markov random field and model the image as a Gibbs distribution.

We use Bayesian methodology to restore images, which requires empirical data and a prior model. In our case, the empirical data is given by the observed image, and the prior is a specified Gibbs distribution. Further, we assume that the random distortion process is known. This allows us to obtain a posterior distribution and to use it to estimate the true scene. The resulting estimate is the most probable configuration of a Markov random field given the observed image.

High dimensionality of the image usually excludes any direct method for obtaining the estimates. However, they can be approximated with the use of stochastic algorithms.

This thesis is a synthesis of the available material on image processing. We put together methods arising from the fields of signal processing and statistics to emphasize the necessity of mathematical modeling. The connection between the methods is established based not only on the consistent notation but most of all on their theoretical evolution.

The outline of the thesis is as follows.

In Chapter 2, we introduce the concept of image enhancement and present few of the most common techniques used in spatial and frequency domains. They include histogram equalization, image filtering and edge detection. The goal is to improve the overall appearance of the observed image based on some subjective criteria.

In Chapter 3, we define random fields and attempt to recover the original image assuming a random model for the degradation process. The restoration techniques presented are the inverse and Wiener filters.

We elaborate on a statistical model in Chapter 4 and represent images in terms of Markov random fields. The equivalence between a Markov random field and a Gibbs distribution is established. This allows us to express a probability distribution of a random field in terms of an energy function used to model spatial interactions. We present two standard models for images, the two-color Ising model and its extension to more levels of grey, the Potts model. Parameter estimation for Gibbs distribution in the special case of the exponential family is discussed.

In Chapter 5, we explain how to obtain samples from the high dimensional Gibbs distribution through simulation of Markov random fields. The simulation algorithms presented are Hasting, Metropolis and Gibbs.

Chapter 6 presents the Bayesian approach to image restoration based on the Markov random field model. There are two possibilities for the estimated true image. One is the configuration with the maximum probability of occurrence as a whole, given the observed image. The other is the configuration in which each individual pixel value has the maximum probability, given the observed image. Due to high dimensionality of data, these Bayesian image estimators must be approximated. We

describe three iterative approximation methods: Markov Chain Monte Carlo, Simulated Annealing and Iterated Conditional Modes.

The experimental results of various numerical and statistical image-processing techniques ¹ are presented in Chapter 7. We design a small synthetic image, corrupt it with noise and compare the performance of the estimators. The statistically recovered images are much better approximation of the original than the numerically filtered images. However, they require more computational time and more memory space.

Finally, in Chapter 8, we summarize some ideas for possible future work.

The fundamentals of two-dimensional discrete signal processing theory are outlined in Appendix A.

¹The source codes for these techniques and all the other examples throughout the thesis were written by the author in MATLAB.

Chapter 2

Image Enhancement

Image enhancement refers to any process intended to subjectively improve the overall appearance so that an image is more useful for display and analysis.

This chapter presents some of the most common methods for image enhancement in spatial and frequency domains. They include histogram equalization, image filtering and edge detection.

We begin by describing the numerical framework for image processing.

2.1 Numerical Representation of Images

We can view a monochrome image as a two-dimensional light intensity function, $x(n_1, n_2)$, where n_1 and n_2 are spatial coordinates and the value of x at (n_1, n_2) is proportional to the brightness of the image at that point.

Figure 2.1 illustrates the concept of digital image representation. A *digital image* is the image x that has been discretized in the spatial coordinates and brightness. It is a finite $(N_1 \times N_2)$ rectangular array of integers whose values correspond to the grey level of the image at each pixel location. A *pixel* (“picture element”) is the basic unit of which an image is composed. Each pixel has an integer location or address that consist of the row and column numbers (later summarized into a single “site” number) and an integer value, called the gray level.

Figure 2.1 A physical image and a corresponding digital image representation.

The range of intensity present in an image is often quantized to 256 levels. The levels are denoted by integers from 0 to 255, which correspond to the intensities from the darkest to the brightest as shown in Figure 2.2.

Figure 2.2 A 256-level grey scale.

2.2 Point Operations

Point operations belong to the class of basic processing techniques that modify the grey level content of an image. The procedure is a mapping of the grey level values present in the input image into new grey level values in the output image. This is

done “pixel-by-pixel” according to a specified grey-scale transformation function f :

$$y(n_1, n_2) = f(x(n_1, n_2))$$

where $x(n_1, n_2)$ denotes a pixel value at the coordinates (n_1, n_2) in the input image. Similarly, $y(n_1, n_2)$ represents the same for the output image.

2.2.1 Grey Level Histogram

The intensity content of a monochrome image can be summarized in a histogram, which specifies the number of pixels at each grey level. One might note that when an image is condensed into a histogram all the spatial information is lost. Thus the histogram is unique for any given image, but the reverse is not true.

The histogram conveys information about the apparent brightness and contrast of the corresponding image. For example, a histogram with a narrow base would indicate that the grey level values are similar throughout the whole image. This would correspond to poor contrast as illustrated in Figure 2.3.

Obviously, any intensity manipulations within the image must be reflected in changes in its histogram. This section exploits the converse of the above property: we will use the histogram itself to enhance the contrast in the image.

2.2.2 Histogram Equalization

Histogram equalization technique uses point operations to obtain an image with the same number of pixels at each gray level. The goal is to design a gray-scale transformation function f so that the resulting image yields a uniform histogram.

Consider the grey level as a continuous random variable taking values in $[0, 1]$. We denote it by G and assign to it the probability distribution p_G :

$$p_G(g) = \begin{cases} 1, & 0 \leq g \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

The cumulative distribution is given by

$$F_G(g) := P[G \leq g] = \begin{cases} 0, & 0 < g \\ g, & 0 \leq g \leq 1 \\ 1, & g > 1 \end{cases}$$

Define a random variable $U := F_G(G)$. Since F_G is a continuous distribution function, which is increasing on $[0, 1]$, it has a unique inverse: $F_G^{-1}(u) = \inf\{g : F_G(g) \geq u\}$.

Then, for $0 \leq u \leq 1$, we can write:

$$\begin{aligned} P[U \leq u] &= P[F_G^{-1}(U) \leq F_G^{-1}(u)] \\ &= P[F_G^{-1}(F_G(G)) \leq F_G^{-1}(u)] \\ &= P[G \leq F_G^{-1}(u)] \\ &= F_G[F_G^{-1}(u)] = u \end{aligned}$$

That is, $U := F_G(G)$ is uniformly distributed over $[0, 1]$. Hence, the cdf F_G can be used as the grey-scale transformation function.

In the case of digital images with 256 possible levels of grey, we consider G to be a discrete random variable. The empirical probability of a pixel taking on a grey level value γ can be then defined based on the input image as

$$P[G = \gamma] = \frac{\text{number of pixels with grey level value } \gamma \text{ in the image}}{\text{total number of pixels in the image}}$$

or in terms of a histogram h

$$P[G = \gamma] = \frac{h(\gamma)}{\sum_{j=0}^{255} h(j)}, \quad \gamma \in \{0, 1, \dots, 255\}$$

where $h(\gamma)$ denotes the number of pixels in the input image with a grey level value $\gamma \in \{0, 1, \dots, 255\}$.

The cumulative distribution function of the discrete-valued G is then

$$F_G(\gamma) = P[G \leq \gamma] = \sum_{j=0}^{\gamma} P[G = j], \quad \gamma \in \{0, 1, \dots, 255\}$$

Now if we let f to be the cdf F_G as given in the above equation, the pixel values in the output image will be in the range from $F_G(0)$ to 1 just like the distribution function itself.

In order to requantize the grey level values to fill the range from 0 to 255, one could use the following equation [Jain, 1986] :

$$F^*(\gamma) = \text{Int} \left[\frac{(F(\gamma) - F(0))}{(1 - F(0))} 255 + 0.5 \right] \quad \gamma \in \{0, 1, \dots, 255\}$$

where $\text{Int}[x]$ denotes the integer part of x which defines the desired grey-level transformation function f . That is:

$$y(n_1, n_2) = F^*(x(n_1, n_2))$$

Unfortunately, we cannot expect a perfectly equalized histogram. This is because G is not continuous, hence $F_G(\gamma)$ is not uniformly distributed.

Figure 2.3 shows an image before and after applying the histogram equalization technique.

Figure 2.3 *Histogram equalization.*

2.3 Image Filtering

Image filtering is one of the most common image processing operations. In this section we describe the approaches in spatial and frequency domains.

2.3.1 Spatial Domain Filters

Many image enhancement techniques are based on spatial operations performed on local neighborhoods of input pixels. These procedures are carried out by convolving the original image with an appropriate spatial mask to produce a filtered image.

Spatial Averaging Filter

In the spatial averaging, each pixel is replaced by a weighted average of its neighbors. For a window of size $2k + 1$ we can express this operation as a convolution

$$y(n_1, n_2) = \sum_{m_1=n_1-k}^{n_1+k} \sum_{m_2=n_2-k}^{n_2+k} x(m_1, m_2)h(n_1 - m_1, n_2 - m_2)$$

where x and y are input and output images respectively and $h(n_1, n_2)$ are the filter weights.

An often used spatial averaging filter, called the mean filter, is given by the following mask

$$\begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}$$

It replaces each pixel by the average of its nearest four pixels according to

$$y(n_1, n_2) = \frac{1}{4} [x(n_1 - 1, n_2) + x(n_1 + 1, n_2) + x(n_1, n_2 - 1) + x(n_1, n_2 + 1)]$$

Figure 2.4 illustrates the effects of applying the mean filter to a corrupted image.

Figure 2.4 *Mean filtering.*

Median Filtering

The median filter belongs to the group of so-called rank-order filters. These filters depend on the ranking of pixels inside a window according to their grey values. Here, each input pixel is replaced by the median of the pixels around it that are contained in the window.

In most cases, the window size is chosen so that it contains an odd number of pixels. If it is even, then the median is taken as the grey level that is centered between the two middle values.

For a 3×3 window we have

$$y(n_1, n_2) = \text{median}\{x(n_1 - 1, n_2), x(n_1 + 1, n_2), x(n_1, n_2 - 1), x(n_1, n_2 + 1)\}$$

The results of applying the median filter to a degraded image can be seen in Figure 2.5.

Figure 2.5 *Median filtering.*

2.3.2 Frequency Domain Filters

In practice, when dealing with 256×256 images, applying a convolution can be very time consuming. However, there exists a faster method that involves the use of Fourier transform. Refer to Appendix A for further discussion on signal transformations.

A monochrome image is a two dimensional signal. Since usually the most distinguished information is hidden in the frequency content of the signal, the Fourier transformation of the image will not just speed up the filtering operations, but primarily it will allow us to obtain further information that is not easily available in the spatial domain. We find that the frequency information contained in an image is directly related to its appearance. For example, low frequencies correspond to gradual changes in intensity while high frequencies correspond to rapid changes such as presence of edges or noise corruptions.

Filtering in the frequency domain can be used to pass unchanged certain frequency components in a signal and to block other frequency components completely. This is equivalent to multiplying the Fourier transform of the image with a filter function.

We denote the discrete Fourier transform of a signal $x(n_1, n_2)$ by $\mathcal{F}_x(\omega_1, \omega_2)$.

Low Pass Filter

As the name suggests, low pass filters eliminate high frequencies and leave original low frequency components intact. This results in a decrease of noise but at the cost of blurring and edge reduction.

The transfer function of an ideal 2-D low pass filter can be expressed as

$$\mathcal{F}_h^{lp}(\omega_1, \omega_2) = \begin{cases} 1 & , \text{ if } \sqrt{\omega_1^2 + \omega_2^2} \leq r_0 \\ 0 & , \text{ otherwise} \end{cases}$$

where $r_0 > 0$ is the *cut-off frequency*. An example of a low pass filter with the cut-off frequency 0.3 is shown in Figure 2.6. The frequency spectrum of this ideal low-pass filter is 1 inside a circle of radius $r_0 = 0.3$, and 0 elsewhere.

Figure 2.6 *Ideal Low Pass filter.*

High-Pass Filter

Conversely, high pass filters eliminate low frequencies without disturbing high frequency information. This results in the loss of overall contrast, enhancement of small details and sharpening of edges.

The transfer function of an ideal high pass filter is given by

$$\mathcal{F}_h^{hp}(\omega_1, \omega_2) = 1 - \mathcal{F}_h^{lp}(\omega_1, \omega_2) = \begin{cases} 0 & , \text{ if } \sqrt{\omega_1^2 + \omega_2^2} \leq r_0 \\ 1 & , \text{ otherwise} \end{cases}$$

Similar approach can be taken to design filters that pass or block all frequency components between r_1 and r_2 (bandpass or bandstop filters).

Figure 2.7. illustrates the high pass filter with the cut-off frequency 0.03. The frequency spectrum of this ideal high-pass filter is 1 everywhere, except inside a circle of radius $r_0 = 0.03$, where it is 0.

Figure 2.7 *Ideal High Pass filter.*

Homomorphic Filter

Homomorphic filtering is useful for improving the appearance of an image by simultaneous compression of its intensity range and contrast enhancement.

An image can be expressed as a product of a reflectance function $r(n_1, n_2)$ and an illumination function $i(n_1, n_2)$:

$$x(n_1, n_2) = r(n_1, n_2)i(n_1, n_2)$$

The illumination component is assumed to be the main contributor to the dynamic range of an image and to vary slowly. The reflectance represents the details in an object, it is assumed to be the main contributor to local contrast and to vary rapidly. Consequently, to reduce the dynamic range and increase local contrast, $r(n_1, n_2)$ should be increased and $i(n_1, n_2)$ reduced.

Taking the logarithm of the above equation will separate the two factors. The two effects become then additive:

$$\ln x(n_1, n_2) = \ln r(n_1, n_2) + \ln i(n_1, n_2)$$

Based on the above assumptions, low-pass filtering of $\ln x(n_1, n_2)$ will result in $\ln i(n_1, n_2)$, and high-pass filtering of $\ln x(n_1, n_2)$ will result in $\ln r(n_1, n_2)$. Once the two factors are separated, $\ln i(n_1, n_2)$ can be attenuated to reduce the dynamic range and $\ln r(n_1, n_2)$ emphasized to increase the local contrast.

2.4 Edge Detection

Edges are very important in the object identification process. They provide powerful visual clues - most of the elements in an image can be recognized based only on their edges. Thus, accurately identified edges allow us to locate objects within an image and to measure their shape properties. For this reason edge detection is an essential tool in object identification and classification. Hence, it is one of the most commonly used operations in image analysis.

Generally, an edge can be regarded as a boundary between two neighboring regions in an image which differ significantly in some physical aspect such as intensity, color or texture. Here, only changes in intensity will indicate edges. We must realize however, that the significance of an intensity change is application dependent: a difference that would be classified as an edge in some applications might not be considered an edge in other applications. For example, when extracting edges in a picture of a building we might only want to see the major contours but not the boundaries of the light reflection from the windows, cracks in the walls or other exterior damage and color variations.

Another problem, which greatly affects performance of an edge detection process, is the presence of distortion in images. For example, changes in intensity in a blurred image are very gradual and it is almost impossible to decide on a point at which a change starts to occur. Noise, on the other hand, introduces random variations in intensity level throughout the whole image including the “edge areas”. This will certainly result in an increase of misclassification errors: the total number of actual edge pixels classified as non-edge points and the total number of actual non-edge pixels classified as edge points. Since all images are subjected to noise and other corruptions, the continuous lines of ideal edges are never encountered in practice. However, the results can be improved using suitable image enhancement techniques to increase the contrast against the background so that the edges become more visible.

2.4.1 Determining the Edge Points

An edge can be defined as a region in an image that consists of a connected set of pixels forming an arc. Such pixels are called edge points and are characterized by the property that their grey level values are significantly different than the grey level values of at least one of their nearest neighbors. In the edge detection process, one measures the intensity changes between neighboring pixels and classifies them as the edge points if warranted.

An edge is defined by a change in grey level. Hence, the neighborhood of an edge must exhibit a difference in the intensities. There are two characteristics of interest:

the slope and the direction of transition, which respectively provide the information on how significant the change is and if the possible edge is horizontal or vertical.

There are several methods to perform edge detection. The most popular techniques can be grouped into two categories based on the concepts of gradient and Laplacian. The gradient methods detect edges by looking for the extrema in the first derivative of the image. The Laplacian methods search for zero-crossings in the second derivative of the image. We illustrate these points in Figure 2.8.

Figure 2.8 Edge detection technique.

Let us consider, as a simple example, a differentiable one-dimensional intensity function $f(u)$ with $u \in \mathbb{R}$. If u_0 is the edge point then the first derivative $f'(u)$ will

have a local extremum at u_0 and the second derivative $f''(u)$ will cross the zero axis, that is change its sign, at u_0 .

In addition, $f'(u)$ can be used to estimate the strength and direction of the edge: a large value of $|f'(u)|$ would indicate a rapid change in intensity, $f'(u) > 0$ would imply that the intensity value has increased.

These techniques are illustrated in Figure 2.8. The artificial image consists of three rectangular areas of different intensity. There are two vertical edges to be detected, one at $u = 30$ and the other at $u = 60$. The “Intensity Across the Image” is a graph of the intensity versus the horizontal position. The graph of the first derivative shows the rate at which intensity changes with position. The spikes indicate the position of edges. The graph of the second derivative illustrates the rate at which the gradient changes with position. The zero-crossings indicate the position of edges.

Thus, the first step in one-dimensional edge detection process is to compute $f'(u)$ and $|f'(u)|$ from $f(u)$. Next, for a certain threshold t , it should be tested if $|f'(u_0)| > t$ if it is not true then u_0 is rejected to be an edge point. If in fact $|f'(u_0)| > t$, then u_0 is a candidate to be an edge point. The validation of the above condition is not sufficient to conclude that u_0 is an edge point due to the following problem: if all the values of u such that $|f'(u)| > 0$ were detected as edges, and edge would turn out to be a line rather than a point. To avoid this problem, it is required so that $|f'(u)|$ has a local maximum at u_0 . Only if this condition is satisfied u_0 is classified as an edge point. The choice of the threshold depends on the application and usually involves some trial and error.

Since we think of images as two-dimensional functions, intensity changes in more than one direction should be considered. Straightforward generalizations using the two orthogonal directions are the gradient

$$\nabla f(u, v) = \left(\frac{\partial f(u, v)}{\partial u}, \frac{\partial f(u, v)}{\partial v} \right)$$

and the second gradient

$$\nabla^2 f(u, v) = \left(\frac{\partial^2 f(u, v)}{\partial u^2}, \frac{\partial^2 f(u, v)}{\partial v^2} \right)$$

Clearly, the differentiation process cannot be performed in the usual way since a

digital image is not really a function of u and v but a matrix of integers. It seems reasonable to use differences over some local region to approximate the derivative at a pixel $x(n_1, n_2)$.

The simplest approximation is given by the following operator

$$\begin{aligned}\nabla_H x(n_1, n_2) &= x(n_1, n_2) - x(n_1 - 1, n_2) \\ \nabla_V x(n_1, n_2) &= x(n_1, n_2) - x(n_1, n_2 - 1)\end{aligned}$$

where $\nabla_H x(n_1, n_2)$ corresponds to $\frac{\partial f(u, v)}{\partial u}$: the differences in the horizontal (column) direction and similarly, $\nabla_V x(n_1, n_2)$ corresponds to $\frac{\partial f(u, v)}{\partial v}$: the differences in the vertical (row) direction.

In both cases, the strength, or edge response, at a given pixel is $\sqrt{\left(\frac{\partial f(u, v)}{\partial u}\right)^2 + \left(\frac{\partial f(u, v)}{\partial v}\right)^2}$ and the edge direction is $\tan^{-1} \left(\frac{\partial f(u, v)}{\partial u} \setminus \frac{\partial f(u, v)}{\partial v} \right)$.

2.4.2 Sobel Edge Detection

Most of the methods of edge detection are based on a convolution of the original image with a set of directional derivative masks. Edge detection operators examine each pixel neighborhood and quantify the slope, and the direction of the grey level transition. In this section, we consider the Sobel edge operator, which is one of the most commonly used gradient filters. It uses two spatial convolution kernels to calculate the first order intensity gradients for each pixel in the image: one gives the gradient for the horizontal direction (horizontal operator) and the other gives the gradient for the vertical direction (vertical operator).

The Sobel edge operator masks have the following values:

horizontal operator	vertical operator
$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

Hence, for a pixel at coordinates (i, j) , the horizontal operator is

$$\begin{aligned} x(i-1, j+1) + 2x(i, j+1) + x(i+1, j+1) - x(i-1, j-1) \\ - 2x(i, j-1) - x(i+1, j-1) \end{aligned}$$

and the vertical operator is

$$\begin{aligned} x(i+1, j+1) + 2x(i+1, j) + x(i+1, j-1) - x(i-1, j+1) \\ - 2x(i-1, j) - x(i-1, j-1) \end{aligned}$$

The procedure involves convolving the original image with the horizontal and vertical direction operators separately obtaining the horizontal and vertical gradients. Then combining these partial results to produce the final result: the total gradient magnitude at each pixel in the image. This is done using the following formula:

$$\text{total gradient magnitude} = \sqrt{(\text{horizontal gradient})^2 + (\text{vertical gradient})^2}$$

Usually, the gradients of images are displayed as binary edge contours. This is obtained by selecting a threshold for the total gradient magnitude and setting the pixel values in the gradient image to 255 if they are smaller than the threshold or to 0 otherwise. The value of the threshold is typically selected so that 5 to 10% of pixels with largest gradients are detected as edges.

There is no general technique to measure the performance of an edge detection procedure. If we had the knowledge of the actual number of edge points and their positions in the image, we could base the evaluation on the number of misclassified pixels. Another possible way is given by the quantity [Jain, 1986]:

$$\frac{\sum_{i=1}^{N_f} \left(\frac{1}{1+\alpha d(i)^2} \right)}{\max\{N_f, N_a\}}$$

where N_f is the number of edge points found by the scheme, N_a is the actual number of edge points in the image, $d(i)$ is the distance between a pixel found to be an edge and the nearest actual edge pixel, and α is a constant used for scaling.

An example of the Sobel edge detection is shown in Figure 2.9.

Figure 2.9 *Sobel edge detection.*

Chapter 3

Image Restoration

Image restoration is a process that reconstructs a degraded image. The restoration techniques assume a mathematical model for degradation and apply the inverse process, suggested by this model, to retrieve the original image. Corruptions in an image can be caused by defects of optical lenses, non-linearity of electro-optical sensor, graininess of the film material, relative motion between object and camera or wrong camera focus.

The potential use of image restoration is limitless. The application which probably has had the greatest impact on our lives is in the field of medicine, for example, breast cancer detection. There are also applications on home electronics, astronomy, biology, physics, agriculture, geography, defense, anthropology, and many other fields.

3.1 Model for Degraded Image

To ensure that images can be improved, a restoration scheme must incorporate some knowledge of the degradation process and possibly include other properties of the original and distorted images. This information must be summarized into appropriate models, which are now crucial to achieving good results.

The system used to model the degradation and restoration processes is illustrated in Figure 3.1. The original image $x(n_1, n_2)$ is blurred by a linear shift-invariant operator $b(n_1, n_2)$ and an image-independent noise $\eta(n_1, n_2)$ is added. A restoration

filter $h(n_1, n_2)$ is the inverse of the degradation process. At this stage some knowledge about the degradation is necessary. We then convolve the degraded image $y(n_1, n_2)$ with $h(n_1, n_2)$ to produce the restored image $\hat{x}(n_1, n_2)$.

Figure 3.1 System to model image degradation and restoration.

Typical image degradations include both blurring and noise. Let us first look at each of the processes separately and then combine them into one mathematical model.

An image degraded only by blurring can be modeled as a convolution of the original image with a point spread function (impulse response), b :

$$\begin{aligned}
 y(n_1, n_2) &= x(m_1, m_2) * b(m_1, m_2) \\
 &= \sum_{m_1} \sum_{m_2} x(m_1, m_2) b(n_1 - m_1, n_2 - m_2)
 \end{aligned}$$

The function b , also called the *blurring function*, is assumed to be shift-invariant. The mean filter, given in section 2.3, is an example of such function.

It is usually assumed that the noise process is additive, that is, the noise for each pixel is independent and identically distributed. An example of a common noise model is Gaussian white noise with mean 0 and variance σ_η^2 :

$$p_\eta(\nu(n_1, n_2)) = \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp \left\{ -\frac{(\nu(n_1, n_2))^2}{2\sigma_\eta^2} \right\}$$

An image corrupted by an additive random noise can be modeled as a sum of the original image and the noise η :

$$y(n_1, n_2) = x(m_1, m_2) + \eta(n_1, n_2)$$

The degradation process can now be summarized into the following equation

$$y(n_1, n_2) = x(n_1, n_2) * b(n_1, n_2) + \eta(n_1, n_2) \quad (3.1)$$

3.2 Image as a Random Field - Power Spectrum

In this section we introduce some statistical concepts related to image processing that are needed to define the power spectrum. These concepts will be further developed in the subsequent chapters.

Recall that the model for a degraded image is defined as a sum of the convolution of the original with a blurring function and a random noise. We note that, blurring, as a local operation, must create some local dependence of the pixel values. In addition, since noise is a random variable, each pixel value in the observed image must be itself a random variable. We then can think of such image as a sample from an ensemble of images and define it by a joint density function of all the random variables concerned. Equivalently, we consider an image as an instantiation of a random field.

Definition 3.1 *Random Field*

A random field on $\mathbb{N} = \{(n_1, n_2), 1 \leq n_i \leq N_i, i = 1, 2\}$ is a collection $X = \{X(n_1, n_2), (n_1, n_2) \in \mathbb{N}\}$ of random variables $X(n_1, n_2)$ with values $x(n_1, n_2) \in \Gamma$. A realization of X is a configuration $x = \{x(n_1, n_2), (n_1, n_2) \in \mathbb{N}\}$ such that $x \in \Omega$.

In the above definition, $\Omega := \Gamma^{N_1 N_2}$, and Γ is the set of all the grey level values in an image.

A random field X is completely characterized by the joint probability density p_X of all the random variables $X(n_1, n_2), (n_1, n_2) \in \mathbb{N}$.

For a particular $(n_1, n_2) \in \mathbb{N}$, $X(n_1, n_2)$ is a random variable with the expectation that depends on the position (n_1, n_2)

$$\mu_X(n_1, n_2) = \mathbb{E}[X(n_1, n_2)] = \sum_{x(n_1, n_2) \in \Gamma} x(n_1, n_2) p_X(x(n_1, n_2)) \quad (3.2)$$

where $p_X(x(n_1, n_2))$ denotes the distribution of $X(n_1, n_2)$.

Two random variables $X(n_1, n_2)$ and $X(m_1, m_2)$ with $(n_1, n_2), (m_1, m_2) \in \mathbb{N}$, that appear in the same random field X can be related by their *auto-correlation function*, R_X , which is the expectation of the product of the two random variables

$$\begin{aligned} R_X(n_1, n_2; m_1, m_2) &= \mathbb{E}[X(n_1, n_2)X(m_1, m_2)] \\ &= \sum_{x(n_1, n_2) \in \Gamma} \sum_{x(m_1, m_2) \in \Gamma} x(n_1, n_2)x(m_1, m_2)p_X(x(n_1, n_2), x(m_1, m_2)) \end{aligned} \quad (3.3)$$

The *auto-covariance function*, C_X , of a random field X is defined as

$$\begin{aligned} C_X(n_1, n_2; m_1, m_2) &= \mathbb{E}[(X(n_1, n_2) - \mathbb{E}[X(n_1, n_2)])(X(m_1, m_2) - \mathbb{E}[X(m_1, m_2)])] \\ &= \mathbb{E}[X(n_1, n_2)X(m_1, m_2)] - \mathbb{E}[X(n_1, n_2)]\mathbb{E}[X(m_1, m_2)] \\ &= R_X(n_1, n_2; m_1, m_2) - \mathbb{E}[X(n_1, n_2)]\mathbb{E}[X(m_1, m_2)] \end{aligned}$$

In most cases in this chapter it is assumed that a random field is “*zero-mean*” , which means

$$\mathbb{E}[X(n_1, n_2)] = 0, \quad \forall (n_1, n_2) \in \mathbb{N}$$

This implies

$$R_X(n_1, n_2; m_1, m_2) = C_X(n_1, n_2; m_1, m_2)$$

Another important property is homogeneity of a random field.

Definition 3.2 Homogenous Random Field

A random field X is said to be homogenous if the joint density of any partial sequence $\{x(n_1, n_2); 1 \leq n_1 \leq k_1, 1 \leq n_2 \leq k_2\}$ is the same as that of the shifted sequence $\{x(n_1 + \tau_1, n_2 + \tau_2); 1 \leq n_1 \leq k_1, 1 \leq n_2 \leq k_2\}$ for any integer vector (τ_1, τ_2) and any length vector (k_1, k_2) .

The implications of homogeneity are

- The expectation does not depend on the position in the space, that is $\mu_X(n_1, n_2) = \mu$ for all $(n_1, n_2) \in N$;
- The autocorrelation function is translation invariant, that is it depends only on the relative shifting of the positions at which the values of the random field are calculated :

$$R_X(\tau_1, \tau_2) = \mathbb{E}[X(n_1, n_2)X(n_1 + \tau_1, n_2 + \tau_2)]$$

Note that the autocorrelation $R_X(\tau_1, \tau_2)$ is equal to $R_X(-\tau_1, -\tau_2)$ as

$$\begin{aligned} R_X(\tau_1, \tau_2) &= \mathbb{E}[X(n_1, n_2)X(n_1 + \tau_1, n_2 + \tau_2)] \\ &= \mathbb{E}[X(n_1 - \tau_1, n_2 - \tau_2)X(n_1, n_2)] \\ &= \mathbb{E}[X(n_1, n_2)X(n_1 - \tau_1, n_2 - \tau_2)] = R_X(-\tau_1, -\tau_2) \end{aligned}$$

and for a zero-mean homogenous random field, $R_X(0, 0) = \sigma_X^2(n_1, n_2)$

Definition 3.3 *Power Spectrum*

The power spectrum of a homogenous random field $X(s)$ is the Fourier transform of the auto-correlation function $R_X(n_1, n_2)$:

$$P_X(\omega_1, \omega_2) = \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} R_X(n_1, n_2) e^{-i\omega_1 n_1} e^{-i\omega_2 n_2} \quad (3.4)$$

where $i := \sqrt{-1}$.

Some of the properties of the power spectrum are that $P_X(\omega_1, \omega_2)$ is equal to its conjugate $P_X^*(\omega_1, \omega_2)$ and that $P_X(\omega_1, \omega_2) = P_X(-\omega_1, -\omega_2)$. Furthermore, $P_X(\omega_1, \omega_2)$ is always real and $P_X(\omega_1, \omega_2) \geq 0$ for all (ω_1, ω_2) .

Let us now consider a special case of so-called *white noise* random field.

Definition 3.4 White Noise

If

$$\begin{aligned} R_X(n_1, n_2; m_1, m_2) &= \mathbb{E}[X(n_1, n_2)X(m_1, m_2)] \\ &= \begin{cases} \sigma_X^2(n_1, n_2), & (n_1, n_2) = (m_1, m_2) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

then the random field X is called *white noise*.

For a homogenous white noise, we have:

$$\begin{aligned} R_X(\tau_1, \tau_2) &= \mathbb{E}[X(n_1, n_2)X(n_1 + \tau_1, n_2 + \tau_2)] \\ &= \sigma^2 \delta(\tau_1, \tau_2) \end{aligned}$$

and the power spectrum is constant for all frequencies:

$$P_X(\omega_1, \omega_2) = \sigma^2 \quad \forall (\omega_1, \omega_2)$$

3.3 Inverse Filter

In this section we assume that the blurring function b is known and use the inverse filtering algorithm to restore the original image.

Thus, given a known image function $x(n_1, n_2)$, and a blurring function, $b(n_1, n_2)$, we want to recover $x(n_1, n_2)$ from the convolution:

$$y(n_1, n_2) = x(n_1, n_2) * b(n_1, n_2)$$

or equivalently in the frequency domain, from the product:

$$\mathcal{F}_y(\omega_1, \omega_2) = \mathcal{F}_x(\omega_1, \omega_2) \mathcal{F}_b(\omega_1, \omega_2)$$

The obvious solution would be to multiply the Fourier transform of the blurred function by the reciprocal of $\mathcal{F}_b(\omega_1, \omega_2)$:

$$\frac{\mathcal{F}_y(\omega_1, \omega_2)}{\mathcal{F}_b(\omega_1, \omega_2)} = \frac{\mathcal{F}_x(\omega_1, \omega_2) \mathcal{F}_b(\omega_1, \omega_2)}{\mathcal{F}_b(\omega_1, \omega_2)} = \mathcal{F}_x(\omega_1, \omega_2)$$

This is commonly referred to as the inverse filtering method where the frequency-domain inverse filter $\mathcal{F}_h$ is given by

$$\mathcal{F}_h(\omega_1, \omega_2) = \frac{1}{\mathcal{F}_b(\omega_1, \omega_2)} \quad (3.5)$$

We note that $\mathcal{F}_h(\omega_1, \omega_2)$ does not exist if $\mathcal{F}_b(\omega_1, \omega_2)$ has any zeros.

In case if there are only few values of $\mathcal{F}_b(\omega_1, \omega_2)$ which are zero (or close to zero), the ideal inverse filter can be approximated with a stabilized version of the inverse filter

$$\mathcal{F}_h^{st}(\omega_1, \omega_2) = \begin{cases} \frac{1}{\mathcal{F}_b(\omega_1, \omega_2)} & \text{if } \mathcal{F}_b(\omega_1, \omega_2) \neq 0 \text{ (or } |\mathcal{F}_b(\omega_1, \omega_2)| > \text{threshold}) \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

This works well if few elements of $\mathcal{F}_b$ have a magnitude below the threshold, but if too many elements are lost, the frequency content of resulting $\mathcal{F}_x^{st}$ will be much lower than that of $\mathcal{F}_x$ and the image will appear distorted.

Another problem with inverse filtering is that it does not perform well when applied to a noisy image. Let us revise the observed image model by including the noise term:

$$y_\eta(n_1, n_2) = y(n_1, n_2) + \eta(n_1, n_2)$$

The corresponding 2-dimensional Fourier transform representation would be

$$\mathcal{F}_y^\eta(\omega_1, \omega_2) = \mathcal{F}_y(\omega_1, \omega_2) + \mathcal{F}_\eta(\omega_1, \omega_2)$$

The inverse filter output can be then written as

$$\widehat{\mathcal{F}}_x^\eta(\omega_1, \omega_2) = \frac{\mathcal{F}_y(\omega_1, \omega_2)}{\mathcal{F}_b(\omega_1, \omega_2)} + \frac{\mathcal{F}_\eta(\omega_1, \omega_2)}{\mathcal{F}_b(\omega_1, \omega_2)}$$

Note that even when $\mathcal{F}_\eta(\omega_1, \omega_2)$ is small, it is possible that $\frac{\mathcal{F}_\eta(\omega_1, \omega_2)}{\mathcal{F}_b(\omega_1, \omega_2)}$ will be large, hence the noise will be amplified in $\widehat{\mathcal{F}}_x^\eta$. In such instances, it may be necessary to use a different type of filter implementation.

3.4 Wiener Filter

Wiener filtering is a technique of restoring images with a presence of blur and noise. We assume that the noise is additive and uncorrelated with the signal.

The aim is to obtain an estimate $\hat{x}(n_1, n_2)$ of $x(n_1, n_2)$ from $h(n_1, n_2)$ such that the mean square error

$$MSE := \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (x(n_1, n_2) - \hat{x}(n_1, n_2))^2 \quad (3.7)$$

is minimized.

The resulting Wiener filter in the Fourier domain can be expressed as

$$\mathcal{F}_h(\omega_1, \omega_2) = \frac{\mathcal{F}_b^*(\omega_1, \omega_2) P_x(\omega_1, \omega_2)}{|\mathcal{F}_b(\omega_1, \omega_2)|^2 P_x(\omega_1, \omega_2) + P_\eta(\omega_1, \omega_2)} \quad (3.8)$$

or

$$\mathcal{F}_h(\omega_1, \omega_2) = \frac{\mathcal{F}_b^*(\omega_1, \omega_2)}{|\mathcal{F}_b(\omega_1, \omega_2)|^2 + \frac{P_\eta(\omega_1, \omega_2)}{P_x(\omega_1, \omega_2)}} \quad (3.9)$$

where P_x and P_η represent the power spectra of the original image and of noise respectively, and $\mathcal{F}_b$ is a blurring matrix.

In the real world, $\mathcal{F}_b$ is not known precisely. Engineers can guess at the blurring function for a given circumstance, but determination of a good blurring function requires a lot of trial and error.

Figure 3.2. illustrates the results of applying the Wiener filter to an image degraded with blur $\mathcal{F}_b$ and white Gaussian noise.

In the absence of blur ($\mathcal{F}_b = 1$), the Wiener filter reduces to:

$$\mathcal{F}_b(\omega_1, \omega_2) = \frac{P_x(\omega_1, \omega_2)}{P_x(\omega_1, \omega_2) + P_\eta(\omega_1, \omega_2)} \quad (3.10)$$

which we call a *Wiener smoothing* filter.

Figure 3.2 *Wiener filter.*

3.5 Power Spectrum Approximation

Let us consider the following generalization of the previously discussed filters given by

$$\mathcal{F}_h(u, v) = \left[\frac{\mathcal{F}_b^*(\omega_1, \omega_2)}{|\mathcal{F}_b(\omega_1, \omega_2)|^2} \right]^\alpha \left[\frac{\mathcal{F}_b^*(\omega_1, \omega_2)}{|\mathcal{F}_b(\omega_1, \omega_2)|^2 + \beta \frac{P_n(\omega_1, \omega_2)}{P_x(\omega_1, \omega_2)}} \right]^{1-\alpha} \quad (3.11)$$

where α and β are positive constants. By varying α between 0 and 1 we may emphasize the relative effect of each filter. If $\alpha = 1$, Equation (3.11) reduces to inverse filter, and when $\alpha = 1/2$ and $\beta = 1$ it reduces to the so-called *Power Spectrum Equalization* (PSE) filter. In addition, if $\alpha = 1/2$, Equation (3.11) will define a filter that is a geometric mean between the inverse and Wiener filters. In general, the selection of β decides on the amount of Wiener smoothing.

In order to implement such filters in practice for $\alpha \neq 1$, the power spectra of the original image and the additive noise have to be known or approximated.

Here, the partial knowledge of the original image and the noise function becomes essential to the problem. For white additive noise, the power spectrum is equal to the variance of the noise. If the image and noise are uncorrelated, then the noise variance is the difference of the observed and original component variances.

Many methods can be applied to estimate the power spectrum of the original image. A good bet is to assume that the input image power spectrum is a reasonable estimate of the undegraded image power spectrum.

The observed image power spectrum may be written in terms of the original image, the blurring function and the noise, as:

$$P_x(\omega_1, \omega_2) = |\mathcal{F}_b(\omega_1, \omega_2)|^2 P_y(\omega_1, \omega_2) + P_\eta(\omega_1, \omega_2)$$

After subtracting the noise contribution, $P_x(\omega_1, \omega_2)$ may be estimated from the observed $P_y(\omega_1, \omega_2)$. Another possibility is to use the periodogram estimate, given by

$$\hat{P}_x = \frac{1}{N_1 N_2} [\mathcal{F}_y(\omega_1, \omega_2) \mathcal{F}_y^*(\omega_1, \omega_2)]$$

or estimate the power spectrum by

$$\hat{P}_x = \frac{P_y - P_\eta}{|\mathcal{F}_b|^2}$$

Since, frequently in practice, reliable information for the power spectra is not available, instead of searching for good estimators, the following substitute to the Wiener filter can be used

$$\mathcal{F}_h(\omega_1, \omega_2) = \frac{1}{\mathcal{F}_b(\omega_1, \omega_2)} \left[\frac{|\mathcal{F}_b(\omega_1, \omega_2)|^2}{|\mathcal{F}_b(\omega_1, \omega_2)|^2 + K} \right] \quad (3.12)$$

where K is a constant chosen, usually by trial and error, to visually optimize the result.

Chapter 4

Markov Random Fields

A digital monochrome image is simply a large matrix of integers representing the intensities or grey level values of the pixels at various sites. Usually, there are 256 possible shades of grey, and at least 256×256 many sites.

In Chapter 2, we introduced the concept of image enhancement. The goal was to improve the overall appearance of the available image based on some subjective criteria. Chapter 3 presented a different approach. There, we thought what the original image could be and attempted to recover it assuming a specified random model.

Hence, we elaborate on a statistical model for images based on the theory of Markov random fields.

4.1 Random Fields as Models for Images

We begin by introducing the theoretical framework for statistical image processing.

Let S be a finite set of sites. For every site s in S , $X(s)$ is a discrete random variable taking values in a finite *set of grey levels* denoted by Γ . A collection $X = \{X(s), s \in S\}$ of the random variables $X(s)$ is called a *random field* on S with values in Γ . The random field X can be also considered as a random variable taking its values in the configuration space $\Omega = \Gamma^{|S|}$. A realization of X is a *configuration* $x = \{x(s), s \in S\}$

in Ω with *grey level values* $x(s) \in \Gamma$. It is assumed that $\mathbb{P}[X = x] > 0$ for all $x \in \Omega$.

Suppose that A is a subset of S . Then the set of grey levels in a given configuration x at sites belonging to A will be denoted by $x(A)$. That is, $x(A) := \{x(s), s \in A\}$. If we let $S \setminus A$ to represent the complement of A in S then $x = \{x(A), x(S \setminus A)\}$, which is just another way of writing x . Particularly, for a fixed site $s \in S$ one has $x = \{x(s), x(S \setminus s)\}$.

We also let p_X^θ denote the probability distribution of X which depends on some parameter vector θ and belongs to the family: $\{p_X^\theta, \theta \in \Theta\}$.

The definitions below relate to specific subsets of sites in S , which are of great importance in the theory of random fields.

Definition 4.1 Neighborhood System

A neighborhood system on S is a family $\mathcal{N} = \{\mathcal{N}_s, s \in S\}$ of subsets of S such that $\forall s \in S$:

1. $s \notin \mathcal{N}_s$;
2. $t \in \mathcal{N}_s \iff s \in \mathcal{N}_t$.

Examples of neighborhood systems are shown in Figure 4.1. Suppose the site s is in the row n_1 and column n_2 , then the so-called 4-point neighborhood of $s = (n_1, n_2)$ consists of 4 sites: $(n_1 - 1, n_2)$, $(n_1 + 1, n_2)$, $(n_1, n_2 - 1)$ and $(n_1, n_2 + 1)$. The 8-point neighborhood of s in addition contains sites $(n_1 + 1, n_2 - 1)$, $(n_1 + 1, n_2 + 1)$, $(n_1 - 1, n_2 - 1)$ and $(n_1 - 1, n_2 + 1)$.

Figure 4.1 Example of a 4 and 8-point neighborhoods

The sites $t \in \mathcal{N}_s$ are called neighbors of s . The two-element set $\{s, t\}$ such that t is a neighbor of s will be denoted as $\langle s, t \rangle$.

Definition 4.2 Clique

A clique is a set of sites $C \subset S$ which are all neighbors of each other, that is

$$\forall s, r \in C, \quad r \in \mathcal{N}_s$$

The set of all cliques in S will be represented by $\mathcal{C}$.

Figure 4.2 shows cliques for the 4 and 8-point neighborhood systems. Note that any singleton $\{s\}$ is a clique. The last example is not a clique as it contains sites which are not all neighbors of each other.

Figure 4.2 Example of cliques for the 4-point neighborhood (only cliques (1)-(3)) and 8-point neighborhood (cliques (1)-(10)).

4.2 Markov Random Fields and Gibbs Distributions

The Markov property for random fields states that the probability of a certain value at site $s \in S$, given the values at all the other sites, depends only on the neighbors of s . This local dependence is formalized in the following two definitions.

Definition 4.3 *Markov Random Field (MRF)*

X is a Markov Random Field with respect to the neighborhood system $\mathcal{N}$ on S if and only if

1. $\mathbb{P}[X = x] > 0, \forall x \in \Omega$;
2. $\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \mathbb{P}[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] \quad \forall s \in S$ and $x \in \Omega$.

Definition 4.4 *Local Characteristic*

Conditional probability of the form

$$\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)]$$

is called local characteristic at site s .

In case of a Markov random field, the local characteristic at site s is given by

$$\mathbb{P}[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)]$$

It can be shown that a random field is uniquely determined by the local characteristics for all $s \in S$.

We now proceed to describe Gibbs distribution, which will be used to represent random fields. The idea originated in statistical mechanics and was introduced by Gibbs in 1902. A relevant feature of this model is that it concerns equilibrium states of large physical systems with energy expressed in terms of potential functions based on local interactions.

Consider the following distribution function

$$p_X^T(x) = \frac{1}{Z(T)} \exp \left\{ -\frac{1}{T} U(x) \right\}, \quad x \in \Omega$$

where the so-called temperature parameter T is positive, the normalizing constant $Z(T) = \sum_{y \in \Omega} \exp \left\{ -\frac{1}{T} U(y) \right\}$ and $U(x)$ is the energy function of the configuration x . Note that $p_X^T(x) > 0$ for all $x \in \Omega$.

Definition 4.5 Gibbs Potential

Gibbs potential is a collection $\{V_C\}_{C \in \mathcal{C}}$ of maps $V_C : \Omega \rightarrow \mathbb{R} \cup \{+\infty\}$ such that

1. $V_C(x) := 0$ if C is not a clique;
2. For all configurations x_1 and x_2 in Ω and for all cliques $C \in \mathcal{C}$ one has

$$(x_1(C) = x_2(C)) \implies (V_C(x_1) = V_C(x_2))$$

where $V_C(x) := V_C(x(C))$.

The energy function $U(x)$, $U : \Omega \rightarrow \mathbb{R}$ is said to be derived from the potential $\{V_C\}_{C \in \mathcal{C}}$ if

$$U(x) = \sum_{C \in \mathcal{C}} V_C(x)$$

Definition 4.6 Gibbs Distribution

A probability distribution, p_X^T , is called Gibbs if

$$p_X^T(x) = \frac{1}{Z(T)} \exp \left\{ -\frac{1}{T} \sum_{C \in \mathcal{C}} V_C(x) \right\}, \quad x \in \Omega$$

where $V_C(x) := V_C(x(C))$ is the Gibbs potential.

For most cases in this thesis, the temperature parameter T will be set to 1.

A Gibbs distribution with a potential corresponding to a certain neighborhood system is a distribution of a Markov random field with respect to the same neighborhood system. Conversely, a Markov random field can be represented by a Gibbs distribution for an appropriate potential. Hence, the Markov random field X can be modeled in terms of a Gibbs distribution.

These important results are summarized in the following theorem ¹.

Theorem 4.1 (The Hammersley-Clifford Theorem)

Consider the neighborhood system $\mathcal{N}$ on S and assume that for all configurations x , $\mathbb{P}[X = x] > 0$. Then

$$\begin{aligned} &X \text{ is a MRF with respect to } \mathcal{N} \\ &\text{if and only if} \\ &\mathbb{P}[X = x] \text{ is a Gibbs distribution with respect to } \mathcal{N} \end{aligned}$$

Proof of The Hammersley-Clifford Theorem.

Part 1

First we prove the statement: If X is a random field with the Gibbs distribution function p_X relative to a neighborhood system $\mathcal{N}$, then X is a Markov random field relative to the same $\mathcal{N}$.

Using the definition of conditional probability,

$$\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \frac{\mathbb{P}[X(s) = x(s), X(S \setminus s) = x(S \setminus s)]}{\mathbb{P}[X(S \setminus s) = x(S \setminus s)]}$$

The numerator is simply $\mathbb{P}[X = x]$ (recall the notation in Section 4.1). By the law of total probability, summing over the set of all possible grey values, the denominator can be written as $\sum_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma, X(S \setminus s) = x(S \setminus s)]$.

Hence we obtain

$$\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \frac{\mathbb{P}[X = x]}{\sum_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma, X(S \setminus s) = x(S \setminus s)]} \quad (4.1)$$

¹ Various versions of the statement and proof of the Hammersley-Clifford theorem can be found in: [Bremaud, 1999], [Geman, Geman, 1994], [Geman, 1980], [Kinderman, Snell, 1980] and [Winkler, 1995].

Note that the set $\mathcal{C}$ consists of cliques that either contain site s or not. That is,

$$\mathcal{C} = \{C \in \mathcal{C} : C \cap \{s\} \neq \emptyset\} \cup \{C \in \mathcal{C} : C \cap \{s\} = \emptyset\}$$

or, equivalently but in shorter notation

$$\mathcal{C} = \{\mathcal{C}_s\} \cup \{\mathcal{C}_{S \setminus s}\}$$

Since, by assumption, X is a Gibbs random field, the numerator of Equation (4.1) becomes:

$$\begin{aligned} \mathbb{P}[X = x] &= \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}} V_C(x) \right\} \\ &= \frac{1}{Z} \exp \left\{ - \left(\sum_{C \in \mathcal{C}_s} V_C(x) + \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(x) \right) \right\} \\ &= \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(x) - \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(x) \right\} \end{aligned}$$

Similarly, for a fixed configuration x with γ being the grey value at site s , the denominator of Equation (4.1) is:

$$\begin{aligned} \mathbb{P}[X(s) = \gamma, X(S \setminus s) = x(S \setminus s)] &= \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}} V_C(\gamma, x(S \setminus s)) \right\} \\ &= \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(\gamma, x(S \setminus s)) - \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(\gamma, x(S \setminus s)) \right\} \end{aligned}$$

But if a clique C does not contain the site s then the potential V_C is independent of the value at site s . That is $V_C(\gamma, x(S \setminus s)) = V_C(x)$ and consequently,

$$\begin{aligned} \mathbb{P}[X(s) = \gamma, X(S \setminus s) = x(S \setminus s)] &= \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(\gamma, x(S \setminus s)) \right\} \exp \left\{ - \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(x) \right\} \end{aligned}$$

Hence, Equation(4.1) becomes

$$\begin{aligned} \mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] &= \frac{\frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(x) \right\} \exp \left\{ - \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(x) \right\}}{\sum_{\gamma \in \Gamma} \frac{1}{Z} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(\gamma, x(S \setminus s)) \right\} \exp \left\{ - \sum_{C \in \mathcal{C}_{S \setminus s}} V_C(x) \right\}} \end{aligned}$$

This reduces to

$$\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \frac{\exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(x) \right\}}{\sum_{\gamma \in \Gamma} \exp \left\{ - \sum_{C \in \mathcal{C}_s} V_C(\gamma, x(S \setminus s)) \right\}} \quad (4.2)$$

We recall that for a clique $C \in \mathcal{C}$ and any two sites $t, s \in S$, if $t \in C$ and $s \in C$, then either $t = s$ or $t \in \mathcal{N}_s$. For this reason we can say that the potential V_C depends now on x only through $x(s)$ and $x(\mathcal{N}_s)$. Hence $\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \mathbb{P}[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)]$ and X is a Markov random field. This proves the first part of the theorem.

Part 2.

Conversely, we suppose that X is a Markov random field relative to a neighborhood system $\mathcal{N}$. We want to prove, this implies that X follows the Gibbs distribution function relative to the same $\mathcal{N}$. The proof is based the Mobius formula ², which is stated next.

Lemma 4.2 (Mobius Formula)

Let Φ and Ψ be two functions defined on the collection of subsets of a finite set E . Then

$$\Phi(A) = \sum_{B \subset A} (-1)^{|A-B|} \Psi(B), \quad \forall A \subset E \quad (4.3)$$

if and only if

$$\Psi(A) = \sum_{B \subset A} \Phi(B), \quad \forall A \subset E \quad (4.4)$$

where $|A|$ denotes the number of elements in the set A .

◇

²Proof of the Mobius formula can be found in [Bremaund, 1999] and [Winkler, 1995].

Suppose 0 is a fixed element of the grey set Γ . We let O denote the configuration with all phases equal to 0, that is, $O = \{x(s) : x(s) = 0, s \in S\}$, and assume $\mathbb{P}[X = O] > 0$.

Suppose x is a fixed configuration and A is a subset of S . We let x^A represent a configuration of Ω that agrees with x on A and has phase 0 outside A . That is, $x^A = \{x(A), O(S \setminus A)\}$ or for each site s in S

$$x^A(s) = \begin{cases} x(s) & , \text{ if } s \in A \\ 0 & , \text{ if } s \notin A \end{cases}$$

Let us now define the following potential for the subset $A \subset S$ and configuration $x \in \Omega$

$$V_A(x) = \sum_{B \subset A} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^B]} \quad (4.5)$$

Letting $\Phi(A) = V_A(x)$ and $\Psi(B) = \ln \frac{\mathbb{P}[X=O]}{\mathbb{P}[X=x^B]}$ in Equation (4.3) of the Mobius Formula we obtain

$$\ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^A]} = \sum_{B \subset A} V_B(x)$$

Set $A = S$ and $B = A$, then the above equality becomes

$$\ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x]} = \sum_{B \subset S} V_B(x)$$

This can be rewritten as

$$\mathbb{P}[X = x] = \mathbb{P}[X = O] \exp \left\{ - \sum_{B \subset S} V_B(x) \right\}$$

Hence if V_A is a Gibbs potential, then X is a Gibbs random field.

Based on the definition of Gibbs potential, we need to prove then that V_A depends only on the values in A , and that $V_A := 0$ if A is not a clique in the neighborhood system $\mathcal{N}$ on S .

Suppose there are two configurations $x, y \in \Omega$ such that $x(A) = y(A)$. Then for any $B \subset A$, $x^B = y^B$. Using Equation (4.5) we have

$$\begin{aligned} V_A(x) &= \sum_{B \subset A} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^B]} \\ &= \sum_{B \subset A} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = y^B]} \\ &= V_A(y) \end{aligned}$$

Since $V_A(x)$ equals $V_A(y)$, we conclude that the potential V_A is independent of the sites outside A in S .

For an arbitrary site $t \in A$, Equation (4.3) can be rewritten as

$$\begin{aligned} V_A(x) &= \sum_{B \subset A, B \ni t} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^B]} + \sum_{B \subset A, B \not\ni t} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^B]} \\ &= \sum_{B \subset A-t} (-1)^{|A-B|} \left(\ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^B]} - \ln \frac{\mathbb{P}[X = O]}{\mathbb{P}[X = x^{B+t}]} \right) \end{aligned}$$

That is,

$$V_A(x) = \sum_{B \subset A-t} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X = x^{B+t}]}{\mathbb{P}[X = x^B]} \quad (4.6)$$

Now, if t is not in $B \subset A$, then

$$\frac{\mathbb{P}[X = x^{B+t}]}{\mathbb{P}[X = x^B]} = \frac{\mathbb{P}[X(t) = x^{B+t}(t) | X(\mathcal{N}_t) = x^{B+t}(\mathcal{N}_t)]}{\mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)]}$$

Hence

$$V_A(x) = \sum_{BCA-t} (-1)^{|A-B|} \ln \frac{\mathbb{P}[X(t) = x^{B+t}(t) | X(\mathcal{N}_t) = x^{B+t}(\mathcal{N}_t)]}{\mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)]}$$

Using similar calculations to those that led to (4.6):

$$V_A(x) = \sum_{BCA} (-1)^{|A-B|} \ln \mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)]$$

Note that if A is not a clique then there exists a site $s \in A$ such that s is not a neighbor of t for $t \in A$.

Let us then fix such an s , and split the above sum as follows:

$$\begin{aligned} V_A(x) &= \sum_{BCA-s-t} (-1)^{|A-B|} \ln \mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)] \\ &\quad - \sum_{BCA-s-t} (-1)^{|A-(B+t)|} \ln \mathbb{P}[X(t) = x^{B+t}(t) | X(\mathcal{N}_t) = x^{B+t}(\mathcal{N}_t)] \\ &\quad - \sum_{BCA-s-t} (-1)^{|A-(B+s)|} \ln \mathbb{P}[X(t) = x^{B+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)] \\ &\quad - \sum_{BCA-s-t} (-1)^{|A-(B+t+s)|} \ln \mathbb{P}[X(t) = x^{B+t+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)] \\ &= - \sum_{BCA-s-t} (-1)^{|A-B|} \times \\ &\quad \ln \frac{\mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)] \mathbb{P}[X(t) = x^{B+t+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)]}{\mathbb{P}[X(t) = x^{B+t}(t) | X(\mathcal{N}_t) = x^{B+t}(\mathcal{N}_t)] \mathbb{P}[X(t) = x^{B+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)]} \end{aligned}$$

But since $s \neq t$ and $s \notin \mathcal{N}_t$ we have

$$\mathbb{P}[X(t) = x^B(t) | X(\mathcal{N}_t) = x^B(\mathcal{N}_t)] = \mathbb{P}[X(t) = x^{B+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)]$$

and

$$\mathbb{P}[X(t) = x^{B+t+s}(t) | X(\mathcal{N}_t) = x(\mathcal{N}_t)] = \mathbb{P}[X(t) = x^{B+t}(t) | X(\mathcal{N}_t) = x^{B+t}(\mathcal{N}_t)]$$

Hence $V_A(x) = 0$.

This finishes the proof of part 2, and the Hammersley-Clifford theorem. ■

4.3 Standard Models for Images

In this section we introduce two standard Markov random field models with relatively simple structure that involves small number of locally interacting variables. For this reason, these models will be used to illustrate various concepts throughout this thesis.

4.3.1 Ising Model

The Ising model was developed by the physicist E. Ising in 1925 to explain theoretically the phase transition in ferromagnetic materials.

In this model, the two-element phase space $\Gamma = \{+1, -1\}$ relates to the possible directions of a dipole (spin “up” or “down”) and the neighborhood is the 4-point system as shown in Figure 4.1(1).

The probability distribution of the random field X is of the form

$$p_X(x) = \frac{1}{Z} \exp \{-U(x)\}$$

where $Z = \sum_y \exp \{-\frac{1}{kT}U(y)\}$ is the normalizing constant.

The energy function is given by

$$U(x) = -J \sum_{\langle s, t \rangle} x(s)x(t) - mH \sum_{s \in S} x(s)$$

where $x(s) \in \Gamma$ indicates the direction of a spin at position s and $\langle s, t \rangle$ denotes a 2-point clique $\{s, t\}$ with $t \in \mathcal{N}_s$.

Note that the energy function can be specified in terms of Gibbs potentials:

$$V_{\{s, t\}}(x) = -J \sum_{\langle s, t \rangle} x(s)x(t)$$

and

$$V_{\{s\}}(x) = -mH \sum_{s \in S} x(s)$$

The first potential $V_{\{s, t\}}(x)$ is a sum taken over all pairs $\langle s, t \rangle$ (each pair counted only once), and it represents the energy caused by interaction of the spins.

The constant J represents a property of the material. If $J > 0$ then spins tend to have the same direction defining a ferromagnet, on the other hand if $J < 0$ then we have an antiferromagnet.

The second potential, $V_{\{s\}}(x)$, is a sum taken over all points s in S , and it represents the effect of the external magnetic field. The positive constant m depends on the material and is minimal if all spins are parallel to the external field. Finally, H denotes the intensity of the external magnetic field.

For a more thorough discussion of the physical model see [Kindermann and Snell, 1980].

In this thesis, we consider the simplest case of the Ising model with no external field and the energy function given by

$$U(x) = -\beta \sum_{\langle s,t \rangle} x(s)x(t) \quad (4.7)$$

where $x(s)$ is a grey level value at site $s \in S$, $x(s) \in \Gamma$ and $\Gamma = \{+1, -1\}$.

The probability of a particular configuration $x \in \Omega$ is

$$\begin{aligned} \mathbb{P}[X = x] &= \frac{\exp \{-U(x)\}}{\sum_{y \in \Omega} \exp \{-U(y)\}} \\ &= \frac{\exp \left\{ \beta \sum_{\langle s,t \rangle} x(s)x(t) \right\}}{\sum_{y \in \Omega} \exp \left\{ \beta \sum_{\langle s,t \rangle} y(s)y(t) \right\}} \\ &= \frac{1}{Z} \exp \left\{ \beta \sum_{\langle s,t \rangle} x(s)x(t) \right\} \end{aligned}$$

and for a site $s \in S$ we have

$$\begin{aligned} \mathbb{P}[X(s) = x(s)] &= \frac{\exp \left\{ \beta \sum_{t \in \mathcal{N}_s} x(s)x(t) \right\}}{\sum_{y \in \Omega} \exp \left\{ \beta \sum_{t \in \mathcal{N}_s} y(s)y(t) \right\}} \\ &= \frac{1}{Z_s} \exp \left\{ \beta \sum_{t \in \mathcal{N}_s} x(s)x(t) \right\} \end{aligned}$$

with the normalizing constant Z_s .

Let us now derive the local characteristic $\mathbb{P}[X(s) = g(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)]$.

Note that since X follows Gibbs distribution, X is also a Markov random field and the Markov property applies.

Hence,

$$\begin{aligned} \mathbb{P}[X(s) = x(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)] &= \mathbb{P}[X(s) = x(s)|X(S \setminus s) = x(S \setminus s)] \\ &= \frac{\mathbb{P}[X(s) = x(s), X(S \setminus s) = x(S \setminus s)]}{\mathbb{P}[X(S \setminus s) = x(S \setminus s)]} \\ &= \frac{\mathbb{P}[X = x]}{\mathbb{P}[X(S \setminus s) = x(S \setminus s)]} \\ &= \frac{\exp\{-U(x)\}}{\sum_{\gamma \in \Gamma} \exp\{-U(\gamma, x(S \setminus s))\}} \end{aligned}$$

Now, for the energy function $U(x) = -\beta \sum_{t \in \mathcal{N}_s} x(s)x(t)$ we have

$$\mathbb{P}[X(s) = x(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)] = \frac{\exp\{\beta \sum_{t \in \mathcal{N}_s} x(t)x(s)\}}{\sum_{\gamma \in \Gamma} \exp\{\beta \sum_{t \in \mathcal{N}_s} x(t)\gamma\}}$$

or

$$\begin{aligned} \mathbb{P}[X(s) = x(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)] &= \\ &= \frac{\exp\{x(s)\beta \sum_{t \in \mathcal{N}_s} x(t)\}}{\exp\{\beta \sum_{t \in \mathcal{N}_s} x(t)\} \exp\{-\beta \sum_{t \in \mathcal{N}_s} x(t)\}} \end{aligned} \tag{4.8}$$

since for fixed $s \in S$, $x(s) = 1$ or $x(s) = -1$.

In general, we can write a local characteristic in terms of the energy function as

$$\mathbb{P}[X(s) = x(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)] = \frac{\exp\{-U(x(s))\}}{\sum_{\gamma \in \Gamma} \exp\{-U(\gamma, x(\mathcal{N}_s))\}}$$

The clustering tendency of pixels of the same color is directly related to the value of the parameter β , called the “texture parameter”. Figure 4.3 shows samples from the Ising model with the energy function given by equation (4.7). We observe large uniformly-colored regions for high values of β .

Figure 4.3 *Samples from an Ising model for β values: 0.2, 0.5, 1.0, 1.5, 2 and 4.*

4.3.2 Potts Model

The Ising model is only useful in binary problems, otherwise the use of discrete variables with more than two states is required. The direct extension to a larger number of states is given by the *Potts model* with the energy function

$$U(x) = -\beta \sum_{\langle s,t \rangle} I_{\{x(s)=x(t)\}} \quad (4.9)$$

where the indicator function $I_{\{x(s)=x(t)\}}$ equals 1 if the grey values at sites s and t are the same and 0 if they are different.

The probability of a particular configuration $x \in \Omega$ using the Potts model is

$$\begin{aligned} \mathbb{P}[X = x] &= \frac{\exp\{-U(x)\}}{\sum_{y \in \Omega} \exp\{-U(y)\}} \\ &= \frac{\exp\left\{\beta \sum_{\langle s, t \rangle} I_{\{x(s)=x(t)\}}\right\}}{\sum_{y \in \Omega} \exp\left\{\beta \sum_{\langle s, t \rangle} I_{\{y(s)=y(t)\}}\right\}} \\ &= \frac{1}{Z} \exp\left\{\beta \sum_{\langle s, t \rangle} I_{\{x(s)=x(t)\}}\right\} \end{aligned}$$

and the local characteristic is

$$\begin{aligned} \mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] &= \frac{\exp\{-U(x)\}}{\sum_{\gamma \in \Gamma} \exp\{-U(\gamma, x(S \setminus s))\}} \\ &= \frac{\exp\left\{\beta \sum_{t \in N_s} I_{\{x(s)=x(t)\}}\right\}}{\sum_{\gamma \in \Gamma} \exp\left\{\beta \sum_{t \in N_s} I_{\{\gamma=x(t)\}}\right\}} \end{aligned} \quad (4.10)$$

Figure 4.4 shows sample configurations from the Potts model (four levels of grey) for different values of the texture parameter β .

Figure 4.4 Samples from a Potts model for β values: 0.2, 0.5, 1.0, 1.5, 2 and 4.

4.4 Parameter Estimation for Gibbs Distribution

When the described statistical models are to be put into practical use, one needs to estimate the unknown parameters. Several illustrative examples of how sensitive the models are to the different values of parameters were given in figures 4.3 and 4.4.

Thus, given a realization x of a Markov random field X , we assume it follows a Gibbs distribution from the family $\{p_X^\theta, \theta \in \Theta\}$ and want to estimate the true value of θ .

Recall that

$$p_X^\theta(x) = \frac{1}{Z_\theta} \exp\{-U_\theta(x)\}$$

where Z_θ is the normalizing constant and $U_\theta(x)$ is the energy function of the configuration x .

We will first discuss the standard method of the *maximum likelihood estimation (MLE)*, and then give its modification called the *maximum pseudo-likelihood estimation (MPLE)*.

The maximum likelihood estimate $\hat{\theta}$ is the value of θ in Θ that maximizes the probability of occurrence of x , or $\mathbb{P}(X = x)$. It is more convenient to consider the log-likelihood function instead

$$L(\theta) = \ln p_X^\theta(x)$$

and define $\hat{\theta}$ as

$$\hat{\theta} = \arg \max_{\theta \in \Theta} L(\theta) \tag{4.11}$$

The computational difficulty stems from the fact that the normalizing constant Z_θ cannot be described in closed form.

There exists however a modification of the MLE which is computationally tractable. It is based on sitewise maximizations of the conditional probabilities given the realization x of X .

The so-called *pseudo log likelihood* function is given by

$$PL(\theta) = \sum_{s \in S} \ln \mathbb{P}^\theta[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)]$$

and the corresponding *maximum pseudo-likelihood estimate (MPLE)* is

$$\hat{\theta} = \arg \max_{\theta \in \Theta} PL(\theta) \quad (4.12)$$

Note that the normalizing constants used by MPLE are now defined locally on $\mathcal{N}_s$, hence generally tractable.

We now consider a special case of the exponential family $\{p_X^\theta, \theta \in \Theta\}$.

Definition 4.7 Exponential Family

Consider a family $\{p_X^\theta(x) : \theta \in \Theta\}$ of probability density functions, where $\Theta = \{\theta : \gamma < \theta < \delta\}$, γ and δ are known constants.

A probability density function of the form

$$p_X^\theta(x) = a(x) \exp\{\Phi(\theta)t(x) + b(\theta)\}$$

is said to belong to the one-parameter exponential family of distributions if

1. The range of x is independent of θ ;
2. $\Phi(\theta)$ is a nontrivial continuous function of θ , $\gamma < \theta < \delta$;
3. $t(x)$ is a nontrivial continuous function of x on the range of x .

In the k -parameter case we have:

$$p_X^\theta(x) = a(x) \exp\left\{\sum_{i=1}^k \theta_i x_i + b(\theta)\right\} \quad (4.13)$$

where $x = (x_1, \dots, x_k)'$ and $\theta = (\theta_1, \dots, \theta_k)'$.

Further we assume that $U^\theta(x)$ depends linearly on the parameter θ , that is, there exists a vector $U^\theta = (U_1, \dots, U_d)$ such that $U^\theta(x) = -\sum_{k=1}^d \theta_k U_k(x)$.

Thus the distribution p_X^θ has the form:

$$p_X^\theta(x) = \frac{\exp\left\{\sum_{k=1}^d \theta_k U_k(x)\right\}}{\sum_{x \in \Omega} \exp\left\{\sum_{k=1}^d \theta_k U_k(x)\right\}} \quad (4.14)$$

Proposition 4.1 *Let the family $\{p_X^\theta, \theta \in \Theta\}$ be exponential, then the gradient of the likelihood function is given by*

$$\frac{\partial L(\theta)}{\partial \theta_i} = U_i(x) - \mathbb{E}_\theta[U_i(X)] \quad (4.15)$$

Proof. The likelihood function is

$$L(\theta) := \ln p_X^\theta(x) = \sum_k \theta_k U_k(x) - \ln \sum_{x \in \Omega} \exp \left\{ \sum_k \theta_k U_k(x) \right\}$$

and its derivative with respect to θ_i , $i = 1, \dots, d$ is:

$$\begin{aligned} \frac{\partial L(\theta)}{\partial \theta_i} &= U_i(x) - \left[\frac{\sum_{x \in \Omega} \exp \left\{ \sum_{k=1}^d \theta_k U_k(x) \right\} U_i(x)}{\sum_{x \in \Omega} \exp \left\{ \sum_{k=1}^d \theta_k U_k(x) \right\}} \right] \\ &= U_i(x) - \sum_{x \in \Omega} \left[\frac{\exp \left\{ \sum_{k=1}^d \theta_k U_k(x) \right\}}{\sum_{x \in \Omega} \exp \left\{ \sum_{k=1}^d \theta_k U_k(x) \right\}} U_i(x) \right] \\ &= U_i(x) - \sum_{x \in \Omega} U_i(x) p_X^\theta(x) \\ &= U_i(x) - E_\theta[U_i(X)] \end{aligned}$$

■

Proposition 4.2 *Let the family $\{p_X^\theta, \theta \in \Theta\}$ be exponential, then the gradient of the pseudo-likelihood function is given by:*

$$\frac{\partial PL(\theta)}{\partial \theta_i} = \sum_{s \in S} [U_i(x(s)) - \mathbb{E}_\theta[U_i(X(s), X(S \setminus s)) | x(S \setminus s)]] \quad (4.16)$$

Proof.

The local characteristic is given by:

$$\mathbb{P}_\theta[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] = \frac{\exp \{-U(x(s))\}}{\sum_{\gamma \in \Gamma} \exp \{-U(\gamma, x(S \setminus s))\}}$$

Hence, for the exponential family we have

$$\mathbb{P}_\theta[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] = \frac{\exp \left\{ \sum_{k=1}^d \theta_k U_k(x(s)) \right\}}{\sum_{\gamma \in \Gamma} \exp \left\{ \sum_{k=1}^d \theta_k U_k(\gamma, x(S \setminus s)) \right\}}$$

The pseudo-likelihood function is then

$$\begin{aligned} PL(\theta) &:= \sum_{s \in S} \ln \mathbb{P}_\theta[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] \\ &= \sum_{s \in S} \ln \frac{\exp \left\{ \sum_{k=1}^d \theta_k U_k(x(s)) \right\}}{\sum_{\gamma \in \Gamma} \exp \left\{ \sum_{k=1}^d \theta_k U_k(\gamma, x(S \setminus s)) \right\}} \\ &= \sum_{s \in S} \left[\sum_{k=1}^d \theta_k U_k(x(s)) - \ln \sum_{\gamma \in \Gamma} \exp \left\{ \sum_{k=1}^d \theta_k U_k(\gamma, x(S \setminus s)) \right\} \right] \end{aligned}$$

and its derivative with respect to θ_i , $i = 1, \dots, d$ is:

$$\begin{aligned} \frac{\partial PL(\theta)}{\partial \theta_i} &= \sum_{s \in S} \left[U_i(x(s)) - \left(\frac{\sum_{\gamma \in \Gamma} \exp \left\{ \sum_k \theta_k U_k(\gamma, x(S \setminus s)) \right\} U_i(\gamma, x(S \setminus s))}{\sum_{\gamma \in \Gamma} \exp \left\{ \sum_k \theta_k U_k(\gamma, x(S \setminus s)) \right\}} \right) \right] \\ &= \sum_{s \in S} \left[U_i(x(s)) - \sum_{\gamma \in \Gamma} U_i(\gamma, x(S \setminus s)) \mathbb{P}_\theta[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] \right] \\ &= \sum_{s \in S} [U_i(x(s)) - \mathbb{E}_\theta[U_i(X(s), X(S \setminus s)) | x(S \setminus s)]] \end{aligned}$$

■

Using Propositions 4.1 and 4.2, the MLE $\hat{\theta}_i$ is a root of the equation

$$U_i(x) - \mathbb{E}_\theta[U_i(X)] = 0$$

where the expectation term cannot be directly derived (it can however be approximated by employing one of the sampling algorithms described in next chapter).

On the other hand, the MPLE $\hat{\theta}_i$ is a root of the equation

$$\sum_{s \in S} [U_i(x(s)) - \mathbb{E}_{\theta}[U_i(X(s), X(S \setminus s)) | x(S \setminus s)]] = 0$$

where the conditional expectations can be computed directly, given that the size of the neighborhood system is small.

Chapter 5

Simulation of Random Fields

Suppose we want to obtain samples from a high dimensional Gibbs distribution. Since direct sampling is intractable, we can use a Markov chain procedure to simulate Markov random fields.

In this chapter, we will explain how the long-run realizations of a field-valued Markov chain are sampled from the Gibbs distribution. Several transition algorithms of the chain, known in the literature as simulation or sampling algorithms, will be presented. We will also describe the Markov chain Monte Carlo (MCMC) method for random fields.

5.1 Review of Markov Chain Theory

Consider a sequence of discrete random variables $\{X_n\}$, where $n = 0, 1, 2, \dots$ corresponds to discrete units of time. Let Γ denote the set of possible values that X_n can take. Such values are called the *states* of the system and Γ is the *state space*.

The sequence $\{X_n\}_{n \geq 0}$ is called a *Markov Chain* if it satisfies the Markov property. That is, if for all time points n and all states $i_0, i_1, \dots, i_n, j \in \Gamma$,

$$\mathbb{P}[X_{n+k} = j | X_0 = i_0, X_1 = i_1, \dots, X_n = i_n] = \mathbb{P}[X_{n+k} = j | X_n = i_n], \quad k \geq 1$$

The probability that X at time $n + 1$ will be in state j given that at time n it is in state i is called *one-step transition probability* and is denoted by $P^{n,n+1}(i, j)$. If the

transition probabilities do not depend on the actual value of the time index n , then the Markov Chain is said to be *stationary*, or *time homogenous*. $P^{n,n+1}(i, j)$ is then written as $P(i, j)$, that is

$$P(i, j) = \mathbb{P}[X_{n+1} = j | X_n = i], \quad n = 0, 1, 2, \dots$$

Suppose that the state space is finite, say $d = |\Gamma| < \infty$. Then the transition probabilities can be arranged in a $d \times d$ matrix P . The matrix $P = (P(i, j), i, j \in \Gamma)$ is stochastic and called the *transition probability matrix*. If P has the property that when raised to some power k , all of its elements are strictly positive, then P and the corresponding Markov Chain are called *regular*.

To describe a Markov chain, we need the joint probability of the random variables $X_n, n \geq 0$. But since

$$\mathbb{P}[X_0 = i_0, X_1 = i_1, \dots, X_n = i_n, X_{n+1} = j] = \mathbb{P}[X_0 = i_0]P(i_0, i_1)P(i_1, i_2)\dots P(i_n, j)$$

the Markov chain is completely specified by the one-step probability matrix and the probability distribution at initial time.

The essential part in the analysis of Markov Chains are the *n-step transition probabilities*, $P^{(n)}(i, j)$. The probability that the process moves from state i to state j in n units of time is

$$P^{(n)}(i, j) := \mathbb{P}[X_n = j | X_0 = i]$$

A state is called *aperiodic* if the probability of returning to it in n transitions is positive for all sufficiently large n . Analogously, if there exists an $n > 0$, such that

$$\mathbb{P}[X_n = i | X_0 = i] > 0 \quad \forall i \in \Gamma$$

then transition matrix P is said to be aperiodic. This means that the state space cannot be split into subsets that would be visited in a periodic fashion.

A chain $\{X_n\}_{n \geq 0}$, or a transition matrix P , is called *irreducible* if all states communicate with each other. That is, the probability of getting from one state to any

other within a finite number of transitions is always positive, or there exists $n > 0$, such that

$$\mathbb{P}[X_n = j | X_0 = i] > 0 \quad \forall i, j \in \Gamma$$

Let us now look at the long run behavior of Markov chains as $n \rightarrow \infty$.

Suppose $\{X_n\}$ is a regular Markov Chain, then

$$\lim_{n \rightarrow \infty} \mathbb{P}[X_n = j | X_0 = i] = \pi(j), \text{ for all } j \in \Gamma$$

or, in terms of the transition probability matrix

$$\lim_{n \rightarrow \infty} P^{(n)}(i, j) = \pi(j), \text{ for all } j \in \Gamma$$

That is, in the long run, the probability that the Markov chain will be in state j is approximately $\pi(j)$ with $\pi(j) > 0$ for all $j \in \Gamma$, and it does not depend on the initial state. The vector $\pi = [\pi(0), \pi(1), \dots, \pi(d)]$ with $\sum_{j=1}^d \pi(j) = 1$, forms the *limiting distribution* for the Markov chain $\{X_n\}$.

Another key concept in the study of Markov chains is that of the invariant distribution. If any distribution π is such that it satisfies

$$\sum_{i \in \Gamma} \pi(i) P(i, j) = \pi(j) \quad \forall j \in \Gamma$$

or in matrix notation

$$\pi P = \pi$$

then π is said to be *invariant* or *stationary* for P .

This implies that, if the marginal distribution at any step n is π (i.e., $X_n \sim \pi$) then the distribution at the next step $n + 1$ is $\pi P = \pi$ (i.e., $X_{n+1} \sim \pi$). Thus once the chain reaches the stage where π is the invariant distribution, this distribution is retained for all the subsequent stages.

Suppose, P and π are in *detailed balance*, that is

$$\pi(i)P(i, j) = \pi(j)P(j, i) \quad \forall i, j \in \Gamma$$

Then for every column of πP we have

$$(\pi P)_i = \sum_{j \in \Gamma} \pi(j)P(j, i) = \sum_{j \in \Gamma} \pi(i)P(i, j) = \pi(i) \sum_{j \in \Gamma} P(i, j) = \pi(i)$$

where the last steps follows from the facts that P is stochastic. This proves the following lemma.

Lemma 5.1 *If P and π are in detailed balance then π is invariant for P .*

Theorem 5.2 (Convergence to a steady state) *Let P be irreducible and aperiodic and suppose P has an invariant distribution π . Let λ be any distribution.*

Let $\{X_n\}_{n \geq 0}$ be a Markov chain with the transition matrix P and initial distribution λ .

Then

$$\lim_{n \rightarrow \infty} \mathbb{P}[X_n = j] = \pi(j) \quad \forall j \in \Gamma$$

In particular

$$\lim_{n \rightarrow \infty} \mathbb{P}[X_n = j | X_0 = i] = \pi(j) \quad \forall i, j \in \Gamma$$

That is, assuming irreducibility and aperiodicity, independently of the initial distribution, the probability distribution of X_n approaches π as $n \rightarrow \infty$. Hence, generating a Markov chain is equivalent to sampling from the distribution π .

5.2 From Markov Chains to Random Fields

The previous section outlined the Markov chain theory that is relevant to simulation. The crucial result was that the distribution of X_n approaches the invariant distribution π as $n \rightarrow \infty$. Thus, in the long run, generating a Markov chain is equivalent to

simulating π . This solves the problem of direct sampling from a high dimensional distribution. All that needs to be done is to design an irreducible and aperiodic Markov chain with transition probability matrix P for which π is invariant.

When processing images, direct sampling from Gibbs fields is not possible. The underlying configuration space Ω is too large: a typical image consists of 256×256 pixel sites, with one out of 256 possible grey values assigned to each site. This results in $256^{256^2} \simeq 10^{157,826}$ possible configurations of grey values. The approach via Markov chain seems to be a solution to this computational problem. Is it however appropriate for simulating images?

In the context of Markov chains, we considered X to be a random variable taking on values in the set of real numbers according to the transition function defined on $\Gamma \times \Gamma$.

In the context of random fields, X is no longer a random variable but a large matrix of random variables. A state corresponds then to a particular configuration and the transition probability matrix is now defined on $\Omega \times \Omega$. It specifies the probabilities of a random field X changing from one configuration to the other. Note that a transformation of just one random variable $X(s)$ for some site $s \in S$ implies a change in the configuration.

We index a family of random fields by n and consider it as a stochastic process $\{X_n\}_{n \geq 0}$ where $X_n = \{X_n(s), s \in S\}$ and $X_n(s) \in \Gamma$. The state at iteration n is a random field X_n on S with phases in Ω .

The Markov chain procedures can be then used on the configuration space Ω to allow successive sampling from random fields whose limiting distributions get closer and closer to the target Gibbs distribution. We now denote this distribution by $\pi(x)$. That is:

$$\pi(x) = \frac{1}{Z} \exp \{-U(x)\}, \quad x \in \Omega$$

where $Z = \sum_{y \in \Omega} \exp \{-U(y)\}$ is the normalizing constant and $U(x)$ is the energy function of the configuration x .

The sampling is accomplished through a design of appropriate transition probabilities $P(x, y)$ with $x, y \in \Omega$ such that the resulting Markov chain is irreducible,

aperiodic and it preserves $\pi(x)$ through the subsequent stages.

If we define the transition probabilities in terms of local characteristics of $\pi(x)$, then only a small number of neighbors need be considered. Hence, the time required for such computations will be reduced.

Let A be a subset of S . Consider the following probability distribution

$$P_A(x, y) = \begin{cases} \frac{1}{Z_A} \exp \{-U(y(A), x(S \setminus A))\} & \text{if } y(S \setminus A) = x(S \setminus A) \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

where

$$Z_A = \sum_{x(A)} \exp \{-U(x(A), x(S \setminus A))\}$$

Probabilities of this form will be called *local characteristics* on $A \subset S$.

Lemma 5.3 *The Gibbs distribution π and its corresponding local characteristics P_A satisfy the detailed balance equation. That is for all $x, y \in \Omega$ and $A \subset S$,*

$$\pi(x)P_A(x, y) = \pi(y)P_A(y, x) \quad (5.2)$$

Proof of Lemma 5.3.

If $y(S \setminus A) \neq x(S \setminus A)$ then both sides of Equation(5.2) cancel out.

If $y(S \setminus A) = x(S \setminus A)$ then $x = (x(A), y(S \setminus A))$ and we have

$$\begin{aligned} \pi(x)P_A(x, y) &= \pi(x) \left(\frac{1}{Z_A} \exp \{-U(y(A), x(S \setminus A))\} \right) \\ &= \frac{1}{Z} \exp \{-U(x)\} \frac{\exp \{-U(y(A), x(S \setminus A))\}}{\sum_{x(A)} \exp \{-U(x(A), x(S \setminus A))\}} \\ &= \frac{1}{Z} \exp \{-U(x(A), y(S \setminus A))\} \frac{\exp \{-U(y(A), y(S \setminus A))\}}{\sum_{x(A)} \exp \{-U(x(A), y(S \setminus A))\}} \\ &= \frac{1}{Z} \exp \{-U(x(A), y(S \setminus A))\} \frac{\exp \{-U(y)\}}{\sum_{x(A)} \exp \{-U(x(A), y(S \setminus A))\}} \\ &= \frac{1}{Z} \exp \{-U(y)\} \frac{\exp \{-U(x(A), y(S \setminus A))\}}{\sum_{x(A)} \exp \{-U(x(A), y(S \setminus A))\}} \\ &= \pi(y) \left(\frac{1}{Z_A} \exp \{-U(x(A), y(S \setminus A))\} \right) \\ &= \pi(y)P_A(y, x) \end{aligned}$$

■

Hence, by Lemma 5.3, a Gibbs distribution $\pi(x)$ is invariant for its local characteristics $P_A(x, y)$ where $A \subset S$.

Let us now consider an enumeration $S = \{s_1, s_2, \dots, s_d\}$ of S , where $d = |S|$. We call this a *visiting scheme*. Suppose that $A = \{s\}$ where s is a single site in S .

We can then construct a homogenous Markov chain which evolves as follows: an initial configuration x is chosen at random according to some distribution λ . In the first step, x is updated at site s_1 based on a single site characteristic $P_{\{s_1\}}(x, (\cdot, x_{S \setminus \{s_1\}}))$. This results in a new configuration $y = (y_{\{s_1\}}, x_{S \setminus \{s_1\}})$. The configuration y is then updated at site s_2 and so on until all the sites in S are updated. Only then the first cycle of the process is completed. The transition probability is thus a composition of single site local characteristics:

$$P(x, y) = P_{\{s_1\}} P_{\{s_2\}} \dots P_{\{s_d\}}(x, y)$$

It was shown that the Gibbs fields are invariant with respect to their local characteristics and hence also with respect to the composition P of local characteristics. We can also argue that P is aperiodic and irreducible as in each $s \in S$ the transition from $x(s)$ to $x(s)$ is always possible. We can then translate the convergence to a steady state theorem to the case of random fields and expect that after a large number of steps we will obtain a sample from a distribution close to $\pi(x)$.

5.3 Simulation Algorithms

The simulation algorithms provide a general construction for a Markov chain with transition probabilities $P(x, y)$ such that $\pi P = \pi$. The first two algorithms presented are due to Hasting and Metropolis. They both take the approach of simulating the entire random field. Their “sitewise” versions follow. Finally, at the end, the so-called Gibbs sampler is described. It is based on the site-by-site visiting scheme and is probably the most popular simulation algorithm in image applications.

Let $Q = \{q(i, j)\}_{i, j \in \Omega}$ be an arbitrary irreducible transition matrix on $\Omega \times \Omega$. The matrix Q is called a *candidate-generating matrix* and its $(i, j)^{th}$ entry $q(i, j)$

denotes the probability of choosing the next tentative configuration j when the present configuration is i .

Let $\alpha(i, j)$ be the probability of accepting j as the next state when the present state is i and $j \neq i$. If $j = i$ the acceptance probability is 1.

Then the probability of moving from configuration x to y is

$$\mathbb{P}[X_{n+1} = y | X_n = x] = q(x, y)\alpha(x, y) \quad \text{for } x, y \in \Omega, \quad y \neq x$$

and

$$\mathbb{P}[X_{n+1} = x | X_n = x] = 1 - \sum_{y \in \Omega} q(x, y)\alpha(x, y), \quad x, y \in \Omega, \quad y \neq x$$

where $n \geq 0$ is the time or iteration index.

It remains to design the acceptance probabilities $\alpha(x, y)$ as functions of π and $q(x, y)$.

5.3.1 Hasting Algorithm

Let

$$\alpha(x, y) = \frac{u(x, y)}{1 + t(x, y)}$$

where $U = \{u(x, y)\}_{x, y \in \Omega}$ is a symmetric positive matrix selected such that $\alpha(x, y) \in [0, 1]$ and

$$t(x, y) = \frac{\pi(x)q(x, y)}{\pi(y)q(y, x)}$$

Note that the algorithm requires knowledge of the target distribution π only up to the normalizing constant, which cancels out in the ratio $\pi(x)/\pi(y)$.

The condition $\alpha(x, y) \in [0, 1]$, implies

$$0 \leq \frac{u(x, y)}{1 + t(x, y)} \leq 1$$

This holds when

$$u(x, y) \leq 1 + t(x, y) \quad \text{and} \quad u(x, y) \leq 1 + t(y, x)$$

since for all x and y , $u(x, y) \geq 0$, and $u(x, y) = u(y, x)$.

The above two inequalities are equivalent with

$$u(x, y) \leq 1 + \min\{t(x, y), t(y, x)\}$$

If we let

$$u(x, y) = 1 + \min\{t(x, y), t(y, x)\}$$

then, since $t(x, y) \geq 0$:

$$\begin{aligned} \alpha(x, y) &= \frac{u(x, y)}{1 + t(x, y)} \\ &= \frac{\min\{1 + t(x, y), 1 + t(y, x)\}}{1 + t(x, y)} \\ &= \min \left\{ \frac{1 + t(x, y)}{1 + t(x, y)}, \frac{1 + t(y, x)}{1 + t(x, y)} \right\} \\ &= \min \left\{ \frac{1 + t(x, y)}{1 + t(x, y)}, \frac{1 + t(y, x)}{1 + t(x, y)} \right\} \\ &= \min \left\{ 1, \frac{1 + t(y, x)}{1 + t(x, y)} \right\} \end{aligned}$$

Finally, using the definition of $t(x, y)$, we obtain the acceptance probability for the Hasting algorithm :

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \right\}$$

Thus, the transition probability matrix $P = \{P(x, y)\}$ is defined by

$$\begin{aligned} P(x, y) &= \begin{cases} q(x, y)\alpha(x, y), & x \neq y \\ 1 - \sum_{y \neq x} P(x, y), & x = y \end{cases} \\ &= \begin{cases} \min \left\{ q(x, y), \frac{\pi(y)q(y, x)}{\pi(x)} \right\}, & x \neq y \\ 1 - \sum_{y \neq x} P(x, y), & x = y \end{cases} \end{aligned} \tag{5.3}$$

Let us now verify that the detailed balance equation holds.

For $x, y \in \Omega$, $y \neq x$, we have:

$$\begin{aligned}\pi(x)\mathbb{P}[X_{n+1} = y|X_n = x] &= \pi(x) \left(q(x, y) \left(\min \left\{ 1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \right\} \right) \right) \\ &= \min \{ \pi(x)q(x, y), \pi(y)q(y, x) \}\end{aligned}$$

Similarly,

$$\begin{aligned}\pi(y)\mathbb{P}[X_{n+1} = x|X_n = y] &= \pi(y) \left(q(y, x) \left(\min \left\{ 1, \frac{\pi(x)q(x, y)}{\pi(y)q(y, x)} \right\} \right) \right) \\ &= \min \{ \pi(y)q(y, x), \pi(x)q(x, y) \}\end{aligned}$$

That is $\pi(x)\mathbb{P}[X_{n+1} = y|X_n = x] = \pi(y)\mathbb{P}[X_{n+1} = x|X_n = y]$. Hence, the algorithm indeed generates a Markov chain for which π is the stationary distribution.

◇

The steps of the Hasting algorithm are:

1. Set the counter $n = 1$, and choose an arbitrary initial configuration x_0 ;
2. Generate a new value y from $q(x_{n-1}, \cdot)$;
3. Evaluate $\alpha(x_{n-1}, y)$;
4. Generate an independent uniform quantity u :
 - if $u \leq \alpha(x_{n-1}, y)$ accept the move, i.e. set $x_n = y$;
 - if $u > \alpha(x_{n-1}, y)$ reject the move, i.e. set $x_n = x_{n-1}$;
5. Set n to $n + 1$ and return to step 2 until convergence is reached.

When convergence is reached, the resulting configuration x_n is drawn from π .

5.3.2 Metropolis Algorithm

The Metropolis algorithm is a special case of Hasting when Q is a symmetric matrix. This causes the acceptance probability to be reduced to

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\}$$

The transition probability matrix $P = \{P(x, y)\}$ is then given by

$$\begin{aligned}
 P(x, y) &= \begin{cases} q(x, y)\alpha(x, y), & x \neq y \\ 1 - \sum_{y \neq x} P(x, y), & x = y \end{cases} \\
 &= \begin{cases} \min \left\{ q(x, y), q(x, y) \frac{\pi(y)}{\pi(x)} \right\}, & x \neq y \\ 1 - \sum_{y \neq x} P(x, y), & x = y \end{cases} \\
 &= \begin{cases} q(x, y) \frac{\pi(y)}{\pi(x)}, & \pi(y) < \pi(x) \text{ and } x \neq y \\ q(x, y), & \pi(y) \geq \pi(x) \text{ and } x \neq y \\ 1 - \sum_{y \neq x} P(x, y), & x = y \end{cases}
 \end{aligned} \tag{5.4}$$

5.3.3 Sitewise Hasting and Metropolis

The main idea of the sitewise approach is to assign candidate generating matrices Q_s to each individual $X(s)$, for all $s \in S$. The entry $q_s(x, y)$ denotes the probability at a site s of choosing the next tentative state y when the present state is x . Suppose x is the current configuration, then Q_s proposes a replacement $y(s)$ for the site s leaving the rest of the sites unchanged.

For a given site s , the acceptance probability for the sitewise Hasting algorithm is

$$\alpha_s(x, y) = \min \left\{ 1, \frac{\pi(y)q_s(y, x)}{\pi(x)q_s(x, y)} \right\}$$

and the probability of changing the grey level value at site s from $x(s)$ to $y(s)$ (that is, the probability of moving from configuration x to y at site s) is

$$\begin{aligned}
 \mathbb{P}[X_{n+1}(s) = y(s) | X_n(s) = x(s)] &= q_s(x, y)\alpha_s(x, y) \\
 &= \min \left\{ 1, \frac{\pi(y)q_s(y, x)}{\pi(x)q_s(x, y)} \right\} q_s(x, y)
 \end{aligned}$$

The sitewise Metropolis algorithm assumes $q_s(y, x) = q_s(x, y)$, which results in

$$\mathbb{P}[X_{n+1}(s) = y(s) | X_n(s) = x(s)] = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\} q_s(x, y)$$

Note that no rule has been given to decide on a current site $s \in S$. According to Geman and Geman (1984), the visiting scheme is not important in practice, as long

as each site gets visited a large number of times. The possibilities may be to visit each site sequentially in some order or to choose a site randomly at each step. Let us denote the probability of choosing site s by r_s (in the second case, $r_s = \frac{1}{|S|}$ for all $s \in S$), then $\{X_n\}$ is a Markov chain with transition probabilities multiplied by r_s . We can also assume that some sitewise visitation schedule has been fixed, such that for every site $s \in S$, $s_k = s$ infinitely often, where $\{s_k, k = 1, 2, 3, \dots\} \in S$ denotes the sequence in which the sites are visited for updating.

5.3.4 Gibbs Sampler

The Gibbs sampler is a very popular simulation scheme for image models that allows to build a Markov chain with a given limiting distribution. The scheme originated in statistical physics as a “heat-bath method” and was used to simulate a Gibbs distribution.

In the context of image processing, the distribution of interest is also Gibbs:

$$\pi(x) = \frac{1}{Z} \exp\left\{-\frac{U(x)}{T}\right\}$$

where $U(x) = \sum_{c \in C} V_c(x)$, $Z = \sum_x \exp\{-U(x)/T\}$ and T is so-called temperature parameter (set to 1 for most cases in this thesis).

Since the procedure was originally to sample from the local characteristics of a Gibbs field, the authors of the scheme (Geman and Geman, 1984) named it “Gibbs Sampler”. This might be misleading as in fact, the scheme can be applied to many other distributions.

Let us assume that the local characteristics $\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)]$ are completely known and can be sampled from. The transition from $X_n = x$ to $X_{n+1} = y$ is made by changing (or not) the value at only one site. If s is the chosen site at step (iteration) n and x is the realization of X_n , then the realization of X_{n+1} is obtained by replacing $x(s)$ by $y(s)$ in x .

The new phase at site s is selected with probability:

$$\mathbb{P}[X(s) = y(s) | X(S \setminus s) = x(S \setminus s)]$$

where $\mathbb{P}[X(s) = \cdot | X(S \setminus s) = x(S \setminus s)] = \mathbb{P}[X(s) = \cdot | X(\mathcal{N}_s) = x(\mathcal{N}_s)]$

Suppose the probability that s is the chosen site at time n is r_s (independently of the past) where $r_s > 0$ for all $s \in S$.

Then the probability of the transition from x into $y = (y(s), x(S \setminus s))$ is:

$$\mathbb{P}[X_{n+1} = y | X_n = x] = r_s \mathbb{P}[X(s) = y(s) | X(S \setminus s) = x(S \setminus s)] I_{\{y(S \setminus s) = x(S \setminus s)\}} \quad (5.5)$$

and the transition probability matrix $P = \{P(x, y)\}$ is defined as:

$$P(x, y) = \begin{cases} r_s \mathbb{P}[X(s) = y(s) | X(S \setminus s) = x(S \setminus s)] I_{\{y(S \setminus s) = x(S \setminus s)\}}, & x(s) \neq y(s) \\ 1 - \sum_{y \neq x} P(x, y), & x(s) = y(s) \end{cases} \quad (5.6)$$

It can be shown that such a designed chain is irreducible, aperiodic and that it preserves the distribution π .

In order to prove that π is the stationary distribution of the above Markov chain we verify the detailed balance equation.

The Gibbs algorithm states that for all $x, y \in \Omega$, $x(S \setminus s) = y(S \setminus s)$, therefore we can write:

$$\pi(y) = \pi(y(s), x(S \setminus s))$$

Multiplying both sides by the ratio $\frac{\pi(x)}{\pi(x(S \setminus s))}$ gives

$$\pi(y) \frac{\pi(x)}{\pi(x(S \setminus s))} = \pi(y(s), x(S \setminus s)) \frac{\pi(y)}{\pi(x(S \setminus s))}$$

Since $\pi(y) = \pi(y(s), y(S \setminus s))$ and $y(S \setminus s) = x(S \setminus s)$, we have

$$\pi(x) \frac{\pi(y(s), x(S \setminus s))}{\pi(x(S \setminus s))} = \pi(y) \frac{\pi(x(s), x(S \setminus s))}{\pi(x(S \setminus s))}$$

which, by the definition of conditional probability, reduces to:

$$\pi(x) \mathbb{P}[X(s) = y(s) | X(S \setminus s) = x(S \setminus s)] = \pi(y) \mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)]$$

Finally, multiplying both sides by the product $r_s I_{\{y(S \setminus s) = x(S \setminus s)\}}$ gives:

$$\begin{aligned} \pi(x) r_s \mathbb{P}[X(s) = y(s) | X(S \setminus s) = x(S \setminus s)] I_{\{y(S \setminus s) = x(S \setminus s)\}} \\ = \pi(y) r_s \mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] I_{\{y(S \setminus s) = x(S \setminus s)\}} \end{aligned}$$

Now, Equation 5.5 can be applied to both sides of the above resulting in

$$\pi(x)\mathbb{P}[X_{n+1} = y|X_n = x] = \pi(y)\mathbb{P}[X_{n+1} = x|X_n = y]$$

This verifies the detailed balance equation, and we conclude that π is the stationary distribution.

◇

The steps of the Gibbs Sampler algorithm are:

1. Set the counter $n = 1$, and an arbitrary initial configuration x_0 ;
2. Pick a new site s in S according to r_s ;
3. Generate $x_n(s)$ according to
$$X_n(s) \sim \mathbb{P}[X(s) = x(s)|X(S \setminus s) = x_{n-1}(S \setminus s)]$$
and form x_n ;
4. Set n to $n + 1$ and return to step 2 until convergence is reached.

When convergence is reached, the resulting configuration x_n is drawn from π .

5.3.5 Example: Ising Model

Let us consider the simple formulation of the Ising model, as described in Section 4.3.1, with the energy function:

$$U(x) = -\beta \sum_{\langle i,j \rangle} x(i)x(j)$$

where $\langle i, j \rangle$ denotes a pair of neighboring sites (cliques) in the 4-point neighborhood system $\mathcal{N}$.

Recall that

$$\begin{aligned}\pi(x) &= \frac{\exp\{-U(x)\}}{\sum_{x \in \Omega} \exp\{-U(x)\}} \\ &= \frac{\exp\left\{\beta \sum_{\langle i,j \rangle} x(i)x(j)\right\}}{\sum_{x \in \Omega} \exp\left\{\beta \sum_{\langle i,j \rangle} x(i)x(j)\right\}} \\ &= \frac{1}{Z} \exp\left\{\beta \sum_{\langle i,j \rangle} x(i)x(j)\right\}\end{aligned}$$

and for a particular site $s \in S$ we have:

$$\begin{aligned}\pi(x(s)) &= \frac{\exp\left\{\beta \sum_{t \in \mathcal{N}_s} x(s)x(t)\right\}}{\sum_{x \in \Omega} \exp\left\{\beta \sum_{t \in \mathcal{N}_s} x(s)x(t)\right\}} \\ &= \frac{1}{Z_s} \exp\left\{\beta x(s) \sum_{t \in \mathcal{N}_s} x(t)\right\}\end{aligned}$$

Finally, the local characteristics in the Ising model are:

$$\begin{aligned}\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] &= \mathbb{P}[X(s) = x(s) | X(\mathcal{N}_s) = x(\mathcal{N}_s)] \\ &= \frac{\exp\left\{\beta x(s) \sum_{t \in \mathcal{N}_s} x(t)\right\}}{\exp\left\{\beta \sum_{t \in \mathcal{N}_s} x(t)\right\} + \exp\left\{-\beta \sum_{t \in \mathcal{N}_s} x(t)\right\}}\end{aligned}$$

We first consider the Gibbs sampler.

If x_{n-1} is the current configuration and s is the chosen site at step n , then

$$X_n(s) \sim \mathbb{P}[X(s) = x(s) | X(S \setminus s) = x_{n-1}(S \setminus s)]$$

That is, $X(s)$ is updated according to the sampling of the associated local conditional distribution, and the probability that the value at site s will change is:

$$P[X_n(s) \neq x_{n-1}(s)] \propto \exp\left\{\beta x_n(s) \sum_{t \in \mathcal{N}_s} x_{n-1}(t)\right\}$$

(Note: $\mathbb{P}[X_n(s) = x_{n-1}(s)] = 1 - \mathbb{P}[X_n(s) \neq x_{n-1}(s)]$)

Hence

- if $x_{n-1}(s) = -1$ then $x_n(s) = 1$ with probability $\propto \exp\left\{\beta \sum_{t \in \mathcal{N}_s} x_{n-1}(t)\right\}$;

- if $x_{n-1}(s) = 1$ then $x_n(s) = -1$ with probability $\propto \exp \left\{ -\beta \sum_{t \in \mathcal{N}_s} x_{n-1}(t) \right\}$.

The sitewise Metropolis algorithm requires first to generate the new proposal value $y(s)$ from Q_s . Since there are only two possible values a random variable $X(s)$ can take, and the algorithm requires that $q_s(x, y) = q_s(y, x)$, the probability that $y(s)$ will be 1 or -1 are both equal to 0.5.

The acceptance probability is evaluated based on the sampling from π . We have:

$$\frac{\pi(y(s))}{\pi(x_{n-1}(s))} = \frac{\exp \left\{ \beta y(s) \sum_{t \in \mathcal{N}_s} x_{n-1}(t) \right\}}{\exp \left\{ \beta x_{n-1}(s) \sum_{t \in \mathcal{N}_s} x_{n-1}(t) \right\}}$$

Hence

- if $x_{n-1}(s) = -1$ then $x_n(s) = 1$ with probability $\exp \left\{ 2\beta \sum_{t \in \mathcal{N}_s} x_{n-1}(t) \right\}$;
- if $x_{n-1}(s) = 1$ then $x_n(s) = -1$ with probability $\exp \left\{ -2\beta \sum_{t \in \mathcal{N}_s} x_{n-1}(t) \right\}$.

◇

5.4 Markov Chain Monte Carlo (MCMC)

Suppose $\{p_X(x) : x \in \Omega\}$ is a probability density such that it is known only up to scale. That is, we can write

$$p_X(x) = \frac{h(x)}{c}, \quad x \in \Omega$$

where $h(x)$ is a completely specified function but the normalizing constant $c = \sum_{y \in \Omega} h(y)$ is not known in closed form (as, for example, the configuration space Ω might be too large to allow direct calculation of c).

The aim is to compute expectations of a given function f under p_X :

$$\mathbb{E}_p[f(X)] = \sum_{x \in \Omega} f(x) p_X(x)$$

when there is no closed form expression.

Note that the form in which the above equation is written allows us also to express the probability of any particular event concerning X . For example, suppose $A \subset \Omega$. Then with $f(x) = I_{\{x \in A\}}$

$$\begin{aligned} \mathbb{P}[X \in A] &= \sum_{x \in A} I_{\{x \in A\}} p_X(x) \\ &= \mathbb{E}_p[I_{\{x \in A\}}] \end{aligned}$$

The ordinary (or i.i.d.) Monte Carlo methods can be used when random samples from the distribution p are available. If a sequence of independent identically distributed random variables $X_1, \dots, X_n$ was obtainable, then, using the Strong Law of Large Numbers, we would have:

$$\mathbb{E}_p[f(X)] = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(X_i)$$

and the empirical average obtained from a finite sample $x_1, \dots, x_n$ generated from p_X could be used to estimate $\mathbb{E}_p[f(X)]$. That is, for large n , the estimate of the expectation is

$$\bar{f}_n = \frac{1}{n} \sum_{i=1}^n f(x_i)$$

In many applications, generation of the i.i.d. random variables X_i is not efficient or is even impossible as p is not known in closed form. This is the foundation for the Markov chain Monte Carlo (MCMC) method.

Theorem 5.4 (Ergodic Theorem) *Let P be irreducible, and f any bounded function $f : \Omega \rightarrow \mathbb{R}$. Then*

$$\mathbb{P} \left[\lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} f(X_i) \right) = \mathbb{E}_\pi(f(X)) \right] = 1$$

where $\mathbb{E}_\pi(f(X)) = \sum_{j \in \Omega} f(j) \pi(j)$ is the expectation of $f(X)$ with respect to π and $\{\pi(j) : j \in \Omega\}$ is the unique invariant distribution.

Suppose $\{X_n\}_{n \geq 0}$ is an irreducible and aperiodic homogenous Markov chain with state space Ω and stationary distribution π . Then the ergodic theorem implies that, for any seed X_0 , the random sequence $\overline{f}_n$ will converge almost surely to the expectation $\mathbb{E}_\pi(f(X))$ as $n \rightarrow \infty$.

The idea is the same as before. We construct a Markov chain with stationary distribution π using the simulation schemes introduced in the previous section. The simulation results are then used to approximate $\mathbb{E}_\pi(f(X))$ by the empirical average $\overline{f}_n$ where the index n refers to the simulation number.

Note that ideally, the first realization X_1 should follow the distribution of interest so that the balance equation would ensure that the marginal distribution of each subsequent X_n is the same. Instead an initial “burn-in” period is allowed.

Chapter 6

Bayesian Image Restoration

In this chapter we will describe the Bayesian approach to image restoration. The model will be based on a Gibbs distribution making use of its equivalence with Markov random fields. Two Bayesian estimators will be given. Due to the high dimensionality of the model, the estimators must be approximated. The techniques of approximation presented will include the Markov chain Monte Carlo technique as well as the Simulated Annealing (first introduced by Metropolis in 1953) and the Iterated Conditional Modes (proposed by Besag in 1983). The Markov chain simulation schemes that were described in Chapter 5 will be used to generate appropriate samples.

6.1 Bayesian Approach to Image Restoration

The problem of Bayesian image restoration is to reconstruct a real scene, or a “true” image, based on the observed data of corrupted image. These observations consist on a configuration y , which is modeled as a random function of the true image x . The random distortion process through which the observed image was formed is assumed to be known.

Furthermore, we assume that there exists a known multivariate distribution $p_X(x)$ which provides data on the value of x at each site and that the values at sites, which are close together tend to be the same or very similar.

Thus, as always in Bayesian inference, we have two types of information: the

prior knowledge and empirical data. The prior represents the probability model for the true (not degraded) image, and so the posterior can be formed after conditioning on the values of the observed (degraded) image.

The approach is to estimate the true scene as the best (or most probable) occurrence x of the non observable Markov random field X on Ω , given the observed configuration y of the Markov random field Y .

6.2 Prior and Posterior Distributions

In the Bayesian image restoration, the prior represents the probability model for the true image X . Since we represent an image by a Markov random field, the prior distribution is chosen so that it belongs to the Gibbs family of distributions. That is

$$\begin{aligned}\mathbb{P}[X = x] &= \frac{1}{Z} \exp\{-U(x)\} \\ &= \frac{1}{Z} \exp\left\{-\sum_{C \in \mathcal{C}} V_C(x)\right\}\end{aligned}\tag{6.1}$$

where $Z = \sum_x \exp\{-U(x)\}$ and $\mathcal{C}$ is the set of all cliques in S .

After deciding on the prior, the next step is to model the corrupted image Y as a random function of the true image X . This is done by forming the conditional distribution

$$\mathbb{P}[Y = y|X = x] \propto \exp\left\{-\sum_{s \in S} V_{\{s\}}(x(s), y(s))\right\}\tag{6.2}$$

Finally, the last step is to form the posterior from the prior after conditioning on the values of the observed image. We have

$$\mathbb{P}[X = x|Y = y] = \frac{\mathbb{P}[Y = y|X = x]\mathbb{P}[X = x]}{\mathbb{P}[Y = y]} \propto \mathbb{P}[Y = y|X = x]\mathbb{P}[X = x]$$

and using formulas 6.1 and 6.2

$$\mathbb{P}[X = x|Y = y] = \frac{1}{Z_y} \exp\left\{-\sum_{C \in \mathcal{C}} V_C(x(C)) - \sum_{s \in S} V_{\{s\}}(x(s), y(s))\right\}\tag{6.3}$$

We note that the energy function is separated into two sums of potentials: first based on the pixel values in the true image x (the prior) and second, on the interaction between pixels in the true and corrupted images (the posterior).

The estimation problem can now be specified in terms of the posterior distribution.

Since the random distortion process through which the observed image was formed is assumed to be known, we can say more about the posterior distribution.

Suppose that the corruptions include blur and noise. The model of the degraded image Y at every site $s \in S$ can be expressed as

$$Y(s) = \phi(b(X(s)), \eta(s))$$

where

- ϕ denotes a possibly not linear transformation, such that for fixed a , the function $b \rightarrow \phi(a, b)$ is invertible. We will denote the inverse by $b \rightarrow \Phi(a, b)$;
- $b(X)$ is a random field obtained by blurring X (the result of convolving X with a point spread function b);
- η is a random noise field, which is assumed to be independent of X .

We note that invertibility of the function ϕ allows us to express the noise field as

$$\eta(s) = \Phi(b(X(s)), Y(s))$$

For example, if η was an additive Gaussian white noise (that is, $\eta = \{\eta(s)\}_{s \in S}$ with each $\eta(s)$ being i.i.d. normal with mean μ and variance σ^2) then the degraded image model would be

$$Y(s) = b(X(s)) + \eta(s), \quad \forall s \in S$$

and

$$\begin{aligned} \eta(s) &= \Phi(b(X(s)), Y(s)) \\ &= Y(s) - b(X(s)) \end{aligned}$$

Let us now use this information and construct the posterior distribution of X given Y .

The definition of conditional probability implies

$$\mathbb{P}[X = x|Y = y] = \frac{\mathbb{P}[Y = y|X = x] \mathbb{P}[X = x]}{\mathbb{P}[Y = y]} \quad (6.4)$$

Assuming that the random variables $\eta(s)$ are i.i.d. normal with mean μ and variance σ^2 , and independent of $X(s)$ for every s in S , we have

$$\begin{aligned} \mathbb{P}[Y(s) = y(s)|X(s) = x(s)] &= \mathbb{P}[\phi(b(X(s)), \eta(s)) = y(s)|X(s) = x(s)] \\ &= \mathbb{P}[\eta(s) = \Phi(b(X(s)), y(s))|X(s) = x(s)] \\ &= \mathbb{P}[\eta(s) = \Phi(b(x(s)), y(s))] \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp \left\{ - \left(\frac{1}{2\sigma^2} \right) (\mu - \Phi(b(x(s)), y(s)))^2 \right\} \end{aligned}$$

and the joint probability for all sites in S is

$$\mathbb{P}[Y = y|X = x] = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp \left\{ - \left(\frac{1}{2\sigma^2} \right) \|\underline{\mu} - \Phi(b(x), y)\|^2 \right\}$$

where $\underline{\mu}$ is a $d \times d$ matrix with all components equal μ , $d = |S|$, and $\|\cdot\|$ is the norm in $\mathbb{R}^d$, that is $\|v\|^2 = \sum_{i=1}^d v_i^2$.

We can now substitute this result into Equation (6.4). Combining all constant terms into Z_y , we obtain

$$\begin{aligned} \mathbb{P}[X = x|Y = y] &= \frac{(2\pi\sigma^2)^{-d/2} \exp \left\{ - \left(\frac{1}{2\sigma^2} \right) \|\underline{\mu} - \Phi(b(x), y)\|^2 \right\} \exp\{-U(x)\}}{Z_y} \\ &= \frac{1}{Z_y} \exp \left\{ - \left(U(x) + \left(\frac{1}{2\sigma^2} \right) \|\underline{\mu} - \Phi(b(x), y)\|^2 \right) \right\} \end{aligned}$$

where $U(x) := \sum_{C \in \mathcal{C}} V_C(x)$.

Hence, the posterior distribution of X given Y is also Gibbs:

$$\mathbb{P}[X = x|Y = y] = \frac{1}{Z_y} \exp\{-U_y(x)\}$$

with the energy function

$$U_y(x) := U(x) + \frac{\|\underline{\mu} - \Phi(b(x), y)\|^2}{2\sigma^2} \quad (6.5)$$

6.2.1 Example: Ising model

Consider the following formulation of the Ising model

$$p_X(x) = \frac{1}{Z} \exp \left\{ \beta \sum_{\langle s, t \rangle} x(s)x(t) \right\}$$

with $\beta > 0$ and suppose that X was corrupted with additive Normal noise η with mean μ and variance σ^2 , that is

$$Y(s) = X(s) + \eta(s) \quad \forall s \in S$$

The presence of data results in a posterior distribution of Gibbsian type

$$\mathbb{P}[X = x|Y = y] = \frac{1}{Z_y} \exp \left\{ \beta \sum_{\langle s, t \rangle} x(s)x(t) + \sum_{s \in S} f(x(s), y(s)) \right\}$$

where the function f depends on the random distortion process, here

$$\sum_{s \in S} f(x(s), y(s)) = - \left(\frac{1}{2\sigma^2} \right) \sum_{s \in S} (\mu - y(s) + x(s))^2$$

The full conditionals are:

$$\begin{aligned} \mathbb{P}[X(s) = x(s)|X(S \setminus s) = x(S \setminus s), Y = y] \\ = \frac{\exp \left\{ \beta \sum_{t \in N_s} x(s)x(t) + f(x(s), y(s)) \right\}}{\exp \left\{ \beta \sum_{t \in N_s} x(t) + f(1, y(s)) \right\} + \exp \left\{ -\beta \sum_{t \in N_s} x(t) + f(-1, y(s)) \right\}} \end{aligned}$$

◇

6.3 Estimation of the True Image

In the previous section it was shown that the posterior distribution is Gibbsian of the form

$$\mathbb{P}[X = x|Y = y] = \frac{1}{Z(y)} \exp \left\{ - \sum_{c \in C} V_c(x(c)) - \sum_{s \in S} V_{\{s\}}(x(s), y(s)) \right\}$$

We now want to estimate the true image X based on this posterior distribution.

The obvious criterion for estimation could be the configuration x that as a whole is the most probable given the data y . This results in the so-called *maximum a posteriori (MAP)* estimator:

$$\hat{x} = \operatorname{argmax}_{x \in \Omega} \mathbb{P}[X = x | Y = y] \quad (6.6)$$

Another possibility could be the configuration x in which the value at each individual pixel is the most probable (independently of each other) given the data y . This results in the so-called *marginal posterior mode (MPM)* estimator:

$$\widehat{x(s)} = \operatorname{argmax}_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma | Y = y] \quad (6.7)$$

Let us now show that both MAP and MPM estimators are Bayesian estimators.

Let $d(Y)$ denote the decision rule. That is, for an observed image y , the function $d(y)$ is the estimate of the true image x : $d(y) = \hat{x}$.

Consider the following loss function

$$\begin{aligned} L_1(x, d(y)) &= I_{\{d(y) \neq x\}} \\ &= \begin{cases} 1 & \text{if } d(y) \neq x \\ 0 & \text{if } d(y) = x \end{cases} \end{aligned}$$

The Bayes risk R for the decision $d(Y)$ is

$$R = \mathbb{E}[L_1(X, d(Y))] = \sum_{y \in \Omega} \sum_{x \in \Omega} L_1(x, d(y)) p(x, y)$$

and the Bayes solution is the decision $d(y)$ that minimizes the risk R .

The risk R is minimized if, for each y , the inner summation of the last equation is minimized.

Minimizing

$$\begin{aligned} \sum_{x \in \Omega} L_1(x, d(y)) p(x, y) &= \sum_{x \neq d(y)} p(x, y) \\ &= \sum_{x \in \Omega} p(x, y) - p(d(y), y) \\ &= p(y) - p(d(y), y) \end{aligned}$$

is equivalent to maximizing $p(d(y), y) = p(d(y)|y)p(y)$ or just maximizing $p(d(y)|y)$.

Hence, the Bayes estimator $\hat{x} = d(y)$ is the one for which $p(\hat{x}|y)$ is maximal over the set of all possible configurations of X . That is the MAP estimator:

$$\hat{x} = \operatorname{argmax}_{x \in \Omega} \mathbb{P}[X = x | Y = y]$$

is a Bayesian estimator.

Now consider the loss function defined as

$$L_2(x, d(y)) = \frac{1}{|S|} \sum_{s \in S} I_{\{x(s) \neq d(y(s))\}}$$

Taking similar steps as above it can be shown that the MPM estimator:

$$\widehat{x(s)} = \operatorname{argmax}_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma | Y = y]$$

is Bayesian for L_2 .

◇

6.4 Techniques of Approximating Image Estimators

In general, direct derivation of both of the above Bayesian estimators is not manageable due to high dimensionality of the models. For this reason iterative sampling procedures are employed to approximate the true image estimators.

6.4.1 MCMC Approximation

Let us consider the following indicator function

$$I_{\{x(s)=\gamma\}} = \begin{cases} 1 & \text{if } x(s) = \gamma \\ 0 & \text{otherwise} \end{cases}$$

where $s \in S$ and $\gamma \in \Gamma$.

We note that the conditional expectation, given the observations y , is equal to the conditional probability of a color γ at site s :

$$\begin{aligned}\mathbb{E}[I_{\{X(s)=\gamma\}}|Y = y] &= \sum_{x(s) \in \Gamma} I_{\{x(s)=\gamma\}} \mathbb{P}[X(s) = x(s)|Y = y] \\ &= \mathbb{P}[X(s) = \gamma|Y = y]\end{aligned}$$

Hence, the maximum posterior mode (MPM) estimate can be obtained by maximizing this expectation with respect to $\gamma \in \Gamma$.

Now, as described in section 5.4, we can generate a sequence of configurations $x_0, x_1, \dots, x_m, \dots, x_n$ such that $x_m, \dots, x_n$ can be considered as samples from a conditional Gibbs distribution $\pi = p_{X|Y}$ for a particular $Y = y$.

Then for some $s \in S$ and $\gamma \in \Gamma$:

$$\begin{aligned}\mathbb{P}[X(s) = \gamma|Y = y] &= E[I_{\{X(s)=\gamma\}}|Y = y] \\ &\approx \frac{\sum_{i=m}^n I_{\{x_i(s)=\gamma\}}}{n - m + 1} \\ &= \frac{\#\{k \in [m, n] : x_k(s) = \gamma\}}{n - m + 1}\end{aligned}$$

Hence, the maximum posterior mode (MPM) estimator, which is the value of $x(s)$ that maximizes the conditional probability $\mathbb{P}[X(s) = x(s)|Y = y]$, can be approximated by the value that has the highest frequency of occurrence, at a pixel s , in the sequence of realizations of a Markov chain:

$$\widehat{x(s)} \approx \max_{\gamma \in \Gamma} \#\{k \in [m, n] : x_k(s) = \gamma\} \quad (6.8)$$

Figure 6.1 shows the results of applying this technique for various values of m and n to a binary image corrupted with Gaussian white noise with mean 0.5 and variance 0.8. The Ising model, with the energy function given by equation (4.7) and $\beta = 1$, was used as the prior. The original and corrupted images are shown in Figures 6.1 (1) and (2). Figures 6.1 (3) - (7) show the MCMC approximations to the original image based on iterations $[m, n] = [10, 20], [15, 30], [25, 50], [50, 100]$ and $[200, 300]$. The last figure is the best approximation, it contains 1.5% of misclassified pixels.

Figure 6.1 *MCMC approximation to MPM estimate applied to a noisy binary image.*

6.4.2 Simulated Annealing

Simulated annealing is a stochastic computational technique that was originally derived in statistical mechanics. It was introduced by Metropolis in 1953 who created an algorithm to simulate the behavior of atoms at equilibrium at a particular temperature. He observed that as the temperature was slowly decreasing the samples were forced towards minimal energy configurations.

Simulated annealing can be used to approximate the Bayesian maximum a posteriori (MAP) estimator. The algorithm is based on local changes at single sites $s \in S$. These changes are made randomly based on sampling from local conditional distributions, hence only the current values of pixels in the neighborhood of s are considered. The fundamental concept is that these local conditional distributions are dependent on the global control parameter T and the local maxima are avoided by setting the initial T high and gradually decreasing it.

In order to formulate a simulated annealing algorithm, the following have to be decided:

1. The objective function to be minimized;
2. The annealing schedule;
3. The way of generating/selecting new configurations.

1. The objective function to be minimized

Note that maximization of $\mathbb{P}[X = \cdot | Y = y]$ with respect to X is equivalent to minimizing the energy function $U(X)$ of $p_{X|Y}$. The idea is then to sample from Gibbs distribution $p_{(X(s)|X(\mathcal{N}_s)Y(s))}^T$ with energy $\frac{U(X)}{T}$ where T is a temperature parameter which slowly decreases to 0.

2. The annealing schedule

The annealing schedule refers to the slow decrease of the temperature parameter T which can be expressed as a decreasing sequence of temperatures $\{T_n\}$ where n is the discrete time index. The schedule includes the initial and final values of T , a rule that defines the number of iterations that should be run for each value of T and a rule that generates the next value of T .

The following theorem (due to Geman and Geman, 1984) prescribes a schedule for lowering T so that a generated configuration will approach the minimal energy.

Theorem 6.1 (Annealing Theorem)

Let

$$\Gamma_0 = \{x \in \Omega : U(x) = \min_{k \in \Omega} \{U(k)\}\}$$

p_0 be the uniform distribution on Γ_0

$\{n_k, k = 0, 1, 2, 3, \dots\}, n_k \in S$ be the sequence in which the sites are visited for updating

Define:

$$\Delta = \max_{x \in \Omega} \{U(x)\} - \min_{x \in \Omega} \{U(x)\}$$

Assume that there exists an integer $k \geq |S|^{1/2} = N$ such that for every $t = 0, 1, 2, 3, \dots$, $S \subseteq \{n_{t+1}, n_{t+2}, \dots, n_{t+k}\}$

Let $\{T_n\}$ be any decreasing sequence of temperatures for which:

(1) $T_n \rightarrow 0$ as $n \rightarrow \infty$;

(2) $T_n \geq \frac{N\Delta}{\log n}$ for all $n \geq n_0$ for some $n_0 \geq 2$.

Then for any starting configuration $k \in \Omega$ and for every $x \in \Omega$

$$\lim_{n \rightarrow \infty} \mathbb{P}[X(n) = x | X(0) = k] = p_0(x)$$

The Annealing theorem prescribes a schedule for lowering the temperature which forces the system into the lowest energy states. Hence, it guarantees convergence to the global maximum of the posterior distribution. The basic idea is that if T_n is the temperature used when executing the n th site replacement and if $T_n \geq \frac{c}{\log(n+2)}$ for all n where c is a sufficiently large constant independent of n , then as $n \rightarrow \infty$ the configuration generated will approach the minimal energy.

3. The way of generating/selecting new configurations

The computational task we considered up to this point was to sample from a distribution π and to compute the expected values, where the latter was accomplished by combining the first with the Ergodic theorem.

Now a new task is to minimize $U(x)$ over Ω .

Note that when π is the posterior distribution $p_{X|Y}$, such minimization of $U(x)$ corresponds to the MAP estimator. This can be achieved by sampling from $\pi := p_{X|Y}$ combined with the Annealing theorem.

We consider the following model

$$p_{X|Y}^T \propto \{p_{Y|X} p_X\}^{1/T}, \quad T > 0$$

It is not possible to simulate Markov random fields directly, but one can construct a transition probability matrix P^T with state space Ω and limiting distribution as $p_{X|Y}^T$ above.

Simulated annealing proceeds by running an associated Markov chain with transition probability matrix P^T , where T is gradually decreased to a value close to 0, according to a prescribed schedule $\{T_n\}$.

Suppose we construct the transition probability matrix $P^T = (P^T(x_{n-1}, x_n))_{x_{n-1}, x_n \in \Omega}$ by the method of Gibbs sampling as seen in Section 5.3.4. Thus, when at step n , the updated value $x_n(s)$ at the chosen site $s \in S$, is assigned according to

$$\{\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x_{n-1}(S \setminus s), Y = y]\}^{1/T_n}$$

Figure 6.2 shows the steps of the Simulated Annealing technique applied to a binary image corrupted with Gaussian white noise with mean 0.5 and variance 0.8. The Ising model with the energy function given by equation (4.7) and $\beta = 1$ was used as the prior. The original and corrupted images are shown in Figures 6.2 (1) and (2). Figures 6.2 (3) - (8) show the configurations of Simulated Annealing after 10, 20, 40, 80, 120 and 200 iterations. The last figure contains the smallest number of misclassified pixels (6.6%).

Figure 6.2 *Simulated Annealing applied to a noisy binary image.*

6.4.3 Iterated Conditional Modes (ICM)

Iterated conditional modes (ICM) is a simple iterative method of image reconstruction proposed by Besag in 1983 as an approximation to the MAP estimate. It is based on the concepts of Bayesian statistics and applied with relatively simple computations not affected by the large scale characteristics of a chosen Markov random field.

The following assumptions need to be made:

1. Given any particular configuration x , the random variables $Y(s), s \in S$ are conditionally independent and each $Y(s)$ has the same conditional distribution $\mathbb{P}(Y(s)|X(s))$ which depends only on $X(s)$ and is known;
2. The true configuration x is a realization of a locally dependent Markov random field with specified distribution p_X .

The ICM method is based on updating a single component at a current site s by choosing the value that has maximal conditional probability given the realization y and the current reconstruction x_n :

$$x_{n+1}(s) = \arg \max_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma | Y = y, X(S \setminus s) = x_n(S \setminus s)] \quad (6.9)$$

Note that the assumptions imply

$$\begin{aligned} \mathbb{P}[X(s) = x(s) | Y = y, X(S \setminus s) = x(S \setminus s)] \\ &= \frac{\mathbb{P}[X(s) = x(s), Y = y, X(S \setminus s) = x(S \setminus s)]}{\mathbb{P}[Y = y, X(S \setminus s) = x(S \setminus s)]} \\ &\propto \mathbb{P}[Y = y | X(s) = x(s), X(S \setminus s) = x(S \setminus s)] \\ &\quad \mathbb{P}[x(s) | X(S \setminus s) = x(S \setminus s)] \mathbb{P}[X(S \setminus s) = x(S \setminus s)] \\ &\propto \mathbb{P}[Y(s) = y(s) | X(s) = x(s)] \mathbb{P}[X(s) = x(s) | X(\mathcal{N}_s) = (x(\mathcal{N}_s))] \end{aligned}$$

Therefore

$$\begin{aligned} x_{n+1}(s) &= \arg \max_{\gamma \in \Gamma} \mathbb{P}[X(s) = \gamma | Y = y, X(S \setminus s) = x_n(S \setminus s)] \\ &\propto \arg \max_{\gamma \in \Gamma} \mathbb{P}[Y(s) = y(s) | X(s) = \gamma] \mathbb{P}[X(s) = \gamma | X(\mathcal{N}_s) = x_n(\mathcal{N}_s)] \end{aligned}$$

The algorithm is implemented at each pixel in turn forming a single cycle. The final estimate of X is produced after applying a finite number of cycles. According to Besag (1986), "convergence to the local maximum of $\mathbb{P}[X = \cdot | Y = y]$ is very rapid with possibly only few changes after a sixth cycle."

6.4.4 Example: Potts Model

Suppose that the conditional density of $Y(s)$ given the values of the true image $X(s) = x(s)$ is Normal with mean $\mu(x(s))$ and variance σ^2 for all s in S .

Let us assume that the prior distribution is given by the Potts model (see Section 4.3.2). Then the local characteristic is given by

$$\mathbb{P}[X(s) = x(s) | X(S \setminus s) = x(S \setminus s)] = \frac{\exp\{\beta \sum_{t \in \mathcal{N}_s} I_{\{x(t)=x(s)\}}\}}{\sum_{\gamma \in \Gamma} \exp\{\beta \sum_{t \in \mathcal{N}_s} I_{\{x(t)=\gamma\}}\}}$$

where β is a fixed parameter (say $\beta = 1.5$) and $\sum_{t \in \mathcal{N}_s} I_{\{x(t)=\gamma\}}$ counts the number of neighbors of a pixel $x(s)$ with value γ ($I_{\{A\}}$ is the indicator function on a set A).

$$\begin{aligned} & \mathbb{P}[X_{n+1}(s) = x(s) | Y = y, X(S \setminus s) = x_n(S \setminus s)] \\ & \propto \left(\frac{1}{2\pi\sigma^2} \right)^{1/2} \exp \left\{ -\frac{(y(s) - \mu(x(s)))^2}{2\sigma^2} \right\} \exp \frac{\{\beta (\sum_{t \in \mathcal{N}_s} I_{\{x_n(t)=x(s)\}})\}}{\sum_{\gamma \in \Gamma} \exp\{\beta (\sum_{t \in \mathcal{N}_s} I_{\{x_n(t)=\gamma\}})\}} \\ & \propto \exp \left\{ -\frac{(y(s) - \mu(x(s)))^2}{2\sigma^2} + \beta \left(\sum_{t \in \mathcal{N}_s} I_{\{x_n(t)=x(s)\}} \right) \right\} \end{aligned}$$

where $\sum_{t \in \mathcal{N}_s} I_{\{x_n(t)=x(s)\}}$ counts the current number of neighbors of a pixel s with value $x(s)$.

Now maximizing $\mathbb{P}[X_{n+1}(s) = x(s) | Y = y, X(S \setminus s) = x_n(S \setminus s)]$ is equivalent with minimizing

$$\frac{(y(s) - \mu(x(s)))^2}{2\sigma^2} - \beta \left(\sum_{t \in \mathcal{N}_s} I_{\{x_n(t) = x(s)\}} \right)$$

with respect to $x(s)$ at each pixel in turn.

Note that, if we consider a 4-point neighborhood system, the values that have to be looked at are only the four neighbors of $x(s)$. These bring at most four possibilities

for a new value of $x(s)$. For instance, if $\mu(x(s)) = \mu$ for all $s \in S$, then the problem would reduce to choosing the most frequent value among the neighbors of $x(s)$.

◇

Figure 6.3 shows the steps of the ICM technique applied to a binary image corrupted with Gaussian white noise with mean 0.5 and variance 0.8. The Ising model with the energy function given by equation (4.7) and $\beta = 1$ was used as the prior. The original and corrupted images are shown in Figures 6.3 (1) and (2). Figures 6.3 (3) - (7) show the configurations of ICM after 2, 4, 6, 10 and 20 iterations. Note that there is no change in the pixel values after 6th iteration where we have 7.8% of misclassified pixels.

Figure 6.3 *ICM applied to a noisy binary image.*

Chapter 7

Experimental Results

In this chapter, we will show and discuss the experimental results of various image processing techniques. The source code for the methods was written in MATLAB based on the algorithms presented in this thesis.

The original image and its histogram are shown in Figure 7.1(1). The image consists of 3 overlapping rectangles and a black text on white background. It was designed so that the observed intensities are equally spaced in the interval from 0 to 255 and there are all possible color combinations of 2-point neighbors.

The degraded image and its histogram are shown in Figure 7.1(2). The image was obtained by adding independent white Gaussian noise to each pixel value in the original image. Section 7.2 will give more details about the degradation process.

The goal of this experiment is to apply different enhancement and restoration algorithms to the corrupted image and to compare the resulting images with the original.

Figure 7.1 *Image degradation: (1) True image and its histogram, (2) Degraded image and its histogram*

The “quality” of the restored images will be measured based on the following criteria:

- The mean-squared error:

$$MSE := \frac{1}{|S|} \sum_{s \in S} (x(s) - y(s))^2$$

where x is the original image, y represents a processed image and S is the set of sites (here $|S| = 64^2$);

- The percentage of misclassified pixels:

$$\%ERROR := \left(\frac{1}{|S|} \sum_{s \in S} I_{\{x(s) \neq y(s)\}} \right) \times 100$$

- The overall appearance:

The ideal output image should look like the original in figure 7.1 (1). We want to obtain a sharp image with clearly defined edges and uniformly filled objects that are of the same shape and color as those in the original.

In this experiment, the methods of image processing will be divided into two groups. The first group will be called “Numerical” and it will include the methods of ideal low pass, mean, median and Wiener filtering. The other group will be called “Statistical” and it will consist of the Bayesian methods of image restoration.

Note that the pixel values in the statistically restored images will be integers belonging to the same set of intensities as the original image. However, the pixel values of the numerically processed images will not necessarily be integers.

It is then meaningless to look at the misclassification rates (%error) for the numerically processed images, as even very close approximations would be considered as errors. Furthermore, due to the different grey level scales, neither of the numerical measures of quality can be used to compare the results between the two groups.

The obvious solution would be to use the same intensity scale for all the images. Consequently, we will map the processed pixel values into the four original intensities: 0, 85, 170 and 255.

The error rates between the original and the degraded images are listed in Table 7.1.

Table 7.1 *Error rates between original and degraded images.*

MSE	%ERROR
2941.20	97.58

7.1 Numerical Methods of Image Processing

In this part of the experiment, the following filters were used

1. Ideal Low Pass Filter with the cut-off frequency 0.5 (ILP-0.5);
2. Ideal Low Pass Filter with the cut-off frequency 0.8 (ILP-0.8);
3. Mean Filter;
4. Median Filter;
5. Wiener Filter.

The results of applying the filters to the degraded image are shown in Figure 7.2. The error rates are listed in Table 7.2.

Table 7.2 *Results of numerical image processing.*

FILTER	MSE
ILP - 0.5	3214.50
ILP - 0.8	2853.40
Mean	3588.20
Median	3024.50
Wiener	3055.20

The ideal low-pass filter with the cut-off frequency of 0.8 produced an image that appears to be the sharpest and has the smallest mean squared error (2853.40), which is smaller than the MSE between the original and the noisy images (2941.20 - see Table 7.1). The MSEs between the original and the other filtered images were all greater than 2941.2. Note that the Wiener filter, which was the only model-based technique that we applied, did not result in a well restored image as one could expect. This might be caused by the general algorithm that assumed presence of noise and blur, both with unknown parameters.

Figure 7.2 Numerically processed images using the following filters: (1) ILP-0.5, (2) ILP-0.8, (3) Mean, (4) Median, (5) Wiener.

The histogram equalization technique was applied next in order to enhance the contrast in the filtered images. The results are given in Figure 7.3 and Table 7.3.

Figure 7.3 Numerically processed images after histogram equalization: (1) ILP-0.5, (2) ILP-0.8, (3) Mean, (4) Median, (5) Wiener.

Table 7.3 Results of numerical image processing after histogram equalization.

FILTER	MSE
ILP - 0.5	2749.10
ILP - 0.8	2651.10
Mean	2938.00
Median	2614.20
Wiener	2481.30

Finally, the pixel values present in the processed images were mapped into the four original intensities: 0, 85, 170 and 255.

The transformations were based on the information contained in the histograms of the processed images (see Figure 7.2). In each case, the intensity range was divided into four intervals. All the pixels falling into the first interval were assigned the value of 0, those falling in the second interval - 85, and so on. The intervals were chosen so that they were approximately centered at the histogram's local maxima, and their end points would fall at the local minima.

The resulting images are shown in Figure 7.4 and the numerical results are listed in Table 7.4.

Figure 7.4 Numerically processed images after intensity transformations: (1) ILP-0.5, (2) ILP-0.8, (3) Mean, (4) Median, (5) Wiener.

Table 7.4 Results of numerical image processing after intensity transformations.

FILTER	MSE	%ERROR
ILP - 0.5	1594.60	20.31
ILP - 0.8	1326.50	17.77
Mean	1626.30	20.39
Median	993.08	10.25
Wiener	919.00	12.43

7.2 Statistical Methods of Image Restoration

The methods used in this part of the experiment were based on the Bayesian approach with the Potts model as a prior.

To express the image in terms of the Potts model, the original pixel values were labeled as it is shown in Table 7.5.

Table 7.5 *Labels for the pixel values used in the Potts model*

COLOR	PIXEL VALUE	LABEL
Black	0	1
Dark Gray	85	2
Gray	170	3
White	255	4

The degraded image was obtained by adding an independent Gaussian noise with mean 0 and variance 0.3 to the labeled image. This resulted in pixel values in the range from -0.93 to 5.86. It was assumed that the corrupted image represents the observed data, and that the distribution of noise is known (which was not the case when we dealt with the Wiener filter).

Three techniques of approximating Bayesian image estimators were used, namely:

1. MCMC approximation to the marginal posterior mode (MPM) estimator;
2. Simulated Annealing (SA) approximation to the maximum a posteriori (MAP) estimator;
3. Iterated Conditional Modes (ICM) approximation to the MAP estimator.

In each case, the Gibbs sampling scheme was employed to obtain samples from posterior distributions. Each pixel was updated 200 times resulting in $200 \times 64 \times 64$ iterations for the MCMC (with the burn-in period of 100 cycles) and Simulated Annealing methods. In neither of the cases, convergence was reached but the number

of pixels changing their values in the final configurations was very small. For the ICM method convergence was obtained after approximately 12 cycles.

The values of the texture parameter β used in the models were selected based on the overall appearance and the actual error rates. Figure 7.4 displays the MSE obtained for various values of β .

The chosen values were: $\beta = 2.5$ for Simulated Annealing and MCMC, and $\beta = 1.5$ for ICM.

Figure 7.5 *MSE versus different values of the parameter β for the three restoration techniques.*

The images restored from the noisy data are shown in Figure 7.6. The numerical results are listed in Table 7.6.

Figure 7.6 *Statistically restored images using: (1)MCMC, (2)Simulated Annealing, (3)ICM.*

Table 7.6 *Results of statistical image restoration.*

TECHNIQUE	MSE	%ERROR
MCMC	44.0979	0.61
SA	86.4319	0.98
ICM	331.6162	4.37

All of the statistical techniques resulted in good approximations to the original image. The MCMC method produced almost perfect approximation. The drawback of this method is that the resulting configurations have to be stored in order to decide on the most common values. The image obtained by the Simulated Annealing technique contained less than 1% of misclassified pixel values, and did not require much computer memory. The ICM estimate appears to be the worst - approximately 4% of pixels were assigned an incorrect value. The advantage of this method is its fast rate of convergence.

Chapter 8

Conclusion

The following list is the summary of the theory and results presented in this thesis.

- We introduced the numerical techniques of image enhancement and restoration.
- We constructed a random model for images based on the theory of Markov random fields.
- We presented several sampling schemes to obtain realizations of a given Markov random field.
- We described the Bayesian approach to image restoration.
- We presented the experimental results from applying numerical and statistical image processing techniques.
- After visual inspection of the experimental results, we concluded that the statistically recovered images approximate the original image much better than the numerical filters. However, they require more computational time and more memory space.

The ideas for possible future work are outlined below.

- The obvious extension to the work presented in this thesis would be to consider a true scene, not a small-size, four-color, artificial image. This would require us to design a model that would work efficiently with 256 colors. One should also try to statistically approximate all the parameters involved.
- Another suggestion is to develop a model which would assume the existence of so-called “edge-sites” between each pair of adjacent pixels (see [Geman, Geman, 1984]). This model could be then used in real-life applications that require edge detection or enhancement, for example, cancer detection in medical imaging.
- Finally, one should investigate the convergence rates of the stochastic algorithms to know the number of iterations needed to obtain samples from the desired distribution.

Appendix A

Fundamentals of Digital Signal Processing

Since a monochrome picture is a 2-dimensional signal, it is necessary to introduce the mathematical concepts that are used in the study of signals.

A.1 Signals

A *signal* is a function of independent variables such as time, distance, position or temperature that contains information about the state or behavior of a physical system.

Signal processing consists of mathematical representation of the signal and the algorithmic operation carried out to extract needed information. The techniques of signal processing take usually the form of a transformation of the original signal into another signal that, in some sense, is more desirable.

One of the simplest discrete-time signals is the *unit impulse*, which is defined as:

$$\delta(n_1, n_2) = \begin{cases} 0 & (n_1, n_2) \neq (0, 0) \\ 1 & (n_1, n_2) = (0, 0) \end{cases} \quad (\text{A.1})$$

where n_1 and n_2 are the spatial coordinates.

A second basic discrete-time signal is the discrete-time *unit step*:

$$u(n_1, n_2) = \begin{cases} 0 & n_1, n_2 < 0 \\ 1 & n_1, n_2 \geq 0 \end{cases} \quad (\text{A.2})$$

The relationship between discrete-time unit impulse and unit step is given by the following two equations:

$$\delta(n_1, n_2) = u(n_1, n_2) - u(n_1 - 1, n_2) - u(n_1, n_2 - 1) + u(n_1 - 1, n_2 - 1) \quad (\text{A.3})$$

$$u(n) = \sum_{m_1=-\infty}^{n_1} \sum_{m_2=-\infty}^{n_2} \delta(m_1, m_2) = \sum_{k=-\infty}^0 \delta(n - k) = \sum_{k=0}^{\infty} \delta(n - k) \quad (\text{A.4})$$

The unit impulse sequence can be used to sample the value of a signal at $(n_1, n_2) = (0, 0)$. It follows that:

$$x(n_1, n_2)\delta(n_1, n_2) = x(0, 0)\delta(n_1, n_2)$$

In general, if we consider a unit impulse $\delta(n_1 - k_1, n_2 - k_2)$, then

$$x(n_1, n_2)\delta(n_1 - k_1, n_2 - k_2) = x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2)$$

That is, multiplying $x(n_1, n_2)$ by a unit impulse samples the value of the signal at the point at which the impulse is located. This result is known as the sampling property of the unit impulse.

Note that an arbitrary sequence can be expressed as a sum of scaled delayed unit samples:

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2) \quad (\text{A.5})$$

A.2 Spatial Domain Representation of 2-D Digital Signal

A.2.1 Linear Shift-Invariant Systems

A system is a unique transformation or operator, here denoted by T , that maps an input sequence $x(n_1, n_2)$ into an output sequence $y(n_1, n_2)$:

$$y(n_1, n_2) = T(x(n_1, n_2)) \quad (\text{A.6})$$

The system T is called *linear* if and only if any linear combination of two inputs, $x_1(n_1, n_2)$ and $x_2(n_1, n_2)$ results in the same combination of the outputs $y_1(n_1, n_2)$ and $y_2(n_1, n_2)$.

That is, for any constants a and b

$$\begin{aligned} T(ax_1(n_1, n_2) + bx_2(n_1, n_2)) &= aT(x_1(n_1, n_2)) + bT(x_2(n_1, n_2)) \\ &= ay_1(n_1, n_2) + by_2(n_1, n_2) \end{aligned}$$

Another important property is the systems response to translation. A system is called *shift* (or spatially) *invariant* if and only if, the shift in position of the input causes the same shift of the output.

That is

$$T(x(n_1 - k_1, n_2 - k_2)) = y(n_1 - k_1, n_2 - k_2)$$

Finally a system is called *linear shift-invariant (LSI)* if and only if, it is both linear and shift-invariant.

A.2.2 The Impulse Response

Consider a linear shift-invariant system T :

$$\begin{aligned}
 y(n_1, n_2) &= T(x(n_1, n_2)) \\
 &= T\left(\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)\right) \\
 &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) T(\delta(n_1 - k_1, n_2 - k_2))
 \end{aligned}$$

The response of the system to the impulse $\delta(n_1, n_2)$ is denoted by $h(n_1, n_2)$ and called the *impulse response*:

$$h(n_1, n_2) = T(\delta(n_1, n_2)) \quad (\text{A.7})$$

By the definition of shift-invariance, we have

$$h(n_1 - k_1, n_2 - k_2) = T(\delta(n_1 - k_1, n_2 - k_2))$$

Hence, for the LSI system

$$\begin{aligned}
 y(n_1, n_2) &= T(x(n_1, n_2)) \\
 &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)
 \end{aligned}$$

Thus the output of the LSI system, to any input can be determined based only on the knowledge of $h(n_1, n_2)$. For this reason, we say that any linear shift-invariant system is completely characterized by the impulse response $h(n_1, n_2)$.

The above equation is called the *convolution* of the input with the input response. If we denote the convolution operator by “*”, the equation can be rewritten as

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$$

A.3 Frequency Domain Representation of 2-D Signals

In practice, when dealing with 256×256 images, applying a convolution can be very time consuming. However, there exists a faster method that involves the use of the Fourier transform.

A transform is a mapping from one set of coordinates to another. The most common transform is the Fourier transform - it takes spatial data and converts it into frequency data. The spatial and frequency domains are just alternative ways of representing signals and the Fourier transform provides the mathematical relationship between these two representations.

It should be mentioned that the mathematical transformations are applied to signals not just to speed up some operations, but primarily to obtain a further information from that signal that is not easily available in the spatial domain. In many cases, the most distinguished information is hidden in the frequency content of the signal

A.3.1 Two-Dimensional Discrete Fourier Transform

Any surface can be expressed as a sum of sine and cosine curves - an image in the frequency domain consists of these sine and cosine components stored in a vector at each pixel.

Such vector can be also represented as a complex number, which by definition consist of two separate (real and imaginary) parts. The exponential of the imaginary number is the desired sum of sine and cosine

$$e^{i\omega} = \cos(\omega) + i \sin(\omega)$$

where $i = \sqrt{-1}$.

In image processing applications, one deals with finite-extent sequences which can be represented by a finite number of values. The discrete-time Fourier transform (DFT) represents such sequence in the frequency domain.

For every spatial domain waveform there is a corresponding frequency domain waveform, and vice versa. Waveforms that correspond to each other in this manner are called Fourier transform pairs.

The DFT pair of an $N_1 \times N_2$ -point sequence is given in the following two equations.

The DFT of $x(n_1, n_2)$ is:

$$\mathcal{F}_x(\omega_1, \omega_2) = \begin{cases} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-i\frac{2\pi}{N_1}\omega_1 n_1} e^{-i\frac{2\pi}{N_2}\omega_2 n_2}, & 0 \leq \omega_1 \leq N_1 - 1 \\ & 0 \leq \omega_2 \leq N_2 - 1 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.8})$$

and the inverse DFT (IDFT) of $\mathcal{F}_x(\omega_1, \omega_2)$ is:

$$x(n_1, n_2) = \begin{cases} \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(\omega_1, \omega_2) e^{i\frac{2\pi}{N_1}\omega_1 n_1} e^{i\frac{2\pi}{N_2}\omega_2 n_2}, & 0 \leq n_1 \leq N_1 - 1 \\ & 0 \leq n_2 \leq N_2 - 1 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.9})$$

We write: $x(n_1, n_2) \longleftrightarrow \mathcal{F}_x(\omega_1, \omega_2)$.

One of the major advantages of Fourier transform is that it obeys the convolution theorem.

Theorem A.1 (The Convolution Theorem) *The Fourier transform of the convolution of two function is equal to the product of the Fourier transforms of the two functions. That is*

$$f * g = \mathcal{F}_f \mathcal{F}_g$$

A.3.2 The Frequency Response

The frequency spectrum of a signal consists of frequency components of that signal.

In order to obtain the frequency content of a signal (that is, to find out how much of each frequency exists in it), the Fourier transform (FT) of that signal in spatial domain is taken.

Consider a linear shift-invariant system with impulse response $h(n_1, n_2)$ and a complex exponential input:

$$x(n_1, n_2) = e^{i\omega_1 n_1} e^{i\omega_2 n_2}, \quad -\infty < n_1, n_2 < \infty, \quad -\pi \leq \omega_1, \omega_2 < \pi \quad (\text{A.10})$$

The output can be then obtained as:

$$\begin{aligned} y(n_1, n_2) &= h(n_1, n_2) * x(n_1, n_2) \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) e^{i\omega_1(n_1-k_1)} e^{i\omega_2(n_2-k_2)} \\ &= e^{i\omega_1 n_1} e^{i\omega_2 n_2} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) e^{-i\omega_1 k_1} e^{-i\omega_2 k_2} \end{aligned}$$

Now let us define the so-called *frequency response* function as the frequency response of the system whose unit sample response is $h(n_1, n_2)$:

$$\mathcal{F}_h(\omega_1, \omega_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) e^{-i\omega_1 k_1} e^{-i\omega_2 k_2}$$

The system's output can now be written as

$$y(n_1, n_2) = \mathcal{F}_h(\omega_1, \omega_2) e^{i\omega_1 n_1} e^{i\omega_2 n_2} \quad (\text{A.11})$$

A.3.3 Fast Fourier Transform

The Discrete Fourier Transform (DFT) is one of the most important tools in Digital Signal Processing due to the following reasons.

First of all, it serves as an easy calculation of the signal's frequency spectrum. This allows direct examination of information encoded in the frequency, phase, and amplitude of the component sinusoids.

Secondly, with the use of DFT, the system's frequency response can be obtained from the system's impulse response, and vice versa. This allows systems to be analyzed in the frequency domain, just as convolution allows systems to be analyzed in the time domain.

Finally, there exists an efficient implementation of the DFT. It is called the Fast Fourier Transform (FFT) and allows to reduce the computational workload by a significant factor.

The Fast Fourier Transform is a method for calculating the DFT. While it produces the same result as the traditional approach, it is much more efficient in terms of the computation time.

We assume that $N = 2^m$ is an even integer. The algorithm is based on the principle of decomposing the DFT sequence into successively smaller sequences not more than $m = \log_2 N$ times. This results in Nm complex multiplications and additions. The factor by which the computational workload is reduced is N^2/Nm^2 for one dimensional sequence.

Bibliography

- [1] Barone, P., Figessi, A., Piccioni, M., Stochastic Models, Statistical Methods and Algorithms in Image Analysis - Proceedings, Rome 1990, *Lecture Notes in Statistics*, Springer-Verlag, New York, 1992.
- [2] Bremaund, P., *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues*, Springer, New York, 1999.
- [3] Besag, J., On the Statistical Analysis of Dirty Pictures, *Journal of the Royal Statistical Society B* (1986), 48, No. 3, pp. 259-302.
- [4] Besag, J., *Markov Chain Monte Carlo for Statistical Inference*, Center for Statistics and the Social Sciences, Working Paper No.9, University of Washington, 2000.
- [5] Besag, J., Green, P., Higdon, D., Mengersen, K., Bayesian Computation and Stochastic Systems, *Journal of the Royal Statistical Society B* (1986), 48, No. 3, pp. 259-302.
- [6] Bouman, C. A., *Markov Random Fields and Stochastic Image Models*, Tutorial Presented at 1995 IEEE International Conference on Image Processing, Washington, D.C., 1995.
- [7] Casella, G., Berger, R. L., *Statistical Inference*, Duxbury Press, Belmont, California, 1990.
- [8] Castelman, K. R., *Digital Image Processing*, Prentice Hall, Toronto, 1996.

- [9] Chellappa, R., Jain, A. K., *Markov Random Fields; Theory and Application*, Academic Press, Boston, 1993.
- [10] Embree, P. M., *C++ Algorithms for Digital Signal Processing*, Prentice Hall, N.J., 1999.
- [11] Gamerman, D., *Markov Chain Monte Carlo - Stochastic Simulation for Bayesian Inference*, Chapman and Hall, New York, 1997.
- [12] Geman, S., Geman, D., Stochastic Relaxation, Gibbs Distributions, and the Bayesian restoration of Images, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, No. 6, Nov. 1984.
- [13] Geman, D., Random Fields and Inverse Problems in Imaging, *Lecture Notes in Mathematics*, Springer-Verlag, New York, 1980.
- [14] Gibbs, A. L., *Bounding the Convergence Time of the Gibbs Sampler in Bayesian Image Restoration*, University of Toronto, Department of Statistics, Technical Report number 9808, 1998.
- [15] Gibbs, A. L., *Convergence Of Markov Chain Monte Carlo Algorithms with Applications to Image Restoration*, PhD thesis, University of Toronto, 2000.
- [16] Gilks, W. R., Richardson, S., Spiegelhalter, D.J., *Markov Chain Monte Carlo in Practice*, Chapman and Hall, New York, 1996.
- [17] Guttorp, P., *Stochastic Modeling of Scientific Data*, Chapman and Hall, New York, 1995.
- [18] Hanson, K. M., Introduction to Bayesian Image Analysis, *Medical Imaging: Image Processing*, Proc. SPIE 1896, pp. 716-731, 1993.
- [19] Hanson, K. M., Cunningham, G.S., Posterior Sampling with Improved Efficiency, *Medical Imaging: Image Processing*, Proc. SPIE 3338, pp. 371-382, 1998.
- [20] Hogg, R. V., Craig, A. T., *Introduction to Mathematical Statistics*, 5th ed., Prentice Hall, Inc, Toronto, 1995.

- [21] Hunt, B. R., Bayesian Methods in Nonlinear Digital Image Restoration, *IEEE Trans. Comp.*, Vol. C-26, No. 3, March 1977.
- [22] Inoue, J., *Statistical Mechanics of Image Restoration - A Short Note*, Information Engineering Laboratory Class Notes, Graduate School of Engineering, Hokkaido University, Japan, 2000.
- [23] Jain, A. K. *Fundamentals of Digital Image Processing*, Holden-Day, San Francisco, 1968.
- [24] Jenkins, G. M., Watts, D. G., *Spectral Analysis and Its Applications*, Prentice Hall, Toronto, 1989.
- [25] Kinderman, R., Snell, J. L., Markov Random Field and Their Applications, *Contemporary Mathematics*, v.1, American Mathematical Society, 1980.
- [26] Lim, J. S., *Two Dimensional Signal and Image Processing*, Prentice Hall, Toronto, 1990.
- [27] Oppenheim, A. V., *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1975.
- [28] Parker, J. R., *Algorithms for Image Processing and Computer Vision*, John Wiley and Sons, Inc., Toronto, 1997.
- [29] Perez, P., Markov Random Fields and Images, *Signals and Images, CWI Quarterly*, VII, N4, 1998, pp. 413-437.
- [30] Petrou, M., Bosdogianni, P., *Image Processing , The Fundamentals*, John Willey and Sons Ltd., Toronto, 1999.
- [31] Taylor, H. M., Karlin, S., *An Introduction to Stochastic Modeling*, Academic Press, Toronto, 1984.
- [32] Winkler, G., *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Springer, New York, 1995.