Application of Random Sets to Image Analysis

Nial Patrick Friel

A Dissertation Submitted to the

University of Glasgow

for the degree of

Doctor of Philosophy

Department of Statistics

November 1999

Preface

Image analysis can be simply thought of as the extraction of information from data in the form of a picture. An image represents a large highly structured data set, with strong spatial dependency between picture elements or pixels. For example a typical image will represent a data set of size at least 256^2. The challenge in image analysis then should be to use and enhance this knowledge in making inference about the information contained in the image. With this in mind it would seem that statistics has a large and important part to play in image analysis. However the contrary is in fact more evident.

For instance, numerical measures of dissimilarity between images, termed image metrics, which are the subject of Chapter 3, are commonly examined by just exploring pixel-by-pixel differences. In this way, all spatial information contained in neighbouring pixels in the image is ignored. The main justification in disregrading this information being mathematical simplicity and computational convenience. Image metrics are however a crucial and fundamental component of many imaging algorithms. In fact this philosophy pervades much of image analysis algorithms. Namely, if a technique is seen to work adequately on a small number of test examples, and if it is easy to implement, then it is deemed suitable.

The main aim of this thesis is to place such algorithms or techniques in the realm of statistics. The hope is that in so doing these techniques will have a much more solid theoretical basis, which may in turn lead to future improvements and developments. The primary tool used is random set theory.

If the image to be analysed is random then random sets provides a natural setting in which to make inference. For instance if a sample of images is to be analysed, then it may be appropriate to model this sample as a realisation of a random set. This approach could be suitable to average images or to filter images. This is the setting in Chapter 5, where sequences of images are 'averaged' locally with the intention of smoothing the image sequence, a process we call set-valued regression.

On the other hand, random sets can be used to treat a deterministic grey-scale image as a random binary image. This idea is illustrated in Chapter 1.

Chapter 1 introduces some notation and conventions used throughout. In particular it introduces the notion of a random set model corresponding to a grey-scale image, which is used extensively in Chapters 2 and 3.

Chapter 2 explains the thresholding problem, beginning with a summary of techniques presented throughout the literature. A new thresholding technique is presented based on expectations of the random set model outlined in Chapter 1.

Chapter 3 concerns the problem of designing image metrics or measures of dissimilarity between images. This is a vital concept in many area of image processing. This chapter begins with a discussion of commonly used image metrics. A new approach to finding image metrics is illustrated, based on exploring distances between distributions of random sets, using the random set model from Chapter 1.

Chapter 4 concerns the Bayesian image restoration problem,that is, restoring a noisy image from knowledge of the noise degradation and prior information about the true image. We comment on widely used approaches to solving this problem. In particular we show how image metrics may be used in this context.

Chapter 5 examines a problem which we term set-valued regression. This can be conveniently thought of as a regression problem where the response variable is now set-valued. We use a loss function approach to examining this problem, illustrating with examples.

All algorithms were implemented in C++ using CLIP, a C++ library for image processing, written by Adri Steenbek from CWI (Amsterdam). All statistical calculations were carried out using `S-Plus`.

Much of the research presented in this thesis has already been published. The new thresholding technique described in Section 2.4 has been presented in [18]. While the grey-scale image metric described in Section 3.4 appears in [17]. This grey-scale metric has been applied to the Bayesian image restoration problem in Section 4.5. Details of this research have been presented in [16].

Acknowledgements

It is my great pleasure to record thanks to the many people who helped in many ways to contribute to this thesis.

I am deeply indebted to my supervisor, Prof. Ilya Molchanov, for all the time and energy he gave to the completion of this thesis. I have gained so much from his help and expertise. Without him this thesis would not have been possible.

I would also like to thank Prof. Adrian Bowman and all the members of the Statistics department for making me feel so welcome during my stay here.

During the past three years I have had the pleasure of beginning many valuable friendships. Thanks to everyone for helping me in so many different ways.

I would like to thank my brother and sisters for all their support and encouragement. Finally I want to express thanks to my parents for all their love and support throughout the years.

Ba mhaith liom an tràchtas seo a thiomnù do mo mhuintir.

Do mo mhàthair agus do m'athair.

Contents

List of Tables

List of Figures

Chapter 1

Preliminaries

1.1 Images and functions

An image f is simply a digital representation of a scene. When a picture is digitised, so that it can be interpreted by a computer, it undergoes a quantisation process whereby the picture is split into small picture elements or pixels. Each pixel is then given a single intensity (or brightness) which best represents that part of the picture. Grey-scale images use intensities represented by various shades of grey. Zero intensity is seen as black, low intensities as dark grey, with the shade of grey lightening as the intensity increases. The maximal intensity is perceived as white. Black pixels are typically given the value 0, and white pixels a maximal value of, say 1. All intermediate pixels are assigned values between 0 and 1. In applications maximum values are typically 255.

For colour images, each pixel is given a three dimensional vector, representing the intensity of the three colours red, green and blue, which compose the colour of the pixel. Throughout this thesis we are concerned solely with grey-scale images.

Further we typically assume that each image f is defined on a window W which is a subset of a Euclidean space, and that each point $x \in W$ is assigned a grey value $f(x)$ which lies between 0 and 1. This is similar to the situation in fuzzy set theory [37], where $f(x)$ is called the degree of membership of x in f.

We further assume that f is upper semicontinuous.

Definition 1.1 (see [47]). A function $f(x)$, $x \in W$ is **upper semicontinuous** when for every x and every $t \geq f(x)$ there exists a neighbourhood V_x of x such that $f(y) < t$ for every $y \in V_x$.

Similarly we may define lower semicontinuity:

Definition 1.2. A function $f(x)$, $x \in W$ is **lower semicontinuous** when for every x and every $t \leq f(x)$ there exists a neighbourhood V_x of x such that $f(y) > t$ for every $y \in V_x$.

Upper semicontinuity ensures the measurability of f and guarantees that the subgraph (often called the hypograph or umbra) of f is a topologically closed set [47].

Definition 1.3. A **subgraph** Γ_f for an image f may be defined as:

$$\Gamma_f = \{(x,t) : x \in W, t \in [0,1] \text{ and } f(x) \geq t\}.$$

Simply, this is the set of all pairs $(x,t) \in W \times [0,1]$ lying between the graph of f and the plane $t = 0$. See Figure 1.1 below.

Working in the continuous framework permits the use of analytical tools, while always allowing the possibility of 'discretising' the problem. This is of course the situation in most practical situations where W is a discretised window or set of pixels and f takes discrete grey levels from a set $GL = \{0, 1, \ldots, l-1\}$ (typically $l = 256$).

The *cumulative histogram* of f is denoted $H_f(t)$. In the discrete setting $H_f(t_2) - H_f(t_1)$ equals the number of pixels with grey levels from the interval

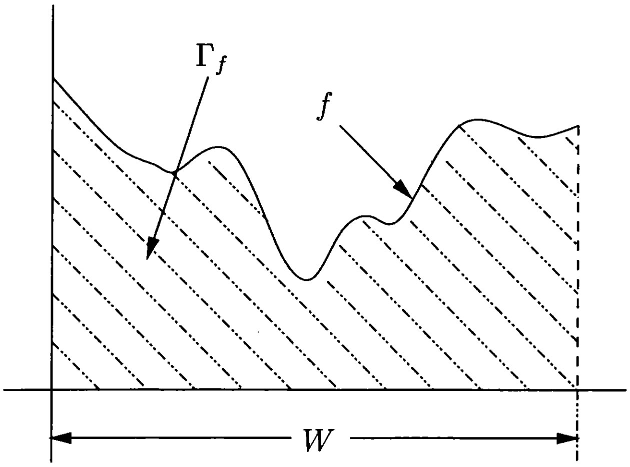


Figure 1.1. Hypograph of image f

$(t_1, t_2]$. In the continuous framework the number of pixels is replaced by a measure of the subset. For W being a subset of the Euclidean space $\mathbb{R}^d$, the d-dimensional Lebesgue measure μ_d is the most natural choice for the basic reference measure. This corresponds to area if $d = 2$ and to volume if $d = 3$.

For each $0 \leq t \leq 1$ define,

$$H_f(t) = \mu_d(\{x \in W : f(x) \leq t\}).$$

It is clear that $H_f(t)$, $0 \leq t \leq 1$, is a non-decreasing right-continuous function and $H_f(1) = \mu_d(W)$. If for some function h_f,

$$H_f(t) = \int_0^t h_f(s)ds,$$

then the function h_f is called the *histogram* of f. Of course in the discrete framework, the integral is replaced by a sum and $h_f(t)$ equals the number of pixels with grey level t. In fact $H_f(\cdot)/\mu_d(W)$ may be viewed as a cumulative distribution function of a random variable ζ_f taking values in $[0,1]$, thus $h_f(\cdot)/\mu_d(W)$ (if $h_f(\cdot)$

exists) becomes its probability density function.

A binary image f differs from a grey-scale image in that it only contains two grey levels, conventionally denoted 0 and 1. Thus f may be identified uniquely with the closed subset $F \subseteq W$, defined as:

$$F = \{x \in W : f(x) = 1\}.$$

Thus while a grey-scale image may be thought of mathematically as a function, a binary image may be considered as a set $F \subseteq W$.

1.2 Histogram equalisation

Often a displayed image may contain some details of interest which are not clearly visible. The grey levels in this area may, for example, all have low values, so that subtle differences between pixel values may not be so apparent. In this case a transformation of the grey-scale range of the image may enhance the contrast in the image.

An anamorphosis [47, p.435] is defined as a transformation τ of the grey-scale range:

$$\tau : [0,1] \rightarrow [0,1],$$

defined such that τ is increasing and continuous. Some examples of anamorphoses are, $\tau[f(x)] = [f(x)]^2$, $\tau[f(x)] = \sqrt{f(x)}$.

Histogram equalisation [42] is a further example of an anamorphosis. It is a transformation of the grey-scale range of an image, which aims to transform the grey-scale range of an image in order to produce an image with equal or uniformly distributed grey values. Histogram equalisation enhances contrast for grey values close to the histogram maxima and decreases contrast near the histogram minima. Figure 1.2 shows an input image and its corresponding equalised version.

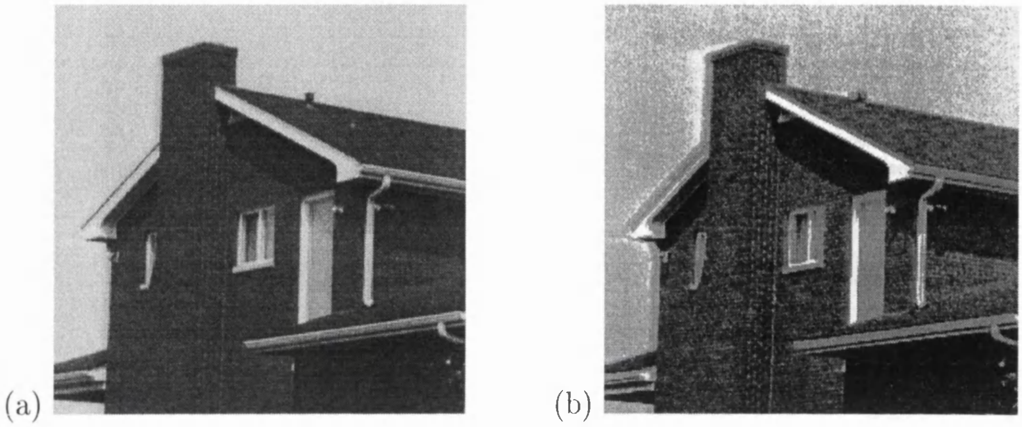


Figure 1.2. (a) house image, (b) equalised house image.

We derive this transformation as follows. Denote the input histogram, as before, by $h_f(t)$. Our intention is to find a monotonic grey-scale transformation, $t = \tau(s)$, such that the desired equalised histogram $h_f^*(t)$, is uniform over the entire grey-scale range. Thus we require,

$$\int_0^s h_f(x)dx = \int_0^t h_f^*(x)dx, \tag{1.1}$$

where $h_f^*(t)$ has the constant value $\mu_d(W)$. Thus (1.1) becomes,

$$H_f(s) = \mu_d(W)t.$$

Solving the above we see that the transformation τ may be derived as,

$$t = \tau(s) = \frac{1}{\mu_d(W)} H_f(s). \tag{1.2}$$

In practice where we work in the discrete setting, (1.2) is modified so that $\mu_d(W)$

corresponds to the number of pixels in the window and $H_f(s) = \int_0^s h_f(x)dx$ is approximated by a sum. Therefore we see that the resulting histogram is not ideally equalised. Figure 1.3 below shows the histograms of the original and equalised images from Figure 1.2, while Figure 1.4 shows the cumulative histograms of both images. It is clear that the equalised histogram is much more equally spread over the grey-scale range than the original histogram, although it is not exactly constant.

It is worth noting that histogram equalisation has parallels to the so-called probability integral transform. If X is a continuous random variable with cumulative distribution function F_X, then the random variable $F_X(X)$ is distributed uniformly on $[0, 1]$. This transformation is not possible where X is a discrete random variable, analogous to the situation described in Figure 1.3 where exact equalisation is not possible when working with discrete images.

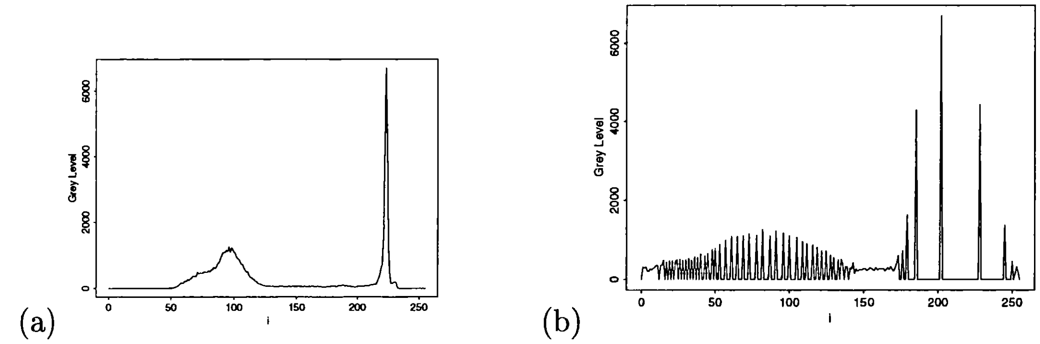


Figure 1.3. Histogram of house image (a) before and (b) after equalisation.

1.3 Distance transform

The distance transform (DT) is a widely used tool in the analysis of binary images. This idea was first introduced by Rosenfeld and Pfaltz [43] and later extended by Borgefors [5, 6]. Subsequently distance transforms have been widely

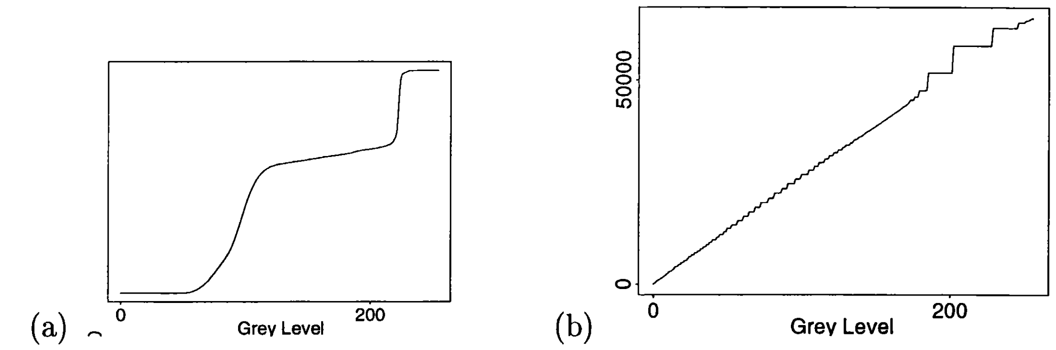


Figure 1.4. Cumulative histogram of house image (a) before and (b) after equalisation.

used in many areas of images processing, for example, in image thresholding [18], image metrics [2, 17], averaging of random sets [3] and image restoration [19, 44].

For any closed set $X \in \mathbb{R}^d$, all points in $\mathbb{R}^d$ can be classified according to their positions with respect to X. This can be achieved by considering the distance of each point to X. The following definition is a generalisation.

Definition 1.4 (see [3]). A function $d(x, X)$ with the first argument being a point $x \in \mathbb{R}^d$ and the second one being a closed subset $X \subset \mathbb{R}^d$ is said to be a **(generalised) distance function** if it is lower semicontinuous with respect to its first argument, measurable with respect to its second argument, and satisfies the following two conditions.

1. If $X_1 \subset X_2$, then $d(x, X_1) \geq d(x, X_2)$ for all $x \in \mathbb{R}^d$ (monotonicity).

2. $X = \{x : d(x, X) \leq 0\}$ (consistency).

The following are examples of distance functions, some which will be used later:

Example 1.1. The **Euclidean distance function** $d(x, X)$ is equal to the distance from $x \in \mathbb{R}^d$ to the nearest point of X. More precisely:

$$d(x, X) = \rho(x, X) = \inf\{\|x - y\| : y \in X\}, \quad x \in \mathbb{R}^d .$$

Example 1.2. The **signed distance function** is defined as:

$$d(x,X) = \begin{cases} \rho(x,X)\,, & x \notin X\,, \\ -\rho(x,X^c)\,, & x \in X\,. \end{cases}$$

Here X^c denotes the complement of X in $\mathbb{R}^d$. Note that a mathematical analysis of signed distance functions can be found in [13].

Example 1.3. The **square distance function** is defined as:

$$d(x,X) = \rho^2(x,X).$$

Note that in Examples 1.1-1.3 the metric ρ may not necessarily be the Euclidean metric, although henceforth it will be assumed to be so.

A binary image f corresponds uniquely to the closed set F defined as:

$$F = \{x \in W : f(x) = 1\},$$

so that F is the set of foreground pixels in the image f. The **distance transform** of the binary image f (or the corresponding set F) is then the grey-scale image having pixel values

$$d(x,F), \quad \text{for all } x \in W,$$

where $d(\cdot,F)$ could be, for example, any one of Examples 1.1-1.3 above. So $d(x,F)$ is then the (generalised) distance from pixel x to the nearest foreground pixel of F.

In Figure 1.5 below we see a binary image and its corresponding Euclidean distance transform.

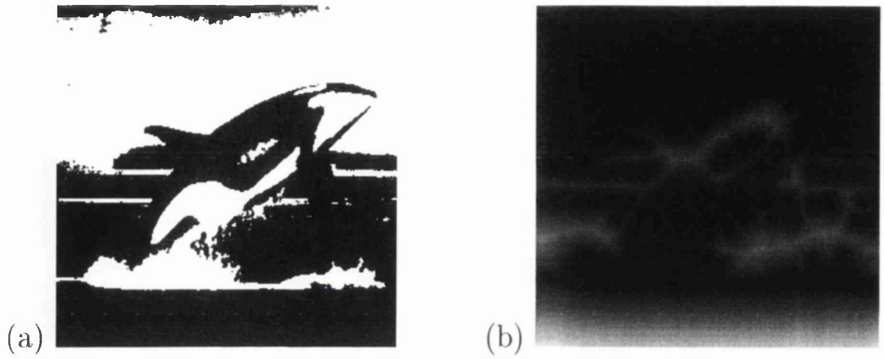


Figure 1.5. Binary image and corresponding distance transform

Working in the discrete set-up necessitates estimating Euclidean distances with suitable approximations. The reason is twofold. Firstly because pixel values of the transformed image must be given integer values and secondly because this leads to a much faster implementation.

Borgefors [5, 6] studied several discrete approximations to the Euclidean distance. Two such choices, known as the chamfer $(3, 4)$ and chamfer $(5, 7, 11)$ have maximum relative errors of 8.09% and 2.02% respectively. Each such approximation is implemented in a two stage algorithm.

DT's are implemented as follows. Object pixels are denoted by the value 0 and background pixels by the value 255. Suppose for the present that vertical and horizontal distances from background to object pixels are denoted by the value 3 and similarly that diagonal distances are denoted by the value 4. Figure 1.6 illustrates this scheme.

The distance transform shown in Figure 1.6 is known as the $(3, 4)$ chamfer DT. The first step in the computation of a DT is to define its corresponding distance matrix. Figure 1.7 shows the distance matrix for the $(3, 4)$ chamfer DT,

(a)

255	255	255	255	255
255	255	0	255	255
255	255	0	0	255
255	0	0	255	255
255	255	255	255	255

(b)

7	4	3	4	7
6	3	0	3	4
4	3	0	0	3
3	0	0	3	4
4	3	3	4	7

Figure 1.6. (a) Binary image (b) Corresponding distance transform

and the (5, 7, 11) chamfer DT.

(a)

4	3	4
3	0	3
4	3	4

(b)

	11		11	
11	7	5	7	11
	5	0	5	
11	7	5	7	11
	11		11	

Figure 1.7. (a) Distance matrix corresponding to the (3, 4) chamfer DT (b) (b) Distance matrix corresponding to the (5, 7, 11) chamfer DT

The DT is implemented by performing two passes (forward and backward) over the input image, thus computing distance values for all the pixels. The forward pass is left to right, top to bottom and the backward pass is right to left, bottom to top. Half of the distance matrix is used in each pass, as illustrated in Figure 1.8.

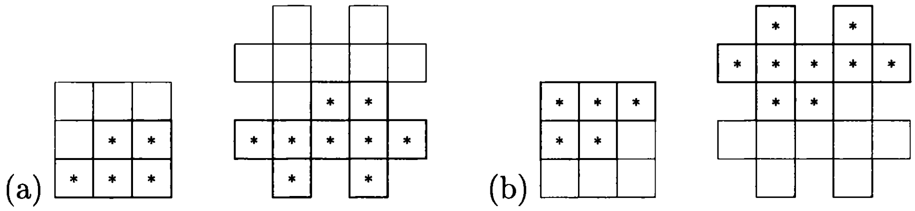


Figure 1.8. (a)Top half of distance matrices used in forward pass (b)Bottom half of distance matrices used in backward pass

At successive stages of the forward and backward passes, a series of sums is formed. These sums are computed by adding the value of a pixel in the image to the corresponding value in the top (bottom) half of the distance matrix. The pixel in the image corresponding to centre pixel in the distance matrix is replaced with minimum of these sums. After the two passes the resultant image is the distance transform of the original image.

1.4 Random sets generated by grey-scale images

This section outlines an interpretation of a grey-scale image as the distribution of a random set. This section is fundamental to much of the discussion in the subsequent chapters.

A random closed set may be thought of as a random element whose values are closed sets. It is easy to appreciate many examples, including, for example, a random disc centred at a random point with a random radius, a convex hull of random points from a certain distribution. For instance in an imaging context, assign to each pixel a random grey value from a given set of grey levels. The resultant set of pixels is a random set. Many more examples abound.

To mathematically define a random closed set in $\mathbb{R}^d$, the first step is to equip the space $\mathcal{F}$ of closed sets in $\mathbb{R}^d$ with a σ-algebra. This σ-algebra, $\sigma_{\mathcal{F}}$, is the minimum σ-algebra containing the family of sets,

$$\mathcal{F}_K = \{F \in \mathcal{F} : F \cap K \neq \emptyset\},$$

where K is any compact set in $\mathbb{R}^d$. So $\mathcal{F}_K$ is the family of closed sets which intersect K. A random closed set is then a measurable map from a given probability space to the space $\mathcal{F}$. The reader is referred to [31] for a discussion of random sets.

Define the thresholded set of an image f at level $t \in [0, 1]$ as,

$$F_t = \{x \in W : f(x) \geq t\}. \tag{1.3}$$

This is a closed set since f is assumed to be upper semicontinuous [47, p.426]. The key observation is that this deterministic thresholded set becomes a random closed set if the threshold is chosen at a random level. That is, replace the index t in (1.3) with a random variable U distributed on $[0, 1]$, the set of grey levels. In effect this allows us to treat deterministic grey-scale images as random binary images.

Proposition 1.1. *Let $K \subset \mathbb{R}^2$ be any compact set. Then $\{F_U \cap K \neq \emptyset\}$ is measurable with respect to a σ-algebra generated by U so that F_U is a random closed set.*

Proof. $\{F_U \cap K \neq \emptyset\} = \{\sup_{x \in K} f(x) \geq U\}$ is measurable since U is a random variable. □

The distribution of the random variable U determines the type of weighting associated with the random set model. For example choosing a uniformly distributed random variable on $[0, 1]$ ensures that all thresholded sets are given equal weighting. We call this a *uniformly weighted* random set model. Another useful model arises if U is distributed according to the histogram of the image, so that,

$$\mathbf{P}\{U \geq t\} = \mu_d(F_t)/\mu_d(W)$$

where μ_d is the d-dimensional Lebesgue measure. This is called the *histogram weighted* random set model. Another possibility could be to assume a probability density proportional to the $(d-1)$-dimensional Hausdorff measure of the boundary of F_t. Of course many other possibilities exist.

In fact the distribution of the uniformly weighted model is immediately related to the image f as follows:

$$\mathbf{P}\{x \in F_U\} = \mathbf{P}\{U \leq f(x)\} = f(x), \quad x \in W$$

The histogram weighted model is advantageous since the distribution of F_U is invariant with respect to all increasing continuous transformations of grey values, that is, anamorphoses. It is important to note that for this model F_U is almost surely not empty. This is not true for the uniformly weighted model, which can lead to random sets with possible empty values.

There is an immediate connection between the uniform and histogram weighted random set models. The histogram random set model of an image f is equivalent to the uniformly weighted random set model of the equalised image f. This is clear since equalisation transforms the grey-scale image to one with uniformly distributed grey levels.

Chapter 2

Thresholding and Random Sets

2.1 Introduction

Thresholding is an important technique in image processing. It is typically used to separate objects in an image from its background. In automatic thresholding an appropriate threshold level t^* is chosen and all pixels satisfying $f(x) < t^*$ are then typically given the value 0 and classified as background pixels. All pixels satisfying $f(x) \geq t^*$ are typically given the value 1 and classified as foreground pixels. So it is seen that thresholding is an operation which transforms a grey-scale image into a binary image.

Many techniques have been presented for thresholding. These techniques can be placed roughly into two categories, namely global and local thresholding. Local thresholding divides an image into several subimages and then finds threshold levels for each subimage. Global thresholding on the other hand is one that finds a single threshold level for the entire image. We concentrate throughout on global thresholding.

This chapter begins with an overview of thresholding techniques which have appeared throughout the literature. From Section 2.4 onwards a new thresholding technique is introduced which is based on the representation of an image via its random level set interpretation.

2.2 Survey of thresholding techniques

This section includes a summary of popular thresholding methods which have appeared in the literature. Commonly these are based solely on the histogram of the image often using little other information contained in the image [27, 28, 46]. We classify these as histogram based techniques. An obvious criticism of these methods is that they each give the same threshold level for different images with similar histograms. For instance, each of these methods is of little use in the situation where the image histogram is flat (or uniform).

Still other methods use information from local changes in pixel intensities, thus utilising more information contained in the image [9, 36]. Specifically these methods use information from the co-occurrence matrix $M = [m_{ij}]_{l \times l}$, where m_{ij} represents the frequency of occurrence for two neighbouring pixels with grey-levels i and j in some predefined manner. Usually only the four adjacent north, south, east and west neighbours are considered. This dramatically reduces computational complexity and in fact has been noted not to adversely effect resultant thresholded images.

The following survey is not intended to be in any way exhaustive.

2.2.1 Histogram based techniques

Threshold selection is easier for images that have bimodal histograms, for example, Figure 2.1. In this instance the object is clearly distinguishable from the background and the threshold level is simply chosen in the valley of the two peaks. Of course in practice most image histograms are not bimodal. For example, the

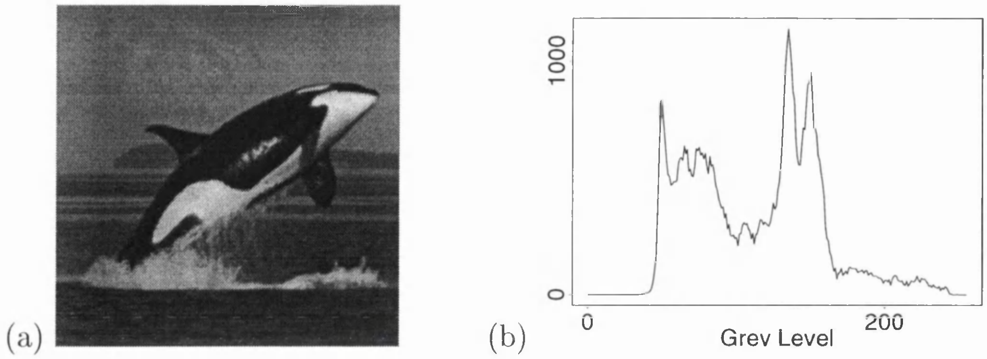


Figure 2.1. (a) `orca` image, (b) histogram of `orca` image.

`airport` image in Figure 2.2 has a multimodal histogram. Many techniques have been presented to modify (or transform) the input histogram to the situation where it becomes nearly bimodal. For instance, this can be done by weighting the histogram in a certain manner so that its peaks become pronounced and its valley deeper. These methods are in general quite ad-hoc.

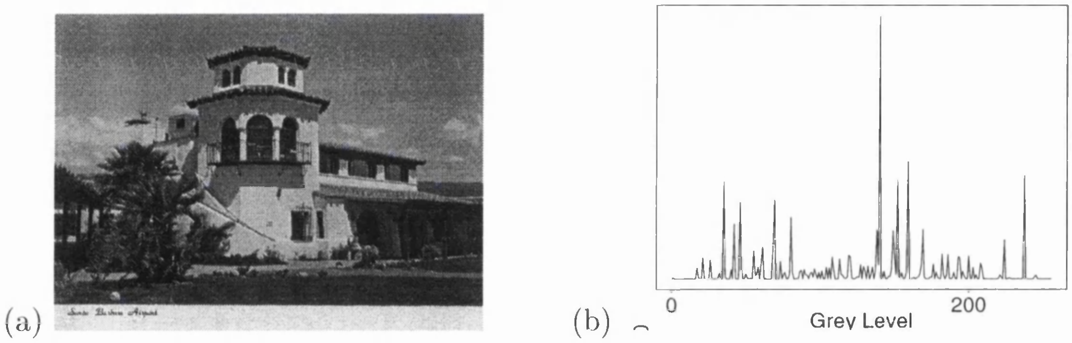


Figure 2.2. (a) `airport` image, (b) histogram of `airport` image.

Suppose an image is known to consist of distinct object and background regions. If it is assumed that $100(1-p)\%$ of pixels in an image f are object pixels, then it is straightforward from the histogram of the image, to find the threshold level t such that the cumulative distribution function of the image is as close as possible to $1-p$. This method is known as the *p-tile* method. This technique could be used to threshold, for example, a text image, where it known that the characters in the image typically occupy 15% of the image, so that $p = 0.15$. Of course this method has severe limitations.

Entropic thresholding Suppose it is known that the object pixels in an image occupy a proportion p of the image and the background pixels $1 - p$. Suppose a pixel $x \in W$ is chosen at random. Following Shannon [48], the uncertainty about whether x is a object or background pixel is measured by its entropy:

$$H(x) = -p \log(p) - (1-p)\log(1-p).$$

Here and throughout this chapter $\log(\cdot)$ denotes logarithm to base 2. As more information becomes available, such as grey-level of pixel x, $H(x)$ should decrease. Entropic thresholding has received much attention in the literature. Below we describe some results.

Entropic thresholding based solely on the histogram of the image relies on the assumption that values for object and background pixels follow two distinct discrete probability distributions. Here image f is assumed to take discrete values from $\{0, 1, \ldots, l-1\}$. As before denote the histogram of the input image at level s by, $h_f(s)$, the number of pixels in f with grey level s. Further let,

$$p_i = \frac{h_f(i)}{n(W)}, \quad i = 0, 1, \ldots, l-1$$

where $n(W)$ equals the number of pixels in the window. Thus $\sum_0^{l-1} p_i = 1$.

Let us suppose that the image is thresholded at level t, so that the a posteriori frequency of grey levels in the foreground and background of the image are P_t

and $1 - P_t$ respectively, where,

$$P_t = \sum_{i=0}^{t} p_i.$$

The a posteriori entropy of this thresholded image at level t is calculated as:

$$H(t) = -P_t \log P_t - (1 - P_t) \log (1 - P_t). \tag{2.1}$$

This entropy value gives a measure of the information content of the thresholded image. Clearly the value t which maximises this entropy would serve as an appropriate criteria. However it is straightforward to show that this occurs when $P_t = 1 - P_t = 1/2$, giving an equal number of foreground and background pixels, a naive threshold.

Pun [39] redressed this situation by maximising an upper bound of (2.1) formulated as:

$$\tilde{H}(t) = -\frac{\sum_0^t p_i \log p_i}{\log[\max(p_0, p_1, \dots, p_t)]} \log P_t - \frac{\sum_{t+1}^{l-1} p_i \log p_i}{\log[\max(p_{t+1}, p_{t+2}, \dots p_{l-1})]} \log (1 - P_t). \tag{2.2}$$

Since

$$-P_t \geq -\frac{\sum_0^t p_i \log p_i}{\log[\max(p_0, p_1, \dots, p_t)]}$$

and

$$-(1 - P_t) \geq -\frac{\sum_{t+1}^{l-1} p_i \log p_i}{\log[\max(p_{t+1}, p_{t+2}, \dots p_{l-1})]},$$

it is clear that $\tilde{H}(t) \leq H(t)$. The level t which maximises $\tilde{H}(t)$ is chosen as the threshold level. However no justification is given as to why this is an appropriate criterion function. In fact it is pointed out in [36] that the maximum value of (2.2) may correspond to an inappropriate value in view of (2.1).

In another algorithm, Pun [40] defines an anisotropy coefficient α, as:

$$\alpha = \frac{\sum_{i=0}^{m} p_i \log(p_i)}{\sum_{i=0}^{l-1} p_i \log(p_i)},$$

where m is the smallest integer satisfying

$$\sum_{i=0}^{m} p_i \geq 0.5.$$

Therefore m is the least grey level dividing the histogram of the image into two (almost) equal parts. The threshold level t is chosen so that

$$\sum_{i=0}^{t} p_i = \begin{cases} 1-\alpha, & \text{if } \alpha \leq 0.5, \\ \alpha, & \text{if } \alpha > 0.5. \end{cases}$$

It has been noted in [27], that this always results in a threshold with $t \geq m$. That is a thresholded image with less object pixels than foreground pixels, thus introducing a bias.

Suppose t is the threshold level which separates the background and foreground pixels into two distinct regions, F and B respectively. Then the probability distributions for F and B may be denoted as:

$$\begin{aligned} F &: \frac{p_0}{P_t}, \frac{p_1}{P_t}, \ldots, \frac{p_t}{P_t} \\ B &: \frac{p_{t+1}}{1-P_t}, \frac{p_{t+2}}{1-P_t}, \ldots, \frac{p_{l-1}}{1-P_t}. \end{aligned}$$

Kapur et al. [27] compute the sum of the entropies associated with each distribution. This sum is then maximised to give the appropriate threshold level.

The entropy associated with the foreground region is as follows:

$$H(F) = -\sum_{i=0}^{t} \frac{p_i}{P_t} \log\left(\frac{p_i}{P_t}\right).$$

Similarly the entropy associated with the background may be formulated as follows:

$$H(B) = -\sum_{i=t+1}^{l-1} \frac{p_i}{1-P_t} \log\left(\frac{p_i}{1-P_t}\right).$$

Define

$$H(t) = H(F) + H(B).$$

Maximising $H(t)$ obtains the maximum information between foreground and background distributions in the image.

It is interesting to note here that if both distributions are uniform, so that the image is ideally equalised ($p_i = 1/l - 1$), then this gives the trivial result that the threshold level is $t = l/2$. This is of course not intuitively appealing. The distance threshold method which we will meet in Section 2.4, does not encounter any such problems.

Otsu's method After thresholding an image at threshold level t suppose that pixels are partitioned into regions C_0 and C_1 (object and background pixels). Otsu [35] used ideas from discriminant analysis to find the rule (grey level) which best separates the means from the two classes. Specifically the criterion function

$$C(t) = P_t(1 - P_t)(\mu_1 - \mu_0)^2$$

is evaluated for $t = 0, 1, \ldots, l-1$. The value of t at which this function is maximised is set as the threshold level. Here, as previously $P_t = \sum_{i=0}^{t} p_i$ and

$$\mu_0 = \frac{\sum_{i=0}^{t} ip_i}{P_t}, \quad \mu_1 = \frac{\sum_{i=t+1}^{l-1} ip_i}{1-P_t}.$$

Maximising the criterion function $C(t)$ has the effect of classifying the object and background pixels into two classes, such that the means of their respective classes are separated as far as possible. The threshold level is found to be midway between the two means.

Minimum error thresholding Kittler and Illingworth [28] approached the thresholding problem by assuming that grey levels of both object and background pixels follow normal distributions $f_o(t) \sim N(\mu_0, \sigma_0^2)$, $f_b(t) \sim N(\mu_1, \sigma_1^2)$ respectively. The overall probability distribution function (or normalised histogram of the entire image) f_T is a mixture of the two normal distributions and is assumed to be of the form,

$$f_T(t) = \alpha f_o(t) + (1-\alpha) f_b(t), \quad 0 \leq \alpha \leq 1, s \in GL. \tag{2.3}$$

Here α is the so-called mixing ratio (proportion of pixels in C_0). Kittler and Illingworth [28] described an iterative algorithm to find estimates of the parameters $(\hat{\mu}_0, \hat{\mu}_1, \hat{\sigma}_0, \hat{\sigma}_1, \hat{\alpha})$, while simultaneously finding the value t satisfying

$$\hat{\alpha}\hat{f}_0(t) = (1-\hat{\alpha})\hat{f}_1(t),$$

where $\hat{f}_0(t) \sim N(\hat{\mu}_0, \hat{\sigma}_0^2)$ and similarly for $\hat{f}_1(t)$. This method minimises the number of misclassified pixels from the object and background distributions.

An obvious criticism of this approach is the assumption that the object and background pixels are distributed as the mixture of two Gaussian distributions. Certainly from inspection, many image histograms would seem to deviate largely from a mixture of two Gaussian distributions. Indeed it is unclear how this algorithm would deal with the case of an image with a uniform histogram (an equalised image).

Some papers [23, 29] have examined the performance of the above histogram based techniques and others by generating a series of synthetic histograms, rather than images themselves. These histograms are assumed to a mixture of two Gaussian distribution functions. Thus a wide variety of differing histograms may be synthesised given various values for the parameters $(\alpha, \mu_0, \sigma_0, \mu_1, \sigma_1)$, namely the mixing ratio, mean and variance of the object pixel distribution, mean and variance of the background pixel distribution.

2.2.2 Region based techniques

As has been pointed out previously, the major drawback of histogram based techniques is the fact that they do not take into account the spatial distribution of the grey values in the image. Different images with similar histograms will be given the same threshold value. Region based methods utilise the spatial dependency of grey levels. Typically these techniques are based on entropy considerations. We comment on two such methods.

In common with both is the notion of a co-occurrence matrix. This is defined as an $l \times l$ matrix, $M = [m_{ij}]_{l \times l}$, where entry m_{ij} represents the transition in the image from grey level i to grey level j in a specified fashion.

Let us suppose that given an image of dimension $M \times N$, $f(l, k)$ represents the grey level of f at pixel (l, k).[1] Following [9, 36], we define the (i, j)th entry

[1] Here we deviate from convention used before, by denoting pixels as pairs of coordinates.

in the co-occurrence matrix as:

$$m_{i,j} = \sum_{l=1}^{M}\sum_{k=1}^{N}\delta(l,k), \tag{2.4}$$

where

$$\begin{array}{ll} \delta(l,k) = 1, & \text{if} \begin{cases} f(l,k) = i & \text{and} \quad f(l,k+1) = j, \\ & \text{or} \\ f(l,k) = i & \text{and} \quad f(l+1,k) = j; \end{cases} \\ \delta(l,k) = 0, & \text{otherwise.} \end{array}$$

Thus it is seen that m_{ij} considers transitions from grey levels i to j between adjacent pixels horizontally right and vertically below. It is possible to make the matrix M symmetric by also considering horizontally left and vertically above transitions. However it has been noted in [36] that this does not significantly improved the information content.

Each entry in the co-occurrence matrix may be normalised to obtain the transition probability:

$$p_{ij} = m_{ij} \Bigg/ \left(\sum_{i=0}^{l-1}\sum_{j=0}^{l-1} m_{ij}\right). \tag{2.5}$$

It is clear that a threshold t divides the matrix into four submatrices as shown in Figure 2.3.

Quadrants A and C represent local transitions within background and object respectively, while quadrants B and D represent transitions from background to object and vice versa. The probabilities associated with each quadrant may be

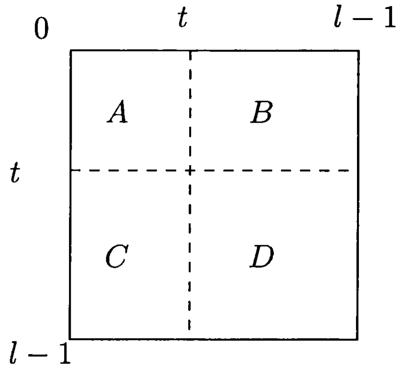


Figure 2.3. Co-occurrence matrix partitioned by threshold t.

formulated as:

$$\begin{aligned} P_A(t) &= \sum_{i=0}^{t}\sum_{j=0}^{t} p_{ij}, & P_B(t) &= \sum_{i=0}^{t}\sum_{j=t+1}^{l-1} p_{ij}, \\ P_C(t) &= \sum_{i=t+1}^{l-1}\sum_{j=t+1}^{l-1} p_{ij}, & P_D(t) &= \sum_{i=t+1}^{l-1}\sum_{j=0}^{t} p_{ij}. \end{aligned} \tag{2.6}$$

Finally each entry in each quadrant may be further normalised by the quadrant probabilities (2.6) as follows:

$$p_{ij}^A = p_{ij}/P_A \quad \text{for} \quad 0 \le i \le t,\ 0 \le j \le t, \tag{2.7}$$

$$p_{ij}^B = p_{ij}/P_B \quad \text{for} \quad 0 \le i \le t,\ t+1 \le j \le l-1, \tag{2.8}$$

$$p_{ij}^C = p_{ij}/P_C \quad \text{for} \quad t+1 \le i \le l-1,\ t+1 \le j \le l-1, \tag{2.9}$$

$$p_{ij}^D = p_{ij}/P_D \quad \text{for} \quad t+1 \le i \le l-1,\ 0 \le j \le t. \tag{2.10}$$

So p_{ij}^A for $0 \le i \le t$, $0 \le j \le t$, represents a probability of background to background transitions. Similarly for (2.8), (2.9), (2.10). Pal and Pal [36], and Chang et al.[9] use these formulations as the basis for their various entropy methods.

Local and conditional entropy Pal and Pal[36] define the local entropies of background and object regions respectively as

$$H_B(t) = -\frac{1}{2}\sum_{i=0}^{t}\sum_{j=0}^{t} p_{ij}^{A} \log p_{ij}^{A} \tag{2.11}$$

$$H_O(t) = -\frac{1}{2}\sum_{i=t+1}^{l-1}\sum_{j=t+1}^{l-1} p_{ij}^{C} \log p_{ij}^{C}. \tag{2.12}$$

Both (2.11) and (2.12) correspond to a posteriori entropies at threshold levels t and so are functions of t. The local entropy of the image is now defined as the sum of the local entropies of the foreground (object) and background and is formulated as:

$$H(t) = H_B(t) + H_O(t) \tag{2.13}$$

The value of t which maximises (2.13) is chosen as the appropriate threshold level.

Quadrants B and D correspond to probabilities of transitions of adjacent pixels from grey levels in the background to the object and vice versa. In effect the entropy associated with the probabilities p_{ij}^{B} for $0 \leq i \leq t$, $t+1 \leq j \leq l-1$ describes the average amount of information gained from the object given that one has viewed the background. It is formulated as:

$$H(O|B) = -\sum_{i=0}^{t}\sum_{j=t+1}^{l-1} p_{ij}^{B} \log p_{ij}^{B}. \tag{2.14}$$

Similarly,

$$H(B|O) = -\sum_{i=t+1}^{l-1}\sum_{j=0}^{t} p_{ij}^{D} \log p_{ij}^{D}. \tag{2.15}$$

The conditional entropy of the image is then defined as:

$$H(t) = \left(H(O|B) + H(B|O)\right)/2. \tag{2.16}$$

The maximiser of (2.16) serves as the appropriate threshold level.

Relative entropy Given two probability distributions p_i and p_i^*, $0 \leq i \leq l-1$, the relative entropy of p relative to p^* is defined as,

$$L(p, p^*) = \sum_{i=0}^{l-1} p_i \log \frac{p_i}{p_i^*}$$

In essence $L(p, p^*)$ determines a measure of distance between the two distributions. Note that in conventional statistics this distance is termed the Kullback-Leiber distance.

Chang et al. [9] extended some ideas from [36] incorporating the idea of relative entropy. Their criteria for thresholding an image is to find a thresholded image best matching the original image in some sense. Specifically they consider the co-occurrence matrix $[m_{ij}]_{l \times l}$ as corresponding to a transition probability distribution of the original image, and similarly define a co-occurrence matrix for a thresholded image at the same level t. Thus the relative entropy between two such distributions serves as a measure of how the threshold image matches the original image.

An appropriate criteria is then to find the grey level t such that the relative entropy is minimised.

2.3 Thresholding and expectations of random sets

Each image f is comprised of a nested family of thresholded sets $\{F_t\}$, $0 \leq t \leq 1$. In Section 1.4 it was shown that F_U is a random set if U is a random variable. Now the problem of finding the appropriate threshold level may be stated as a problem of choosing a set F_{t^*} from the sample of thresholded sets $\{F_t\}$, $0 \leq t \leq 1$, which best summarises this sample (or indeed image f). In classical statistics, populations or samples are summarised by means or expectations. In turn this suggests that the appropriate thresholded set might be found by exploring expectations of F_U.

In general the problem of defining an expectation of a random set $X \subset \mathbb{R}^d$ is difficult. The usual approach is to embed the space of sets of interest into a linear space. It is now possible to define an expectation in this space, and then try to map this expectation back to the space of sets of interest. This idea is explained graphically in Figure 2.4.

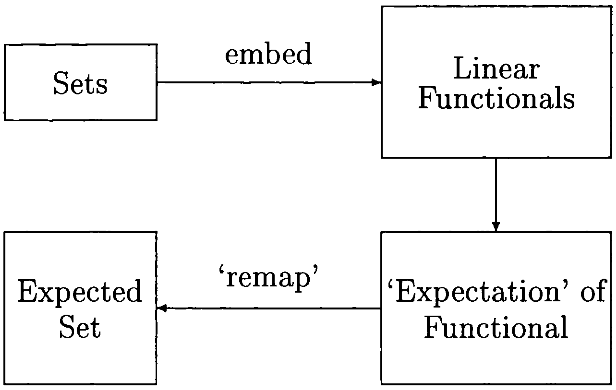


Figure 2.4. Approach to defining expectations of random sets

For examples of expectations of random sets, see [32, 49].

2.3.1 Naive thresholds

The *Vorob'ev expectation* [49, p.113] is based on measures of sets and is defined as follows. Let $\mathbf{1}_X(x)$ be the *indicator function* (or characteristic function) of a random set X in $\mathbb{R}^d$, that is, it is equal to 1 for $x \in X$ and to 0 otherwise. Then

$$\mathbf{E}\mathbf{1}_X(x) = p_X(x) = \mathbf{P}\{x \in X\}$$

is called the *coverage function* of X. Assume that $\mathbf{E}\mu_d(X) < \infty$, where μ_d is the d-dimensional Lebesque measure. Then the *Vorob'ev expectation*, $\mathrm{E}_\mathrm{V}(X)$, is defined by

$$X_t = \{x \in \mathbb{R}^d : p_X(x) \geq t\} \tag{2.17}$$

for t which is determined from the equation,

$$\mu_d(X_t) = \mathbf{E}\mu_d(X).$$

If this equation is not satisfied for any X_t, (for example, if $\mu_d(X_t)$ is discontinuous), then X_t is chosen from the inequality

$$\mu_d(X_t) \leq \mathbf{E}\mu_d(X) \leq \mu_d(X_q), \text{ for all } q < t.$$

We now show that finding the *Vorob'ev expectation* of the random set $X = F_U$ generated by an image f, using both the uniformly and histogram weighted models, yields naive thresholds.

If the random set model $X = F_U$ is uniformly weighted, then,

$$p_{F_U}(x) = \mathbf{P}\{x \in F_U\} = \mathbf{P}\{U \leq f(x)\} = f(x).$$

In this case the coverage function coincides with the image and the set X_t coincides with the thresholded set, F_t, of the image at level t, as introduced earlier. Further,

$$\mathbf{E}\mu_d(X) = \mathbf{E}\mu_d(F_U) = \int\limits_W \mathbf{P}\{x \in F_U\}\, dx = \int\limits_W f(x)\, dx.$$

Thus we see that the *Vorob'ev expectation* of the random set F_U is the thresholded set F_t defined so that

$$\mu_d(F_t) = \int\limits_W f(x) dx\,,$$

if such a set exists. If not, then we choose F_t such that $\mu_d(F_t) \le \int_W f(x)dx \le \mu_d(F_q)$ for all $q < t$. So we see that F_t is the thresholded set whose area is the closest to the integral of f over the entire window. This is a naive approach to thresholding.

Now consider the situation where the random set model is histogram weighted, so that $U = \zeta_f$ is distributed according to the histogram h_f of the image f (assuming that the histogram exists). Then,

$$\mathbf{E}\mu_d(F_U) = \int\limits_W \mathbf{P}\{U \le f(x)\}\, dx = \int\limits_W \int\limits_0^{f(x)} \frac{h_f(t)}{\mu_d(W)}\, dt\, dx\,.$$

Changing the order of integration above we see that

$$\mathbf{E}\mu_d(F_U) = \frac{1}{\mu_d(W)} \int\limits_0^1 h_f(t)\, dt \int\limits_{F_t} ds = \frac{1}{\mu_d(W)} \int\limits_0^1 h_f(t)\mu_d(F_t)\, dt. \tag{2.18}$$

It may be seen that $\mu_d(F_t) = \int_t^1 h_f(s)\, ds$. Inserting this into (2.18) and integrating, we see that $\mathbf{E}\mu_d(F_U) = \mu_d(W)/2$. Thus we see that the *Vorob'ev expectation* for the histogram weighted case yields the thresholded set F_t, determined by $\mu_d(F_t) = \mu_d(W)/2$, if such a set exists. If not, we choose F_t such that $\mu_d(F_t) \leq \mu_d(W)/2 \leq \mu_d(F_q)$ for all $q < t$. So we see that F_t is the thresholded set whose area is the closest to half the area of the window.

2.3.2 Distance average

We have seen in the previous section that the Vorob'ev expectation is determined by the measure of X and its coverage function. In particular it ignores sets of measure zero, for example, isolated points or curves in the continuous setup. We now describe an expectation which has no such problems. It is based on the idea that random sets can be represented by their distance functions rather than indicator functions used to define the Vorob'ev expectation.

Let us suppose that X is a general non-empty random closed set. Let $d(x, X)$ be the generalised *distance function* of X (defined in Section 1.3). Lower semicontinuity of d implies that $d(x, X)$ is a random variable. Assuming that $d(x, X)$ has a (possibly infinite) expectation, define the *expected distance function* as

$$\bar{d}_f(x) = \mathbf{E}d(x, X).$$

In general this is itself not a distance function, see [3]. However, it is sensible in many cases to search for a deterministic set (or binary image) such that its distance function (or distance transform) is the closest to $\bar{d}_f(x)$ in some sense.

Since it is difficult to search through all possible closed sets, it is possible to restrict the choice of possible 'candidates' onto those sets which appear as thresholds of $\bar{d}_f(x)$. A suitable level set of the expected distance function serves as the *distance average* of X [3]. To find this level, $\bar{d}_f(x)$ is thresholded to get a family of sets

$$X(\varepsilon) = \{x : \bar{d}_f(x) \leq \varepsilon\}, \quad \varepsilon > 0.$$

Then the *distance average* $\bar{X}$ is the set $X(\varepsilon)$ chosen to minimise

$$||\bar{d}_f(\cdot) - d(\cdot, X(\varepsilon))||, \tag{2.19}$$

where $\bar{d}_f(\cdot)$ and $d(\cdot, X(\varepsilon))$ are considered to be functions of their arguments designated by dots. The norm in (2.19) could be any one of numerous norms (or more generally metrics) in function spaces, e.g L_∞, L_2, L_1. Note that if h is a numerical function on $W \subset \mathbb{R}^m$, then the L_∞ norm of h is given by

$$||h||_\infty = \sup_{x \in W} |h(x)|,$$

and the L_p norm of h for $p \geq 1$ is defined by

$$||h||_p = \left(\int_W |h(x)|^p \, dx \right)^{1/p} .$$

We will see later that for our specific purposes some norms serve better than others.

2.4 The distance threshold

In computing the distance average of a random set X, a set-valued mean is obtained by identifying a deterministic set whose distance function is the closest to the expected distance function, in some sense. So we are faced with the problem of how to choose various candidates from which the distance average is selected.

Note that in the previous section for a general random set X, the expected distance function $\bar{d}_f(x)$ is thresholded at different levels, so that these thresholded sets become the 'candidates' for the corresponding expectation. Therefore, for the case of a random set F_U generated from an image, the distance average of $X = F_U$ is not a thresholded set (although it could be considered as an 'integral threshold').

However, if instead of searching through the sets $X(\varepsilon)$, the distance average is chosen from the thresholded sets F_t, then this results in the distance average becoming a thresholded set of f. In other words, the threshold is chosen as a set F_t such that its distance function best 'mimics' the expected distance function $\bar{d}_f(x)$. This obtained binary image will be called the *distance threshold.*

Let the maximum and minimum grey levels in the grey-scale image f defined on the window W be denoted by t_1 and t_2 respectively. It is clear that the interval $[t_1, t_2]$ describes the effective range of grey levels in the image. Further, note that F_t is an empty set and consequently $d(\cdot, F_t)$ ill-defined for grey levels $t > t_1$. We formulate our algorithm as follows.

1. Evaluate $d(x, F_t)$ for all grey levels $t \in [t_1, t_2]$ and $x \in W$. This ensures that F_t will always be a non-empty set. These functions form a collection of grey-scale images obtained as distance transforms of F_t for different threshold levels t.

2. Compute

$$\bar{d}_f(x) = \mathbf{E}[d(x, F_U)]$$

for all $x \in W$. (The subscript f is introduced to emphasis the dependency of f on $\bar{d}(x)$.) Here U is a random variable distributed on the effective range of grey levels $[t_1, t_2]$. Two basic options are to let U have the uniform distribution on $[t_1, t_2]$ or use the histogram weighted model where U is distributed according the histogram of f. The resulting function $\bar{d}_f(x)$ itself can be represented as a grey-scale image on W.

3. For the chosen norm, evaluate $||\bar{d}_f(\cdot) - d(\cdot, F_t)||$, for example,

$$||\bar{d}_f(\cdot) - d(\cdot, F_t)|| = \begin{cases} \sup_{x \in W} |\bar{d}_f(x) - d(x, F_t)| & L_\infty \text{ norm}, \\ \left(\int_W |\bar{d}_f(x) - d(x, F_t)|^2 \, dx\right)^{1/2} & L_2 \text{ norm}, \end{cases} \tag{2.20}$$

for each grey level $t \in [t_1, t_2]$. The set F_t corresponding to the value of t, which minimises the left-hand side of (2.20) as a function of t is chosen as the *distance threshold.* Note that there are further natural choices for the norm in (2.20). This step involves minimisation of a function of one variable (the level t).

It should be noted that in practical implementation one does not require storage of all distance transforms from step 1. Instead, they are being accumulated and averaged successively as the threshold level moves up. Figure 2.5 below shows an example of an image together with its corresponding expected distance function $\bar{d}_f(x)$ for the uniformly weighted model. This grey-scale image is called the *grey-scale distance transform*, see [33]. It has been noted in [33] that there is a one to one correspondence between discrete images and grey-scale distance transforms. It is unclear if this assertion holds in the continuous case, that is, for upper semi-continuous functions.

We henceforth use the notation $T(f)$ to denote the distance threshold of an image f. If the distance threshold level is t (obtained by minimisation of (2.20)), then

$$T(f) = F_t = \{x \in W : f(x) \geq t\},$$

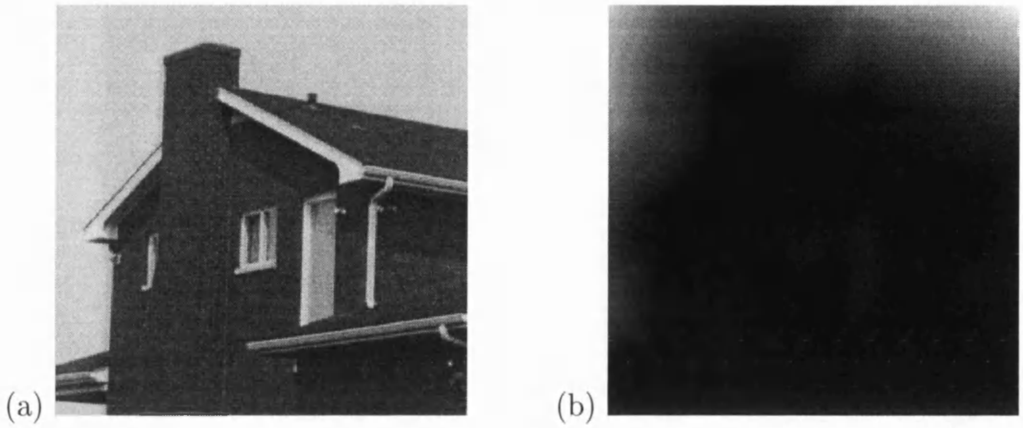


Figure 2.5. (a) house image, (b) grey-scale distance transform of house image.

the set of object pixels in the thresholded binary image.

It should be noted that the distance threshold depends on a number of parameters:

- the random variable U, (or random set model corresponding to the grey-scale image), for example uniformly weighted or histogram weighted random set model.
- The choice of distance transform used, for example, Euclidean distance transform, or signed distance transform.
- The choice of norm chosen to minimise (2.20).

We don't suggest a fixed approach to take for the situation where there are several grey levels minimising (2.20). Some approaches may work better for different situations. The following result, however, shows that an advantage in using the L_∞ norm is that we can characterise the set of levels t which minimise (2.20).

Theorem 2.1. *If for $t' < t''$,*

$$\|\bar{d}_f(x) - d(x, F_{t'})\|_\infty = \|\bar{d}_f(x) - d(x, F_{t''})\|_\infty, \tag{2.21}$$

then $\|\bar{d}_f(x) - d(x, F_t)\|_\infty \leq \|\bar{d}_f(x) - d(x, F_{t'})\|_\infty$ *for all* $t' \leq t \leq t''$*. In particular if* t', t'' *both minimise*

$$\|d(\cdot) - d(\cdot, F_t)\|_\infty\,, \quad t_1 \leq t \leq t_2\,, \tag{2.22}$$

then t *will also minimise (2.22) for all* $t' \leq t \leq t''$*.*

Proof. Since $d(x, F_t), 0 \leq t \leq 1$, is increasing for each fixed x, the following inequality holds:

$$\bar{d}_f(x) - d(x, F_{t''}) \leq \bar{d}_f(x) - d(x, F_t) \leq \bar{d}_f(x) - d(x, F_{t'}).$$

Thus,

$$|\bar{d}_f(x) - d(x, F_t)| \leq \max(\, |\bar{d}_f(x) - d(x, F_{t'})|\,,\, |\bar{d}_f(x) - d(x, F_{t''})|\,).$$

Taking suprema over $x \in W$, yields:

$$\|\bar{d}_f(\cdot) - d(\cdot, F_t)\|_\infty \leq \|\bar{d}_f(\cdot) - d(\cdot, F_{t'})\|_\infty = \|\bar{d}_f(\cdot) - d(\cdot, F_{t''})\|_\infty.$$

Thus t satisfies (2.21), as required. □

Corollary 2.1. $\|\bar{d}_f(\cdot) - d(\cdot, F_t)\|_\infty$ *viewed as a function of* t *has either a single local minimum or is minimised over an interval of values of* t*.*

2.4.1 Properties

The following section introduces examples of some desirable properties of the distance threshold.

1. Let $\tilde{f}$ be an *inverse* image of f obtained by replacing all pixels with grey level $t \in [0,1]$ with grey level $1-t$. If the *signed distance function* is used in computing the distance threshold, then in this instance

$$T(\tilde{f}) = W \setminus T(f).$$

So the distance threshold for the 'inverse' image of f is identical with the complement of the distance threshold of f in W. This may be shown as follows:

Denote the corresponding level sets of $\tilde{f}$ by $\tilde{F}_t = \{x : \tilde{f}(x) \geq t\}$. Thus,

$$\tilde{F}_t = \{x : 1 - f(x) \geq t\} = W \setminus F_{1-t}$$

This implies that $d(x, \tilde{F}_t) = -d(x, F_{1-t})$, where d is the signed distance function, uniformly weighted. Hence

$$\bar{d}_{\tilde{f}}(x) = \mathbf{E}d(x, \tilde{F}_U) = -\mathbf{E}d(x, F_{1-U}) = -\bar{d}_f(x)\,,$$

where $\bar{d}\tilde{f}(x)$ is the expected distance function of $\tilde{f}$. Thus, $||\bar{d}_{\tilde{f}}(x) - d(x, \tilde{F}_t)|| = ||\bar{d}_f(x) - d(x, F_{1-t})||$, giving the desired result.

Note that this result is specific to the signed distance and in particular does not hold for the Euclidean distance function. This result says that the corresponding thresholding operation is not symmetric with respect to a swap of foreground and background pixels.

2. The image cf is obtained from f by replacing every grey value t with grey value ct. Here we must ensure that the transformed grey-scale range does not lie outside $[0,1]$. The following property holds, for the uniformly weighted case.

$$T(cf) = T(f).$$

Begin by denoting the level sets of cf by $F_t^* = \{x : cf(x) \geq t\}$. Clearly

$$F_t = F_{ct}^*. \tag{2.23}$$

Further,

$$\begin{aligned}\bar{d}_f(x) &= \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} d(x, F_t)dt \\ &= \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} d(x, F_{ct}^*)dt.\end{aligned}$$

Introducing the change of variables:

$$\begin{aligned}\bar{d}_f(x) &= \frac{1}{c(t_2 - t_1)}\int_{ct_1}^{ct_2} d(x, F_t^*)dt \\ &= \bar{d}_{cf}(x). \qquad (2.24)\end{aligned}$$

So the expected distance function of f and cf coincide. Both (2.23) and (2.24) combine to give the result.

The result above does not tell us that the thresholded levels in both images are equal, rather that the thresholded sets are identical. This property is trivial for the histogram weighted case since as above the equalised variants of cf and f coincide.

3. $T(f|_K) \subset K$, where

$$f|_K(x) = \begin{cases} f(x)\,,\ x \in K \\ 0\,,\ x \notin K \end{cases}$$

This property states that the distance threshold of an image restricted to some subset K of the window W, and zero elsewhere, is contained in the set K.

This is trivially true since the thresholded sets of $f|_K$, which we denote by F_t^*, are such that, $F_t^* \subset K$ for all grey levels t. The distance threshold is will be chosen from one of these sets, and so the assertion is true.

2.4.2 Examples

Model images

To compute the distance threshold for a model 2-dimensional image, by hand, is computationally intensive. It is for this reason that we begin by considering sample images in 1-dimension, i.e. where the window, W, is a subset of the real line.

Example 2.1. The image f in Figure 2.6 is comprised of three grey levels, namely 0, h and 1. Consequently the image is interpreted as being comprised of two distinct level sets F_1 and F_2. A uniformly weighted random set model together with the signed distance transform and L_∞-norm are used to compute the distance threshold. Here,

$$\bar{d}_f(x) = \begin{cases} -x + ha + (1-h)b, & 0 \leq x < (a+c)/2, \\ x(2h-1) + (1-h)b - hc, & (a+c)/2 \leq x < (b+c)/2, \\ x - c, & (b+c)/2 \leq x \leq d. \end{cases}$$

Further,

$$\begin{aligned} \|\bar{d}_f(\cdot) - d(\cdot, F_1)\|_\infty &= (1-h)|b-a|, & (2.25) \\ \|\bar{d}_f(\cdot) - d(\cdot, F_2)\|_\infty &= h|b-a|. & (2.26) \end{aligned}$$

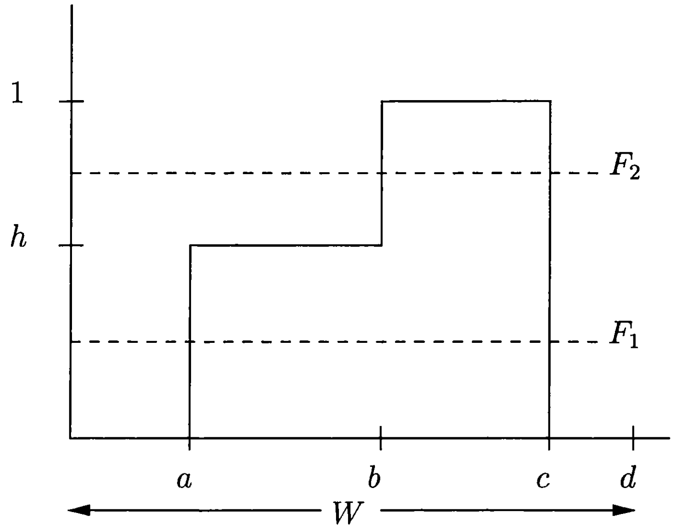


Figure 2.6. Model image with three grey levels.

Whichever F_1 or F_2 corresponds to the minimum of (2.25), (2.26) determines the threshold level. Clearly the threshold level is F_1 if and only if $h > 1/2$ and F_2 otherwise.

Example 2.2. Figure 2.7 shows an image f comprised of three distinct level sets, F_1, F_2 and F_3 corresponding to the grey levels a, $a+b$ and 1. For this example we use the uniformly weighted random set model, L_∞-norm and Euclidean distance transform.

Here it may be seen that,

$$\bar{d}_f(x) = \begin{cases} (1-a-b)(2e-x), & 0 \leq x < e, \\ b(x-e) + (1-a-b)(2e-x), & e \leq x < 3e/2, \\ (1-a)(2e-x), & 3e/2 \leq x \leq 2e, \\ 0, & 2e \leq x \leq 3e. \end{cases}$$

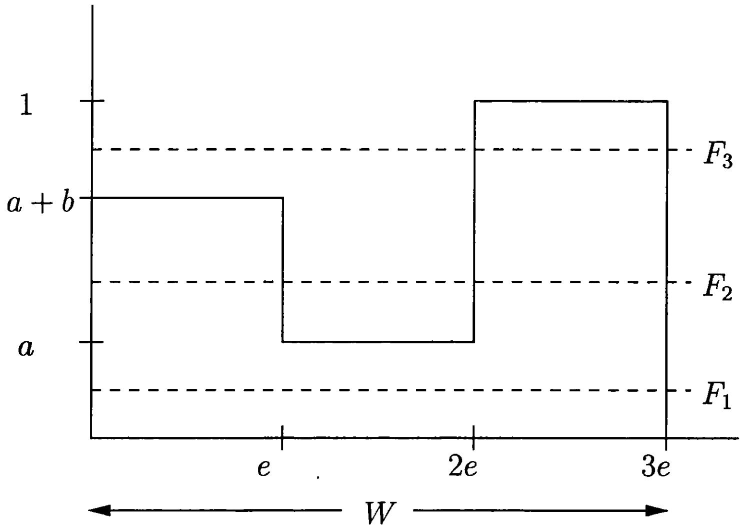


Figure 2.7. Model image with three grey levels.

Hence we find,

$$\begin{aligned}
\|\bar{d}_f(\cdot) - d(\cdot, F_1)\|_\infty &= \max\left\{(1-a-b)\,2e\,,\ \left(\frac{1-a}{2}\right)e\right\}, && (2.27)\\
\|\bar{d}_f(\cdot) - d(\cdot, F_2)\|_\infty &= \max\left\{(1-a-b)\,2e\,,\ \left(\frac{a}{2}\right)e\right\}, && (2.28)\\
\|\bar{d}_f(\cdot) - d(\cdot, F_3)\|_\infty &= (a+b)2e. && (2.29)
\end{aligned}$$

Whichever F_1, F_2, F_3 corresponds to the minimum of (2.27), (2.28), (2.29) determines the threshold level. For example, the threshold level is F_3 if $a+b < \max\{1-a-b, (1-a)/4\}$.

We note here that if we alter the image above so that it's histogram remains unchanged, for example, by interchanging pixels with grey level a with those of grey level $a+b$, then we get a set of conditions different from (2.27), (2.28), (2.29). This should appeal to our intuition as it tells us that the distance threshold takes into account spatial distribution of pixels, rather than just frequencies of grey levels.

Practical images

In this section we present results of the distance threshold for various grey-scale images. Our algorithm must be modified to cater for the discrete set-up. It is usual to work with 256 grey levels, $\{0, 1, \ldots, 255\}$. We denote t_1 and t_2 by the minimum and maximum grey levels respectively, in the image.

1. Compute $d(x, F_t)$, for each grey level $t = t_1, t_1 + 1, \ldots, t_2$, for some chosen distance transform.

2. For the case of a uniformly weighted model, $\bar{d}_f(x)$ is simply an arithmetic mean of the distance functions for each grey level between the minimum and maximum grey levels,

$$\bar{d}_f(x) = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} d(x, F_t)\,.$$

   For the histogram weighted model,

$$\bar{d}_f(x) = \sum_{t=t_1}^{t_2} \frac{h_f(t)}{n(W)}\, d(x, F_t)\,,$$

   where $n(W)$ equals the number of pixels in the window W and $h_f(t)$ denotes the histogram.

3. For the chosen norm, evaluate

$$\|\bar{d}_f(\cdot) - d(\cdot, F_t)\| = \begin{cases} \max_{x\in W} |\bar{d}_f(x) - d(x, F_t)| & L_\infty \text{ norm}\,, \\ \left(\sum_{x\in W} |\bar{d}_f(x) - d(x, F_t)|^2\right)^{1/2} & L_2 \text{ norm}\,, \end{cases} \tag{2.30}$$

   for each grey level $t = t_1, \ldots, t_2$. The set F_t corresponding to the value of t minimising this norm is chosen as the *distance threshold*. If the L_∞ norm is used, then using Corollary 2.1, we see that we need only search through values of t in increasing order until we get a minimum value.

The typical performance time on SUN Ultra 1 (133 MHz) to compute the distance threshold for a 256×256 image is 10-15 seconds to compute the grey-scale distance transform $\bar{d}_f(x)$ and the further 10 seconds to solve the minimisation problem.

Below in Table 2.1 and Figure 2.8 we present results of the distance threshold for a single image (Lenna image) for various choices of parameters. Trends apparent in this example, for various parameter choices, are consistent with trends for a wide variety of different images. For example, it is may be said that the Euclidean DT gives a rather high threshold level, resulting in a binary image which is 'too dark'. While in all instances the signed DT seems to have performed much better, by comparison. It can also be argued that histogram weighting performed marginally better than uniform weighting. Further it is clear that the choice of norm did not have a dramatic effect on the distance threshold level for given random set model and distance transform.

We have found that for all images encountered, the best visual performance (from the author's subjective viewpoint) among all combination of parameters are, a histogram weighted random set model, together with the signed distance function and L_∞ norm. It is hard to give a theoretical justification for these choices. It is apparent however, that in employing the histogram weighted random set model we are enhancing information contained in the image.

Below in Figures 2.9 - 2.12 we present various images together with their corresponding histograms, we display distance thresholds for each image using a histogram weighting, signed distance function and L_∞ norm. The images are grouped into various categories according to the shape of their histograms, which

Random set model	Distance transform	Norm	Threshold level
Histogram weighted	Signed DT	L_∞	133 Figure. 2.8(c)
Histogram weighted	Signed DT	L_2	121 Figure. 2.8(d)
Histogram weighted	Euclidean DT	L_∞	148 Figure. 2.8(e)
Histogram weighted	Euclidean DT	L_2	148 Figure. 2.8(e)
Uniformly weighted	Signed DT	L_∞	133 Figure. 2.8(c)
Uniformly weighted	Signed DT	L_2	138 Figure. 2.8(f)
Uniformly weighted	Euclidean DT	L_∞	169 Figure. 2.8(g)
Uniformly weighted	Euclidean DT	L_2	161 Figure. 2.8(h)

Table 2.1. Results of distance threshold for various parameter choices for Lenna image in Figure 2.8(a)

are bimodal, multimodal and unimodal. In this way we aim to display the performance of the distance thresholds on a diverse range of images.

Multimodal histograms We classify multimodal histograms as those having no clear modes. Such images would not lend themselves easily to valley seeking methods. Typically these images have a relatively even distribution of grey levels over the entire grey level range. For this reason histogram equalisation does not dramatically alter the shape of the histogram. This in turn tells us that the distance threshold for such images does not change significantly when use either uniformly or histogram weighted random set models.

For the Lenna image in Figure 2.8(a) we see that the grey level histogram is multimodal. Thus most histogram methods wouldn't work successfully here. This threshold picked up most of the features on Lenna's face and much of the feathers in her hat. The threshold level of 133 (Figure 2.8(c)), for the stated choice of parameters, compares favourably with the threshold level of 128 which the authors in [46] suggest as giving the best visual performance among all threshold levels.

We see that the distance threshold for the image of a postcard in Figure 2.9

also performs well. Much of the detail in the original image is preserved in the thresholded image.

Unimodal histograms We classify histograms as being unimodal if there is one distinct mode. Again valley seeking methods would be inconclusive here. For the tank image in Figure 2.10 below we see from the corresponding histogram that most of the grey level values are centred between levels 110 and 150. The distance threshold level of $t = 110$ produced a reasonable threshold, although some of the details on the tank is missing.

The histogram of the image in Figure 2.11 shows that much of the grey levels are centred between levels 150 and 200. The distance threshold level of $t = 166$ lies in this range. The thresholded image picked out detail on the gulls beaks and eyes.

Bimodal histograms Images with bimodal histograms are easiest to threshold. Generally a good threshold level is chosen between the two modes. We have found that for all images examined with bimodal histograms the corresponding distance threshold levels are indeed situated at the valley of the two peaks.

We see that this is the case with the image in Figure 2.12, where the threshold level of $t = 126$ lies between the two peaks of the histogram.

Texture images The image in Figure 2.13 represents a texture image from the Brodatz album of textures. The distance threshold level of $t = 161$ produces a binary image which preserves much of the features of the original image.

2.5 Multithresholding

Multithresholding is an extension of thresholding. Thresholding partitions an image into two regions, while multithresholding generalises this to partition an image f into $k+1$ distinct regions. Denote the multithresholded image by f_k. The idea is to choose an increasing sequence of k grey levels, say $\{t_1, t_2, \ldots, t_k\}$ from the set of grey levels following some optimality criterion, and then to partition f following the rule:

$$f_k(x) = \begin{cases} 0\,, & \text{if } f(x) < t_1, \\ 1\,, & \text{if } t_1 \leq f(x) \leq t_2, \\ & \vdots \\ k-1\,, & \text{if } t_{k-1} \leq f(x) \leq t_k, \\ k\,, & \text{if } f(x) > t_k. \end{cases}$$

The grey levels $\{0, 1, \ldots, k\}$ of the multithresholded image are chosen arbitrarily. They could, for example be chosen equally spaced in the interval $[0, 255]$.

This problem may be restated in the following manner. Given an image f with corresponding random set model, or equivalently a sample of thresholded sets F_t, (one for each grey level), choose k thresholded sets, say $F_{t_1}, F_{t_2}, \ldots, F_{t_k}$ which best summarise all the information contained in the sample.

In this setting the problem has some parallels to that of finding the k-mean for a random set $X \subset \mathbb{R}^d$, which is discussed [12]. Here the aim is to find a set $H \subset \mathbb{R}^d$ with k elements, and then to associate each $x \in X$ with an element $\pi(x)$ from H. The choice of k points is determined in such a way that the discrepancy between x and $\pi(x)$ is minimised in some manner.

Recall that in the implementation of the distance average, the first stage was to represent an image f by its grey-scale distance transform,

$$\bar{d}_f(x) = \mathbf{E}_U d(x, F_U),$$

and then to find a realisation of F_U, F_t say, whose distance function is closest to $\bar{d}_f(\cdot)$. A similar idea could be applied to multithresholding. Suppose f is multithresholded into k grey levels $\{t_1, t_2, \ldots, t_k\}$. Denote the grey-scale distance transform of the multithresholded image by,

$$\bar{d}_{f_k}(x) = \begin{cases} n(W)^{-1} \sum_{i=1}^{k} h_{f_k}(i) d(x, F_{t_i}) & \text{(histogram weighted)}, \\ k^{-1} \sum_{i=1}^{k} d(x, F_{t_i}) & \text{(uniform weighted)}. \end{cases}$$

Here $h_{f_k}(i)$ denotes the histogram of f_k. Note that the thresholded level set of f_k at level i coincides with

$$F_{t_i} = \{x : f(x) \geq t_i\},$$

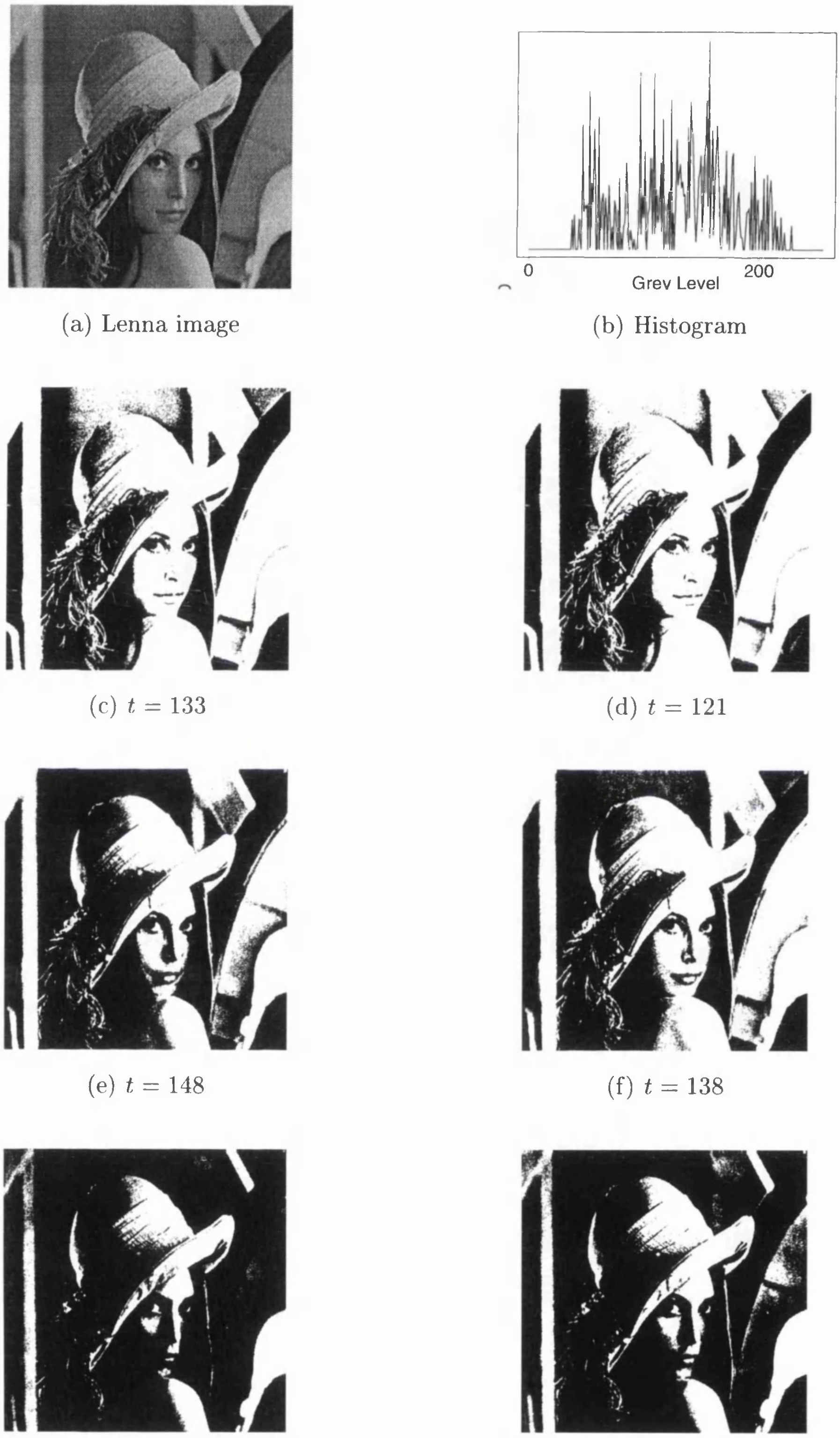
the thresholded set of $f(x)$ thresholded at level t_i

The choice of $\{t_1, t_2, \ldots, t_k\}$ minimising the distance between $\bar{d}_f(\cdot)$ and $\bar{d}_{f_k}(x)$ determines the multithreshold:

$$\{t_1, t_2, \ldots, t_k\} = \arg\min_{\{t_i\}} ||\bar{d}_f(\cdot) - \bar{d}_{f_k}(\cdot)||. \tag{2.31}$$

Clearly when $k = 1$, this reduces to the distance threshold.

It should be noted that this scheme may be computationally expensive. If, for example, it is required to multithreshold a image with 256 grey levels into an image with k grey levels, then there are $256(255) \ldots (256 - k + 1)$ possible choices of grey level to minimise (2.31).



(a) Lenna image

(b) Histogram

(c) $t = 133$

(d) $t = 121$

(e) $t = 148$

(f) $t = 138$

(g) $t = 169$

(h) $t = 161$

Figure 2.8. Lenna image, corresponding histogram, and results of distance threshold for various parameter choices.

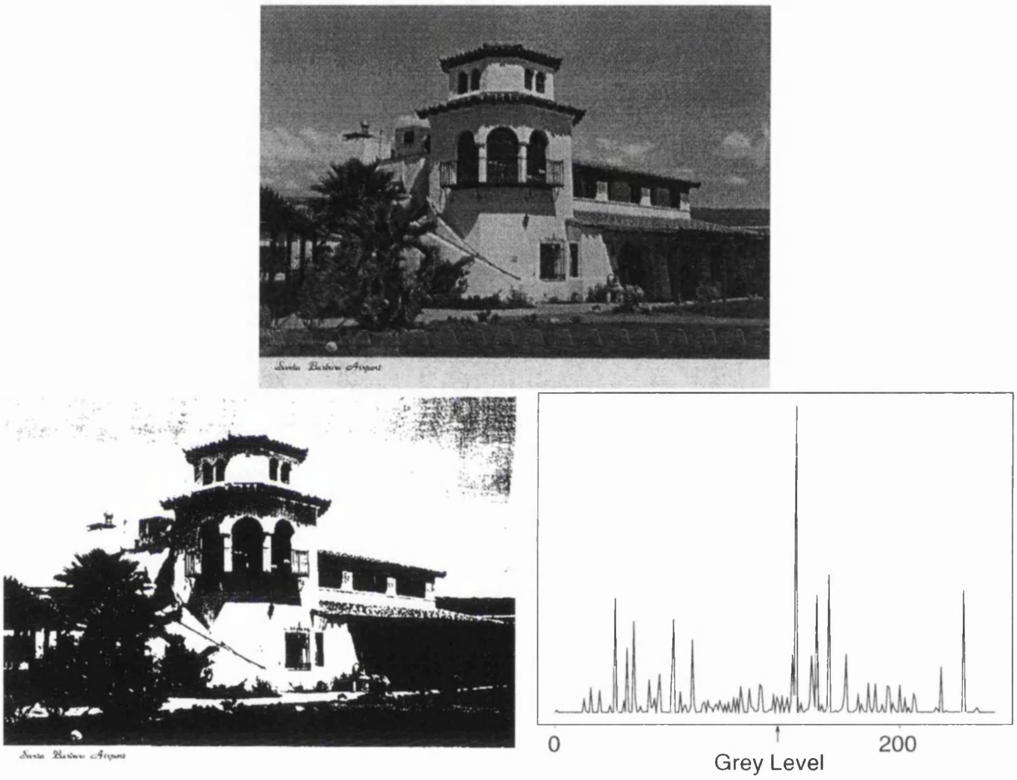


Figure 2.9. `Airport` image, thresholded image and grey level histogram showing the position of distance threshold ($t = 132$).

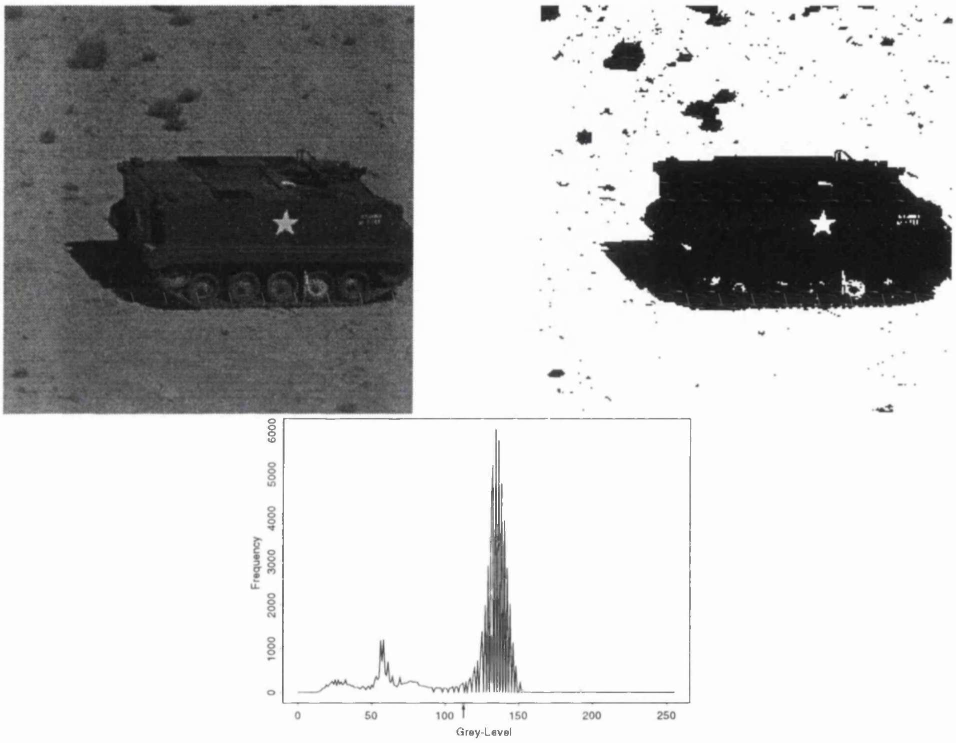


Figure 2.10. Tank image, thresholded image and grey level histogram showing the position of distance threshold ($t = 110$) .

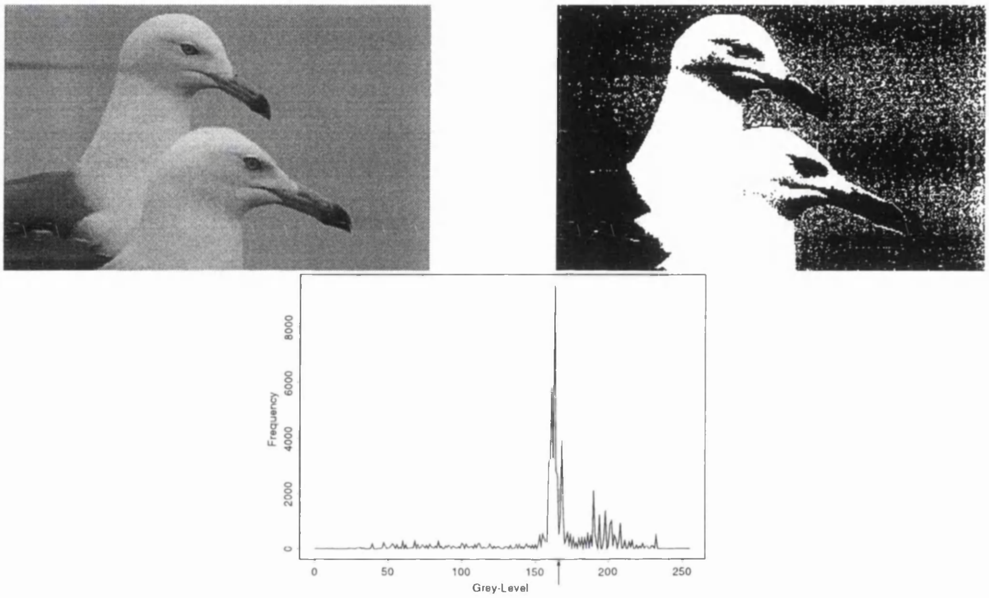


Figure 2.11. Gulls image, thresholded image and grey level histogram showing the position of the distance threshold ($t = 166$).

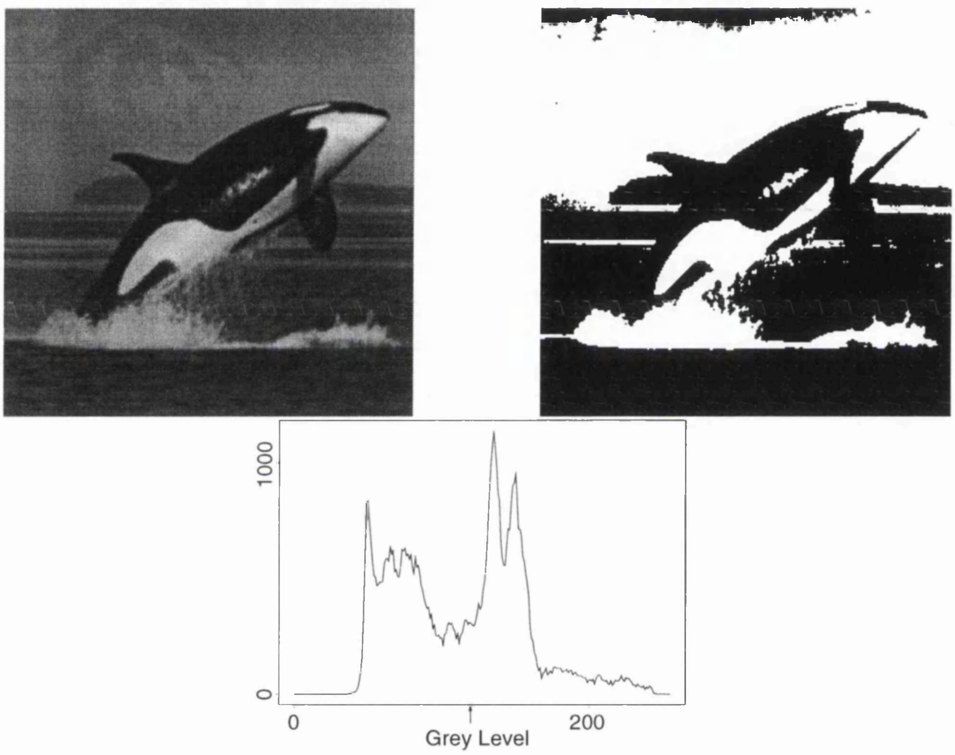


Figure 2.12. An image with a bimodal histogram, thresholded image and grey level histogram showing the position of the distance threshold ($t = 126$).

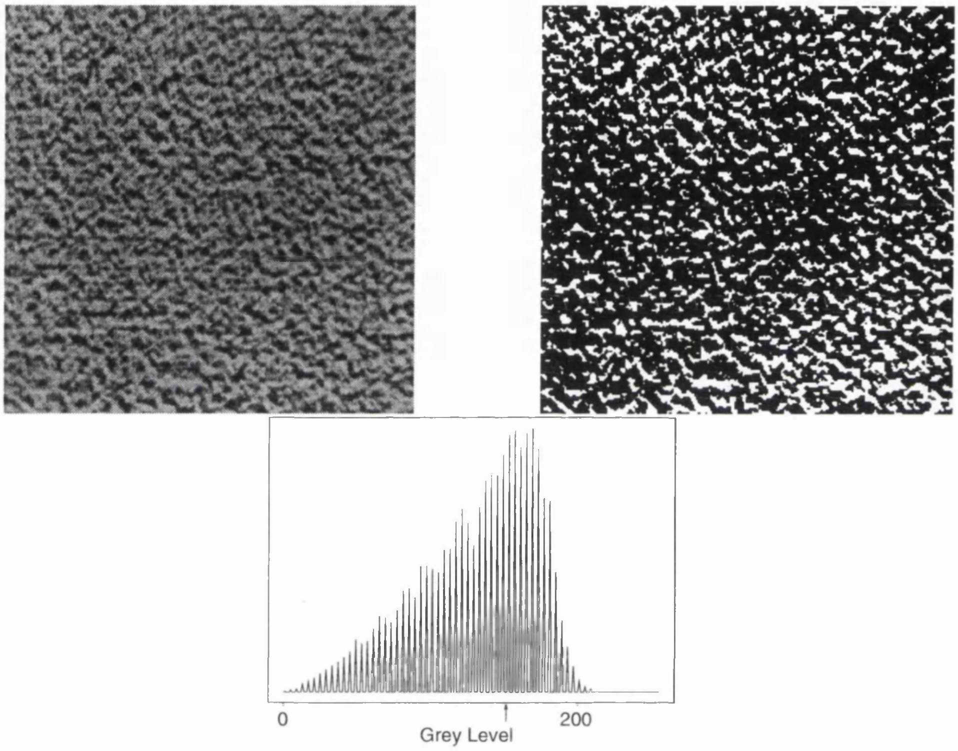


Figure 2.13. Texture image, thresholded image and grey level histogram showing the position of the distance threshold ($t = 161$).

Chapter 3

Image Metrics

3.1 Introduction

A measure of comparison of the similarity/dissimilarity between two images is a vital concept in image processing. Often algorithms are needed to estimate a 'true' image as accurately as possible. This is the situation in image reconstruction, encoding, edge detection etc. In such scenarios it is of interest to assess numerically the discrepancy between two images, so that a number $d(f, g)$ characterises the distance between f and g. In this way we can objectively test the performance of the algorithm, diminishing the effect of human subjectivity, reducing it to a choice of the disimilarity measure.

It is often desirable that $d(f, g)$ takes the form of a metric on the space of images. This is certainly more attractive from a mathematical viewpoint. In this instance we term $d(\cdot, \cdot)$ an image metric.

Often many methods used to assess distances between grey-scale images are formulated on an ad hoc basis. One of the most widely used method of comparison is the root mean squared (RMS) distance, which computes the sum of squared differences between corresponding pixels:

$$\mathrm{RMS}(f,g) = \left(\frac{1}{n(W)} \sum_{x \in W} (f(x) - g(x))^2 \right)^{1/2} . \tag{3.1}$$

In the continuous set-up RMS can be identified with the L^2 distance

$$\|f - g\|_2 = \left(\int_{x \in W} (f(x) - g(x))^2 \right)^{1/2} .$$

The RMS method has the obvious advantage that it is computationally efficient, however it is accepted that it and other similar pixel by pixel methods do not always always give a good measure of visual similarity. One plausible reason for this may lie in the fact that RMS ignores locations of pixel by pixel errors. Suppose two 'estimated' images had the same number and intensity of pixel errors, where in the first all the errors were heavily localised in one area of the image, while in the second all the errors were evenly spread throughout the image. Both estimated images would return the same RMS distance, however the visual quality of the first would appear to be much poorer than the second. As an illustration consider Figure 3.1 below. Both estimated image 1 and 2 return the same RMS value when compared to the true image, however estimated image 1 would seem to be visually more similar to the true image than estimated image 2. All of the erroneous pixels in estimated image 1 are spread evenly throughout the image, while in estimated image 2 all errors are localised in the 'face', that is, eyes, nose and mouth are missing. In fact the double integral distance, which we shall meet later in this chapter, and which does take into account spatial considerations returns more realistic distances of 0.017 and 0.158 when the true image is compared with estimated images 1 and 2, respectively. (Note that the double integral distance is normalised here to return values between 0 and 1.)

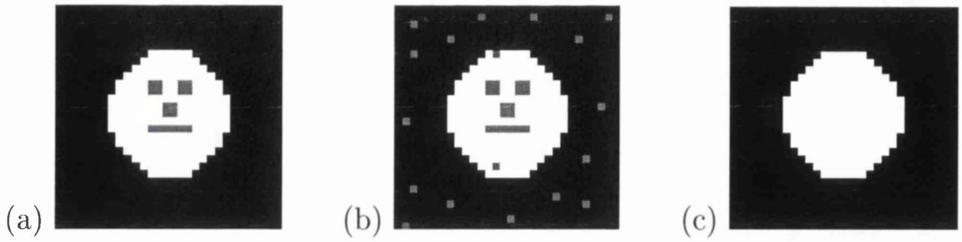


Figure 3.1. (a) True image, (b) Estimated image 1, (c) Estimated image 2

Throughout this chapter we will be concerned solely with discussing methods of comparison between images of similar dimension and size, further we will assume that the grey-scale range of both are identical. This is not as restrictive as it may seem, since for most applications where ones works with 'true' and 'estimated' images, these assumptions hold true.

This chapter begins with a discussion of binary image metrics in present use, followed by the situation for grey-scale images. We conclude with a discussion of new grey-scale image metrics.

3.2 Binary image metrics

3.2.1 Pixel-by-pixel differences

Recall that a binary image f may be uniquely described by the compact set F defined as:

$$F = \{x \in W : f(x) \neq 0\}.$$

Thus F is the set of pixels in the image f taking the value 1. It is clear that the problem of finding a measure of distance between two binary images f, g amounts to finding an appropriate distance between the corresponding sets F, G. Without further notice for binary images we interchange the notation f, g

with F, G respectively.

Many metrics commonly used for binary image comparison are very simple in nature. A frequently used metric is the pixel misclassification error rate:

$$\varepsilon(f,g) = \frac{1}{n(W)} \sum_{x \in W} \mathbf{1}_{[f(x) \neq g(x)]}, \tag{3.2}$$

where $\mathbf{1}_{[f(x) \neq g(x)]}$ is the usual indicator function taking the value 1 when $f(x) = g(x)$ and 0 otherwise. Note that (3.2) takes values between 0 and 1 and is equivalent for binary images to RMS. In the continuous set-up it is equivalent, up to a constant $1/n(W)$, to the L_1 norm of $(f - g)$. Indeed (3.2) has the advantage that it is computationally efficient, and displays some good theoretical properties, however it has been noted that often such pixel by pixel methods don't accurately convey a visual sense of similarity [2]. Distances between F and G are measured by the number of disagreements, without regard to their location. For example, errors such as the displacement of the boundary of F, involving a large number of pixels, but not severely affecting its 'shape' are given high ε values. As an illustration consider Figure 3.2. Here Image B is obtained by rigid motion of Image A. The misclassification error rate between the two images has the value 0.315. The delta metric distance, which we will meet in the subsequent section, gives a much more realistic value of 0.07.

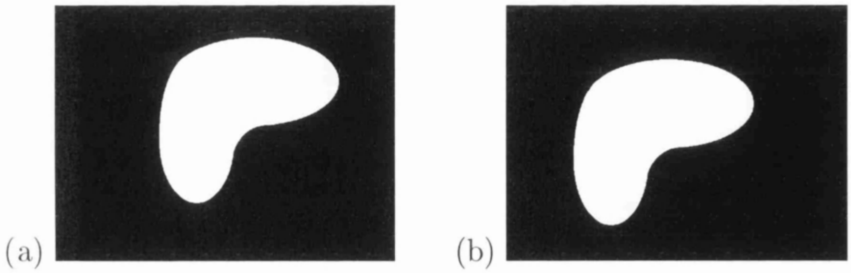


Figure 3.2. (a) Image A, (b) Image B

Pratts 'figure of merit' (FOM) has also been proposed as a dissimilarity measure for binary images [1, 38]. It is defined as,

$$\mathrm{FOM}(F,G) = \frac{1}{\max\{n(F),n(G)\}} \sum_{x\in G} \frac{1}{1+k\rho(x,F)^2}, \tag{3.3}$$

where $\rho(\cdot,\cdot)$ is the Euclidean distance function and k is a scaling constant, usually set to 1/9 when $\rho(\cdot,\cdot)$ is normalised so that the smallest non-zero distance between neighbouring pixels equals 1. Note that FOM is implemented by setting the true image to correspond to the compact set F and the estimated image to correspond to the set G. It is clear that FOM is not itself a metric, in fact $\mathrm{FOM}(F,G) \neq \mathrm{FOM}(G,F)$. Also $0 < \mathrm{FOM}(F,G) \leq 1$ and $\mathrm{FOM}(F,G) = 1$ if and only if $f(x) = g(x)$ for all $x \in W$. It has also been noted that FOM doesn't always convey an accurate measure of distance.

3.2.2 Baddeley's delta metric

A recently presented binary error metric which circumvents some of the difficulties mentioned in the above error measures is Baddeley's delta metric [2]. We begin with a brief discussion of the Hausdorff metric, on which the delta metric is based.

Definition 3.1. Given two non-empty sets (or binary images) F, G in $\mathbb{R}^d$ (or W), the Hausdorff distance is defined as:

$$H(F,G) = \max\left\{\sup_{x\in F} \rho(x,G), \sup_{x\in G} \rho(x,F)\right\}. \tag{3.4}$$

In other words, the Hausdorff distance is the maximum distance from a point in one set to the nearest point in the other as shown in Figure 3.3.

As before $\rho(\cdot,\cdot)$ is the Euclidean distance function. The Hausdorff distance

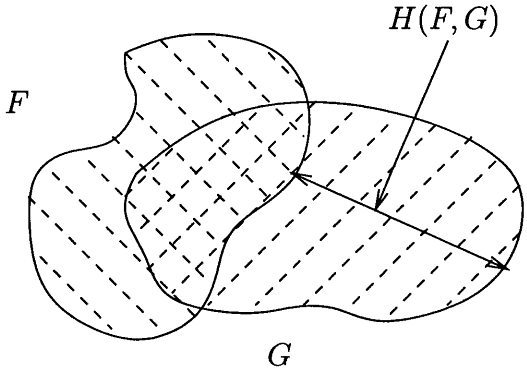


Figure 3.3. Hausdorff distance between F and G

may be equivalently formulated as:

$$H(F,G) = \sup_{x \in W} |\rho(x,F) - \rho(x,G)|. \tag{3.5}$$

The Hausdorff metric is a theoretically attractive metric. It generates the myopic topology, which Serra [47] argues is the most appropriate topology for the medium of binary images. It is however practically unusable, since it is highly sensitive to background noise. The changing of a even a single background pixel to a foreground pixel or vice versa, can drastically affect the Hausdorff distance. Figure 3.4 displays two similar sets F and G. However the presence of the point $x \in F$ gives a large value of $H(F,G)$. In the absence of x, $H(F,G)$ would give a more realistic measure of distance.

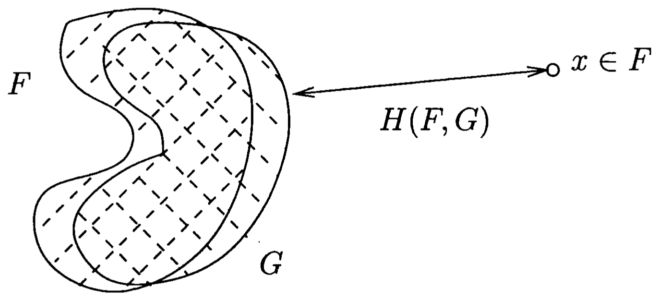


Figure 3.4. Sensitivity of $H(F,G)$ to noise

It should be noted that different metrics may induce the same topology. Baddeley [2] proposed an alternative metric to the Hausdorff metric which in its continuous form, is topologically equivalent to the Hausdorff metric, but which is less sensitive to noise. This metric is obtained by replacing the supremum in (3.5) with an L^p average, and bounding the distance function by some appropriate concave transformation. The delta metric between two binary images f, g is defined as:

$$\Delta_c^p(f,g) = \left\{ \frac{1}{n(W)} \sum_{x \in W} |\rho^*(x,F) - \rho^*(x,G)|^p \right\}^{1/p} . \tag{3.6}$$

Here $\rho^*(x,F) = \min\{\rho(x,F), c\}$, so that in (3.6) pixels greater than a distance c, the so-called cut-off value, from the nearest foreground pixel will be given the value c. In this way the effect of background noise is limited.

By construction, letting $p \to \infty$ and $c \uparrow \infty$, reduces Δ_c^p to the Hausdorff metric. It is also worth noting that if the cut-off value c is less than the smallest distance between any two distinct pixels, then Δ_c^p is related to the misclassification error rate:

$$(\Delta_c^p(f,g)/c)^p = \frac{1}{n(W)} \sum_{x \in W} 1_{[f(x) \neq g(x)]} .$$

The delta metric has been applied to a Bayesian restoration problem [19, 44], with successful results. We will explore this restoration problem for grey-scale images in the subsequent chapter.

3.3 Grey-scale metrics

The problem of finding an error metric between grey-scale images is somewhat more complex than that of binary images. Grey-scale images have the added dimension of pixel grey level, while error metrics for binary images essentially amounts to finding an appropriate distance between compact sets in some (discrete) subset (or window) of a Euclidean plane.

There are many approaches to take towards finding 'good' grey-scale image metrics. One approach might be to extend useful binary image metrics to the hypographs (subgraphs) of grey-scale images, for example [51]. One may also apply metrics between fuzzy sets to images, since an image may be considered as representing a fuzzy set. The approach taken in [17] is to formulate distances between images via probability metric distances between the random set models generated by the images. Probability metrics [41] are used to determine distances between random variables, and can be generalised for random sets [31]. One such probability metric is modified so that it is equivalent to the delta metric, when applied to binary images, and then this metric is extended to the more general case of grey-scale images.

3.3.1 Usual measures of comparison

In most applications involving comparison of images, methods to assess discrepancies are usually quite ad hoc in nature. Many use methods based on simple pixel by pixel differences. Two of the most widely used include the root-mean-squared (RMS) distance,

$$\mathrm{RMS}(f,g) = \left\{ \frac{1}{n(W)} \sum_{x \in W} (f(x) - g(x))^2 \right\}^{1/2}, \tag{3.7}$$

which is equivalent, up to a constant, to the L_2 distance between f and g, and the signal-to-noise ratio (SNR),

$$\mathrm{SNR}(f,g) = \frac{\sum_{x\in W} g(x)^2}{\sum_{x\in W}(f(x)-g(x))^2}. \tag{3.8}$$

Here $f(x)$ represents the 'true' image and $g(x)$ the 'estimated' noisy image. It is assumed that $g(x) = f(x) + e(x)$, where $e(x)$ represents the amount of noise contained in pixel x. The denominator then aims to calculate the total amount of noise in the 'estimated' image. It is clear that SNR is not a metric.

RMS and other distances are widely used in many applications, image compression/decompression, restoration etc., however it is accepted that they do not convey an accurate sense of discrepancy. Part of the downfall of RMS and SNR lies in the fact that each are invariant to localisation of errors. If pixel by pixel differences were heavily localised in a certain area of an image, then visual quality would seem to be much poorer than if the same number and intensity of errors were spread over the entire image. That is, the effect of all the pixel differences would be 'diluted' throughout the image, see Figure 3.1 as an example. A further complication lies in the fact that equal differences in pixel grey levels do not have the same visual impact — the retina is known to to have different sensitivities to different grey levels.

It is also possible to transfer metrics between grey-scale images from the spatial domain to the frequency domain via Fourier transforms. Parseval's identity gives an immediate connection between RMS in the spatial and frequency domains. Let $\hat{f}(u)$ be the Fourier transform of an image f, for each pixel u in the frequency domain U, then Parseval's identity says that:

$$\sum_{x\in W} f(x)^2 = \frac{1}{n(U)}\left\{\sum_{u\in U} |\hat{f}(u)|^2\right\}, \tag{3.9}$$

which in turn tells us that RMS is equivalent in both the frequency and spatial domains. That is,

$$\left\{\frac{1}{n(W)}\sum_{x\in W}(f(x)-g(x))^2\right\}^{1/2} = \left\{\frac{1}{(n(U))^2}\sum_{u\in U}|\hat{f}(u)-\hat{g}(u)|^2\right\}^{1/2}$$

However this explains the idea that error measures need not always be defined in the spatial domain. The Sobolev metric of order δ, where $0 < \delta < 1$ can be abstracted by modifying (3.9) as

$$\mathrm{Sob}(f,g) = \left\{\frac{1}{(n(U))^2}\sum_{u\in U}(1+|\eta_u|^2)^\delta|\hat{f}(u)-\hat{g}(u)|^2\right\}^{1/2}.$$

Here η_u is the two dimensional frequency vector associated with position u in the frequency domain U. The Sobolev metric first appeared in the context of image comparison in [51].

3.3.2 Fuzzy metrics

A fuzzy subset of a set W is a mapping α from W into $[0,1]$. For $x \in W$, $\alpha(x) \in [0,1]$ gives the 'degree of membership' of x in α. If α takes only the values 0 or 1, then such a membership function is called crisp. The support of α is defined as $\mathrm{supp}(\alpha) = \{x : \alpha(x) > 0\}$.

The concept of a fuzzy set is similar to that of a grey-scale image. Here the approach is to treat grey levels as degrees of membership for various pixels, so that

$$f : W \to [0,1],$$

becomes its membership function. The connection between the random set model for an image f, introduced earlier, is obvious. Recall that given an image f with corresponding uniformly weighted random set model F_U, $\mathbf{P}(x \in F_U) = f(x)$ for $x \in W$. That is, given a pixel x, the probability that it is contained in the random set is simply $f(x)$. Analogously in fuzzy set terminology, $f(x)$ is the degree of membership of x. Throughout this section we will equivalently speak about fuzzy sets and images.

It is clear that metrics defined between fuzzy sets may be applied to define distances between grey-scale images. Examples of fuzzy set metrics appear in [8, 10]. Both are based on representing a fuzzy set via level sets (or t-cuts), which in image processing language are nothing but thresholded sets. The level set of the fuzzy set (or thresholded set of the image) may be denoted (as before) by

$$F_t = \{x : f(x) \geq t\}, \ \ t \in [0,1].$$

Chaudhuri et al. [10] define a metric $M(\cdot,\cdot)$ between fuzzy sets (images) f and g as the weighted integral of Hausdorff distances between corresponding level sets:

$$M(f,g) = \frac{\int_0^1 w(t)H(F_t, G_t)\,dt}{\int_0^1 w(t)\,dt}, \tag{3.10}$$

where $w(t) = t$ and $H(\cdot,\cdot)$ is the Hausdorff metric. However this metric is only computable if the fuzzy sets (images) f, g have the same maximum value. Otherwise, if $f(x) = t^*$ for some $x \in W$, such that t^* is larger than the maximum value of g, then G_{t^*} is empty. Thus in (3.10) we would be computing the Hausdorff distance to an empty set, which isn't well defined.

Chaudhuri et al. [10] addressed this problem by modifying both fuzzy sets so that the maximal values of f and g in both sets is 1 and occurs just at the points where the maximal value in each occurred previously. The resultant fuzzy sets are denoted f', g'. In imaging terminology this amounts to modifying each image so that the set of pixels with the largest grey level are now given the maximal grey level value 1. However $M(f', g') = 0$ does not imply that $f = g$. A correction term is introduced to avoid this:

$$e(f,g) = \frac{\int_W |f(x) - g(x)|\,dx}{\mu_d(W)}.$$

The fuzzy metric is now defined as:

$$M^*(f,g) = M(f',g') + e(f,g). \tag{3.11}$$

It is suggested that this metric and a similar metric proposed in [8], could be used in an imaging framework. We outline a few drawbacks in this scenario. First of all, the Hausdorff metric is inappropriate as a basis for this method, considering its instability to background noise, as mentioned earlier. A more sensible approach would seem to be to apply better binary metrics here, for example, the delta metric. Also no justification is given for the choice of weights $w(t) = t$ in (3.10).

It is clear that the above fuzzy set metrics could also be formulated using the random set model for grey-scale images. Suppose we have two images f, g with corresponding random sets F_U, G_U, then all of the above metrics are particular cases of the metric given by,

$$\mathbf{E}m(F_U, G_U),$$

Here $m(\cdot,\cdot)$ is some metric defined between binary images (or compact sets), for example the delta metric. As with (3.11), care may need to be taken if both f and g don't reach the same maximal value and so F_U or G_U may be empty.

In this setting the metrics in (3.10) and (3.11) are generated if U has cumulative distribution function:

$$\mathbf{P}(U \leq s) = \frac{\int_0^s t dt}{\int_0^1 t dt} = s^2.$$

This choice of distribution is rather arbitrary.

More interesting metrics may be generated if the distribution U, or the corresponding random set model, is more appropriate. We consider some choices below. Let us suppose that f and g both attain the maximal grey value 1. (Otherwise f and g should be adjusted accordingly.)

We could consider a uniform weighting, so that

$$M(f,g) = \int\limits_0^1 m(F_t, G_t)\, dt,$$

or maybe a distribution corresponding to the histogram of the 'true' image, if either f or g can be interpreted as such. For example if f is 'true' image, then we may consider a metric of the form,

$$M(f,g) = \frac{1}{\mu(W)} \int\limits_0^1 h_f(t) m(F_t, G_t)\, dt.$$

3.3.3 Binary metrics applied to grey-scale images

Recall that a subgraph Γ_f for a grey-scale image f is defined as:

$$\Gamma_f = \{(x,t) : x \in W, t \in [0,1] \text{ and } f(x) \geq t\},$$

so that this is the set of all pairs $(x,t) \in W \times [0,1]$ lying between the graph of f and the plane $t = 0$.

In this way we may consider a 2-dimensional grey-scale image as a 3-dimensional binary image, incorporating the grey values as an additional dimension. Any binary metric may be applied to Γ_f and Γ_g to define distances between grey-scale images.

The aim in [51] was to use the delta metric to calculate the distance between Γ_f, Γ_g, for two images f, g. It is clear then that the crux of this problem is to define an appropriate measure of distance, $d((x,t), \Gamma_f)$, between points $(x,t) \in W \times [0,1]$ and the subgraph Γ_f of f. (Once this has been done, analogous to the delta metric, this new metric will then be an L_p average of such distances, over all points (x,t).) A first step is to define distances between pairs of points in $W \times [0,1]$. Wilson et al. [51] propose:

$$d((x,t),(x',t')) = \max\left(\rho(x,x'), |t-t'|\right), \tag{3.12}$$

where $\rho(x,x')$ is some metric defining distances between points in the plane.

Using the same notation as previous, the thresholded set for an image f at any grey level t is defined as,

$$F_t = \{x : f(x) \geq t\}.$$

The distance between points in the plane and this set is defined as the distance to the nearest point in the set:

$$d(x, F_t) = \inf\{\rho(x,x') : x' \in F_t\}. \tag{3.13}$$

Using (3.12) and (3.13), the distance from a point $(x,t) \in W \times [0,1]$ to the subgraph of f, Γ_f may now be defined, after some calculations as:

$$d((x,t),\Gamma_f) = \inf_{t' \in [0,1]} \{\max[d(x,F_{t'}), |t-t'|]\}\,. \tag{3.14}$$

Analogous to the formulation of the delta metric, this distance is truncated by a distance c, to suppress the influence of noise as:

$$d^*((x,t),\Gamma_f) = \min\left(d((x,t),\Gamma_f), c\right).$$

The desired metric Δ_c^g, between the two subgraphs Γ_f and Γ_g is now defined as,

$$\begin{aligned} \Delta_c^g(f,g) &= \Delta_c^g(\Gamma_f,\Gamma_g) \\ &= \left\{ \int\limits_{(x,t)\in W\times[0,1]} |d^*((x,t),\Gamma_f) - d^*((x,t),\Gamma_g)|^p \right\}^{1/p}. \end{aligned}$$

In the discrete setting, working with a discrete set of grey levels GL, Δ_c^g may be written as:

$$\Delta_c^g(f,g) = \left\{ \frac{1}{n(W)n(GL)} \sum_{(x,t)\in W\times GL} |d^*((x,t),\Gamma_f) - d^*((x,t),\Gamma_g)|^p \right\}^{1/p}.$$

where $n(W)n(GL)$ is the total number of pairs $(x,y) \in W \times GL$. This metric is shown to perform adequately on a wide variety of images. One major drawback to this metric however, is the vast computation time required to calculate this distance and additional storage requirements arising from the necessity to store distance transforms of all thresholded sets as noted in (3.14).

3.3.4 Kantorovich distance

An image f may be viewed as a probability distribution function defined such that for $A \subseteq W$,

$$\mathbf{P}(A) = \frac{\int_A f(x)\,dx}{\int_W f(x)\,dx}.$$

A natural tool to explore distances between probability distribution functions are probability metrics. Kantorovich [26] introduced one such probability metric. We begin with a description of this metric in an abstract setting, and then show how it may be applied to grey-scale images [25].

Let (U, δ) be a separable metric space and let P_1, P_2 be two probability measures defined on this metric space. Define $\Theta(P_1, P_2)$ as the set of all probability measures P on $U \times U$ with fixed marginals P_1, P_2 and put

$$K_1(P_1, P_2) = \inf \left\{ \int_{U \times U} \delta(x, y) P(dx, dy) : P \in \Theta(P_1, P_2) \right\}. \qquad (3.15)$$

The fact that $K_1(\cdot, \cdot)$ is a metric follows from the fact that $\delta(\cdot, \cdot)$ itself is a metric.

A dual formulation may also be written. Let

$$\mathrm{Lip}(U) = \sup\{f : U \to \mathbf{R} : |f(x) - f(y)| \leq \delta(x, y)\}, \qquad (3.16)$$

be the family of Lipschitz functions on U. Now define

$$K_2(P_1, P_2) = \sup \left\{ \left| \int_U f(x) P_1(dx) - \int_U f(x) P_2(dx) \right| : f \in \mathrm{Lip}(U) \right\}. \qquad (3.17)$$

The equivalence of (3.15), (3.17) may be stated as a theorem due to Kantorovich [26]:

Theorem 3.1. *If U is compact then*

$$K_1(P_1, P_2) = K_2(P_1, P_2).$$

Kaijser [25] has adopted this metric to the situation where the metric is now defined between images f and g defined on windows W_1 and W_2 respectively. The Kantorovich distance is often interpreted as a mass transportation problem, where the cost-function of the transportation problem depends on the type of distance function used to measure distances between pixels.

To begin, we introduce the notion of a transportation image. A transportation image is a set

$$T = \{(x_n, y_n, m_n), 1 \leq n \leq N\}$$

of a finite number, N say, of three-dimensional vectors. It is assumed that each m_n is strictly positive and that there are never two vectors for which the first two elements are equal. The first two elements are denoted transmitting and receiving pixels respectively.

Given a transportation image, it is possible to define two images, termed a transmitting image, f and a receiving image, g as follows: Let W_1 denote the union of all transmitting pixels and W_2 denote the union of all receiving pixels. Given $x \in W_1$, let $A(x)$ represent the set of indices in the set $\{(x_n, y_n, m_n), 1 \leq n \leq N\}$ for which the transmitting pixel is equal to x. Similarly for $y \in W_2$, $B(y)$ denotes the set of indices for which the receiving pixel is equal to y. The transmitting image is now defined by:

$$f(x) = \sum_{n \in A(x)} m_n, x \in W_1,$$

and the receiving image by

$$g(y) = \sum_{n \in B(y)} m_n, y \in W_2.$$

It is clear from the definition of f, g that the total grey value of f, g defined as $\sum_{x \in W} f(x)$, $\sum_{x \in W} g(x)$ respectively, is equal.

Now suppose we are given two images f and g defined on windows W_1, W_2 respectively, then distances between pixels $x \in W_1$ and $y \in W_2$ may be computed via a distance $\rho(x, y)$, for example the Euclidean metric. (Note that W_1, W_2 may or may not overlap.) Suppose further that both images have equal total grey value. Let $\Theta(f, g)$ denote the set of all transportation images from f to g. Given a particular distance function, we may now define a cost for any given transportation image $T \in \Theta(f, g)$ as:

$$c(T) = \sum_{n=1}^{N} \rho(x_n, y_n) m_n.$$

The Kantorovich distance $d_K(f, g)$ between f and g may now be defined as:

$$d_K(f, g) = \inf\{c(T) : T \in \Theta(f, g)\}. \tag{3.18}$$

Kaijser [25] shows how computation of this distance may be stated in terms of a linear programming problem. Analogous to the situation described above for probability measures, a dual formulation may also be presented. Let $\{\alpha(x), x \in W_1\}$ and $\{\beta(y), y \in W_2\}$ denote sets of variables associated with pixels in W_1 and W_2 called dual variables. As above, let $\rho(\cdot, \cdot)$ denote a distance function between pixels in W_1 and W_2. Let Ψ denote the set of all dual variables satisfying

$$\rho(x, y) - \alpha(x) - \beta(y) \geq 0$$

where, $x \in W_1, y \in W_2$. (This condition is analogous to the Lipschitz condition (3.16) for probability measures.) The dual formulation of $d_K(\cdot,\cdot)$ may now be written as:

$$d_K(f,g) = \sup \left\{ \sum_{x \in W_1} \alpha(x) f(x) + \sum_{y \in W_2} \beta(y) g(y) : (\alpha(x), \beta(y)) \in \Psi \right\}. \quad (3.19)$$

References to proofs of the equivalence between (3.18) and (3.19) are given in [25].

One immediate drawback to this metric is that the total grey value for both input images needs to be identical. The author [25] suggests finding two normalising constants, one for each image, and multiplying pixel values in each image by the corresponding constant, so that the two transformed images will have equal total grey value.

Another disadvantage is the vast computation time required to calculate this distance. Kaijser [25] has produced an algorithm which has less computational complexity than standard algorithms, but which would still seem to be infeasible for many applications.

3.4 Probability metrics

The following section illustrates a new approach towards generating metrics for grey-scale images. This method relies on the usage of the random set model for images. Distances between images may now be formulated via distances between their corresponding random sets. Probability metrics have been used to generate distance between random variables, see [41]. Molchanov [31] generalised this theory to the class of closed random sets.

3.4.1 Probability metrics for random sets

Following Rachev [41], a probability metric $d(\xi,\eta)$ between two random variables ξ and η satisfies the following conditions:

1. $d(\xi,\eta)=0$ implies $\mathbf{P}\{\xi=\eta\}=1$.

2. $d(\xi,\eta)=d(\eta,\xi)$.

3. $d(\xi,\eta)\leq d(\xi,\zeta)+d(\zeta,\eta)$ for each random variable ζ.

Probability metrics are used to determine distances between random variables. Examples of probability metrics include, the *uniform (or Kolmogorov) metric*:

$$U(\xi,\eta)=\sup\{|F_\xi(x)-F_\eta(x)|:x\in\mathbf{R}\}, \tag{3.20}$$

where F_ξ denotes the cumulative distribution function of ξ. Another example is the *Lévy metric*, which is defined as follows:

$$\mathcal{L}(\xi,\eta)=\inf\{\varepsilon>0:F_\xi(x-\varepsilon)-\varepsilon\leq F_\eta(x)\leq F_\xi(x+\varepsilon)+\varepsilon\ \forall x\in\mathbf{R}\}. \tag{3.21}$$

The Kantorovich metric outlined in Section 3.3.4 is also a probability metric.

There are many approaches to generalise probability metrics to the class of random sets, for example, it is possible to generalise some classical probability metrics by replacing distribution functions with capacity functionals of random sets, see [31]. Capacity functionals play the same role in the theory of random sets as probability distribution functions in classical probability theory. The capacity functional of a random set X is defined as

$$T_X(K)=\mathbf{P}\{X\cap K\neq\emptyset\},\ \ K\in\mathcal{K},$$

where $\mathcal{K}$ denotes the family of all compact sets. We see that the capacity functional is the probability that a given compact set K hits the random set X.

The class $\mathcal{K}$ of all compact sets is too large to efficiently define and compute the capacity functional on it. For this reason it may be necessary to reduce the class of test sets in order to compute the capacity functional. That is, we need to fix a certain class $\mathcal{M} \subset \mathcal{K}$ on which we examine $T_X(K)$, bearing in mind that the distribution of X may not be uniquely identified.

The uniform metric (3.20) may be extended to random sets X and Y as follows:

$$\mathcal{U}(X, Y; \mathcal{M}) = \sup\{|T_X(K) - T_Y(K)| : K \in \mathcal{M}\}.$$

The Lévy metric (3.21) can be reformulated for random sets as:

$$\mathcal{L}(X, Y; \mathcal{M}) = \inf\{\varepsilon > 0 : T_X(K) \leq T_Y(K^\varepsilon) + \varepsilon, T_Y(K) \leq T_X(K^\varepsilon) + \varepsilon, \forall k \in \mathcal{M}\},$$

where

$$K^\varepsilon = \cup\{B_\varepsilon(x) : x \in K\} = K \oplus B_\varepsilon(0)$$

is the ε-envelope of K with $\oplus$ being the Minkowski addition and $B_\varepsilon(0)$ being a ball of radius ε centred at the origin. Both metrics depend on the choice of the class $\mathcal{M}$. This class is often chosen to be the class of all singletons, all balls, or all rectangles (parallelepipeds).

Unfortunately, these two metrics (which are useful in the study of general random sets) are not particularly interesting for random sets obtained as thresholds of grey-scale images. If $X = F_U$ and $Y = G_V$ for two grey-scale images f and g and independent random variables U and V are uniformly distributed over $[0, 1]$, then

$$T_{F_U}(K) = \sup_{x\in K} f(x)\,, \quad T_{G_V}(K) = \sup_{x\in K} g(x)\,.$$

Assume that $\mathcal{M}$ contains all singletons. Then

$$\mathcal{U}(F_U, G_V; \mathcal{M}) = \sup |f(x) - g(x)|$$

is the uniform distance between f and g. The Lévy metric is reduced to

$$\begin{aligned}\mathcal{L}(F_U, G_V; \mathcal{M}) = \inf\{\varepsilon > 0 :\ & \Gamma_f \subseteq (\Gamma_g) \oplus (B_\varepsilon \times [0,\varepsilon])\,,\\ & \Gamma_g \subseteq (\Gamma_f) \oplus (B_\varepsilon \times [0,\varepsilon])\}\,,\end{aligned}$$

where Γ_f denotes the subgraph (Definition 1.2) of f (respectively of g) and $B_\varepsilon(x)$, the ball of radius ε, centred at x is defined as,

$$B_\varepsilon(x) = \{y \in W : \rho(x,y) \leq \varepsilon\}.$$

Thus the Lévy metric equals the Hausdorff distance between the subgraphs of the images as subsets of $W \times [0,1]$ where the carrier space is equipped with the metric

$$\rho((x,t),(y,s)) = \max(\rho(x,y), |t-s|)\,, \quad x, y \in W\,, t, s \in [0,1]\,,$$

where $\rho(x,y)$ is the Euclidean distance between pixels x and y.

3.4.2 Integral Metrics

Let ξ and ζ be general random elements in a space $\mathcal{X}$ and let $\mathcal{H}$ denote a family of non-negative measurable functions

$$h : \mathcal{X} \mapsto \mathbf{R}.$$

An approach taken by Müller [34] towards defining a class of metrics between random elements $\xi, \eta \in \mathcal{X}$, called *integral metrics*, is as follows:

$$M(\xi, \eta) = \sup_{h \in \mathcal{H}} |\mathbf{E}[h(\xi)] - \mathbf{E}[h(\eta)]|. \tag{3.22}$$

Note that each expectation in (3.22) may be written explicitly in terms of the corresponding probability measure as:

$$\mathbf{E}[h(\xi)] = \int h(x) dP_{\xi}(x). \tag{3.23}$$

Example 3.1. The uniform (or Kolmogorov) metric between random variables:

$$U(\xi, \eta) = \sup_{t \in \mathbf{R}} |F_{\xi}(t) - F_{\eta}(t)|,$$

is a special case of the integral metric. Since F_{ξ}, the probability distribution function of ξ, may be written as $F_{\xi}(t) = 1 - \int \mathbf{1}_{[t,\infty)} dP_{\xi}$, it is clear that the uniform metric is an integral metric generated by the family of all functions $h(t) = \mathbf{1}_{[t,\infty)}$, $t \in \mathbf{R}$.

Since the integral metric is defined for arbitrary random elements, it is possible to apply it to random sets X, Y:

$$I(X, Y) = \sup_{h \in \mathcal{H}} |\mathbf{E}(h(X)) - \mathbf{E}(h(Y))|.$$

So functions $h \in \mathcal{H}$ are defined on sets. One particular important example is when such functions are sup-measures, so that functions from $\mathcal{H}$ can be identified by their values on singletons.

In particular if

$$h(X) = \sup_{x \in X} h(x),$$

then h is a sup-measure, and this allows a reformulation of the integral metric in terms of Choquet integrals.

Choquet integral

The integral of $h : \mathbb{R}^d \to [0, \infty)$ with respect to the capacity functional $T_X(\cdot)$ is defined as,

$$\int_{\mathbf{R}^d} h \, dT_X = \int_0^\infty T_X(H_t) \, dt, \tag{3.24}$$

where $H_t = \{x : h(x) \geq t\}$. Note that this is called the *Choquet integral* [11, 14].

Following (3.23) and (3.24) the *integral metric* for random sets X and Y may be written as,

$$\begin{aligned} I(X,Y) &= \sup_{h \in \mathcal{H}} \left| \int h \, dT_X - \int h \, dT_Y \right| \\ &= \sup_{h \in \mathcal{H}} \left| \int_0^\infty T_X(H_t) \, dt - \int_0^\infty T_Y(H_t) \, dt \right| . \end{aligned} \tag{3.25}$$

Henceforth we concentrate on applying the integral metric to the special case of random sets generated from images. Consider the situation now where the random set $X = F_U$ and $Y = G_V$ are generated from images f and g. Then the capacity functional takes the following form:

$$T_{F_U}(H_t) = \mathbf{P}\{U \le \sup_{x \in H_t} f(x)\}. \tag{3.26}$$

In the case of a uniformly weighted random set model, the capacity functional may be written as,

$$T_{F_U}(H_t) = \sup_{x \in H_t} f(x)\,. \tag{3.27}$$

Using (3.24) and (3.27) we see that (3.25) can be reformulated as,

$$I(f,g) = I(F_U, G_U) = \sup_{h \in \mathcal{H}} \left| \int_0^\infty \sup_{x \in H_t} f(x)\, dt - \int_0^\infty \sup_{x \in H_t} g(x)\, dt \right| \,. \tag{3.28}$$

Hausdorff-Type Metrics

Suppose now that the images f and g are binary, where each pixel takes, for example, the values 0 or 1. In this case we define $F = \{x \in W : f(x) \neq 0\}$ and similarly for G, so that the random sets F_U and G_U now become deterministic sets F and G respectively.

It may be seen that,

$$\int_0^\infty \sup_{x \in H_t} f(x)\, dt = \int_0^{\sup\{h(x):\, x \in F\}} dt = \sup_{x \in F} h(x)\,.$$

Therefore,

$$I(f,g) = I(F,G) = \sup_{h \in \mathcal{H}} |\sup_{x \in F} h(x) - \sup_{x \in G} h(x)|\,. \tag{3.29}$$

We now show that by choosing the family of functions $\mathcal{H}$ in a certain manner

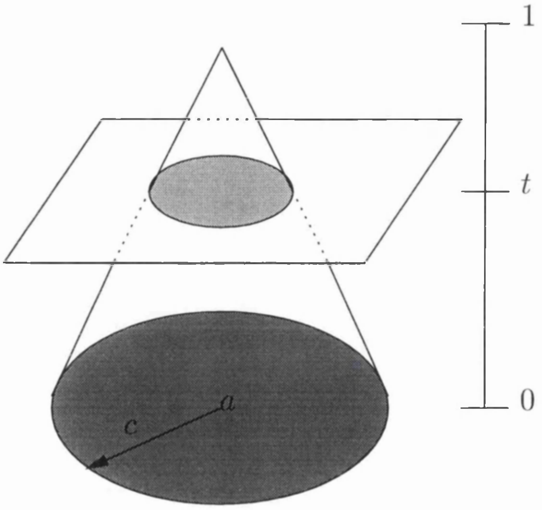


Figure 3.5. Diagram showing the function h^a together with the corresponding level set H_t^a.

the integral metric becomes the Hausdorff metric for binary images. Define

$$\mathcal{H} = \{h^a : a \in W\} \tag{3.30}$$

where

$$h^a(x) = \begin{cases} 1 - \|x - a\|/c & , \|x - a\| \leq 1, \\ 0 & , \text{otherwise}, \end{cases} \tag{3.31}$$

is the cone of height 1 with the circular base of radius c centred at a, see Figure 3.5. We set $h^a(x) = 0$ for all points $x \in W$, not contained in the base of h^a. (We allow for the possibility that the base may not be entirely contained in the window.) Observe that,

$$H_t^a = \{x : h^a(x) \geq t\} = B_{(1-t)c}(a)\,.$$

Theorem 3.2. *Assume that $\mathcal{H}$ is given by* (3.30) *and* (3.31) *with c sufficiently* *large to satisfy*

$$c \geq \operatorname{diam}(W) = \sup\{\|x - y\| : x, y \in W\}.$$

Then

$$I(F, G) = H(F, G)/c.$$

Proof. It may be seen, since c is large enough, that

$$\sup_{x \in F} h^a(x) = \frac{c - \rho(a, F)}{c},$$

whence

$$|\sup_{x \in F} h^a(x) - \sup_{x \in G} h^a(x)| = c^{-1}|\rho(a, F) - \rho(a, G)|.$$

Thus, using (3.29),

$$I(F, G) = c^{-1} \sup_{h^a \in \mathcal{H}} |\rho(a, F) - \rho(a, G)| = c^{-1} \sup_{a \in W} |\rho(a, F) - \rho(a, G)|,$$

as required, using the representation in (3.5). □

It is straightforward to modify (3.29) so that it is equivalent (up to a constant) to the delta metric. To see this, define $\mathcal{H}$ as before, where the radius c of each cone is now taken to be the fixed cut-off value c, as described above. In this instance

$$\sup_{x \in F} h^a(x) = \frac{c - \rho^*(a, F)}{c},$$

where $\rho^*(a, F) = \min(\rho(a, F), c)$, that is the distance transform of F truncated at a value c. Replacing the outer supremum in (3.29) with an L_p average over all points $a \in W$ leaves,

$$I^*(F,G) = \left[\int\limits_W |\sup_{x \in F} h^a(x) - \sup_{x \in G} h^a(x)|^p \, da \right]^{1/p} = \Delta_c^p(F,G)/c. \tag{3.32}$$

3.4.3 Double integral metric

Return now to the situation where F_U, G_U are uniformly weighted random sets generated from images f, g. Generalising (3.32) to grey-scale images leads to a new error metric which we call the *double integral metric*, defined as:

$$\begin{aligned} DI(f,g) &= DI(F_U, G_U) \\ &= \left[\int\limits_W \left| \int\limits_0^1 \sup_{x \in H_t^a} f(x) \, dt - \int\limits_0^1 \sup_{x \in H_t^a} g(x) \, dt \right|^p da \right]^{1/p} . \end{aligned} \tag{3.33}$$

Note that $\sup_{x \in H_t^a} f(x)$ may also be expressed as

$$\sup_{x \in H_t^a} f(x) = (f \oplus B_{(1-t)c})(a).$$

The operator $\oplus$ is termed a dilation operator and the disc $B_{(1-t)c}$ in this situation is termed a structuring element. The idea is that spatial information the image f is obtained as $B_{(1-t)c}$ is translated over the image f. Below in Figure 3.6 is seen an image f dilated by a disc of radius t.

Using (3.33) and changing variables, we get

$$DI_c(f,g) = \frac{1}{c} \left[\int\limits_W \left| \int\limits_0^c [(f \oplus B_t)(a) - (g \oplus B_t)(a)] \, dt \right|^p da \right]^{1/p} . \tag{3.34}$$

The subscript c is introduced into the notation to emphasise the dependence of

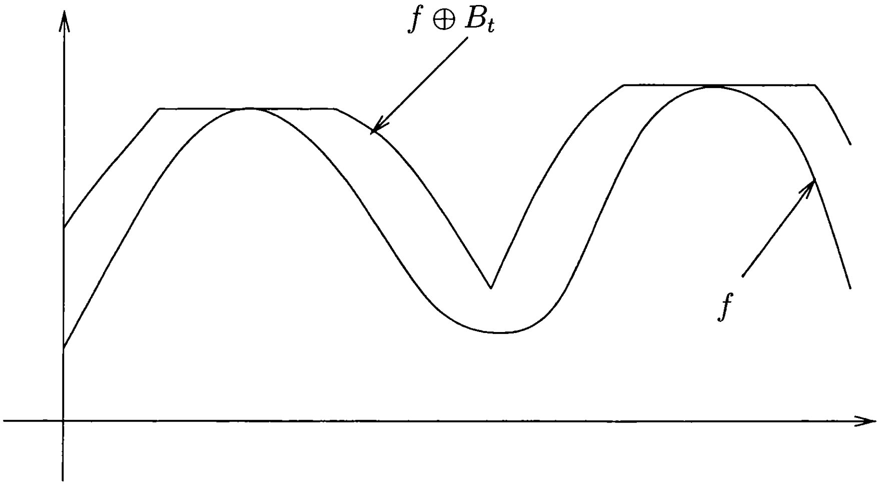


Figure 3.6. Dilation of image f by disc B_t.

the parameter c in the double integral metric. It is interesting to note that letting $c = 0$ in (3.34), we find that $DI_0(f,g) = L_p(f,g)$. That is,

$$DI_0(f,g) = \left[\int_W |f(x) - g(x)|^p \, dx \right]^{1/p}.$$

Note further that $DI_c(f,g) \to |\sup f - \sup g|$ as $c \to \infty$. In general, smaller values of c make the metric more 'local'.

3.4.4 Discrete variant

In all practical applications, where W is a discrete grid of pixels, the double integral metric (3.34) may be discretised as:

$$DI_c(f,g) = \frac{1}{c+1} \left[\sum_{a \in W} \left| \sum_{t=0}^{c} ((f \oplus B_t)(a) - (g \oplus B_t)(a)) \right|^p \right]^{1/p}. \tag{3.35}$$

It is unclear whether in the continuous set-up $DI_c(\cdot,\cdot)$ satisfies the uniqueness axiom of a metric, specifically whether $DI_c(f,g) = 0$ implies $f = g$. However in the discrete framework the following result holds.

Theorem 3.3. *Suppose f, g are defined on a discrete parameter space (or for images, on a grid of pixels), then $DI_c(f,g) = 0$ implies $f = g$, for two upper semicontinuous functions f and g.*

Proof. Let f achieve its maximum at point $a \in W$ and g at point $b \in W$. Assume that $f(a) \geq g(b)$. Note that from (3.35) $DI_c(f,g)$ implies, in particular, that

$$(c+1)^{-1}\sum_{t=0}^{c}(f \oplus B_t)(a) = (c+1)^{-1}\sum_{t=0}^{c}(g \oplus B_t)(a) .$$

If $f(a) > g(a)$, then the left-hand side equals $f(a)$, while the right-hand side is strictly less than $g(b)$. The obtained contradiction shows that $f(a) = g(a) = g(b)$ and so both f and g achieve its maximum at a.

Now let f achieve its maximum on the set $W \setminus \{a\}$ at point a_1, and g at point b_1. Assume that $f(a_1) \geq g(b_1)$. As before (3.35) implies, in particular, that

$$(c+1)^{-1}\sum_{t=0}^{c}(f \oplus B_t)(a_1) = (c+1)^{-1}\sum_{t=0}^{c}(g \oplus B_t)(a_1) . \qquad (3.36)$$

If $f(a_1) > g(a_1)$, and allowing for the fact that point a may or may not lie in the set $B_c(a_1)$, then (3.36) forces a contradiction, and implies that $f(a_1) = g(a_1)$.

Because W is finite (in the discrete setup) we can apply this argument in succession to show that f and g coincide at all points inside W. □

Other variants

It is also possible to formulate other versions of the double integral metric, which may be useful in certain situations. For example a symmetrised variant:

$$\widetilde{DI}_c(f,g) = (DI_c(f,g) + DI_c(\tilde{f},\tilde{g}))/2, \qquad (3.37)$$

where $\tilde{f}(x) = L - f(x)$, with L being the maximum grey level of f. Alternatively a weighted variant of the following form could be considered:

$$DI_{w,c}(f,g) = \frac{1}{c+1}\left[\sum_{a\in W}\left|\sum_{t=0}^{c} w_t((f\oplus B_t)(a) - (g\oplus B_t)(a))\right|^p\right]^{1/p}$$

where w_t's are non-negative weights summing to unity. These weights could be used to emphasise different structuring elements B_t.

3.5 Grey-scale distance transform metric

Recall from Section 2.3.2 that the grey-scale distance transform for an image f is written as:

$$\bar{d}_f(x) = \mathbf{E}d(x, F_U).$$

The subscript f is introduced to denote the dependency on image f. In the discrete case, for an image with 256 grey levels $\{0, 1, \ldots, 255\}$, $\bar{d}_f(x)$ for the uniformly weighted model is written as,

$$\bar{d}_f(x) = \frac{1}{t_2 - t_1 - 1}\sum_{t=t_1}^{t_2} d(x, F_t),$$

where t_1, t_2 are the minimum and maximum grey levels respectively, in f. So in this case $\bar{d}_f(x)$ is simply an arithmetic average of distance functions of each thresholded level set over the effective grey-scale range of the image.

The histogram weighted model leads to the grey-scale distance transform

$$\bar{d}_f(x) = \sum_{t=t_1}^{t_2} \frac{h_f(t)}{n(W)} d(x, F_t).$$

It is noted in [33], that for the discrete case, there is a uniqueness between grey-scale images and corresponding grey-scale distance transforms. It is unclear whether this holds in the continuous case.

The idea of the *grey-scale distance transform metric* is simply to use distances between grey-scale distance transforms as a measure of distance between their underlying images. Thus the distance between images f and g is written as:

$$GDT(f,g) = ||\bar{d}_f(x) - \bar{d}_g(x)||. \tag{3.38}$$

It is seen that $GDT(\cdot,\cdot)$ depends on three choices, namely both the random set model, and distance function used to compute the grey-scale distance transform, and then the choice of norm used to evaluate distances between $\bar{d}_f(x)$ and $\bar{d}_g(x)$.

In fact the $GDT(\cdot,\cdot)$ is used implicitly to find the distance threshold. If a uniformly weighted random set model is used, then for a binary image F, its grey-scale distance transform reduces to the distance transform $d(\cdot, F)$. In this case the distance threshold level is then chosen as the grey level t which minimises $GDT(f, F_t)$. Similarly it is suggested in Section 2.3.6 that the k grey levels chosen to form the multithresholded image f_k be chosen in such a way that $GDT(f, f_k)$ is minimised over all choices of k grey levels.

It is worth noting that $\Delta_p^c(\cdot,\cdot)$ is a special case of $GDT(\cdot,\cdot)$ applied to binary images. Suppose f, g are binary images, then a uniformly weighted random set model implies trivially, that $\bar{d}_f(x) = d(x, F)$. Further if this distance function is a truncated distance function, $d^*(x,F) = \min(d(x,F), c)$, and an L^p norm is used in (3.38), then

$$GDT(f,g) = \left\{ \frac{1}{n(W)} \sum_W |\rho^*(x,F) - \rho^*(x,G)|^p \right\}^{1/p} = \Delta_c^p(f,g).$$

3.6 Some examples

In this section we compare results of the two newly introduced image metrics, DI_c, GDT together with the widely used RMS distance. Throughout we have used the symmetrised version (3.37) of the double integral metric, where the value of p is fixed at $p = 2$. Further for the grey-scale distance transform metric, we have used the histogram weighted variant, with a signed distance function truncated at $c = 8$, and L_2 norm. Each of the three metrics have been rescaled, by multiplying each by an appropriate constant, so that they each give values between 0 and 255.

3.6.1 Experiment 1. Local image distortion

Figure 3.7 shows five images of the letter 'i', where the width of the dot on the 'i' is being gradually reduced. Each of the three metrics compared the original letter 'i' with each of the distorted letters. The aim of this experiment is to investigate how well each metric recognises when the dot on the 'i' was fully removed.

These images are grey-scale images where i.i.d. Gaussian noise of mean 0 and standard deviation 30 grey level units have been added to original images with just two grey levels 0 and 255. All pixels with grey levels greater than 255 and less than 0 were truncated to lie in the grey-scale range $\{0, \ldots, 255\}$. The noise is independent between both the pixels and the images.

From the left graph in Figure 3.8 the $x-$axis shows the number of columns removed from the original 'i' image, before noise was added. The $y-$axis shows for each metric the distance between corresponding images. Each metric shows an increase in distance between the original letter 'i' when compared with successive images showing reduction in the dot. Further GDT and in particular $\widetilde{DI}$ show smaller values than RMS. The right graph in Figure 3.8 plots the differences

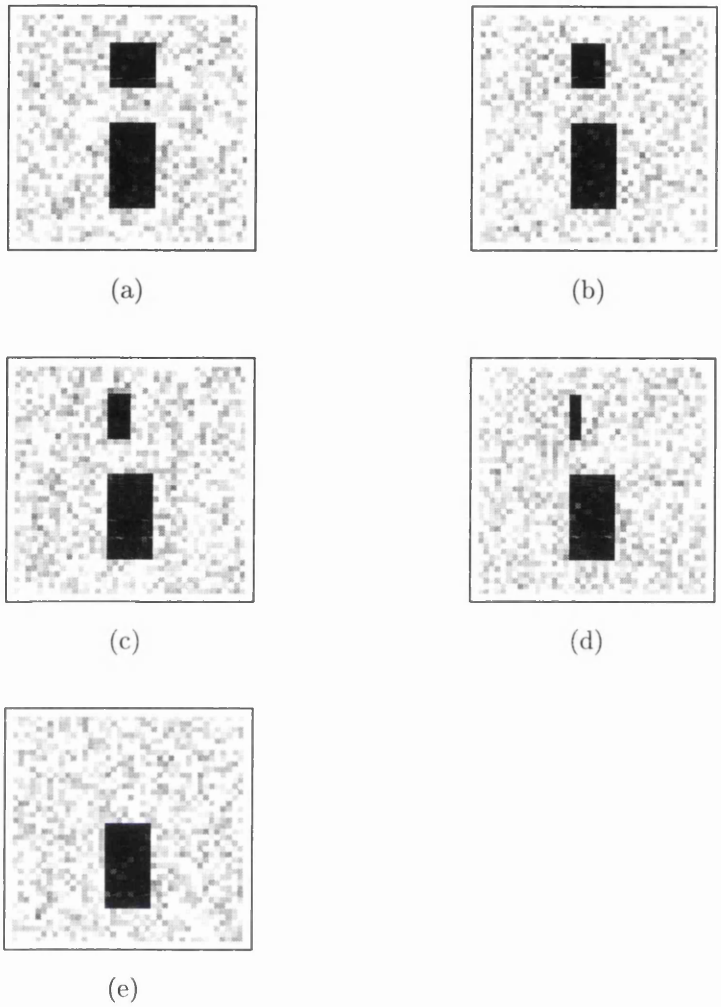

(a) (b) (c) (d) (e)

Figure 3.7. Letter ‘i’ image showing gradual reduction of the width of the dot on the ‘i’

between adjacent results for each distance in the left graph. It is clear that the greatest changes in distance, when the dot on the ‘i’ has been fully removed, has occurred for both $\widetilde{DI}$ and GDT. This strengthens the idea that both account for some spatial information in the image, while the RMS seems to ignore this information.

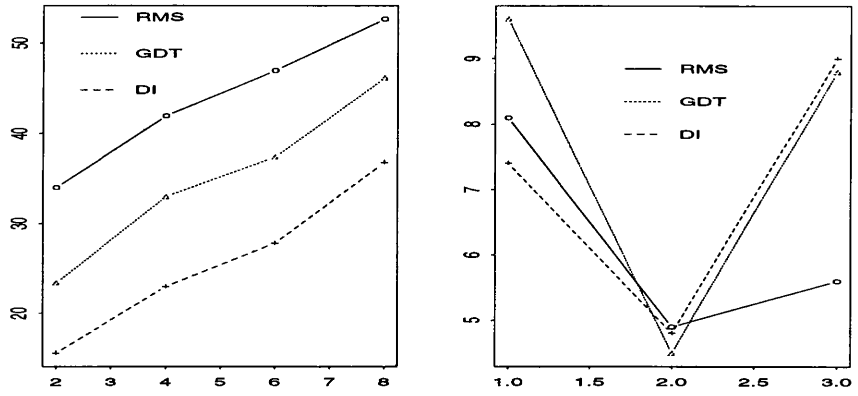
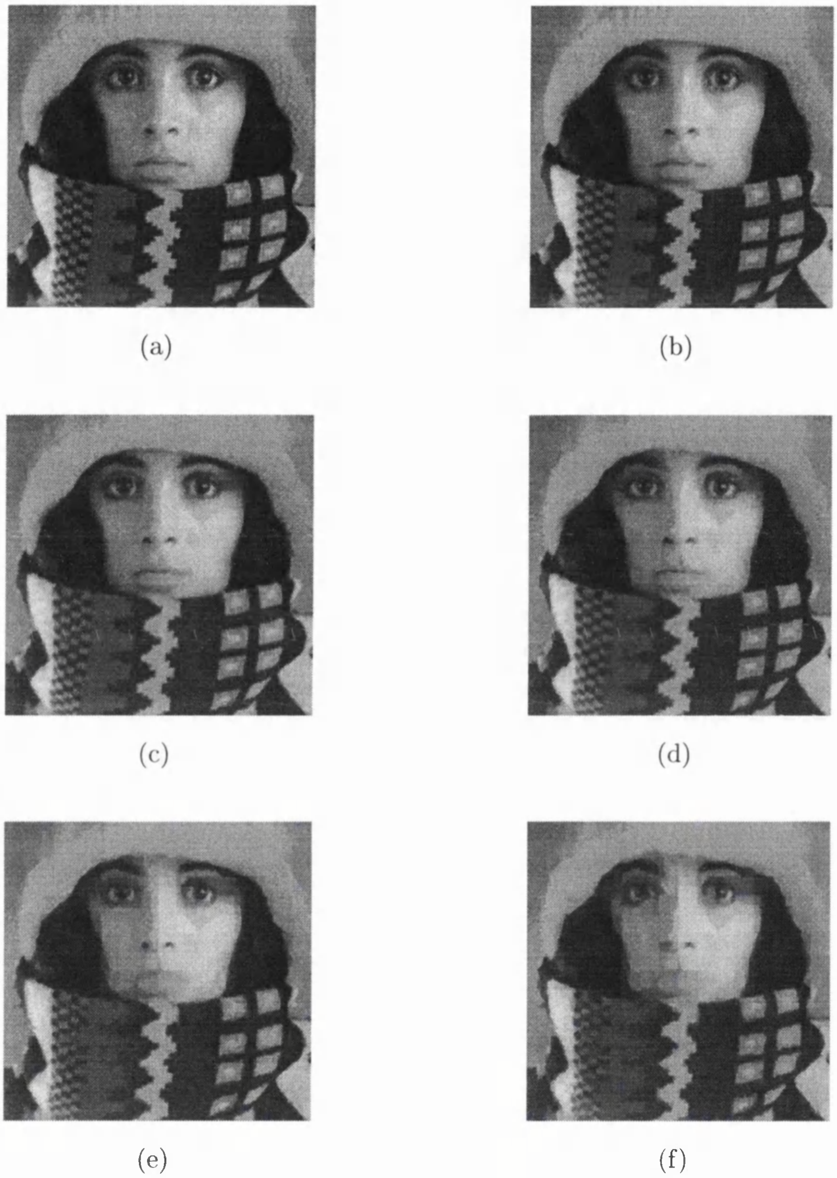


Figure 3.8. The left graph indicates values of each of the three metrics when the original image in Figure 3.7(a) is compared with each of images (b)-(f) in Figure 3.7. The right graph plots the differences between adjacent results for each distance in the left graph.

3.6.2 Experiment 2. Image compression/decompression

In this experiment the trui image in Figure 3.9 (a) was distorted by compressing it by increasing degrees of compression. The original image was then compared with each decompressed image shown in Figure 3.9 (b)-(f). The images were compressed using a fractal image compressor. Note that increased compression leads to a decrease in image fidelity.

Examining the images visually, it would appear that although image quality decreases with increased compression, the greatest decrease in image quality occurs moving from image (d) to images (e) and (f). This is indeed reflected in the $\widetilde{DI}$ values in Figure 3.10. By comparison, RMS shows an almost uniform transition between each image, while GDT finds the largest decrease in image quality between images (c) and (d).



(a) (b) (c) (d) (e) (f)

Figure 3.9. (a) `trui` image, (b)-(f) compressed/decompressed `trui` images

3.6.3 Experiment 3. Image dilation/erosion

This experiment compares visual quality of images distorted by erosions. An image f eroded by a so-called structuring element $S \subset \mathbb{R}^2$ corresponds to the

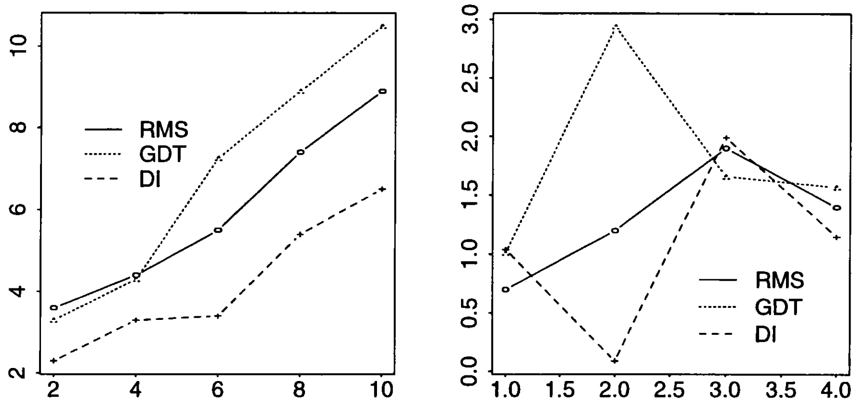


Figure 3.10. The left graph indicates values of each of the three metrics when the original image in Figure 3.9(a) is compared with each of the compressed/decompressed images (b)-(f) in Figure 3.9. The right graph shows differences between adjacent values in the left graph.

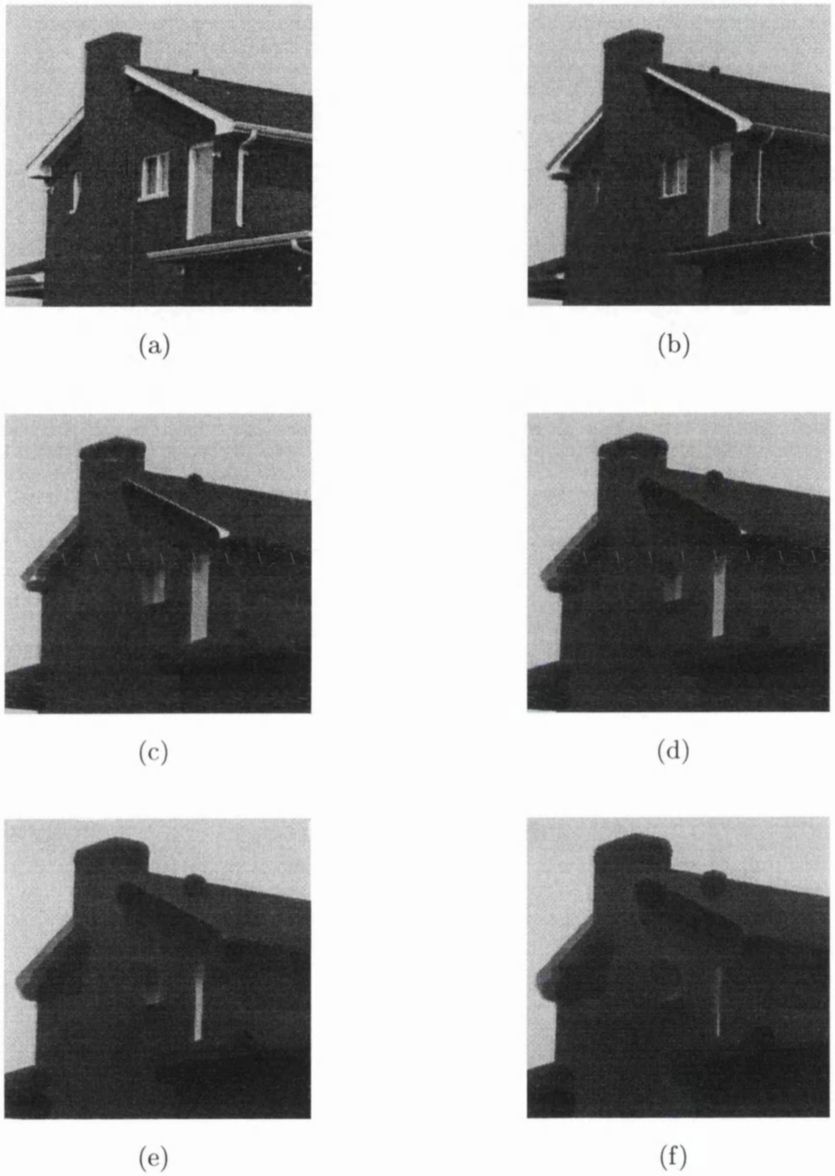
image,

$$(f \ominus S)(x) = \inf_{y \in S+x} f(y).$$

So that this image is formed by taking the minimum of a set of pixel values in the moving window S.

The **house** image in Figure 3.11(a) eroded by a disc of radius 2 is shown in Figure 3.11(b). The radius of the disc has increased by 2 successively for images (c)-(f).

It is seen the image quality decreases as the radius of the disc increases. This is reflected in the left graph in Figure 3.12. However visually it would seem that greatest difference between successive images occurs between image (b) and image (c). Much of the features of the house are missing in image (c) and all subsequent images. In particular $\widetilde{DI}$ seems to notice this change, as is reflected in the right graphs in Figure 3.12, which shows differences in adjacent values from graphs on the left. Both GDT and RMS by comparison, don't show as dramatic a change between image (b) and subsequent images.



(a) (b) (c) (d) (e) (f)

Figure 3.11. (a) house image, (b)-(f) eroded house images

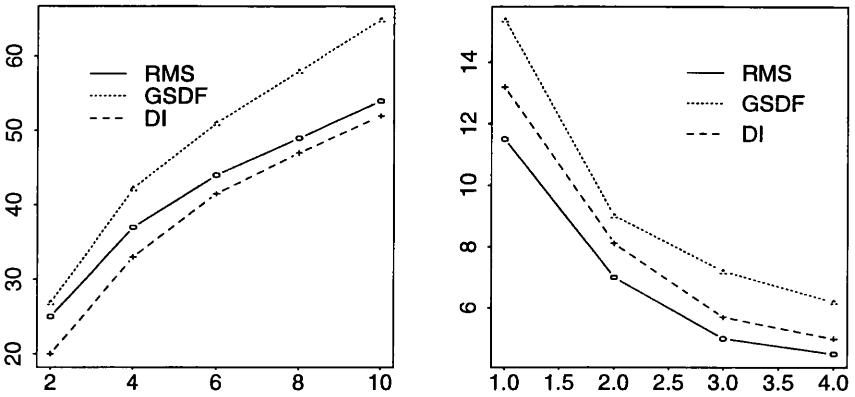
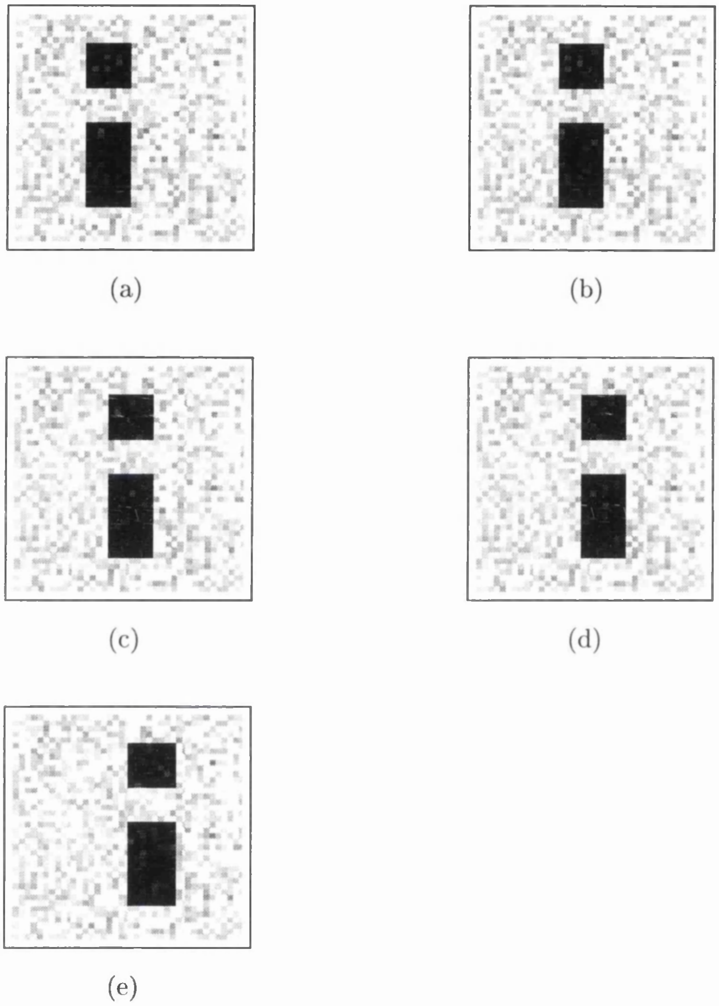


Figure 3.12. The left graph indicates values of each of the three metrics when the original image in Figure 3.11(a) is compared with each of the eroded images (b)-(f) in Figure 3.11. The right graph shows differences between adjacent values in the left graph.

3.6.4 Experiment 4. Image translation

Figure 3.13 shows 5 images of the letter 'i'. These images were obtained by repeatedly translating a binary image, with two values 0 and 255, of the letter 'i', corresponding to Figure 3.13(a) two pixels horizontally to the right. Gaussian i.i.d. noise of mean 0 and standard deviation 30 grey level units was then added to each image. All pixels with grey levels greater than 255 and less than 0 were then truncated to lie in the grey-scale range $\{0, 1, \ldots, 255\}$. The noise is independent between the pixels and the images.

The left graph in Figure 3.14 shows the results of the three metrics when image (a) is compared with each of images (b)-(e). The x-axis shows the number of pixels that have been horizontally translated. The $y-$axis shows for each metric the distance between corresponding images. It is curious to note that *GDT* shows a decrease in distance between image (d) and image (e). This could be explained by random variation in both images. However both *GDT* and in particular $\widetilde{DI}$ show considerably less values than *RMS*, recognising that all five images are highly similar. The right graph in Figure 3.14 shows differences between adjacent values in the left graph.


(a)

(b)

(c)

(d)

(e)

Figure 3.13. (a) i image, (b)-(e) translated images

3.6.5 Summary of results

These above results have attempted to assess the performance of each of the two newly presented image metrics, $\widetilde{DI}$ and GDT, when compared to the usual RMS distance. In each experiment it could be argued that $\widetilde{DI}$ has perfomed better

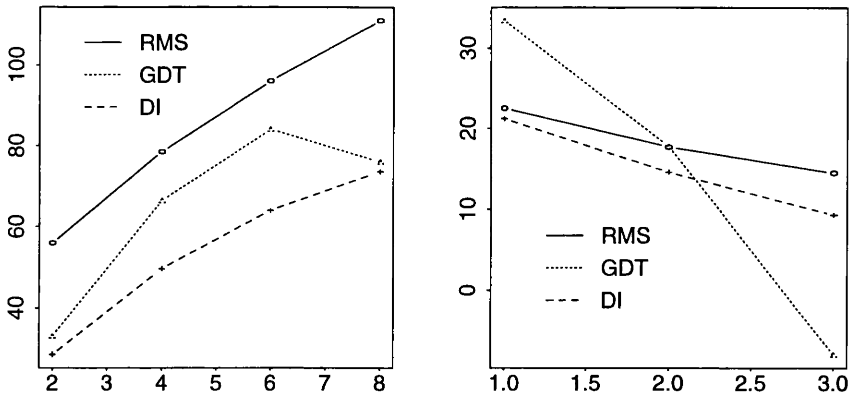


Figure 3.14. The left graph indicates values of each of the three metrics when the original image in Figure 3.13(a) is compared with each of the translated images (b)-(e) in Figure 3.13. The right graph shows differences between adjacent values in the left graph.

than *RMS*. However it should also be noted that *RMS* itself has performed adequately. The performance of *GDT* is a little erratic. In some cases it could be argued that it has performed better than *RMS*, while in other instances this is not true.

Chapter 4

Bayesian Image Restoration

4.1 Introduction

Image restoration involves estimation of some unknown true image scene from a known noisy version of it. In this chapter we concentrate solely on a Bayesian approach. Using Bayesian notation, the true unknown image is denoted by $\mathbf{x}$, and the observed noisy version by $\mathbf{y}$. Here $\mathbf{x}$ is modelled as a random variable with prior distribution $\pi(\mathbf{x})$ and in particular as a (local) Markov random field (MRF). Following a Bayesian approach, inference is based on the posterior distribution,

$$\pi(\mathbf{x}|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}).$$

The noisy data $\mathbf{y}$ is acquired with some known likelihood $\pi(\mathbf{y}|\mathbf{x})$.

Common estimators of $\mathbf{x}$ include that image which maximises the posterior distribution, called the MAP estimator, and the image corresponding to the expectation of the posterior distribution, the PE estimator. In practice the posterior distribution is often analytically intractable and in this instance a solution may be to represent the posterior distribution by samples. As is typical in Bayesian statistics, Markov chain Monte Carlo (MCMC) methods are the most widely used tool for this purpose. We will speak more about these methods later.

Several authors [16, 19, 44, 45] have approached the Bayesian restoration problem from a decision theoretic viewpoint. Here a first step is to select a suitable image metric, or in more usual terminology, loss function $L(\mathbf{x}, \mathbf{z})$. So that this gives a measure of discrepancy between the true image $\mathbf{x}$ and an estimated image $\mathbf{z}$. Following this set-up the optimal Bayes estimator (OBE) is then chosen as the configuration minimising the posterior expectation of this loss (often called the risk),

$$\mathbf{x}^* = R(\mathbf{z}) = \arg\min_{\mathbf{z}} \mathbf{E}_{\mathbf{x}|\mathbf{y}} L(\mathbf{x}, \mathbf{z}). \tag{4.1}$$

While much work has gone into modelling prior distributions, very little, by comparison, has involved loss function modelling. This might be due to the complexity of solving (4.1). In this chapter we show a simple framework for loss function modelling, generalising an algorithm described in [19]. Results of this and other algorithms are presented on real and synthetic data.

4.2 Bayesian framework

Henceforth in this chapter we deviate slightly from notation used previously and adopt the standard notation used in Bayesian literature, representing images as vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. Pixels will be represented by the letters i, j and pixel values or grey levels will be denoted by the letter p, where the grey-scale range equals $GL = \{0, 1, \ldots, l-1\}$.

The true unknown image $\mathbf{x}$ is assumed to be a realisation of a random vector

$$\mathbf{X} = \{\mathbf{X}(i) : i \in W\}. \tag{4.2}$$

While the observed image $\mathbf{y}$ is a realisation of a random vector

$$\mathbf{Y} = \{\mathbf{Y}(i) : i \in W\}, \tag{4.3}$$

caused by some known stochastic degradation of $\mathbf{x}$.

Here we make some assumptions (following closely [4]):

1. The random variables $\{\mathbf{Y}(i) : i \in W\}$ are conditionally independent and have the same conditional density function dependent on $\mathbf{X}$. Thus the joint density function of $\mathbf{y}$ given $\mathbf{X} = \mathbf{x}$ is:

$$\pi(\mathbf{y}|\mathbf{x}) = \prod_{i \in W} \pi(\mathbf{y}(i)|\mathbf{x}). \tag{4.4}$$

   This is essentially a likelihood function of the data, and incorporates knowledge of the stochastic noise model.

2. The true image $\mathbf{x}$ is a realisation of a locally dependent Markov random field. In particular this means that the following two conditions hold:

   (a) $\pi(\mathbf{X} = \mathbf{x}) > 0$ for all $\mathbf{x}$ in the sample space of $\mathbf{X}$.

   (b) $\pi\left(\mathbf{X}(i) = \mathbf{x}(i)|\mathbf{X}(j) = \mathbf{x}(j), j \neq i\right) =$
   $\pi\left(\mathbf{X}(i) = \mathbf{x}(i)|\mathbf{X}(j) = \mathbf{x}(j), j \in \sigma_i\right),$

   where σ_i is some specified neighbourhood structure centred at pixel $i \in W$.

A first-order neighbourhood consists of those pixels immediately adjacent to pixel i. Edge pixels have three rather than four neighbouring pixels, except at corners where they have two. A second-order neighbourhood in addition consists of those pixels diagonally adjacent to pixel i.

Figure 4.3 shows first and second-order neighbourhood structures.

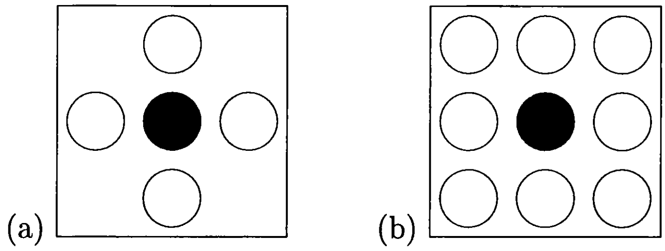


Figure 4.1. (a) first-order neighbourhood (b) second-order neighbourhood

Assumptions 1 and 2 represent our beliefs and knowledge about the observed and true images. This information is merged following Bayes theorem to form the posterior distribution:

$$\pi(\mathbf{x}|\mathbf{y}) = \frac{\pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})}{\int \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})\, d\mathbf{x}}. \tag{4.5}$$

The denominator here is typically difficult to compute and so the posterior distribution is written as:

$$\pi(\mathbf{x}|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}).$$

Of course in practice the posterior distribution will often be analytically intractable and in this instance a solution may be to represent the posterior distribution by samples. As is typical in Bayesian statistics, Markov chain Monte Carlo (MCMC) methods are the most widely used tool for this purpose.

4.3 MCMC methods - an overview

In this section we provide a brief introduction to MCMC methods widely used in Bayesian Statistics. The reader is referred to [22] and references therein for a more comprehensive discussion.

Consider the following problem: How may we compute the posterior expectation of a function f of $\mathbf{X}$? Formally this may be written as

$$\mathbf{E}_{\mathbf{x}|\mathbf{y}}[f(\mathbf{X})] = \frac{\int f(\mathbf{x})\, \pi(\mathbf{y}|\mathbf{x})\, \pi(\mathbf{x})\, d\mathbf{x}}{\int \pi(\mathbf{y}|\mathbf{x})\, \pi(\mathbf{x})\, d\mathbf{x}}. \tag{4.6}$$

As above with (4.5) the normalising constant in (4.6) is difficult to compute and a final resort may be to compute this expectation by drawing samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ from $\pi(\mathbf{x}|\mathbf{y})$ and then approximating

$$\mathbf{E}_{\mathbf{x}|\mathbf{y}}[f(\mathbf{X})] \approx \frac{1}{N} \sum_{t=1}^{N} f(\mathbf{x}_t). \tag{4.7}$$

This is called Monte Carlo integration. This shows that the population mean is approximated by a sample mean. Clearly when the samples $\{\mathbf{x}_t\}$ are independent, laws of large numbers ensure that approximations can be made more accurate by increasing the sample size N. However drawing samples $\{\mathbf{x}_t\}$ independently from $\pi(\mathbf{x}|\mathbf{y})$, in general, not possible. It may happen that the posterior distribution is quite non-standard, so that it would become necessary that, generally speaking, samples from $\pi(\mathbf{x}|\mathbf{y})$ be chosen in the correct proportions. Samples generated by a Markov chain having $\pi(\mathbf{x}|\mathbf{y})$ as its stationary distribution is a solution. This is called Markov chain Monte Carlo (MCMC).

The following shows how a Markov chain may be constructed so that images may be sampled from the posterior distribution. The method which we first examine is based on the work of Hastings [24], which generalises an earlier paper by Metropolis et al. [30]. We describe this in the context of image analysis as follows.

4.3.1 Metropolis-Hastings algorithm

The *Metropolis-Hastings algorithm* [24] is widely used throughout Bayesian statistics. We briefly describe its implementation below. Let us begin by denoting successive states (images) in the Markov chain at time t by $\mathbf{x}_t$. At time t, choose uniformly at random for each pixel i in $\mathbf{x}_t$ a new colour or grey level p from some proposal distribution $q(\cdot|\cdot)$. (The proposal probability $q(p|\mathbf{x}_t(i))$ denotes the probability of changing from grey level $\mathbf{x}_t(i)$ to grey level p.) Denote the proposed new image configuration by $\hat{\mathbf{x}}_t$ and the old configuration by $\mathbf{x}_t$. The move is accepted with probability

$$\min\left(1,\ \frac{\pi(\hat{\mathbf{x}}_t|\mathbf{y})\, q(\mathbf{x}_t(i)|p)}{\pi(\mathbf{x}_t|\mathbf{y})\, q(p|\mathbf{x}_t(i))}\right) \tag{4.8}$$

If a candidate grey level for pixel i is accepted, then the updated image is defined as $\mathbf{x}_t = \hat{\mathbf{x}}_t$. This procedure is repeated until all pixels have been visited just once, at which point the iteration increases from t to $t+1$.

It has been shown in fact that for any proposal distribution $q(\cdot|\cdot)$, convergence to $\pi(\mathbf{x}|\mathbf{y})$ is guaranteed. See [22, chapter 3,4] for details. However the choice of $q(\cdot|\cdot)$ is of importance. A cautious proposal distribution generating small changes in grey-levels will have a high acceptance rate, but will mix slowly, while a bold proposal giving large changes in pixels values will have a low acceptance rate and again will mix slowly. In both instances convergence to the stationary distribution will be slow. A proposal distribution somewhere between these extremes would be the ideal.

The following update mechanism is a particular case of the Metropolis-Hastings algorithm.

4.3.2 Metropolis algorithm

The *Metropolis algorithm* [30] is similar to the Metropolis-Hastings algorithm except that it only considers symmetric proposal distributions. That is distributions having the form $q(p_1|p_2) = q(p_2|p_1)$. It is seen now that the acceptance probability (4.8) reduces to

$$\min\left(1,\ \frac{\pi(\hat{\mathbf{x}}_t|\mathbf{y})}{\pi(\mathbf{x}_t|\mathbf{y})}\right), \tag{4.9}$$

thus avoiding the need to compute the proposal probability for each proposal.

In all examples henceforth, we use the Metropolis algorithm. In most applications we propose grey levels p from a uniform distribution given by:

$$p \sim q(\cdot|x) = U[\mathbf{x}_t(i) - \alpha, \mathbf{x}_t(i) + \alpha]. \tag{4.10}$$

So grey-level p is proposed uniformly from the integers in the closed interval $[\mathbf{x}_t(i) - \alpha, \mathbf{x}_t(i) + \alpha]$. Here α is chosen so that the distribution is neither not too narrow nor too wide, and is, generally speaking, determined by the number of grey levels in the original image. However care needs to be taken to ensure that proposed pixel values don't lie outside the original grey-scale range. Thus if $p \sim q(\cdot|\mathbf{x}_t(i)) \notin [0, l-1]$, the original grey-scale range, p is set equal to $\mathbf{x}_t(i)$. This ensures that $q(\cdot|\cdot)$ is a symmetric proposal distribution.

The Metropolis algorithm may be written algorithmically as:

1. Choose some starting image configuration $\mathbf{x}_0$ arbitrarily.
2. Choose pixel i at random from $\mathbf{x}_t$ and sample a new pixel value p from a symmetric proposal distribution $q(\cdot|\mathbf{x}_t(i))$ (for example, (4.10) as described

above). Set $\hat{\mathbf{x}}_t(i) = p$ and define

$$\hat{\mathbf{x}}_t = \begin{cases} \mathbf{x}_t(j), \; j \neq i \\ \hat{\mathbf{x}}_t(i), \; j = i. \end{cases}$$

(So $\hat{\mathbf{x}}_t$ is identical to $\mathbf{x}_t$ except at pixel i)

3. Calculate the ratio

$$\lambda = \frac{\pi(\hat{\mathbf{x}}_t|\mathbf{y})}{\pi(\mathbf{x}_t|\mathbf{y})}. \tag{4.11}$$

4. Sample a random variable U from a $U(0,1)$, a uniform distribution on $[0,1]$.

5. If $U \leq \lambda$ set $\mathbf{x}_t(i) = \hat{\mathbf{x}}_t(i)$.

6. Repeat from step 2 until all pixels in the window have been visited.

7. Increase iteration counter t and repeat from step 2 until convergence.

It is straightforward to show that (4.11) can be simplified much further, using assumptions 1 and 2, and the fact that the images $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$ differ only by the grey level of pixel i:

$$\frac{\pi(\hat{\mathbf{x}}_t|\mathbf{y})}{\pi(\mathbf{x}_t|\mathbf{y})} = \frac{\pi(\mathbf{y}(i)|p)\, \pi(p|\mathbf{x}_{t,-i})}{\pi(y(i)|\mathbf{x}_t(i))\, \pi(\mathbf{x}_t(i)|\mathbf{x}_{t,-i})}. \tag{4.12}$$

Here $\mathbf{x}_{t,-i}$ denotes the set of all pixel grey levels in $\mathbf{x}_t$ excluding pixel grey level i. Each factor on the left-hand side of (4.12) is straightforward to compute. So it is seen that the Metropolis algorithm is relatively easy to implement.

Let us return briefly to the question posed at the beginning of Section 4.3, namely to the problem of computing the posterior expectation of a function f of $\mathbf{X}$. In light of the above the solution is to first simulate a Markov chain, using for example the Metropolis algorithm, for a sufficiently long number of iterations or burn-in, say m iterations until its distribution is close to the stationary distribution, continuing to obtain dependent samples $\{\mathbf{x}_t, t = m, \ldots, N + m\}$ from the stationary distribution. The expectation $\mathbf{E}[f(\mathbf{X})]$ may now be estimated as:

$$\mathbf{E}_{\mathbf{xy}}\widehat{[f(\mathbf{X})]} = \frac{1}{N} \sum_{t=m+1}^{N+m} f(\mathbf{x}_t) \tag{4.13}$$

This is called an ergodic average. The ergodic theorem establishes convergence to the required expectation, as $N \rightarrow \infty$. See [22, p.45-48] for details.

4.4 Simulated annealing

Previously we have seen methods to sample from complex distributions and to compute expectations. Here we discuss a powerful method for solving global optimisation problems. Simulated annealing was introduced into the realm of statistics by Geman and Geman [20]. Annealing is a term used in Physics which describes a procedure whereby certain chemical systems are driven to their low energy highly regular states.

Simulated annealing works as follows. Let $R(\mathbf{z})$ be a function, of an image $\mathbf{z}$, which we aim to minimise. In an imaging context $\mathbf{z}$ might be some image configuration. The idea is to interpret $\mathbf{z}$ as sampled from a distribution depending on a parameter t, called the temperature:

$$\pi(\mathbf{z}; t) = \frac{1}{C} \exp\{-R(\mathbf{z})\}^{1/T(t)}, \tag{4.14}$$

where C is a normalising constant, ensuring that $\pi(\mathbf{z}; t)$ sums to 1 for all possible $\mathbf{z}$. The constant C is usually difficult to calculate explicitly, so instead $\pi(\mathbf{z}; t)$ is written in the form,

$$\pi(\mathbf{z};t) \propto \exp\{-R(\mathbf{z})\}^{1/T(t)}.$$

As shall be seen later, the constant C is not crucial to any calculations.

The first step is to check that $\pi(\mathbf{z};t)$ induces an MRF or equivalently that $\pi(\mathbf{z};t)$ is a so-called Gibbs distribution. (In [20], the interest was primarily in computing the MAP estimate, that is, the global minimiser of the posterior distribution $\pi(\mathbf{x}|\mathbf{y})$, which is an MRF.)

Our aim now is to find the global maximiser or mode of $\pi(\mathbf{z};t)$. Since $\exp(\cdot)$ is an increasing function, the mode of $\pi(\mathbf{z};t)$ will correspond to the mode of $-R(\mathbf{z})$ and so the minimum of $R(\mathbf{z})$. Generally speaking the problem of maximising $\pi(\mathbf{z};t)$ is computationally intensive. Suppose for instance that $\mathbf{z}$ is an image of size 64×64 with 64 possible colours, then the number of possible image configurations is approximately 2^{1500}. Further, if $\pi(\mathbf{z};t)$ is a complex distribution, with many local maxima, then convergence of $\mathbf{z}$ to the global maximum of $\pi(\mathbf{z};t)$ may be difficult to ensure. Usual iterative search algorithms may get trapped in local maxima. Simulated annealing tends to avoid such problems.

Loosely speaking, the approach is to use MCMC methods to sample from $\pi(\mathbf{z};t)$, which when combined with the temperature schedule $T(t)$ guarantees that if $T(t)$ tends sufficiently slowly to zero, sampling will be from the mode of $\pi(\mathbf{z};t)$, that is, convergence to the global maximum of $\pi(\mathbf{z};t)$. At high temperatures the distribution is essentially uniform, while as the temperature decreases slowly, $\pi(\mathbf{z};t)$ becomes more and more 'spiked' around its global maximum. Thus small values of $T(t)$ exaggerate the mode of $\pi(\mathbf{z};t)$, making it easier to find by sampling. See Geman, Geman [20] for convergence results. The choice of temperature schedule $T(t)$ is of importance. It is stated in [20] that if the temperature schedule satisfies the bound

$$T(t) \geq \frac{c}{\log(1+t)},$$

where c is a constant independent of t, then convergence to the target global maximum is guaranteed. However in practice this schedule may be too slow, and faster schedules may be needed.

Simulated annealing requires only slight modification from, for example, the Metropolis algorithm, and so is straightforward to implement. The algorithm may be written as:

1. Choose some starting image configuration $\mathbf{z}_0$ and cooling schedule $T(t)$, defined so that $T(t)$ tends slowly to zero as iteration t increases.

2. At iteration t choose pixel i at random from $\mathbf{z}_t$ and sample a new pixel value p from a symmetric proposal distribution $q(\cdot|\mathbf{z}_t(i))$ (for example, as described above). Set $\hat{\mathbf{z}}_t(i) = p$ and define

$$\hat{\mathbf{z}}_t = \begin{cases} \mathbf{z}_t(j), \ j \neq i \\ \hat{\mathbf{z}}_t(i), \ j = i. \end{cases}$$

   (So $\hat{\mathbf{z}}_t$ is identical to $\mathbf{z}_t$ except at pixel i)

3. Calculate the ratio

$$\lambda = \frac{\pi(\hat{\mathbf{z}}_t; t)}{\pi(\mathbf{z}_t; t)}. \qquad (4.15)$$

   (This step renders the calculation of the constant C (4.14) unnecessary.)

4. Sample a random variable U from $U(0, 1)$, a uniform distribution on $[0, 1]$.

5. If $U \leq \lambda$ set $\mathbf{z}_t(i) = \hat{\mathbf{z}}_t(i)$.

6. Repeat from step 2 until all pixels in the window have been visited.

7. Increase iteration counter t and repeat from step 2 until convergence to the mode of $\mathbf{z}$.

As with the Metropolis algorithm, the ratio (4.15) may be split into easily computed factors similar to (4.12).

4.5 Loss functions and Bayesian image restoration

Let us return now to the main thrust of this chapter, that of restoring a noisy image $\mathbf{y}$ given information of the noise degradation and prior information of the true image $\mathbf{x}$. Inference is based on the posterior distribution. At this stage all our tools are in place. MCMC methods allow us to both sample from the posterior distribution and to estimate expectations. Further simulated annealing allows us to optimise functions and distributions.

Since we are going to base all our inference on the posterior distribution, plausible estimates for the true image would thus seem to be, for example, that image which maximises the posterior distribution, which is called the maximum a posteriori or MAP estimate. Another suitable estimator might be that image corresponding to the posterior mean which we call the PE estimate. Both of these estimators may be calculated easily via MCMC methods. A simulated annealing algorithm could be used to calculate $\mathbf{x}_{MAP}$, while a Metropolis algorithm could be used to sample images from the posterior which would then used to form the posterior mean estimate $\mathbf{x}_{PE}$, even without simulated annealing.

Suppose now for a moment that the true image is known and that several estimates of it are available, for example $\mathbf{x}_{MAP}$, $\mathbf{x}_{PE}$. How then should we compare the estimates? Following Chapter 3, the answer is simply to choose a suitable image metric, say L, and then find the image which returns the smallest metric distance when compared with the true image. Suppose now that the true image is unknown. We can still use the same basic idea. Compute the posterior expectation of L for each estimate, choosing the estimate returning the least value.

Returning to the present situation, a similar idea still applies. For a given image metric L the optimal Bayes estimator (OBE) is computed by minimising the expectation of this metric,

$$\mathbf{x}^* = \arg\min_{\mathbf{z}} \mathbf{E}_{\mathbf{x}|\mathbf{y}} L(\mathbf{x}, \mathbf{z}). \tag{4.16}$$

In this instance the image metric L is termed a *loss function.* Henceforth we adopt this terminology. This decision theoretic approach has been taken by several authors including [16, 19, 44, 45], with good effect.

4.5.1 Usual Bayesian estimators

Many of the widely used Bayesian estimators can be easily written in the form of (4.16). The MAP estimate which maximises the posterior distribution has as a corresponding loss function:

$$L_{MAP}(\mathbf{x}, \mathbf{z}) = \mathbf{1}_{[\mathbf{x} \neq \mathbf{z}]}. \tag{4.17}$$

Here $\mathbf{1}_{[\mathbf{x} \neq \mathbf{z}]}$ denotes the indicator function, taking the value 1 when $\mathbf{x} \neq \mathbf{z}$ and 0 otherwise. The PE estimate, which corresponds to the mean of the posterior, has a loss function corresponding to:

$$L_{PE}(\mathbf{x},\mathbf{z}) = \sum_{i \in W} (\mathbf{x}(i) - \mathbf{z}(i))^2. \tag{4.18}$$

This is the sum of squared differences between images $\mathbf{x}$ and $\mathbf{z}$. While the mode of the marginal posterior distribution, MPM estimate, can also be written in the form of (4.16), where,

$$L_{MPM}(\mathbf{x},\mathbf{z}) = \sum_{i \in W} \mathbf{1}_{[\mathbf{x}(i) \neq \mathbf{z}(i)]}. \tag{4.19}$$

Note that for binary images, $L_{MPM} = L_{PE}$. Of course for each of (4.17), (4.18) and (4.19), the corresponding OBE's are known explicitly, and the underlying loss functions often only implicitly specified.

While each of the above estimators seems plausible, their corresponding loss functions seem to have many drawbacks. In the case of (4.17), it returns the maximum value of 1 even when there is disagreement at just a single pixel. Indeed both (4.17) and (4.19) don't examine differences in grey levels, rather whether or not pixel values agree or disagree. From this point of view their use as grey-scale image estimators is dubious. In fact Rue [44] and many others have stated that the MAP estimate tends to lead to oversmoothing, mislabelling of pixel values and deletion of fine details in image reconstruction. This indeed agrees with the author's experience. Each of (4.18) and (4.19) are based on pixel by pixel differences and following the discussion in Section 3.2.1, each of the above loss functions may not always work well in practice. In essence, they do not consider any spatial structure in the image, penalising errors on a pixel by pixel basis. It has been written in [44] that the $\mathbf{x}_{PM}$ estimate is too local, at the expense of global detail in the reconstructed image.

The challenge therefore is to explore 'better' loss functions, with the hope that they improve upon existing methods. In the next section a brief discussion of some recent methods is described.

4.5.2 Some improved loss functions

Delta metric

The delta metric, $\Delta_c^p(\cdot,\cdot)$ has been shown in [2] to be a good binary image metric. See also the discussion in Section 3.2.2. Frigessi and Rue [19] successfully implemented this as a loss function for use as an OBE for binary image restoration. It is important to note that when the cut-off value $c = 0$, that $\Delta_c^2 = L_{PE}$. As c increases, more spatial information becomes available, which in turn visibly improves the restoration.

Incorporating prior information into loss functions

Rue [44] also presented another binary loss function. It is based on pixel misclassification rates. Here all inference is based on the pixel misclassification rate, defined as:

$$\mathrm{e}(i) = \begin{cases} 0\,, & \text{if } \mathbf{x}(i) = \mathbf{z}(i), \\ 1\,, & \text{otherwise,} \end{cases}$$

so that $\mathrm{e}(i) = 0$, if pixel i is correctly classified, and 1 otherwise. Since $\mathrm{e}(i)$ does not account for any magnitude of difference in grey-level it is therefore unsuitable for images with many grey levels.

Let D be some subset of W. Denote M_D as the number of misclassification in D. Define polynomials p and q as:

$$p(D) = 1 - \prod_{i \in D}(1 - \mathbf{e}(i)) = \begin{cases} 0\,, & \text{if } M_D = 0 \\ 1\,, & \text{otherwise.} \end{cases}$$

$$q(D) = \prod_{i \in D} \mathbf{e}(i) = \begin{cases} 1\,, & \text{if } M_D = |D| \\ 0\,, & \text{otherwise,} \end{cases}$$

where $|D|$ is the number of pixels in D.

Thus $p(D) = 0$ if all pixels in D are classified correctly and 1 otherwise. So D might represent an area or feature which is important to restore correctly. Similarly $q(D) = 1$ if all pixels in D are misclassified, and 0 even if there is a single pixel misclassified. In this instance D could represent an area which we penalise if completely misclassified.

For $0 \leq k \leq |D|$ define $p_k(D)$ to be the set of all $\begin{pmatrix} |D| \\ k \end{pmatrix}$ subsets ω of D, which are of size k. For example, if $D = \{x_1, x_2, x_3\}$, then $p_2(D) = \{\{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}\}$.

Let s and θ denote translation and rotation operators, respectively, for the set D. Fix some pixel in D to be the origin and denote, $T_s R_\theta(D)$ or $R_\theta T_s(D)$ to be the set D first rotated by θ and then translated to a new origin s.

The idea now is to examine loss functions of the form,

$$L(\mathbf{x}, \mathbf{z}) = L(\mathbf{e}) = \sum_{\omega \subseteq W} t_\omega q(\omega),$$

and

$$L(\mathbf{x},\mathbf{z}) = L(\mathbf{e}) = \sum_{\omega \subseteq W} t'_\omega p(\omega).$$

for appropriate choices of weights t_ω and t'_ω corresponding to specified local sets $\omega \subset W$. In fact Rue [44] has shown that these two forms are equivalent.

The approach is as follows. First choose a set of basis regions $D_1, D_2, \ldots, D_n$, and for each basis region D_i, define weights t_{ij} for the polynomial $P_i(D_i)$:

$$P_i(D_i) = \sum_{j=1}^{|D_i|} t_{ij} \sum_{\omega \in p_j(D_i)} p(\omega). \tag{4.20}$$

The loss function is then defined as:

$$L(\mathbf{x},\mathbf{z}) = L(\mathbf{e}) = \sum_{i=1}^{n} \sum_{s,\theta : T_s R_\theta(D_i) \subseteq W} P_i(T_s R_\theta(D_i)).$$

The second sum is over all rotated and translated basis regions. It is suggested that the number of basis regions chosen is small, and that each are local. Consider the following (trivial) example, similar to one presented in [44], as motivation towards choices of basis regions $\{D_i\}_{i=1,\ldots,n}$ and weights $\{t_{ij}\}$. In Figure 4.2 is seen a binary image of size 15 × 15, where each box is of size 2 × 2 pixels.

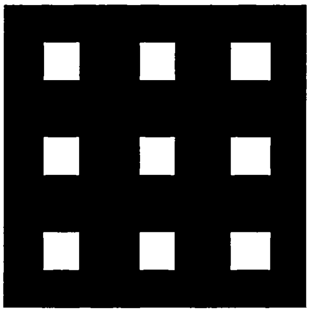


Figure 4.2. True image with regular geometric structure

Suppose this image is degraded by some noise model, and the aim then being to restore the noisy image as accurately as possible. If it is known a priori that the true image contains boxes of size 2×2, and it is required to completely restore each box, then an appropriate basis region might then be, $D_1 = \{2\times2\text{block of pixels}\}$, (number of basis regions equals 1). The choice of weights $t_{1,j}$ in the polynomial $P_1(D_1)$, might be $t_{1,j} = 0$, $j = 1, 2, 3$, and $t_{1,4} = 1$.

This approach to loss function modelling can be seen as both a strength and a weakness. If prior information is available, as to what the true image should look like (2×2 squares on a black background, in the example above), then following a Bayesian philosophy this information should be incorporated into the model. In this case the information is merged in the loss function, and so should be seen as a strength. However in the absence of any information of this nature, then it is unclear how to model the loss function appropriately. This might be seen as a weakness.

In all the examples presented in [44], using knowledge of the true image scene, the choice of loss function outperforms both the MAP and MPM (or equivalently PE estimate for binary images).

A loss function based on sample covariance

An improved grey-scale loss function model has been presented by Rue [45]. He suggested a covariance based loss function with the aim of removing the major weakness of L_{PE} which does not depend on the spatial structure of pixel by pixel differences.

The proposed loss function only depends on errors through pixel by pixel differences. It is convenient to introduce the notation

$$e_i = \mathbf{e}(i) = \mathbf{x}(i) - \mathbf{z}(i),$$

so that image $\mathbf{e} = \mathbf{x} - \mathbf{z}$. This situation contrasts with the previous loss function, which is based solely on a loss function where magnitude of difference in pixel values is unimportant.

Specifically this loss function adds a penalty term to L_{PE}, penalising strong sample covariance of the error e_i in a specified neighbourhood of each pixel. Suppose w_i is the set of 3×3 pixels centred at pixel i. The sample covariance of $\mathbf{e}$ at lag k in w_i gives a strong indication of local spatial structure in the error $\mathbf{e}$. This structure can then be penalised by adding it as a penalty to the pixel by pixel error e_i.

Specifically suppose k is a spatial lag in w_i, and denote by $|w_i|$ the number of pixels in w_i. Note that a spatial lag is simply a specified position for a pixel in some collection of pixels. Let $w_i + k$ equal the set of pixels in w_i translated by pixel k. The sample covariance of $\mathbf{e}$ at lag k in w_i may be defined as:

$$C_{w_i}(k) = \frac{1}{|w_i \cap (w_i + k)|} \sum_{j \in w_i \cap (w_i + k)} (e_j - m_{w_i})(e_{j+k} - m_{w_i}), \tag{4.21}$$

where,

$$m_{w_i} = \frac{1}{|w_i|} \sum_{j \in w_i} e_j \tag{4.22}$$

denotes the sample mean of $\mathbf{e}$ in w_i. Where w_i is the 3×3 set of pixels centred at pixel i. Rue suggests the set of lags Ψ as shown in Figure 4.3. The penalty for local spatial structure in the errors w_i is defined as the sum of $C^2_{w_i}(k)$ over all pixels k in the lag Ψ. This penalty term when combined with the pixel by pixel error terms gives a loss function as:

$$L_{RUE}(\mathbf{x}, \mathbf{z}) = \sum_{i \in W} (\mathbf{x}(i) - \mathbf{z}(i))^2 - \lambda \sum_{i \in W} \sum_{k \in \Psi} C^2_{w_i}(k). \tag{4.23}$$

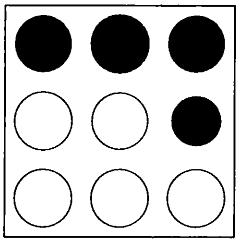


Figure 4.3. Set of lags and corresponding set w_i

The parameter λ controls the relative importance of spatial structure in the error.

Rue [45] presents a two step algorithm to compute OBE's based on this loss function. Much of the following work relies on this algorithm.

4.5.3 Estimating OBE's

We now address the problem as to how OBE's may be calculated for a given class of loss functions. At first glance it would appear that the approach to evaluating

$$\mathbf{x}^* = \arg\min_{\mathbf{z}} R(\mathbf{z}) = \arg\min_{\mathbf{z}} \mathbf{E}_{\mathbf{x}|\mathbf{y}} L(\mathbf{x}, \mathbf{z}),$$

would be to consider this as an ordinary minimisation problem evaluating the expectation for each image configuration $\mathbf{z}$. However to do this in practice would not be feasible. An alternative is to first compute $\hat{R}(\mathbf{z})$, an estimate of $R(\mathbf{z})$, via MCMC methods, and then to minimise $\hat{R}(\mathbf{z})$. In fact this is the route we take.

Consider loss functions of the form:

$$L(\mathbf{x}, \mathbf{z}) = \sum_{i \in W} (\mu(\mathbf{x}; i) - \mu(\mathbf{z}; i))^2, \tag{4.24}$$

where $\mu(\mathbf{x}; i)$ is some function defined locally for image $\mathbf{x}$ around pixel i. Using loss functions of this form allows the risk to be written as,

$$\begin{aligned} R(\mathbf{z}) &= \mathbf{E}_{\mathbf{x}|\mathbf{y}} L(\mathbf{x}, \mathbf{z}) \\ &= \mathbf{E}_{\mathbf{x}|\mathbf{y}} \sum_{i \in W} (\mu(\mathbf{x}; i) - \mu(\mathbf{z}; i))^2 \\ &= \sum_{i \in W} \mu(\mathbf{z}; i) \left[\mu(\mathbf{z}; i) - 2\mathbf{E}_{\mathbf{x}|\mathbf{y}} \mu(\mathbf{x}; i)\right] + \text{constant}. \qquad (4.25) \end{aligned}$$

It is clear that the constant term in (4.25) has no bearing on the minimisation of $R(\mathbf{z})$ and so may be ignored. Notice that such loss functions allows an explicit estimate of the risk, once the posterior expectation of $\mu(\mathbf{x}; i)$ has been estimated. Note also that $\mathbf{x}^*$ may be interpreted as the image having $\{\mu(\mathbf{x}^*; i) : i \in W\}$ closest to $\{\mathbf{E}_{\mathbf{x}|\mathbf{y}} \mu(\mathbf{x}; i) : i \in W\}$. The algorithm to estimate $\mathbf{x}^*$ may be split into two parts:

1. Estimate the posterior expected values of $\mu(\mathbf{x}; i)$ for each $i \in W$, whence an explicit expression for the risk $\hat{R}(\mathbf{z})$ is available.

2. Minimise $\hat{R}(\mathbf{z})$ over all image configurations $\mathbf{z}$.

The posterior expectations of $\mu(\mathbf{x}; i)$ may be calculated using MCMC methods, and then $\hat{R}(\mathbf{z})$ minimised using simulated annealing. The choice of function $\mu(\cdot; i)$ is of crucial importance, since it determines how spatial information around pixel i is to be obtained. In fact some of the loss functions (image metrics) we have met may be restated in the form (4.24). We summarise these below:

$$\begin{aligned} L_{PE} &: \mu(\mathbf{x}; i) = \mathbf{x}(i) \\ \Delta_c^2 &: \mu(\mathbf{x}; i) = \min(d(i, \mathbf{x}), c) \\ DI_c^2 &: \mu(\mathbf{x}; i) = \frac{1}{c+1} \sum_{i=1}^{c} (\mathbf{x} \oplus B_r)(i). \end{aligned}$$

Recall that the symmetrised double integral metric is written as:

$$\widetilde{DI} = \left(DI(f,g) + DI(\tilde{f},\tilde{g})\right)/2,$$

where $\tilde{f}(x) = l - 1 - f(x)$, with $l - 1$ being the largest grey level of f. Thus,

$$R(\mathbf{z}) = \left(\mathbf{E}_{\mathbf{x}|\mathbf{y}} DI(\mathbf{x},\mathbf{z}) + \mathbf{E}_{\mathbf{x}|\mathbf{y}} DI(\tilde{\mathbf{x}},\tilde{\mathbf{z}})\right)/2.$$

So calculation of the OBE for $\widetilde{DI}(f,g)$ first requires estimation of two expectations, $\mathbf{E}\mu(\mathbf{x};i)$ and $\mathbf{E}\mu(\tilde{\mathbf{x}};i)$, where $\mu(\mathbf{x};i) = \frac{1}{c+1}\sum_{i=1}^{c}(\mathbf{x} \oplus B_r)(i)$. An estimate of $R(\mathbf{z})$ is then available, which can be minimised via simulated annealing.

4.6 An alternative approach

We now present an alternative idea, somewhat different to the approach described previously, while still retaining much the same flavour. We call this approach the *level-by-level* method. The idea is based on the simple fact that any image $\mathbf{x}$ with l grey levels may easily be reconstructed from its $l-1$ constituent binary thresholded images. That is,

$$\mathbf{x} = \sum_{t=1}^{l-1} \mathbf{x}^{(t)},$$

where $\mathbf{x}^{(t)}$ is the image $\mathbf{x}$ thresholded at level t. (Note that the notation here for a thresholded image is different from that in Chapter 2.) Here the summation is pixelwise, so that $\mathbf{x}(i) = \sum_{t=1}^{l-1}\mathbf{x}^{(t)}(i)$. Note that each value of $\mathbf{x}^{(t)}$ is either 0 or 1 since it is a binary. Now instead of estimating the true image globally, we try to estimate $\mathbf{x}^{(t)}$, the estimate of the true image thresholded at level t using a ‘good’ binary loss function, for example, $\Delta_c^2(\cdot,\cdot)$, as explained in Section 3.2.1. Then each reconstructed binary image $\hat{\mathbf{x}}^{(t)}, t = 1, \ldots, l-1$ can be used to estimate the true image as,

$$\hat{\mathbf{x}} = \sum_{t=1}^{l-1} \hat{\mathbf{x}}^{(t)}.$$

This situation has some parallels to stack filtering. Here an input noisy image is first decomposed into its constituent binary thresholded images, and each binary image is passed through a binary filter.

What could be the potential advantages to using this method? Certainly binary images are easier to analyse than grey-scale images. Further Δ_c^2 has been shown to work well in a Bayesian context [19]. Also reconstruction of the binary images at each grey level is independent and so may be carried out in parallel. However a disadvantage is that as the number of grey levels increases, the computational complexity increases further. This may suggest that this approach is more suitable for reconstruction of images with few grey levels.

This scheme works as follows: sample N grey-scale images $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ from the stationary distribution $\pi(\mathbf{x}|\mathbf{y})$. Threshold each image at grey level t, to form $(l-1) \times N$ binary images

$$\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_N^{(t)},\ t = 1, 2, \ldots, l-1.$$

This sample $\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_N^{(t)}\}$, for each $t = 1, 2, \ldots l-1$ may then be used to estimate $\mathbf{E}\mu(\mathbf{x}^{(t)}; i)$ as:

$$\widehat{\mathbf{E}\mu(\mathbf{x}^{(t)}; i)} = \frac{1}{N} \sum_{m=1}^{N} \mu(\mathbf{x}_m^{(t)}; i). \tag{4.26}$$

He we use $\mu(\cdot;i) = \min(d(i,\cdot),c)$, so that the estimate of the true image thresholded at level i is calculated using the Δ_c^2 loss function. The estimated expectation (4.26) is then used to calculate

$$\begin{aligned}\hat{\mathbf{x}}^{(t)} &= \arg\min_{\mathbf{z}} \mathbf{E}\Delta_c^2(\mathbf{x}^{(t)},\mathbf{z}) \\ &\approx \arg\min_{\mathbf{z}} \sum_{i\in W} \mu(\mathbf{z};i)\left[\mu(\mathbf{z};i) - 2\mathbf{E}\widehat{\mu(\mathbf{x}^{(t)};i)}\right] + \text{constant.}\end{aligned}$$

Pixelwise addition of each image $\hat{\mathbf{x}}^{(t)}$, results in,

$$\hat{\mathbf{x}} = \sum_{m=1}^{l-1} \hat{\mathbf{x}}^{(t)}.$$

4.7 Simulation experiments

This section display results for both the symmetrised double integral metric and level-by-level method on artificial data. Further, MAP estimates and PE estimates are presented to compare with the newly introduced alternatives. In general we have found that the double integral metric overestimates restored pixels values. However the symmetrised variant generally gives more appealing results.

The prior model introduced by Geman and McClure [21] is used throughout:

$$\pi(\mathbf{x}) \propto \exp\left(-\beta\sum_{\langle i,j\rangle}\Upsilon(\mathbf{x}(i)-\mathbf{x}(j)) - \beta\frac{1}{\sqrt{2}}\sum_{[i,j]}\Upsilon(\mathbf{x}(i)-\mathbf{x}(j))\right).$$

It uses contributions from each pixel and its eight nearest neighbours, The sum $\langle i,j\rangle$ is over all first-order neighbouring pairs and $[i,j]$ is over all second-order neighbouring pairs (see Figure 4.3. The potential function Υ takes the form $\Upsilon(d) = d^2/(d^2+\kappa^2)$, where κ is a scale parameter.

The simulated annealing step uses the cooling schedule $T(t)/T(0) = 0.99^t$, with 10 and 0.005 as the initial and final temperatures. This represents approximately 750 iterations.

The PE estimate was computed by averaging, pixelwise, a sample of 1, 000images from the posterior. The MAP estimate was computed via simulated annealing using the cooling schedule described above.

4.7.1 Experiment 1

The test image in Figure 4.4(a) consists of an image of size 15 × 15 pixels, where the pixel values increase in steps of 3 , there are 9 pixels values in total, $\{0, 1, \ldots, 8\}$. The noisy data 4.4(b) is the true image with i.i.d. Gaussian noise of mean 0 and variance 1.5^2 added to each pixel. Also displayed are the PE and MAP estimates and restored images corresponding to both the symmetrised double integral metric and the level-by-level method.

In the prior model we use $\beta = 0.75$ and $\kappa = 0.5$. We first run a Metropolis algorithm converging to the stationary (posterior) distribution after 5, 000 iterations. At iteration t a new value p for pixel i is chosen uniformly from the integers $[\mathbf{x}_t(i) - 1, \mathbf{x}_t(i) + 1]$. If $p \notin [0, 1, \ldots, 8]$, then $p = \mathbf{x}_t(i)$.

To compute $\widetilde{DI}$ we first compute the posterior expectations of $\mu(\mathbf{x}; i)$ and $\mu(\tilde{\mathbf{x}}; i)$, where $\mu(\mathbf{x}; i) = \frac{1}{c+1} \sum_{i=1}^{c} (\mathbf{x} \oplus B_r)(i)$ from 1, 000 sample from the posterior distribution. Each full sweep visits all pixels in a uniform order.

The level-by-level method uses a sample of 1, 000 images from the posterior distribution, each of which are thresholded to give 9 sample of 1, 000 binary images, one for each of the 8 grey levels. Each sample is then used to compute the expectation (4.26).

In this instance the MAP estimate seems to have over-smoothed the true image, although it does display distinct homogeneous regions. Both the PE estimate and the estimate corresponding to the level-by-level method look quite similar. However the level-by-level method has slightly out-performed it. The estimate corresponding to $\widetilde{DI}$ has tended more than any of the others to preserve the step nature of the true image, although it has tended to overestimate most pixel values.

4.7.2 Experiment 2

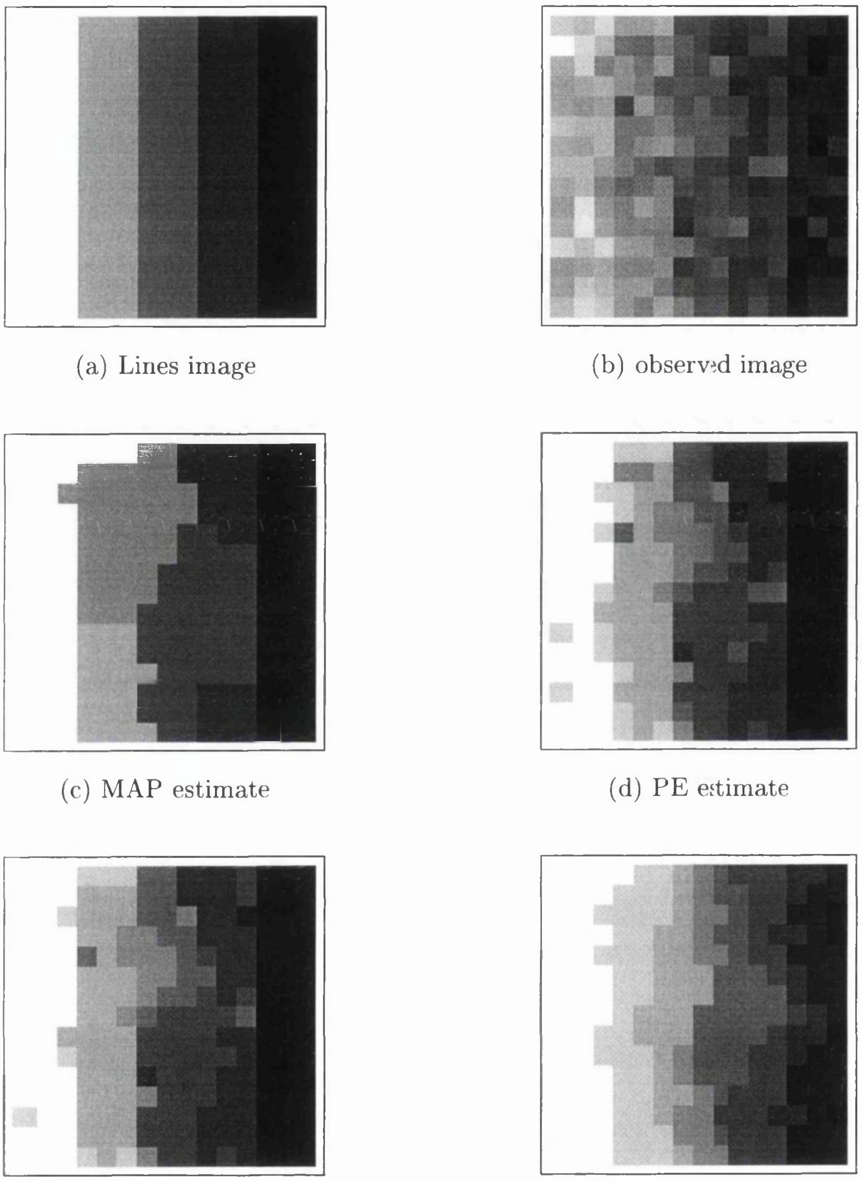
An disadvantage of Bayesian image restoration is that it is computer intensive. This makes it difficult to feasibly examine real-life images. In this example we have taken an image 4.5(a) and cropped an area of the image around the right eye 4.5(b). This now serves as a real-life example. This image has a grey-scale range of $\{0, 1, \ldots, 255\}$. Gaussian i.i.d. noise of variance 20^2 grey levels was added to each pixel.

Again MAP and PE estimates are presented, together with the estimate corresponding to $\widetilde{DI}$. In this example we have chosen not to include the level-by-level method, since the grey-scale range of the true image, is quite large.

In the prior model we use $\beta = 1.1$ and $\kappa = 0.5$. Again a Metropolis algorithm was implemented converging to the stationary (posterior) distribution after $5,000$ iterations. At iteration t a new value p for pixel i is chosen uniformly from the integers $[\mathbf{x}_t(i) - 20, \mathbf{x}_t(i) + 20]$. If $p \not\in [0, 1, \ldots, 255]$, then $p = \mathbf{x}_t(i)$.

To compute $\widetilde{DI}$ we first compute the posterior expectations of $\mu(\mathbf{x}; i)$ and $\mu(\tilde{\mathbf{x}}; i)$, where $\mu(\mathbf{x}; i) = \frac{1}{c+1} \sum_{i=1}^{c} \mathbf{x} \oplus B_r(i)$ from $1,000$ sample from the posterior distribution. Each full sweep visits all pixels in a uniform order.

Looking at Figure 4.5, the following comments can be made. Again the MAP estimate has oversmoothed the data. Both the PE estimate and the estimate corresponding to the $\widetilde{DI}$ loss function provide similar estimates. However again $\widetilde{DI}$ has overestimated pixel values.



(a) Lines image

(b) observed image

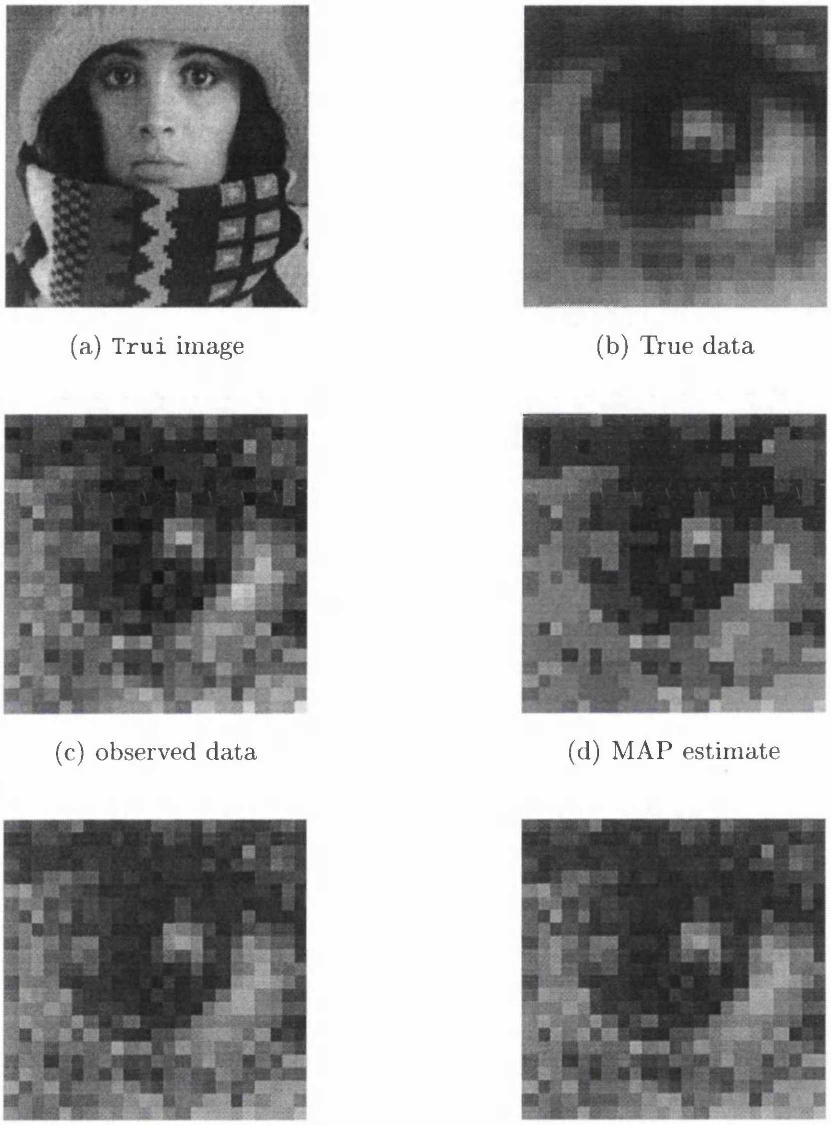
(c) MAP estimate

(d) PE estimate

(e) level-by-level

(f) $\widetilde{LI}$

Figure 4.4. Lines image and various estimates.



(a) Trui image

(b) True data

(c) observed data

(d) MAP estimate

(e) PE estimate

(f) $\widetilde{DI}$

Figure 4.5. Trui image and various estimates.

Chapter 5

Set-valued Regression

5.1 Introduction

Classical regression involves exploring relationships between two types of variables, response and explanatory variables. This chapter aims to extend this theory to the situation where the response variable is set-valued. Of particular interest is when the set is a binary image in $\mathbb{R}^2$. Analogous to classical regression where the response variable is observed with a certain amount of noise, each observed set (or binary image) is assumed to be an inexact measurement of an underlying set (or binary image). Further in this instance, it is useful to interpret the explanatory variable as time.

The ideal now would be to extend concepts and methods from the classical setting to the present situation. This however is not straightforward, for a number of reasons. The family of compact sets in the Euclidean space is not linear, which makes concepts like expectations difficult to formulate. Further, subtraction of sets is unclear, so the problem of finding residuals is difficult.

The problem examined in this chapter may be succinctly stated as follows. Suppose given some observed time sequence of noisy (or inexact measurements

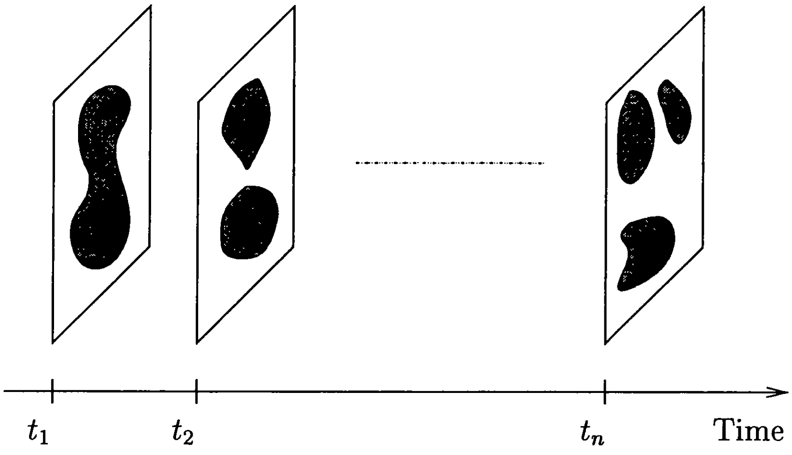


Figure 5.1. Time dependent noisy image sequence

of) sets, find the underlying time evolution of the noise-free sets. We term this problem *set-valued regression.*

It is clear that this problem has applications in many real-life situations. Consider the following as a motivation. Suppose that it is wished to send an image sequence, maybe a movie, via a noisy channel to a receiver. Suppose further that it is costly to send each image in the sequence, so that the receiver obtains an incomplete noisy version of the original sequence. Set-valued regression would help in this situation to recover or estimate the original noise-free complete image sequence. Figure 5.1 illustrates this problem.

This chapter begins with a simple examination of what happens when the response variable is a single closed interval or segment in $\mathbb{R}$. This is the simplest example of a non-trivial convex set in $\mathbb{R}$. Approaches are suggested in Section 5.4 to deal with the case where the response variables are general sets in $\mathbb{R}^2$. We illustrate this approach on some binary images. Finally we outline with examples, an application of this scheme, to image warping which provides a smooth transition between two given images.

5.2 A simple case of set-valued linear regression

Suppose that given an ordered increasing sequence of time points $t_1, t_2, \ldots, t_n$, which correspond to sets Y_{t_i}, $i = 1, \ldots n$. The problem is now to find a smooth evolution of the sets over all time point.

We begin by considering the simple case where Y_{t_i} are convex sets in $\mathbb{R}^1$, that is, closed intervals on the real line. Consider the following regression model:

$$Y_{t_i} = \Gamma \oplus \Xi_{t_i} + \zeta_{t_i}. \tag{5.1}$$

The problem posed by (5.1) is to fit a constant set Γ to the observed sets (segments) Y_{t_i}. Here Γ and Ξ_{t_i} are set-valued where it is assumed that $0 \in \Xi_{t_i}$. Further, ζ_{t_i} is a random variable. The term Ξ_{t_i} represents a 'shape' disturbance of Γ, while ζ_{t_i} denotes a location error. Note that the operation $\Gamma \oplus \Xi_{t_i}$ enlarges Γ. In particular this implies that Γ must be smaller than the length of each of the observed intervals Y_{t_i}, introducing a constraint to the problem.

We introduce the notation $Y_{t_i} = [y^l_{t_i}, y^u_{t_i}]$, $\Gamma = [\gamma^l, \gamma^u]$, $\Xi_{t_i} = [\xi^l_{t_i}, \xi^u_{t_i}]$. Denote $\bar{y}_{t_i} = \frac{y^l_{t_i} + y^u_{t_i}}{2}$, $\tilde{y}_{t_i} = \frac{y^u_{t_i} - y^l_{t_i}}{2}$ and $\bar{\gamma} = \frac{\gamma^l + \gamma^u}{2}$, $\tilde{\gamma} = \frac{\gamma^u - \gamma^l}{2}$. The constraint described above may now be conveniently written as:

$$\tilde{\gamma} \leq \min\{\tilde{y}_{t_i}\}. \tag{5.2}$$

5.2.1 Maximum likelihood estimators

The following two examples illustrate maximum likelihood estimators (MLE) for several possible noise models.

Example 5.1. Consider the following noise model:

$$\Xi_{t_i} = [-\xi_{t_i}, \xi_{t_i}], \quad \text{where } \xi_{t_i} \sim \text{Expo}(\lambda).$$

and

$$\zeta_{t_i} \sim N(0, \sigma^2).$$

The random variables ξ_{t_i} and ζ_{t_i} are assumed to be independent, while the parameters λ and σ are assumed known. The regression model (5.1) may now be expanded as:

$$\begin{aligned} y^u_{t_i} &= \gamma^u + \xi_{t_i} + \zeta_{t_i}, && (5.3) \\ y^l_{t_i} &= \gamma^l - \xi_{t_i} + \zeta_{t_i}. && (5.4) \end{aligned}$$

Now adding (5.3) and (5.4), and subtracting (5.4) from (5.3), gives the two equations:

$$\begin{aligned} \bar{y}_{t_i} &= \bar{\gamma} + \zeta_{t_i}, && (5.5) \\ \tilde{y}_{t_i} &= \tilde{\gamma} + \xi_{t_i}. && (5.6) \end{aligned}$$

By independence of ξ_{t_i} and ζ_{t_i}, we can write the likelihood of obtaining $Y_{t_1}, Y_{t_2}, \ldots, Y_{t_n}$ as,

$$\begin{aligned} L(\bar{\gamma}, \tilde{\gamma}) &= \prod_{i=1}^{n} p_{\zeta_{t_i}}(\bar{y}_{t_i} - \bar{\gamma}) p_{\xi_{t_i}}(\tilde{y}_{t_i} - \tilde{\gamma}) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}\lambda^n} \exp\left\{ -\sum_{i=1}^{n} (\bar{y}_{t_i} - \bar{\gamma})^2 / 2\sigma^2 - \sum_{i=1}^{n} (\tilde{y}_{t_i} - \tilde{\gamma}) \right\}. \end{aligned}$$

Thus it is seen that the maximum likelihood estimator of $\bar{\gamma}$ may be written as:

$$\hat{\bar{\gamma}} = \frac{1}{n} \sum_{i=1}^{n} \bar{y}_{t_i}.$$

The maximum likelihood estimator for $\tilde{\gamma}$ is found by maximising the likelihood subject to the constraint (5.2). Since $\tilde{\gamma} \leq \min\{\bar{y}_{t_i}\}$ this estimator is seen to be:

$$\hat{\tilde{\gamma}} = \min\{\tilde{y}_{t_i}\}.$$

Of course both $\hat{\gamma}$ and $\hat{\tilde{\gamma}}$ may be combined to form an estimate of Γ.

Example 5.2. Consider the following noise model:

$$\Xi_{t_i} + \zeta_{t_i} = [\eta^l_{t_i}, \eta^u_{t_i}],$$

where,

$$\begin{aligned}\eta^l_{t_i} &= \min(\mu^1_{t_i}, \mu^2_{t_i}),\\ \eta^u_{t_i} &= \max(\mu^1_{t_i}, \mu^2_{t_i}).\end{aligned}$$

Here $\mu^1_{t_i}, \mu^2_{t_i}$ are independent identically distributed as $N(0, \sigma^2)$.

In this example the regression model (5.1) reduces to the two equations:

$$\begin{aligned}y^l_{t_i} &= \gamma^l + \eta^l_{t_i},\\ y^u_{t_i} &= \gamma^u + \eta^u_{t_i}.\end{aligned}$$

The probability density function of the end points for Y_{t_i} may be written as:

$$\begin{aligned}p_{\eta^l_{t_i}, \eta^u_{t_i}}(y^l_{t_i} - \gamma^l, y^u_{t_i} - \gamma^u) &= 2p_{\mu^1_{t_i}, \mu^2_{t_i}}(y^l_{t_i} - \gamma^l, y^u_{t_i} - \gamma^u)\\ &= 2p_{\mu^1_{t_i}}(y^l_{t_i} - \gamma^l)p_{\mu^2_{t_i}}(y^u_{t_i} - \gamma^u),\end{aligned}$$

by independence of $\mu^1_{t_i}$, $\mu^2_{t_i}$. Thus the likelihood of obtaining the intervals $Y_{t_1}, Y_{t_2}, \ldots, Y_{t_n}$, may be written as:

$$\begin{aligned} L(\gamma^l, \gamma^u) &= \prod_{i=1}^{n} 2p_{\mu^1_{t_i}}(y^l_{t_i} - \gamma^l) p_{\mu^2_{t_i}}(y^u_{t_i} - \gamma^u) \\ &= 2^n \prod_{i=1}^{n} \frac{1}{2\pi\sigma^2} \exp\left\{ \frac{-(y^l_{t_i} - \gamma^l)^2}{2\sigma^2} \right\} \exp\left\{ \frac{-(y^u_{t_i} - \gamma^u)^2}{2\sigma^2} \right\}. \end{aligned}$$

Re-parameterising the problem in terms of $\bar{\gamma}$, $\tilde{\gamma}$:

$$L(\bar{\gamma}, \tilde{\gamma}) = (\text{const}) \exp\left\{ -\sum_{i=1}^{n} \left((\bar{y}_{t_i} - \bar{\gamma})^2 - (\tilde{y}_{t_i} - \tilde{\gamma})^2 \right) \right\}.$$

Thus the maximum likelihood estimator is then found by minimising the equation,

$$\sum_{i=1}^{n} \left((\bar{y}_{t_i} - \bar{\gamma})^2 + (\tilde{y}_{t_i} - \tilde{\gamma})^2 \right),$$

with respect to $\bar{\gamma}$, $\tilde{\gamma}$, subject to the constraint (5.2).

So it is seen, as in the previous example, the maximum likelihood estimators of both $\bar{\gamma}$ and $\tilde{\gamma}$ may be written respectively as:

$$\hat{\bar{\gamma}} = \frac{1}{n} \sum_{i=1}^{n} \bar{y}_{t_i} \quad \text{and} \quad \hat{\tilde{\gamma}} = \min\{\tilde{y}_{t_i}\}.$$

5.2.2 Least squares type estimators

Now we turn our attention to calculating estimators of Γ of the form:

$$\hat{\Gamma} = \arg\min_{\Gamma} \sum_{i=1}^{n} L(Y_{t_i}, \Gamma), \tag{5.7}$$

where $L(Y_{t_i}, Z)$ is some measure of distance between the closed intervals Y_{t_i} and Γ. So $\hat{\Gamma}$ has analogies to least squares estimators for conventional linear regression. Note that this estimation problem is again constrained by the condition (5.2).

In the following examples, we evaluate $\hat{\Gamma}$ for various choices of $L(\cdot,\cdot)$.

Example 5.3. Let $L(\cdot,\cdot)$ be the Hausdorff metric (3.4),(3.5). Thus,

$$\hat{\Gamma} = \arg\min_{\Gamma} \sum_{i=1}^{n} \max\left\{ |y_{t_i}^u - \gamma^u|, |y_{t_i}^l - \gamma^l| \right\}.$$

This may be reformulated in terms of $\bar{y}_{t_i}$, $\tilde{y}_{t_i}$, $\bar{\gamma}$, $\tilde{\gamma}$ as,

$$\begin{aligned}\hat{\Gamma} &= \arg\min_{\bar{\gamma},\tilde{\gamma}} \sum_{i=1}^{n} \max\left\{ |\bar{y}_{t_i} - \bar{\gamma} - (\tilde{y}_{t_i} - \tilde{\gamma})|, |\bar{y}_{t_i} - \bar{\gamma} + (\tilde{y}_{t_i} - \tilde{\gamma})| \right\} \\ &= \arg\min_{\bar{\gamma},\tilde{\gamma}} \sum_{i=1}^{n} \left\{ |\bar{y}_{t_i} - \bar{\gamma}| + \tilde{y}_{t_i} - \tilde{\gamma} \right\}, \qquad (5.8)\end{aligned}$$

since the constraint (5.2) implies that $\tilde{y}_{t_i} - \tilde{\gamma} > 0$ for all $i = 1, 2, \ldots, n$. Now,

$$\tilde{y}_{t_i} = \tilde{\gamma} + \xi_{t_i}^u - \xi_{t_i}^l,$$

where $\xi_{t_i}^u - \xi_{t_i}^l \geq 0$, which implies that $\hat{\tilde{\gamma}}$, the estimator of $\tilde{\gamma}$ may be written as

$$\hat{\tilde{\gamma}} = \min\{\tilde{y}_{t_i}\}.$$

Finally $\hat{\bar{\gamma}}$ satisfies the equation,

$$\hat{\bar{\gamma}} = \arg\min_{\bar{\gamma}} \sum_{i=1}^{n} |\bar{y}_{t_i} - \bar{\gamma}|.$$

So it seen that,

$$\hat{\bar{\gamma}} = \text{median}\{\bar{y}_{t_i}\}.$$

Example 5.4. Suppose that the choice of $L(\cdot,\cdot)$ in (5.7) is the sum of squared difference between end-points. That is,

$$L(Y_{t_i},\Gamma) = (y^u_{t_i} - \gamma^u)^2 + (y^l_{t_i} - \gamma^l)^2.$$

In this case,

$$\hat{\Gamma} = \arg\min_{\gamma^l,\gamma^u} \sum_{i=1}^{n} ((y^u_{t_i} - \gamma^u)^2 + (y^l_{t_i} - \gamma^l)^2) \tag{5.9}$$

As for each of the previous examples we re-parameterise the problem in terms of $\bar{y}_{t_i}$, $\tilde{y}_{t_i}$, $\bar{\gamma}$ and $\tilde{\gamma}$. After some simplification (5.9) reduces to,

$$\hat{\Gamma} = \arg\min_{\bar{\gamma},\tilde{\gamma}} 2\sum_{i=1}^{n} ((\bar{y}_{t_i} - \bar{\gamma})^2 + (\tilde{y}_{t_i} - \tilde{\gamma})^2). \tag{5.10}$$

Now it is seen using the constraint (5.2), that the estimators of $\bar{\gamma}$ and $\tilde{\gamma}$ are (respectively):

$$\hat{\bar{\gamma}} = \frac{1}{n}\sum_{i=1}^{n} \bar{y}_{t_i}, \quad \text{and} \quad \hat{\tilde{\gamma}} = \min\{\tilde{y}_{t_i}\}.$$

So it is seen that this estimator of Γ coincides with the maximum likelihood estimators in Examples 5.1 and 5.2. This is similar to the situation in conventional linear regression, where the maximum likelihood estimators corresponding to the Gaussian noise model coincides with the usual least squares estimators.

Example 5.5. Suppose now that the $L(\cdot,\cdot)$ in (5.7) is the Delta metric, introduced in Section 3.2.1. The aim now is to estimate the interval,

$$\hat{\Gamma} = \arg\min_{\Gamma} \sum_{i=1}^{n} \Delta_c^2(Y_{t_i}, \Gamma).$$

Unlike each of the previous examples, it is difficult to find a closed form solution to this problem. However it will be seen in he subsequent sections that this estimation problem may be tackled using a simulated annealing approach.

5.3 Non-parametric set-valued regression

This section extends the work of the previous section to the case where the response variable is now a compact set in $\mathbb{R}^2$, specifically a binary image. It will be observed that the subsequent discussion may be easily extended to images or sets in higher dimensions.

In the previous Section the intention was to fit a constant set to the data. Of course in practice this is highly unrealistic, being a crude approximation to any underlying trend in the data. All is not lost however. Non-parametric regression is often used in conventional statistics by fitting constants to the data 'locally'. In this way local constant fits are, in a sense, 'pieced' together giving a regression function which explains much better the underlying trend in the data. The aim now is to extend such non-parametric regression ideas to the case where the response variable is a closed set or binary image in $\mathbb{R}^2$.

5.3.1 Background information

Consider first the situation where we deal with points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, following some model $y_i = f(x_i) + \varepsilon_i$, where (usually) $\mathbf{E}(\varepsilon_i | x_i) = 0$. Kernel smoothing offers a non-parametric approach towards recovering or estimating the underlying function $f(x)$. The reader is referred to [7, 50] for an introduction to non-parametric regression. The idea is simple. Choose some kernel function $k(x;h)$, with bandwidth h, and estimate $f(x)$ as:

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} k(x_i - x; h) y_i}{\sum_{i=1}^{n} k(x_i - x; h)}. \tag{5.11}$$

So $\hat{f}(x)$ is just a weighted average of the data y_i. The kernel function is usually a smooth symmetric function, peaking at 0 and monotonically decreasing as $|x_i - x|$ increases in size. This ensures that most weight is given to points x_i lying close to x. The bandwidth h controls the width of the kernel function, and hence the degree to which neighbouring points influence the estimate. As $h \to \infty$, $\hat{f}(x)$ tends to $n^{-1}\sum_{i=1}^{n} y_i$, the average of all observations, introducing a large bias. On the other hand small values of h give estimates which reproduce the data, that is, $\hat{f}(x_i) = y_i$. In fact (5.11) may be equivalently written in the form:

$$\hat{f}(x) = \arg\min_{\theta_x} \sum_{i=1}^{n} k(x_i - x; h)(y_i - \theta_x)^2, \tag{5.12}$$

so that θ_x can be viewed as a weighted least squares fit of a constant to an unknown function f. Since the weights determined by the kernel are small or vanishing outside a neighbourhood of x, the constant fit is local.

This idea may be extended to fit local polynomials $\sum_{j=1}^{t} \theta_x^j x^j$ to an unknown function f. The value of the function at value x may be estimated as:

$$\hat{f}(x) = \arg\min_{\{\theta_x^j\}} \sum_{i=1}^{n} k(x_i - x; h) \left(y_i - \sum_{j=1}^{t} \theta_x^j \right)^2 .$$

Clearly this increases the computational complexity of the problem.

The reader is referred to a paper by Fan and Gijbels [15], which outlines some good theoretical properties which the local linear regression estimator possesses.

We now concentrate on the situation where the response variables are set-valued.

5.3.2 Extension to set-valued regression

Suppose now that instead of points we are given some time dependent sequence of binary images $\{X_{t_1}, X_{t_2}, \dots, X_{t_n}\}$. How can we obtain a smooth evolution of the given images? The formulations in (5.11) and (5.12) suggest two approaches. In particular (5.11) suggests an approach whereby we estimate X_t, at some time point t, by a weighted 'average' of the binary images $\{X_{t_1}, X_{t_2}, \dots, X_{t_n}\}$, for weights $k(t_i - t; h)$. Averaging or more generally finding expectations of random closed sets however, is not straightforward. We have already seen examples of expectations of random sets in Section 2.3.1, including the Vorob'ev expectation.

The *Aumann expectation* [49] is based on the representation of a set via its support function,

$$h(X, u) = \sup\{\langle x, u\rangle : x \in X\}, \; u \in \mathbf{S}^{d-1},$$

where $\langle x, u\rangle$ is the scalar product and $\mathbf{S}^{d-1}$ is the unit sphere in $\mathbb{R}^d$. The scalar product of $x = (x_1, x_2, \dots, x_d)$ and $u = (u_1, u_2, \dots, u_d)$ is defined as

$$\langle x, u\rangle = x_1u_1 + x_2u_2 + \dots x_du_d.$$

If X is non-convex, then the support function of X corresponds to the convex hull of X. The Aumann expectation of a random set X is defined as the convex set having support function $h(\mathbf{E}X, u) = \mathbf{E}h(X, u)$. So the Aumann expectation is determined by the expected support function of X. This is particularly useful to find expectations of convex sets, since the expectation itself will always produce convex sets.

The *Frèchet mean* is a general concept of an expectation in a metric space $\mathbb{R}^d$. Let X be a random set in $\mathbb{R}^d$ and d a metric defined on $\mathcal{K}$, the space of compact sets in $\mathbb{R}^d$. The set $K_0 \in \mathcal{K}$, which minimises

$$\mathbf{E}d(X, K)^2 \text{ for } k \in \mathcal{K} \tag{5.13}$$

is defined to be the *Frèchet mean* of X. In most practical situations however the minimisation problem (5.13) is difficult to compute.

The *distance average* [3], on which the distance threshold (Section 2.4) is based is another example. Recall that it is based on the representation of a set via its distance function. Here the first step is to form the expected distance function, which in the present case may be written as:

$$\bar{d}_t(x) = \sum_{i=1}^{n} k(t_i - t; h) d(\cdot, X_{t_i}). \tag{5.14}$$

This however is not in general a distance function. The distance average is then chosen as the binary image $\hat{X}_t$ whose distance transform is closest to $\mathbf{E}d(\cdot, X_t)$. In [3] it is suggested that this set or binary image may be found by thresholding $\bar{d}_t(x)$, forming a family of sets,

$$X(\varepsilon) = \{x : \bar{d}_t(x) \geq \varepsilon\}, \tag{5.15}$$

and then defining,

$$\hat{X}_t = X(\hat{\varepsilon}) = \begin{cases} \arg\min_\varepsilon \|d(\cdot, X(\varepsilon)) - d(\cdot, \bar{d}_t(\cdot))\|, & L_\infty \text{ norm} \\ \arg\min_\varepsilon \left\{\sum_W |d(\cdot, X(\varepsilon)) - d(\cdot, \bar{d}_t(\cdot))|^2\right\}^{1/2}, & L_2 \text{ norm.} \end{cases} \tag{5.16}$$

Here $d(\cdot, X)$ could be any one of many distance functions, including

$$d^*(\cdot, X) = \min\{c, \rho(\cdot, X)\},$$

the truncated distance function used to define $\Delta_c^p(\cdot,\cdot)$. A clear advantage of the distance average is its ease of computation.

5.4 Loss functions and non-parametric smoothing

An alternative approach to exploring averages of binary images might be to generalise (5.12), suggesting estimators of the form:

$$\hat{X}_t = \arg\min_{B_t} \sum_{i=1}^{n} k(t_i - t; h) L(X_{t_i}, B_t), \tag{5.17}$$

where $L(\cdot,\cdot)$ is some loss function between sets (or binary images). Note that $L(X_{t_i}, B_t)$ generalises $(y_i - \theta_x)^2$ in (5.12), a non-negative distance between observations and parameter of interest.

We concentrate on such an approach, illustrating how such estimators may be computed. Note that this is a non-parametric approach, and so in contrast to the problem explored in Section 5.2 presupposes no parametric form of the solution, which in turn doesn't induce any constraints on the solution.

Henceforth we apply (5.17) using the delta metric Δ_c^p, so that,

$$\hat{X}_t = \arg\min_{B_t} \sum_{i=1}^{n} k(t_i - t; h) \left(\Delta_c^p(X_{t_i}, B_t)\right)^2. \tag{5.18}$$

For the rest of the discussion we will fix the value of p in Δ_c^p to $p = 2$. We introduce the notation,

$$R(B_t) = \sum_{i=1}^{n} w_i(t) \left(\Delta_c^2(X_{t_i}, B_t)\right)^2, \tag{5.19}$$

to denote the risk of estimating X_t by B_t. The weights $w_i(t) = k(t_i - t; h)$ are determined from some kernel density and it is assumed that for each time point t, the weights $w_i(t)$ are normalised so that $\sum_{i=1}^{n} w_i(t) = 1$. The smooth image at time t will then be that image $\hat{X}_t$ minimising the risk $R(B_t)$, that is, satisfying:

$$\hat{X}_t = \arg\min_{B_t} R(B_t).$$

At first glance (5.18) would appear to be a nasty optimisation problem. However it will be shown that similar to the problem posed in the previous chapter, of estimating the OBE for a given loss function, that the present problem may be tackled in a like manner. Note that (5.19) may be expanded as:

$$\begin{aligned} R(B_t) &= \sum_{i=1}^{n} w_i(t) \sum_{x \in W} (d^*(x, X_{t_i}) - d^*(x, B_t))^2 \\ &= \sum_{i=1}^{n} w_i(t) \sum_{x \in W} d^*(x, B_t) \left[d^*(x, B_t) - 2d^*(x, X_{t_i})\right] + \text{const} \\ &= \sum_{x \in W} d^*(x, B_t) \left[d^*(x, B_t) - 2\sum_{i=1}^{n} w_i(t) d^*(x, X_{t_i})\right] + \text{const}. \end{aligned} \tag{5.20}$$

It is clear that the constant term in (5.20) has no bearing on the minimisation of $R(B_t)$ and so may be ignored. In fact this minimisation problem may be equivalently written as:

$$\begin{aligned}\hat{X}_t &= \arg\min_{B_t} \sum_W d^*(x,B_t)\left[d^*(x,B_t) - 2\sum_{i=1}^n w_i(t)d^*(x,X_{t_i})\right] \\ &= \arg\min_{B_t} \sum_W \left[d^*(x,B_t) - \sum_{i=1}^n w_i(t)d^*(x,X_{t_i})\right]^2 . \qquad (5.21)\end{aligned}$$

So the estimator $\hat{X}_t$ has the nice property that it has truncated distance function $d^*(\cdot,\hat{X}_t)$ closest to $\sum_1^n w_i(t)d^*(\cdot,X_{t_i})$.

The algorithm to compute $\hat{X}_t$ has many similarities to that used in Section 4.5.3, to calculate the optimal Bayes estimator for the image restoration problem. Here the algorithm may be split into distinct parts:

1. Calculate $\sum_1^n w_i(t)d^*(\cdot,X_{t_i})$, where the normalised weights $w_i(t)$ correspond to some kernel density.

2. Minimise $R(B_t)$ (5.20) over all sets $B_t \in \mathbb{R}^d$.

As with the Bayesian restoration problem, minimisation is carried out via simulated annealing. Further, $\sum_1^n w_i(t)d^*(\cdot,X_{t_i})$ may be thought of, in a sense, as an expected distance function of X_t. In this way it is seen to be an analogue of the algorithm to estimate OBE's for certain loss functions (Section 4.5.3).

In fact this scheme also follows closely that of the distance average. Step 1 above is identical to the first step in the computation of the distance average (5.14). Further, Step 2 guarantees convergence to the global minimiser, if the cooling schedule in the simulated annealing is sufficiently slow. Minimisation is over the entire space of binary images, contrasting with the distance average where minimisation is restricted to the family of sets (5.16).

To minimise $R(B_t)$, we use a simulated annealing approach and interpret B_t

as a Markov random field with distribution

$$\pi(B_t; i) \propto \exp(-R(B_t))^{1/T(i)}. \tag{5.22}$$

where $T(i)$ is some decreasing cooling schedule defined such that $T(i)$ tends slowly to 0 as $t \to \infty$. The mode of $\pi(B_t)$ can now be approximated using a Metropolis algorithm within simulated annealing as follows. Start with some initial image B_0 and decreasing temperature schedule $T(i)$. At iteration i, propose for each pixel x the opposite colour, if pixel x is white then propose the colour black and vice versa. Denote the old configuration by B_i and the proposed new configuration by $B_i^{(x)}$. Accept the move with probability

$$\min\left\{1, \exp\left(\frac{1}{T(i)}(R(B_i) - R(B_i^{(x)}))\right)\right\}.$$

After each full sweep of all pixels update the iteration value i. The hope is that if the cooling schedule is sufficiently slow then the resulting image $\hat{Y}_t$ will converge to the global minimiser of $R(B_t)$.

It is worth noting that the above algorithm may be carried out for higher powers of p, although this will increase the computing time dramatically.

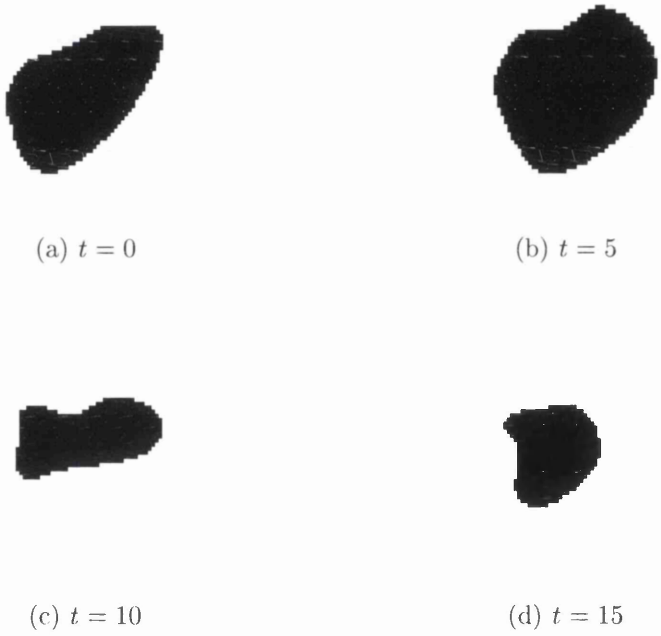
5.5 Early Experimental Results

This section displays some results of the smoothing algorithm on a synthesised binary image sequence in Figure 5.2. All of the images are of size 64 x 64 pixels. Figure 5.3 displays results of the smoothing algorithm at various intermediate time points. The Epanechnikov kernel,

$$k(t) = \frac{3}{4}(1 - t^2)\mathbf{1}_{\{|t| \leq 1\}}$$

was used with bandwidth $h = 0.5$. The cooling schedule $T(i)/T(0) = 0.99^i$ was used with 10 and 0.005 as the initial and final temperatures. This represents approximately 750 iterations for each smooth image.

It is clear from Figure 5.3 that the resultant image sequence is certainly smoother than the original sequence, although an unfortunate aspect is that the resultant images don't maintain the connectivity of the original sequence. This is certainly an area which may require further research.


(a) $t = 0$

(b) $t = 5$

(c) $t = 10$

(d) $t = 15$

Figure 5.2. Original image sequence at time points: (a) Time $t = 0$, (b) Time $t = 5$, (c) Time $t = 10$ (d) Time $t = 15$

5.6 A further application - image warping

Suppose given two sets X, Y both in $\mathbb{R}^d$, *image warping* aims to find a family of sets $\{Z_t, 0 \leq t \leq 1\}$ which interpolate X and Y. When $t = 0$, $Z_t = X$ and when

$t = 1$, $Z_t = Y$. As t increases, Z_t progressively leaves X and goes to Y.

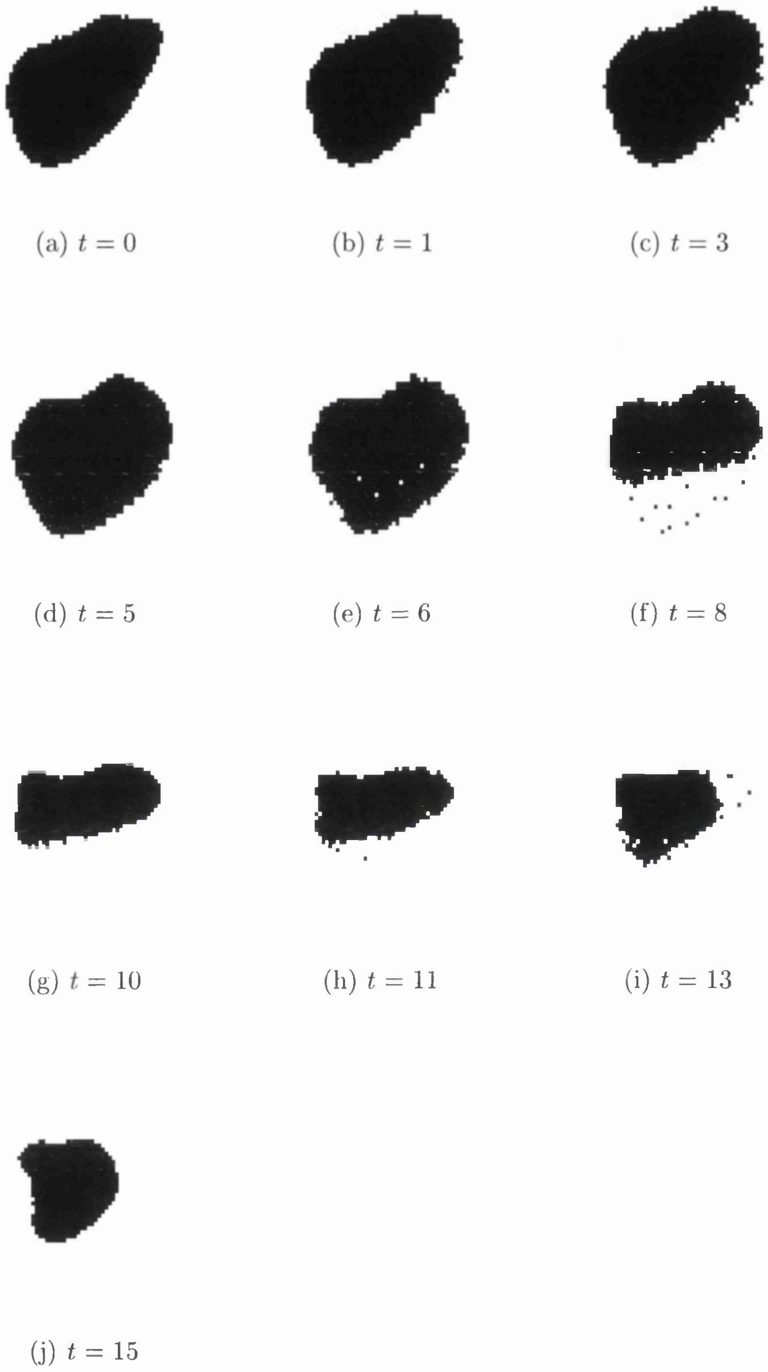
The idea is simply to find for a particular value t lying between 0 and 1, the binary image $\hat{Z}_t$ satisfying:

$$\hat{Z}_t = \arg\min_{B_t} \left[t\Delta_c^2(X, B_t) + (1-t)\Delta_c^2(Y, B_t) \right]. \tag{5.23}$$

It is clear that this is identical to the problem of smoothing image sequences addressed in the previous section. In this case the observed images are just the two images X and Y. The weight function is simply,

$$w(t) = \begin{cases} t, & \text{image } X \\ 1-t, & \text{image } Y. \end{cases}$$

Figure 5.4 displays results of image warping on the input images Figure 5.4(a) and Figure 5.4(f). As with the smoothed image sequence in the previous section, the resultant images don't preserve the connectivity of the input images.


(a) $t = 0$

(b) $t = 1$

(c) $t = 3$

(d) $t = 5$

(e) $t = 6$

(f) $t = 8$

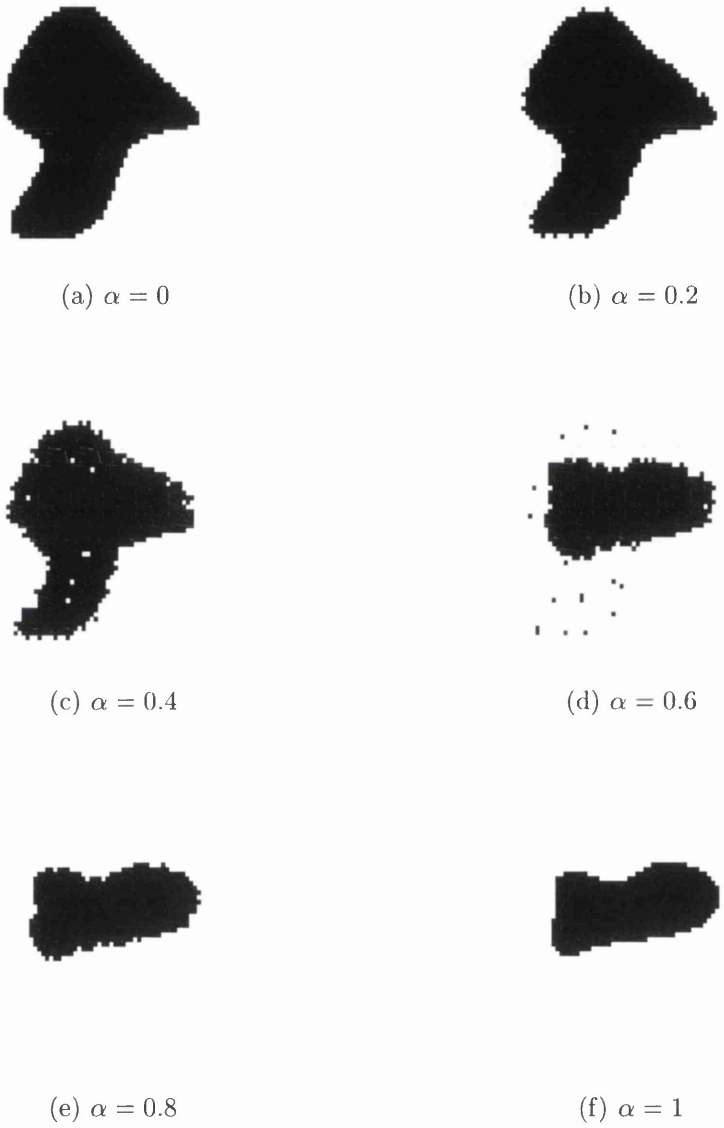
(g) $t = 10$

(h) $t = 11$

(i) $t = 13$

(j) $t = 15$

Figure 5.3. Smooth image sequence at various time points between (a) $t = 0$ and (j) $t = 15$



(a) $\alpha = 0$

(b) $\alpha = 0.2$

(c) $\alpha = 0.4$

(d) $\alpha = 0.6$

(e) $\alpha = 0.8$

(f) $\alpha = 1$

Figure 5.4. Results of image warping between images corresponding to $\alpha = 0$ and $\alpha = 1$

Chapter 6

Conclusions

This thesis has achieved its stated aim of placing many image analysis algorithms in a statistical context. Through the course of this research many further questions have arisen, which could form the basis for future developments. We briefly summarise and conclude the work presented in this thesis.

The distance threshold in Chapter 2 has been shown to work extremely well on a wide variety of image types, certainly improving on many techniques presented in the literature. Indeed it represents an altogether new approach to thresholding. One problem which may be worthwhile exploring in the future is that of the robustness of the threshold with respect to image perturbations. This problem has received very little attention in the literature, and certainly would be a desirable property which any thresholding algorithm should possess.

The newly presented image metrics, double integral metric and grey-scale distance transform metric, in Chapter 3 have shown encouraging results. The design of image metrics is one area of image analysis, which in the author's opinion has received very little attention in the literature, but which is fundamental to many image processing algorithms. One worry with the newly presented image metrics might be the computational time required to calculate these metrics.

They certainly take longer to compute than the usual RMS distance, however this might be counterbalanced with each metrics improved performance. The double integral metric in particular has performed well.

One of the main motivations in exploring new image metrics was to test their performance as loss functions in Bayesian image restoration, outlined in Chapter 4. It is worthwhile that the new double integral metric can be conveniently used to solve the often cumbersome problem of estimating the optimal Bayes estimator. One of the main reasons why new loss functions are rarely applied to this problem is primarily due to the computational difficulty of estimating the corresponding optimal Bayes estimators. The results displayed are encouraging, however one drawback, in common with many Bayesian restoration techniques, is that it is difficult to see global effects of the restored estimates, due to computational difficulties in working with large images.

The set-valued regression problem explored in the final chapter is clearly a very useful problem. It may be seen that this problem might be applied to many different scenarios. For the particular example of set-valued regression on binary images, the results may not be as visually appealing as would be hoped. However it does outline a very promising approach and basis for this problem. A further problem worth exploring might be that of finding appropriate stochastic noise models. This might suggest a maximum likelihood approach to set-valued regression.

Bibliography

[1] I.E. Abdou and W.K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. In *Proceedings of the IEEE, 67*, 1979.

[2] A.J. Baddeley. Errors in binary images and a L^p version of the Hausdorff metric. *Nieuw Archief voor Wiskunde*, 10:157–183, 1992.

[3] A.J. Baddeley and I.S. Molchanov. Averaging of random sets based on their distance functions. *Journal of Mathematical Imaging and Vision*, 8:79–92, 1998.

[4] J. Besag. On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Satistical Society, series B*, 48(3):259–302, 1986.

[5] G. Borgefors. Distance transforms in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[6] G. Borgefors. Distance transforms in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[7] A.W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis.* Oxford university press, 1997.

[8] L. Boxer. On Hausdorff-like metrics for fuzzy sets. *Pattern Recognition Letters*, 18:115–118, 1997.

[9] C. Chang, K. Chen, J. Wang, and M. Althouse. A relative entropy-based approach to image processing. *Pattern Recognition*, 27:1275–1289, 1994.

[10] B.B. Chaudhuri and A. Rosenfeld. On a metric distance between fuzzy sets. *Pattern Recognition Letters*, 11(17):1157–1160, 1996.

[11] G. Choquet. Theory of capacities. *Ann. Inst. Fourier*, 5:131–295, 1953/54.

[12] J.A. Cuesta and C. Matran. The strong law of large numbers for k-means and best possible nets of Banach valued random variables. *Probab. Th. Rel. Fields*, 78:523–534, 1988.

[13] M.C. Delfour and J.-P. Zolésio. Shape analysis via oriented distance functions. *Journal of Functional Analysis*, 123:129–201, 1994.

[14] D. Denneberg. *Non-additive measure and integral.* Kluwer, Dordrecht, 1994.

[15] J. Fan and I. Gijbels. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):2008–2036, 1992.

[16] N. Friel and I.S. Molchanov. Class of error metrics for grey-scale image comparison. In *Proc. of SPIE, Mathematical Modeling and Estimation Techniques in Computer Vision.*, volume 3457, San Diego, July 1998.

[17] N. Friel and I.S. Molchanov. Distances between grey-scale images. In H. Heijmans and J. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, Dordrecht, 1998.

[18] N. Friel and I.S. Molchanov. A new thresholding technique based on random sets. *Pattern Recognition*, 32:1507–1517, 1999.

[19] A. Frigessi and H. Rue. Bayesian image classification with Baddeley's delta loss. *Journal of Computational and Graphical Statistics*, 6:55–73, 1997.

[20] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[21] S. Geman and D.E. McClure. Statistical methods for tomographic image reconstruction. *Bull. Int. Statist. Inst. Bk.*, 4:5–21, 1987.

[22] W.R. Gilks, S. Richarson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice.* Chapman and Hall, 1996.

[23] C. A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing*, 55(6):532–537, 1993.

[24] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[25] T. Kaijser. Computing the Kantorovich distance for images. *Journal of Mathematical Imaging and Vision*, 9:173–191, 1998.

[26] L. Kantorovich. On a problem of Monge. *Uspekhi Mat. Nauk.*, 3(2):225–226, 1948. (in Russian).

[27] J.N. Kapur, P.K. Sahoo, and A.K.C Wong. A new method for grey-level picture thresholding using the entropy of the histogram. *Computer vision, graphics and image processing*, 29:273–285, 1985.

[28] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19:41–47, 1986.

[29] C.K. Leung and F.K. Lam. Maximum segmented image information thresholding. *Graphical Models and Image Processing*, 60(1):57–76, 1998.

[30] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091, 1953.

[31] I.S. Molchanov. *Limit Theorems for Unions of Random Closed Sets*, volume 1561 of Lect. Notes Math. Springer, Berlin, 1993.

[32] I.S. Molchanov. Statistical problems for random sets. In R. Mahler J. Goutsias and H.T. Nguyen, editors, *Applications and Theory of Random Sets*, pages 27–45, Berlin, 1997. Springer.

[33] I.S. Molchanov. Grey-scale images and random sets. In H. Heijmans and J. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 247–257, Dordrecht, 1998.

[34] A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–444, 1997.

[35] N. Otsu. A threshold selection method from grey-level histogram. *IEEE Trans. Systems Man Cybernet.*, SMC-8:62–66, 1978.

[36] N.R. Pal and S.K. Pal. Entropic thresholding. *Signal Processing*, 16:97–108, 1989.

[37] S.K. Pal and A. Ghosh. Fuzzy geometry in image analysis. *Fuzzy sets and systems*, 48:23–40, 1992.

[38] W.K. Pratt. *Digital Image Processing.* John Wiley and Sons, New York, 1977.

[39] T. Pun. A new method for grey-level picture thresholding using the entropy of the histogram. *Signal Processing*, 2:223–237, 1980.

[40] T. Pun. Entropic thresholding: A new approach. *Computer Vision, Graphics and Image Processing*, 16:210–239, 1981.

[41] S.T. Rachev. *Probability Metrics and the Stability of Stochastic Models.* Wiley, Chichester, 1991.

[42] A. Rosenfeld and A. Kak. *Digital Picture Processing.* Academic Press, New York, 1976.

[43] A. Rosenfeld and J.L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[44] H. Rue. New loss functions in Bayesian imaging. *Journal of the American Statistical Association*, 90:900–908, 1995.

[45] H. Rue. A loss function model for the restoration of grey level images. *Scandanavian Journal of Statistics*, 24:103–114, 1997.

[46] C. Sahoo, P. Wilkins and J. Yeager. Threshold selection using Renyi's entropy. *Pattern Recognition*, 30(1):71–84, 1997.

[47] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1982.

[48] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948.

[49] D. Stoyan and H. Stoyan. *Fractals, Random Shapes and Points Fields.* Wiley, Chichester, 1994.

[50] M.P. Wand and M.C. Jones. *Kernel Smoothing.* Chapman and Hall, 1995.