EXTENSIONS OF THE HIT-OR-MISS TRANSFORM FOR FEATURE DETECTION IN NOISY IMAGES AND A NOVEL DESIGN TOOL FOR ESTIMATING ITS PARAMETERS

By

Paul Murray

In the fulfilment of the requirement

for the degree of Doctor of Philosophy

Centre for excellence in Signal & Image Processing

Department of Electronic & Electrical Engineering

University of Strathclyde

Supervised by

Professor Stephen Marshall

Declaration and Copyright

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed:
Date:

Acknowledgements

There are a number of people who I would like to thank for their support throughout the duration of my studies.

First, I must I express my most sincere gratitude to my supervisor, Prof. Stephen Marshall. Steve introduced me to the world of image processing during a three month summer placement which I was fortunate enough to undertake in the summer of 2007. It was during this time that I began to consider the option of working towards a PhD, and I firmly believe that had I not been given that 12 week placement, I would never have taken this path in life. I would like to say big thank you to Steve for his continued help, support and guidance, both in the academic environment, and on a personal level during the time I have spent working under his supervision. I am also grateful to Steve for sending me to a number of fantastic places outside of the University, in order to extend my knowledge and to allow me to learn from a number of inspirational people. Finally, I must thank him for his constant reassurance and countless consultations while I was writing this thesis. I may not have managed to complete this body of work without it.

I must thank Eric Bullinger, now at the University of Liege, for his guidance in the early stages of this work. Thanks also to Des McGhee, department of Maths and Statistics here at the University of Strathclyde, for his guidance on mathematical notation and set theory. His assistance helped secure a publication in the second year of my studies. Thanks also to Ian Glover, Centre for excellence in Signal and Image processing (CeSIP) here at Strathclyde, for a helpful discussion regarding Section 4.5 of this thesis.

Thanks to Ilya Shmulevich, Institute for Systems Biology (ISB), for allowing me to visit him there for three months to carry out research under his supervision. I would like to thank Ilya for his enthusiasm regarding my work, and for his constant sharing of ideas. Thanks also to Ilya for his interesting suggestions and discussions about methods for improving several aspects of my work. I am also grateful to Ilya for

helping to send me to Cold Spring Harbour Laboratory to present work that was carried out in collaboration with his institution.

Thanks to Cecilia Garmendia and Aimée Dudley for allowing me work on an extremely interesting and challenging collaborative project which should provide some interesting results in the near future. Thanks to Noel Blake for helping me acquire sets of biological images, and for his patience when explaining the data acquisition process to me. Thanks also to Nicholas Flann for many interesting discussions regarding image analysis, and other data processing techniques. Thanks to Adrian Ozinsky for his interesting discussions and explanations about the fundamental biological aspects of the work that goes on in his lab, and in other research centres in Seattle and further afield. Special thanks also go to my many friends at ISB who made my time there stimulating, interesting and truly unforgettable…both academically and socially speaking!

This may be a little unorthodox as far as thesis acknowledgements go, however, I would like to thank Walter Galbraith and Simon Whitely for their support during the final years of my undergraduate studies. I believe I have told them this, but here it is in writing…had it not been for their support, I doubt that I would have been able to get as far in my career as I have. Thanks for taking me under your wing guys!

Thanks to Monica Schliemann and Peter Scheurich at the Institute of Cell Biology and Immunology, University of Stuttgart, for supplying the cancer cell images used in a number of Figures in Chapter 4 and 5 of this thesis. I would also like to thank Stefano Fumagalli, Patrick Ejlerskov, and Jonathan A. Coles in the Centre for Biophotonics, University of Strathclyde, and James Brewer at the University of Glasgow, for allowing us to use the images shown in all figures in Section 5.4.1 and in Figure 6.14.

Thanks must also go to Benjamin Perret, at the University of Strasbourg for providing us with the set of astronomical images (used in his paper) and for granting us permission to use this data to produce Figure 5.14 and Figure 6.15. I would also

like to thank Benjamin for the interesting email discussion that we had regarding the Hit-or-Miss Transform. Thanks also to Erik Urbach, CSIRO, for providing the code that implements his own fast algorithm and the competing techniques against which he tested his routine. This saved significant time when implementing and testing the fast algorithm that is described in this thesis. Thanks also to Erik Urbach for granting permission to produce Figure 6.6 and Figure 6.11 shown in Chapter 6.

On a more personal note, I would like to thank my family for always encouraging me to strive for the best in life. For always believing in my ability - especially when I doubt it myself - and for listening to my many concerns and doubts over the years about my getting to where I am today. Thanks!

Finally, I would like to thank all of my friends at the University of Strathclyde, and, in particular, the past and present members of CeSIP for the friendship they have shown over the years. I must also say a big THANK YOU to a few special friends at the University for providing a sympathetic ear when I needed it most (you know who you are…and if you are unsure, you can ask me. I will tell you if you are part of this special group or not ☺).

Abstract

The work presented in this thesis focuses on extending a transform from Mathematical Morphology, known as the Hit-or-Miss transform (HMT), in order to make it more robust for detecting features of interest in the presence of noise in digital images. The extension that is described here requires that a single parameter is determined for correct functionality. A novel design tool which allows this parameter to be accurately estimated is proposed as part of this work. An efficient method for computing the extended transform is also presented.

The HMT is a well known morphological transform that is capable of identifying features in digital images. When image features contain noise, texture or some other distortion, the HMT may fail. Various researchers have extended the HMT in different ways to make it more robust to noise. The most successful, and most recent extensions of the HMT for noise robustness, use rank order operators in place of standard morphological erosions and dilations. A major issue with most of these methods is that no technique is provided for calculating the parameters that are introduced to generalise the HMT, and, in most cases, these parameters are determined empirically.

In this thesis, a new conceptual interpretation of the HMT is presented which uses percentage occupancy (PO) functions to implement the erosion and dilation operators of the HMT. When implemented in this way, the strictness of these PO functions can easily be relaxed in order to allow slacker fitting of the structuring elements. Relaxing the strict conditions of the transform is shown to improve the performance of the routine when processing noisy data.

This thesis also introduces a novel design tool which is derived directly from the operators that are used to implement the aforementioned PO functions. This design tool can be used to determine a suitable value for the only parameter in the proposed extension of the HMT. Further, it can be used to estimate parameters for other generalisations of the HMT that have been described in the literature in order to

improve their noise robustness. The power of the proposed technique is demonstrated and tested using sets of very noisy images. Further, a number of comparisons are performed in order to validate the method that is introduced in this work when compared with the most recent extensions of the HMT.

One drawback with this method is that a direct implementation of the technique is computationally expensive. However, it is possible to implement the proposed method using rank-order filters in place of the percentage occupancy functions. Rank order filters are used in a multitude of image processing tasks. Their application can range from simple pre-processing tasks which aim to reduce/remove noise, to more complex problems where such filters can be used in combination to detect and segment image features. There is, therefore, a need to develop fast algorithms to compute the output of this class of filter in general.

A number of methods for efficiently computing the output of specific rank order filters have been presented over the years. For example, numerous fast algorithms exist that can be used for calculating the output of the median filter. Fast algorithms for calculating morphological erosions and dilations - which, like the median filter, are a special case of the more general rank order filter - have also been proposed. In this thesis, these techniques are extended and combined such that it is possible to efficiently compute any rank, using any arbitrarily shaped window, making it possible to quickly compute the output of any rank order filter. The fast algorithm which is described is compared to an optimised technique for computing the output of this class of filter, and significant gains in speed are demonstrated when using the proposed technique. Further, it is shown that this efficient filtering algorithm can be used to produce an extremely fast implementation of the generalised HMT that is described in this work. The fast generalised HMT is compared with a number of other extensions and generalisations of the HMT that have been proposed in the literature over the years.

Table of Contents

List of Symbols

Fundamentals

$\mathbb{Z}$ ……Set of integers

$\mathbb{R}$ ……Set of real numbers

$\mathbb{Z}^n$ ……n-dimensional discrete space

$\mathbb{R}^n$ ……n-dimensional Euclidean space

$\varnothing$ ……..The empty set

$\in$ ……..Is an element of/belongs to

$\{x|\text{CONDITION}\}$ ……..The set of points x which satisfy some CONDITION

$\forall$ ……..For all

$\exists$ ……..There exists

X ……..Set of n-dimensional Euclidean space E^n

X^c ……The complement of set X

B ……Structuring element (SE)

$\hat{B}$ ……..The reflection of B

B^* ……The dual of B

f, g ……Functions used to represent greyscale/binary images

f^c ……..…..The complement of f

I ……...A greyscale image

x ……Position vector used to index image pixels

b ……..Position vector used to index points of SE

t ……Intensity level of an image pixel in a greyscale image

X_b ……Translation of X by some position vector b

$Card(\cdot)$ ……..Cardinality, i.e. the number of elements in $(\cdot)$

$\text{supp}(\cdot)$ ……...Support/definition domain of $(\cdot)$

$\cap$ ……..Binary intersection

$\cup$ ……Binary union

$\wedge$ ……Point-wise minimum

$\vee$ ……Point-wise maximum

$\setminus$ ……...Set difference operation

$\subset$ ……Is a subset of

$\subseteq$ ……Is a proper subset of

ψ, ϕ ……...General symbols denoting a transform

ρ ……..Scale factor

LS_t ……...Level set obtained by thresholding at level t

Morphological (and related) operations and definitions

$I \ominus B$……..Erosion of I by SE B

$I \oplus B$ ……Dilation of I by SE B

$I \circ B$ ……Opening of I by SE B

$I \bullet B$ ……...Closing of I by SE B

$E_G^{(u)}(F)$ ……Geodesic erosion of size u of F with respect to G

$D_G^{(u)}(F)$ ……...Geodesic dilation of size u of F with respect to G

$R_G^D(F)$ ……Reconstruction by dilation of F with respect to G

$R_G^E(F)$ ……..Reconstruction by erosion of F with respect to G

$\zeta_{B,k}$ ……...Rank order filter of rank k using window/SE B

P……Percentage occupancy parameter of the POHMT

d……Number of shifts needed to obtain right angled PO plot

l_{FG}, l_{BG} ……...Level which resp. the foreground and background SE can reach

O_{FG}, O_{BG} .....Occupancy of resp. the foreground and background SE

PO_{FG}, PO_{BG} .....Percentage occupancy of resp. the foreground and background SE

PO_B .....Value used to decide the output of the POHMT

List of Acronyms

FG……...Image foreground

BG……Image background

SE ……...Structuring element

HMT……...Hit-or-Miss Transform

CHMT……....Extended, constrained HMT proposed by Soille

UHMT……..Extended, unconstrained HMT proposed by Soille

RHMT……Extended HMT proposed by Ronse

KHMT……...Extended HMT proposed by Khosravi and Schafer

BHMT……..Extended HMT proposed by Barrat *et al.*

FHMT……Extended HMT proposed by Perret *et al.*

SHMT……... ……...Synthetic HMT proposed by Doh *et al.*

POHMT…….Percentage occupancy HMT - extended HMT proposed in this thesis

PO……Percentage occupancy

PO plot…….Percentage occupancy plot

LSB……Low surface brightness galaxy

ROC…….Receiver operator characteristic

vHGW…….van Herk/Gil Werman algorithm

LUT……...Look up table

MLE…….Maximum likelihood estimation

SNR…….Signal to noise ratio

AWGN……...Additive white Gaussian noise

1 Introduction

Mathematical Morphology, first introduced by Matheron [1] and Serra [2] and later extended by Heijmans [3], provides an extremely powerful set of tools for image processing. Among these is the Hit-or-Miss Transform (HMT) [2], which is capable of identifying groups of connected pixels that comply with certain geometric properties. The HMT has been used by a large number of researchers in a wide range of application areas, including: medical image processing [4], [5]; optical character recognition [6]; face localisation, [7]; astronomical imaging [8]; and remote sensing [9], [10] and [11].

Although the HMT is widely used, one of the major drawbacks with the standard transform is that it will fail to detect objects of interest if there is noise in a given image, or if image features are extremely textured. In fact, it takes only a single pixel in the foreground of a sought feature to fall to the level of the background (or below it) for the transform fail. Similarly, if just one pixel in the surrounding background of a feature reaches the level of the foreground (or above it) as a result of noise, the transform will fail. This sensitivity makes the standard transform unusable when image data is extremely noisy. The aim of this thesis is to extend the standard HMT in order to overcome these difficulties.

The main motivation for this work was application driven. The author was provided with a set of images containing cancer cells and was asked if it would be possible to detect and segment the cells automatically. The data sets were noisy, and there was some variation in the geometry and intensity of the cells that were visible in the image. However, the cells appeared to be geometrically well defined, and as a result, the standard HMT was tested to determine whether or not it would be capable of locating these cells in the images. Initially, this seemed like a reasonable approach. However, after trying a number of composite structuring elements (SEs), it became clear that it was not possible to detect all of the cells using one pair of SEs due to the levels of noise that were present in the image. As a result, the aim of this PhD thesis was to extend the standard transform such that it would be capable of detecting these

cancer cells, and more generally, to make the transform capable of detecting any features that could be represented by a composite SE, in very noisy data.

There has been significant interest in extending the HMT over many years in order to make it more robust to noise, and a number of authors have presented their own techniques and methods for achieving this. Some of these techniques modify the SEs that are used to search for features of interest, others apply pre-processing to increase the likelihood of successful detection, while others modify the conditions of the transform itself. All of these extensions are thoroughly explained and discussed in the literature review that is provided in Chapter 3. However, it is pointed out here, that one aspect that most of these techniques have in common, no matter which form the extension takes, is that the proposed extended transform generally requires some parameter to be accurately determined for successful operation. In some cases, these parameters are determined empirically. However, in most cases no generic solution for setting these parameters is provided. The values of the parameters can be critical to the detection task and therefore require further study.

This thesis aims to address this issue by introducing a novel extension to the HMT that is accompanied by a robust method for setting its parameters. It will be shown, that in addition to estimating parameters for the method that is proposed in this thesis, this design tool can be used to determine the equivalent parameters for other HMTs that have been proposed in the literature.

1.1 Organisation of Thesis

The remainder of this thesis is set out in the following chapters.

Chapter 2 introduces the fundamental morphological operators of erosion and dilation, as well as the notation, and some more advanced morphological operators that are used in this thesis. This chapter is not intended to provide an exhaustive coverage of mathematical morphology, but it should provide sufficient detail that referring to alternative texts is not required to understand this thesis and its contributions.

Chapter 3 provides an extensive literature review which puts the techniques that are proposed in this thesis in context. It discusses the original definition of the HMT for application to binary images and its numerous extensions for the greyscale case. Techniques that are used to improve the performance of both the binary and greyscale HMTs when image data is noisy are also discussed. Then, a number of techniques which can be used to improve the efficiency of morphological operators and other rank order filters are reviewed.

Chapter 4 defines and discusses a number of existing greyscale HMTs and places them in context. The new conceptual HMT, called the POHMT, is then presented, before it is extended in order to make it more robust in the presence of noise. Two methods for setting the transform's single parameter are provided, and some results are shown.

Chapter 5 introduces a novel design tool that can be used to reliably set the only parameter of the extended HMT that is described in Chapter 4. An explanation of this design tool is provided, and examples are used to show how it may be used for setting the single parameter of the POHMT, and for setting similar parameters of competing techniques. A large number of experimental results are provided in this chapter to demonstrate how this novel design tool and the extended HMT (Chapter 4) can be used to detect features in very noisy data.

Chapter 6 presents a fast algorithm for computing the output of rank order filters using arbitrarily shaped windows. The method is demonstrated for the general case of rank order filtering, before the relationship between the extended HMT and rank order filters is explained. The efficient algorithm is tested under a number of conditions and it is then shown that implementing the extended HMT, using this efficient algorithm, provides significant reductions in the time taken to compute the output of the routine.

Chapter 7 provides some concluding remarks and gives suggestions for further work.

1.2 Original Contributions of the Work

It is believed that the novel contributions of this work are:

- A new conceptual definition of the greyscale HMT (Chapter 4)
- An extension of this greyscale HMT, known as a Percentage Occupancy Hit-or-Miss Transform (POHMT) for improved accuracy in noise (Chapter 4)
- A technique for estimating the parameter of the POHMT using noise models (Chapter 4)
- A novel design tool known as a Percentage Occupancy (PO) plot (Chapter 5)
- A method to make the POHMT operate as a discriminatory filter (Chapter 5)
- A description of an efficient method for computing the output of arbitrarily shaped rank order filters (Chapter 6)
- A set of equations that can be used to compute the "critical points" of an arbitrarily shaped window (Chapter 6)

Each of these contributions will now be explained in more detail.

In Chapter 4 of this thesis, a novel conceptual definition of the standard HMT is presented, and this is placed in the context of a number of existing greyscale HMTs. In the definition proposed here, the traditional morphological operators that are used to implement the HMT are replaced with two PO functions that allow features in the image to be marked if they simultaneously occupy 100% of both SEs. Defining the standard greyscale HMT in this way facilitates the novel extension of the transform (presented in this thesis) which relaxes the strict conditions of the transform itself.

The second novel contribution of this work is the POHMT which is introduced in this thesis in order to overcome the difficulties that are faced by the standard transform when it is presented with noise. An empirical method is used to determine a suitable value for the single parameter of the POHMT at this stage, and then a novel approach, based on analysing noise models (cropped from the image), is used for this purpose. A comparison of these approaches is provided.

The POHMT bears resemblance to some existing methods which aim to overcome the HMT's sensitivity to noise in so far as the POHMT essentially replaces traditional morphological operators with more general rank order filters. The difference here, however, is that a completely original design tool, known as a percentage occupancy plot, is introduced which can be used to determine a suitable value for the single parameter that must be fixed before using the POHMT. Another advantage of the approach taken here is that this novel PO plot can be used by other researchers to set similar parameters for their own routines. This is demonstrated in Chapter 5, where a number of greyscale HMTs are modified and their parameters are estimated using the new design tool. It is also shown that the PO plot can be analysed in such a way that it is possible to make the POHMT function as a discriminatory filter that is capable of selectively marking and discarding features in the image.

In addition to extending the HMT and introducing a novel design tool, this thesis presents an efficient method for computing the output of arbitrarily shaped rank order filters. The efficient algorithm that is presented here extends and combines two techniques that have already been presented in the literature: one for efficiently computing the output of the median filter; and a second for fast computation of morphological operators. The novel contribution in this work lies in describing how these methods can be combined and used to compute the output of any rank order filter defined by an arbitrarily shaped window. Further, a mathematical formulation is introduced which can be used to calculate the required set of so called "critical points" in the SE. As will be explained, these points are used to keep track of values entering and leaving the arbitrary window as it moves around the image.

The new fast algorithm can be used to increase the efficiency of any process that makes use of rank order filters, for example, when pre-processing image data to remove noise, or, in more complex filtering tasks. Of more direct importance here, is that the efficient algorithm for computing rank order filters can be used to implement a fast POHMT. It is shown in Chapter 6 that the single parameter of the POHMT can be used to set the rank parameter of the efficient rank filters before the fast POHMT is then demonstrated to compute the output of the transform in times which cannot be achieved by existing algorithms.

The ideas and techniques that are proposed in this thesis are placed in the context of the leading methods for computing the HMT when image data is noisy. It is shown by example how the new design tool can be used to set parameters for alternative HMTs. Further, the extent to which the setting of these parameters has on improving the noise robustness of these HMTs is quantified and compared to the POHMT. All of the algorithms and ideas that are presented in this thesis are compared with the state-of-the-art techniques in the area.

2 Mathematical Morphology and Related Techniques

This chapter introduces the fundamental theory and background that is required to understand the terms and techniques that are discussed and used in this thesis. The major contribution of this work lies in a new conceptual extension of a transform which exists in the area of Mathematical Morphology. Therefore, this chapter focuses mainly on introducing and defining the fundamental operators that underpin Mathematical Morphology.

Mathematical Morphology aims to analyse features and structures in images based on their shape and size, hence the name morphology. In [12], it is said to be mathematical in the sense that it is defined in terms of set theory, integral geometry and lattice algebra. Morphological transforms are commonly used to perform a number of tasks ranging from image filtering through to segmentation and extraction of image features. The two fundamental morphological operators are known as erosion (a shrinking operator) and dilation (a growing operator). Most other morphological transforms and operators can be derived from these two fundamental operators.

This chapter sets out the notation that will be used in this thesis and introduces some properties of morphological operators. The concept of a structuring element, which is critical in most morphological operations, is then introduced. Erosion and dilation are defined for application to both binary and greyscale images. A number of other transforms that are used in this thesis are also defined and explained by example. The standard Hit-or-Miss Transform which is extended, and is hence the focus of this thesis, is also defined in this section. Finally, this chapter concludes by explaining the relationship between morphological operators and rank order filters which is required for complete comprehension of the extensions made to the HMT in the subsequent chapters.

2.1 Preliminaries and Properties of Mathematical Morphology

In this section some preliminaries are introduced, and properties that facilitate the discussion and analysis of morphological operators, are defined.

2.1.1 Set Representation of Binary Images

Pixels in a binary image are either "on" or "off". By convention, in this thesis, "on" pixels have a value, 1, and off pixels have a value, 0. An example of a binary image is shown in Figure 2.1. Black regions in the image correspond to off pixels which represent the image background (BG), while white pixels correspond to the on pixels which make up the image foreground (FG).

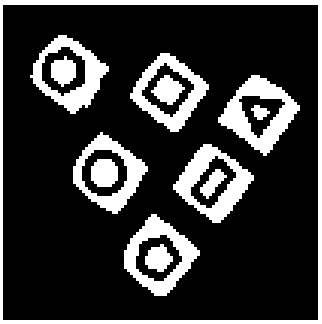


Figure 2.1 Binary Image

Mathematically, a binary image *f* can be described as a mapping of the support of *f*, $\text{supp}(f)$, into $\{0,1\}$, where $\text{supp}(f)$ belongs to $E = \mathbb{Z}^2$. More formally,

$$f: \ \text{supp}(f) \subset E \to \{0,1\} \tag{2.1}$$

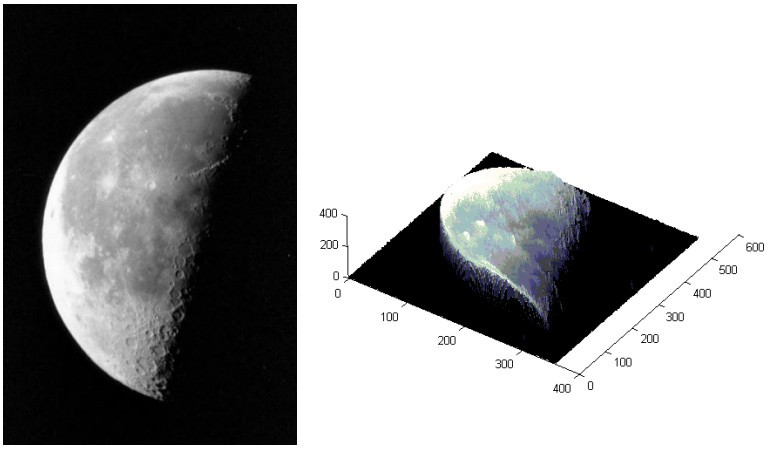
where for all $x \in E$, $f(x)$ is either 0 or 1.

2.1.2 Set Representation of Greyscale Images

The dynamic range of a greyscale image is determined by the bitdepth per image pixel. That is, instead of mapping the support of f into $\{0,1\}$ as was the case with the binary image (where the bitdepth is 1), $\text{supp}(f)$ of a greyscale image is mapped into the bounded set $\{0,1,2,...,2^n-1\}$, where n denotes the number of bits that are used to represent each image pixel. That is,

$$f: \ \text{supp}(f) \subset E \to \{0,1,2,...,2^n-1\}. \tag{2.2}$$

An example of a greyscale image is shown in Figure 2.2(a). In morphology, greyscale images are often interpreted as a topographic surface where the intensity of an image pixel can be thought of as an elevation in the topography. As such, a greyscale image can be viewed as a 3D surface as shown in Figure 2.2(b).



(a) (b)

Figure 2.2 Greyscale Images (a) Greyscale image of the moon. (b) Topographic representation of (a)

Greyscale images, like the one shown in Figure 2.2, commute with threshold decomposition. This means that greyscale images can be thought of as a stack of binary images or level sets (not including the one at $t=0$ for which all pixels have a value of one).

That is,

$$f = \sum_{t=1}^{2^n-1} \mathrm{LS}_t(f) \tag{2.3}$$

where

$$\mathrm{LS}_t(x) = \begin{cases} 1 \text{ if } f(x) \geq t \\ 0 \text{ otherwise} \end{cases}. \tag{2.4}$$

The definition of the greyscale image given in Equation (2.3) implies that there exists a binary image, LS_t (See Equation (2.4)), for each greyscale level that can be assigned to a pixel in the greyscale image. It follows that when $\mathrm{LS}_t(x) = 1$ for some level *t*, that $\mathrm{LS}_t(x)$ will equal 1 for all values between this level *t* and 0. This property has been exploited by a number of researchers [12], [13], [14], and, to some extent, this property is exploited in this thesis.

2.1.3 Translation Invariance

Let ψ denote some arbitrary image transform. A transform, ψ, is translation invariant if: applying the transform to some image *f* and translating the result is equivalent to translating the image prior to applying the transform. That is,

$$\psi \text{ is translation invariant } \Leftrightarrow \psi(f_b) = \left[\psi(f)\right]_b \tag{2.5}$$

where, $f_b = \{ f + b \,|\, b \in E \}$ i.e. the translation of image *f* by some vector *b*.

2.1.4 Scale Invariance

A transform, ψ, is scale invariant if the result of applying ψ to a scaled version of an image, is equivalent to first applying the transform and then scaling the output by the same factor. Let ρ denote some positive, non zero, scale factor, then

$$\psi \text{ is scale invariant } \Leftrightarrow \psi(\rho f) = \rho\left[\psi(f)\right]. \tag{2.6}$$

A number of morphological operations can be scale invariant.

2.1.5 Increasingness

Let *f* and *g* represent two separate images where $\forall f$ and $\forall$g, *g* is less than or equal to *f*. A transform is said to be increasing if it preserves this ordering relationship. That is,

$$\psi \text{ is increasing } \Leftrightarrow \forall f, \forall g, g \le f \Rightarrow \psi(g) \le \psi(f). \tag{2.7}$$

Many morphological operators are increasing.

2.1.6 Extensivity

A transform is extensive if the result of applying the transform to an image *f* is greater than or equal to *f*.

$$\psi \text{ is extensive } \Leftrightarrow f \le \psi(f). \tag{2.8}$$

Adding the same positive greylevel to all pixels in a given image could be considered as an extensive operator.

2.1.7 Anti-extensivity

A transform is said to be anti-extensive if the result of applying the transform to an image *f* is less than or equal to *f*.

$$\psi \text{ is anti-extensive } \Leftrightarrow \ \psi(f) \leq f. \tag{2.9}$$

Subtracting the same positive greylevel from all pixels in a given image could be considered as an anti-extensive operator.

2.1.8 Idempotence

A transform is said to be idempotent if the result of applying it more than once has no further affect on the result of applying it the first time. That is,

$$\psi \text{ is idempotent } \Leftrightarrow \psi(f) = \psi\left[\psi(f)\right]. \tag{2.10}$$

It is entirely redundant to apply an idempotent transform more than once to an image.

2.1.9 Duality

Let ψ and ϕ denote two transforms and f^c denote the complement of *f* where for all *x*, $f^c(x) = \left(2^n - 1\right) - f(x)$. Then ψ and ϕ are dual with respect to complementation if applying ψ to *f* is equivalent to applying ϕ to f^c and taking the complement of the result. That is,

$$\psi \text{ and } \phi \text{ are dual w.r.t. complementation } \Leftrightarrow \ \psi(f) = \left[\phi\left(f^c\right)\right]^c \tag{2.11}$$

Duality is a key property of morphological operators. Every morphological operator has its dual [15].

If some properties of a transform ψ are known then it is possible to infer the following for ϕ [12],

$$\begin{aligned}
&\text{if } \psi \text{ is idempotent} &&\Rightarrow \phi \text{ is idempotent,}\\
&\text{if } \psi \text{ is extensive} &&\Rightarrow \phi \text{ is anti-extensive,}\\
&\text{if } \psi \text{ is anti-extensive} &&\Rightarrow \phi \text{ is extensive,}\\
&\text{if } \psi \text{ is increasing} &&\Rightarrow \phi \text{ is increasing.}
\end{aligned}$$

2.2 Fundamental Morphological Operations

After defining the concept of a structuring element, the elementary morphological operations of erosion and dilation are defined in this section. Opening and Closing which are dual morphological operators, derived from erosion and dilation, are also presented.

Before going any further, the notation that is used throughout the remainder of this thesis will now be defined. For binary operations, let *X* represent a binary set and *B* represent a SE where *X* and *B* are sets in 2D space, $E = \mathbb{Z}^2$. For greyscale images and greyscale operations, let T^E be the set of all greylevel functions from a subspace of *E* to *T* where $T = \mathbb{R} \cup \{+\infty, -\infty\}$ or $T = \mathbb{Z} \cup \{+\infty, -\infty\}$ such that *T* is a complete lattice with respect to the order "$\leq$". Let $I \in T^E$, denote a greyscale image, and $B_{NF} \in T^E$ denote a greyscale SE.

2.2.1 Structuring Elements

In mathematical morphology, a structuring element is a small set (often a subset of the image being processed) that is used to probe the image that is being processed. SEs can be defined as 2D (flat SEs) binary sets, or as 3D (non-flat/greyscale SEs) functions, in which case, greyscale values can be assigned to the points in the SE. Flat (2D) SEs are much more commonly used than SEs of any other dimension. The main reason being that flat SEs are insensitive to variations in image intensity.

SEs must be designed with respect to some common point, known as the origin. This allows the positioning of the SE in an image and it defines the pixel for which the

output of any morphological operator will be calculated. In fact, the origin of the SE is often located in its centre; however, this is not a requirement. The origin of the SE can be placed anywhere inside or outside of the SE. The size and shape of the SE determine exactly what the outcome of the morphological operation will be. For example, the larger the SE, the greater the affect of applying some morphological transform. Some commonly used SEs are shown in Figure 2.3.

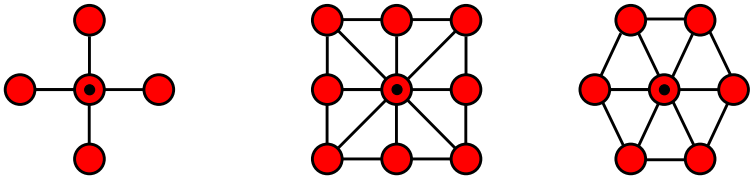


Figure 2.3 Three basic structuring elements (a) Diamond (b) Square (c) Hexagon. The origin of these SEs is marked by a black dot and corresponds to the centre pixel in the SEs shown here.

Of course, structuring elements can be designed to have any arbitrary shape and size, their geometry is not restricted to the examples shown in Figure 2.3. SE selection is usually determined by the structure of features in the image that are to be suppressed or preserved. It should be noted that the SEs shown in Figure 2.3 are flat SEs showing the spatial support of the elementary diamond, square and hexagonal SEs. Although non-flat SEs would have the same support, and hence the same shape that is illustrated in Figure 2.3, this support would be mapped into the bounded set $\{0,1,2,...,2^n-1\}$ in order to make a small greyscale image that can be used to probe the data that is to be processed. Care must be taken when designing non-flat SEs in order to ensure that results are accurate, reliable, and do not go out of range.

2.2.2 Erosion

Computing the erosion of a binary set *X* by SE *B* can be thought of as a process of translating *B* to every point in *X*, and marking in the output image, points in *X* where the SE *B* is completely contained in *X*. The result of eroding a binary set *X* by SE *B* is a set containing the locus of points that correspond to places where *B* fits inside *X*. Adopting the notation used by Gonzalez and Woods in [16], the erosion of *X* by *B*, denoted $X \ominus B$, may be written,

$$X \ominus B = \left\{ x \middle| B_x \subseteq X \right\}. \tag{2.12}$$

Although the notation used in [16] has been adopted here, it should be noted that an equivalent way to denote an erosion of *X* by *B* that is commonly used in the literature is to write $\varepsilon_B(X)$. An example of erosion is shown in Figure 2.4.

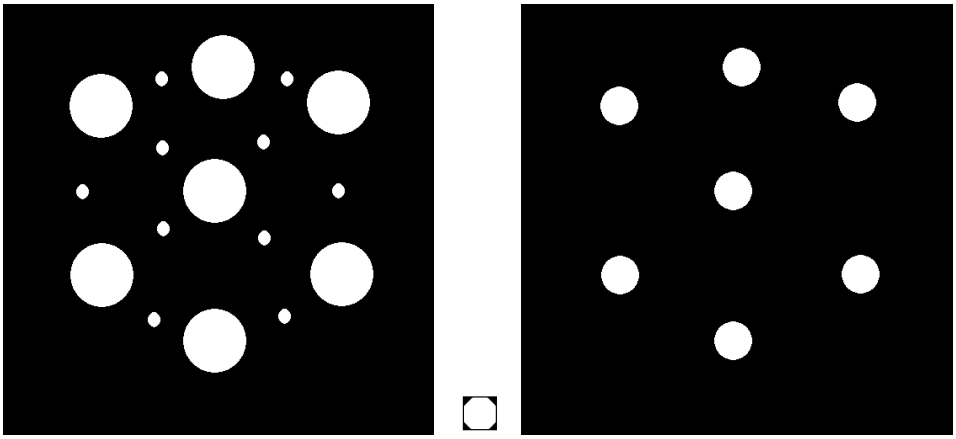


Figure 2.4 Example of binary erosion (a) Original image (512×512), (b) SE used to erode the image shown in (a), (c) Result of eroding the original image (a) by SE (b).

It is clear by comparing Figure 2.4(a) and Figure 2.4(c), that erosion is a shrinking operator which: reduces the dimensions of features that can contain the SE; and removes features that are too small to contain the SE.

An alternative method for implementing erosion is to calculate the intersection of all translations of the set *X* by the negative vectors of *B*,

$$X \ominus B = \bigcap_{b \in B} X_{-b}. \tag{2.13}$$

This definition of erosion is equivalent to Minkowski subtraction. An illustrative example of how this method works is shown in Figure 2.5. The example shown in Figure 2.5 uses 1D profiles to represent a binary image and it shows how the image can be translated before computing the erosion by calculating the intersection of all the image translates. The SE that has been used in Figure 2.5 is a three pixel wide horizontal line with its origin in the centre of the SE.

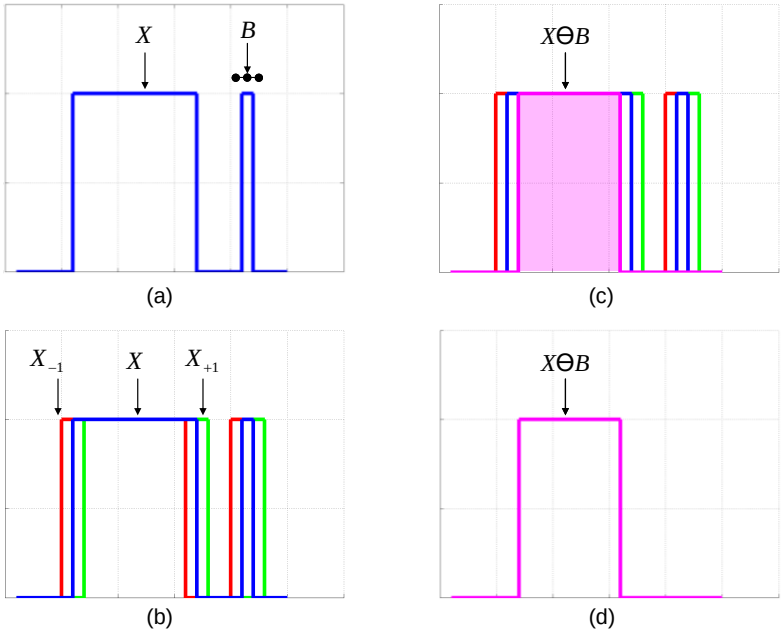


Figure 2.5 Example of binary erosion implemented by translating the image by the points in the SE and computing the intersection of all translates. (a) Original image and SE illustration. (b) Original image and the translations of this image by the negative vectors of the SE. (c) Result of erosion where the shaded magenta profile highlights the intersection of all image translations. (d) Result of eroding *X* by *B*, the leftmost feature is narrower in the eroded result than in *X*. Further, the rightmost feature of *X* has been removed by eroding.

The definition of erosion that is given in Equation (2.13) can be directly extended for processing greyscale images by replacing the intersection operation with the infimum (inf.) and the binary set X with a greyscale image I [12],

$$I \ominus B = \bigwedge_{b \in B} I_{-b}. \tag{2.14}$$

This technique for implementing erosion is demonstrated in Figure 2.6 using 1D intensity profiles. As was the case in Figure 2.5, 1D profiles have been used to illustrate this process where the SE, as before, is a three pixel wide horizontal line, with its origin in the centre.

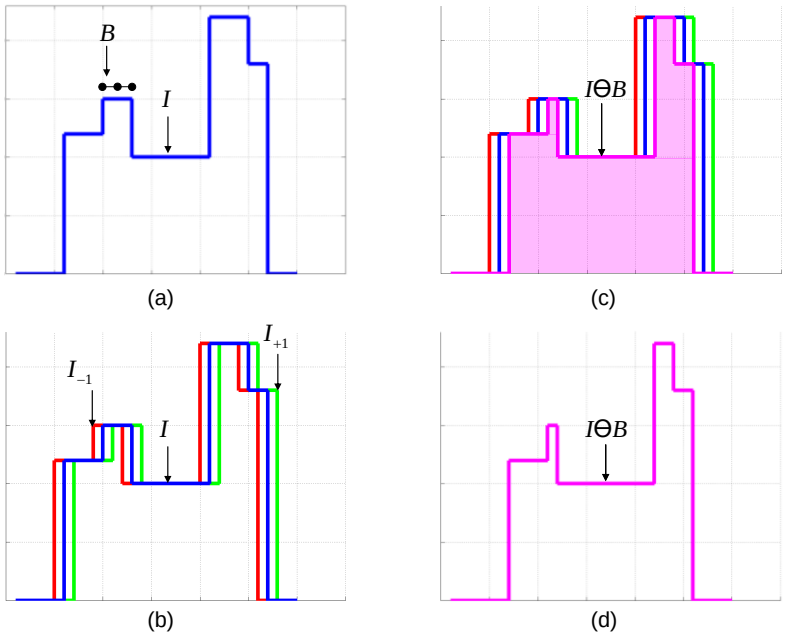


Figure 2.6 Example of greyscale erosion implemented by translating the image by the points in the SE and computing the infimum of all translates. (a) Original image and SE illustration. (b) Original image and the translations of this image by the negative vectors of the SE. (c) Result of erosion where the shaded magenta profile highlights the infimum of all image translations. (d) Final eroded result.

It is also possible to compute a greyscale erosion by translating the SE to all points in the image, and writing in the output image, the minimum value of the pixels that coincide with the SE when its origin is coincident with some point $x \in E$. That is,

$$[I \ominus B](x) = \min_{b \in B} \{ I(x+b) \}. \tag{2.15}$$

By reference of Figure 2.7, it is clear that erosion removes isolated bright points and generally darkens the image. This is extremely apparent by comparing the eyes, and the scarf attached to the hat, in the images shown in Figure 2.7(a) and Figure 2.7(b).

(a) (b)

Figure 2.7 Example of greyscale erosion, (a) Original image (512×512) and (b) the result of eroding the image shown in (a) with a small, flat 3x3 square SE.

Until now, only flat SEs have been used in the definitions of erosion that have been provided. If non-flat SEs are used, then Equation (2.15) can be modified to account for the greylevels in the SE. That is,

$$[I \ominus B_{NF}](x) = \min_{b_{NF} \in B_{NF}} \left\{ I\left(x + b_{NF}\right) - B_{NF}\left(b_{NF}\right) \right\}, \tag{2.16}$$

where, for all points x with which the SE coincides, the greyscale values of the SE are subtracted from the corresponding image pixels before the minimum of the resulting values is computed.

Since only flat SE are used in the applications of the transform that is proposed in this work, a further discussion and/or demonstration of greyscale morphology, using non-flat SEs, is beyond the scope of this thesis. However, a number of the morphological transforms that are defined in this section, and throughout this thesis, are given using both flat and non-flat SEs for completeness.

2.2.3 Dilation

Dilation is the dual of erosion with respect to complementation. It can be described as a process of translating a SE to every point in *X*, and marking in the output image, points in *X* where the SE touches an object in the foreground. The dilation of *X* by *B*, denoted $X \oplus B$, may be written as,

$$X \oplus B = \left\{ x \middle| \left(\hat{B} \right)_x \cap X \neq \varnothing \right\} \tag{2.17}$$

where $\hat{B}$ denotes the reflection of the SE *B*. Some researchers chose to denote the dilation of *X* by *B* using $\delta_B(X)$, however, the $X \oplus B$ notation is used here. An example of dilation is shown in Figure 2.8 using the same image that was eroded in Figure 2.4.

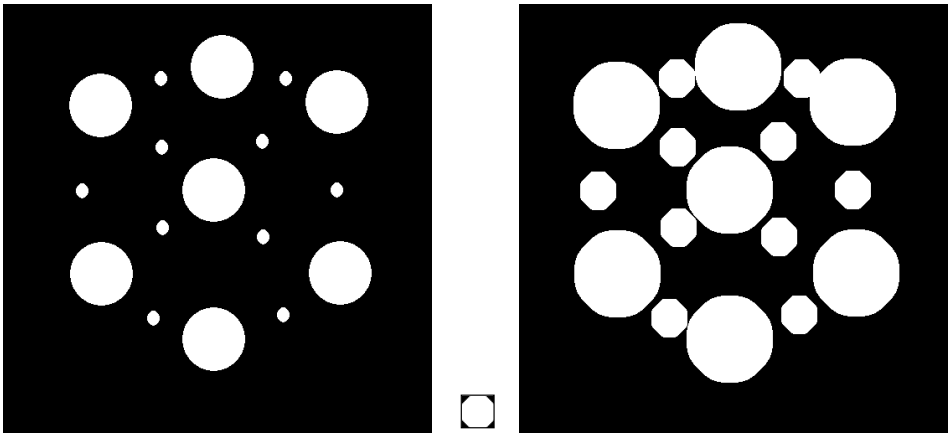


Figure 2.8 Example of binary dilation (a) Original image (512×512), (b) SE used to dilate the image shown in (a), (c) Result of dilating the original image (a) by SE (b).

It is clear by observation of Figure 2.8 that dilation grows/expands image features. It is also clear that the shape of the SE distorts the shape of the features in the dilated image, where the smaller circles in Figure 2.8(c) appear more hexagonal than anything else. This effect can also be observed on the larger circles in the image, however, the extent of the distortion is not quite so obvious on these features. Since it expands image features, dilation is capable of connecting regions that are

unconnected in the original image. This can be observed in the top right of Figure 2.8(c) where a small circle and a large circle have expanded so much that they are connected following dilation. This may or may not be a desired effect when computing the dilation of an image. As such, care should be taken when designing the SE that is used for a particular application.

An alternative method that can be used to implement a morphological dilation is to calculate the union of all translations of the set X by the positive vectors of B,

$$X \oplus B = \bigcup_{b \in B} X_b. \tag{2.18}$$

This definition of dilation is equivalent to Minkowski addition. An illustrative example of how this method works is shown in Figure 2.9. The example in Figure 2.9 shows how the image can be translated before computing the dilation by calculating the union of all image translates. The SE that has been used in Figure 2.9 is a three pixel wide horizontal line with its origin in the centre. It is shown in Figure 2.9(a).

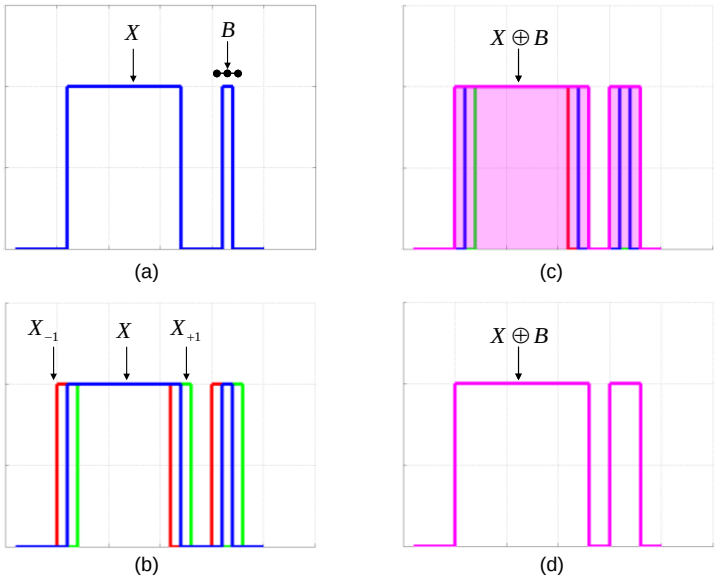


Figure 2.9 Example of binary dilation implemented by translating the image by the points in the SE and computing the union of all translates. (a) Original image and SE illustration. (b) Original image and the translations of this image by the positive vectors of the SE. (c) Result of dilation where the shaded magenta profile highlights the union of all image translations. (d) Result of dilating *X* by *B*, both features in the original image have become wider following dilation.

The definition of dilation that is given in Equation (2.18) can be directly extended for processing greyscale images by replacing the union operator with the supremum (sup.) and the binary set X with a greyscale image I,

$$I \oplus B = \bigvee_{b \in B} I_b. \tag{2.19}$$

Using Equation (2.19) to implement dilation is demonstrated in Figure 2.10 where 1D profiles have been used to represent the image and the SE is a three pixel wide horizontal line with its origin in the centre.

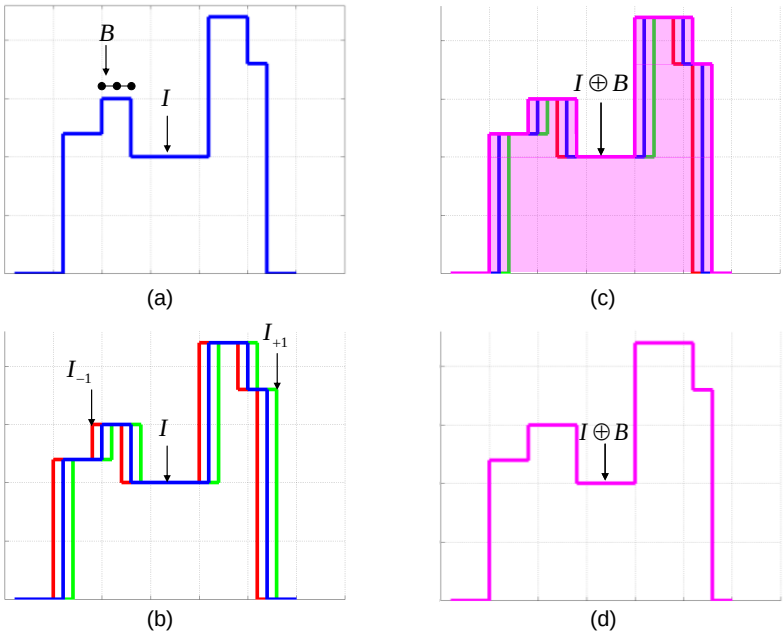


Figure 2.10 Example of greyscale dilation implemented by translating the image by the points in the SE and computing the supremum of all translates. (a) Original image and SE illustration. (b) Original image and the translations of this image by the positive vectors of the SE. (c) Result of dilation where the shaded magenta profile highlights the supremum of all image translations. (d) Final dilated result.

It is also possible to compute a greyscale dilation by translating the SE to all points in the image, and writing in the output image, the maximum value of the pixels that coincide with the SE when its origin is coincident with some point $x \in E$. That is,

$$\left[I \oplus B\right](x) = \max_{b \in B} \left\{ I\left(x - b\right) \right\}. \tag{2.20}$$

By reference of Figure 2.11, it can be seen that dilation enhances/expands isolated bright points and generally brightens the greyscale image. This is particularly clear by close observation of the eyes, the scarf extending from the hat, and the bright spots that can be seen in the right side of the image shown in Figure 2.11(b).

(a) (b)

Figure 2.11 Example of greyscale dilation, (a) Original image (512×512), and (b) the result of dilating the image shown in (a) with a small, flat 3x3 square SE.

If non-flat SEs are used to dilate an image, then Equation (2.20) can be modified to account for the greylevels assigned to the points in the SE. That is,

$$\left[I \oplus B_{NF}\right](x) = \max_{b_{NF} \in B_{NF}} \left\{I\left(x - b_{NF}\right) + B_{NF}\left(b_{NF}\right)\right\}, \tag{2.21}$$

where for all points x that coincide with the SE, the greyscale values of the SE are added to the corresponding image pixels before the maximum of the resulting values is computed.

2.2.4 Opening

There exists no inverse transform to the elementary morphological operations of erosion and dilation [12]. This means that following a morphological erosion it is, in general, impossible to exactly restore the original image from the eroded result. Clearly, any points that are completely removed by the erosion can never be recovered (See Figure 2.4). Further, the exact geometry of features that have been shrunk by the erosion cannot always be accurately be recovered. One approach that can be used to recover as much as possible, is to dilate the eroded image in an attempt to restore the features that have been shrunk in the eroded result. This technique is known as opening.

A morphological opening is defined as the erosion of an image, followed by the dilation of the result, using the same SE. Mathematically, an opening may be written in terms of the appropriate (binary or greyscale) definitions of erosion and dilation that are given in Section 2.2.2 and Section 2.2.3,

$$I \circ B = (I \ominus B) \oplus B, \tag{2.22}$$

where $I \circ B$ denotes the opening of an image I by the SE B.

Opening tends to smooth contours in the image, remove isolated bright points, and break tenuous connections between regions in the image [16]. An example of using opening to separate elliptical features in a binary image is shown in Figure 2.12. The image is a synthetic example where the features of interest (elliptical regions) have been connected by narrow fibres.

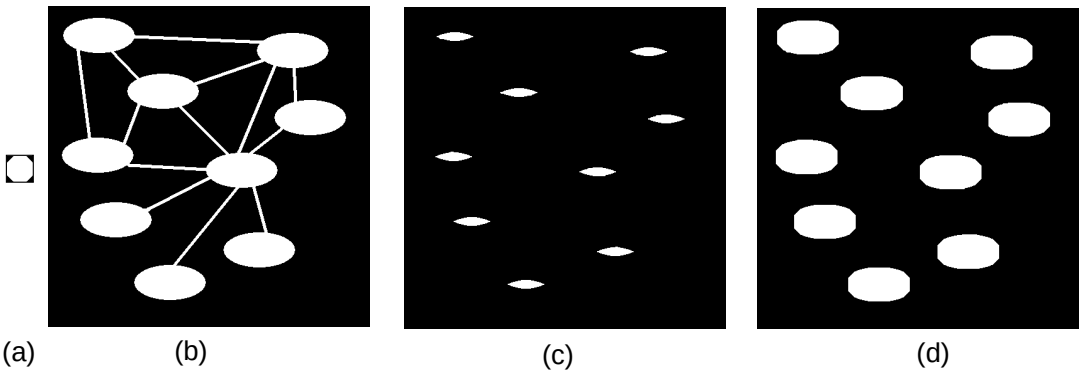


Figure 2.12 Example of opening. (a) Disk SE used for the erosion and dilation (opening). This SE can be included in the elliptical features of interest, but is too large to fit inside the fibres. (b) Original image (512×512) which is to be opened (c) Erosion of the image shown in (b) using the SE shown in (a). (d) Dilation of (c) – Opening of (b) – using the SE shown in (a).

By reference of Figure 2.12 it is clear that the morphological opening has successfully isolated the features of interest. If nothing more, the task of accurately counting the individual features in the image is now trivial following the opening. Clearly, there are 9 elliptical features in the image. However, prior to applying the morphological opening, any simple counting algorithm would have counted only one connected component in the image shown in Figure 2.12(b).

It should be noted, that although the features of interest have been restored to almost their original size by dilating the eroded image, much of the shape information has actually been lost in the opening. The results of the opening could be improved by using a smaller SE. A smaller SE would not distort the elliptical shape of the features of interest quite so much, but it would still be capable of removing the fibres that connect the features in Figure 2.12(b). An alternative method (in this idealised case) would be to use the feature itself as a SE – this would actually guarantee that the features of interest could be exactly reconstructed, without distortion, while still removing the fibres. The SE that has been used in Figure 2.12 is clearly not optimal for this problem, however, it has been used to allow this discussion of opening and SE selection.

2.2.5 Closing

Morphological closing is the dual of opening with respect to complementation. Closing is defined as a dilation of an image, followed by an erosion of the dilated image, where the same SE is used for both operations. Mathematically, closing may be written,

$$I \bullet B = \left(I \oplus B\right) \ominus B \tag{2.23}$$

where, $I \bullet B$ denotes the closing of *I*, by SE *B*. Any of the definitions of erosion and dilation that are provided in Section 2.2.2 and Section 2.2.3 may be substituted into Equation (2.23) provided that comply with the image data that is being processed.

Closing tends to smooth image contours, remove holes in the image foreground, and to join any disconnected regions in an image [16]. A simple example that uses closing to reconnect lines that have been broken as a result of thresholding is shown in Figure 2.13.

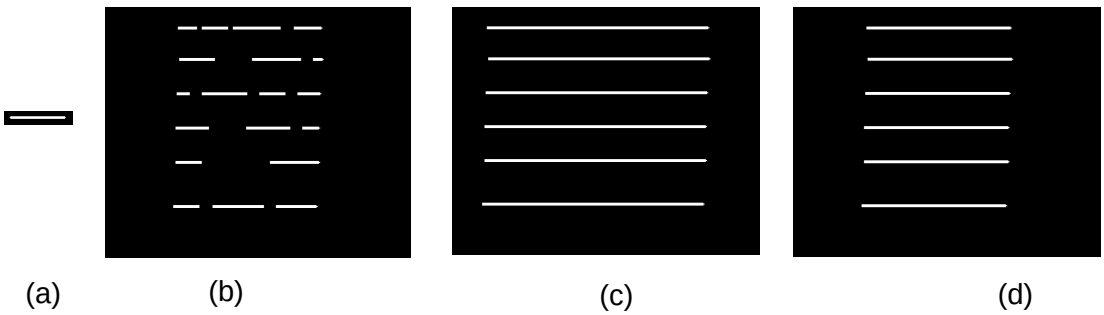


Figure 2.13 Example of image closing. (a) Horizontal line SE used for the dilation and erosion (closing) which can be used to reconnect the broken lines in the image that have been introduced by thresholding. (b) Original image (512×512) which is to be closed (c) Dilation of the image shown in (b) using the SE shown in (a) to connect the lines. (d) Erosion of (c) – Closing of (b) – using the SE shown in (a). The morphological closing reconnects the lines and restores them to their original width.

Since there is no noise in Figure 2.13 and there are no features present other than fragments of the broken lines, the problem is a simple one that can be solved by closing. The horizontal line SE was chosen to be wider than the greatest gap between the line fragments. This means that the dilation can successfully close the gaps with

the minor drawback that the width of the line is extended after dilation. Eroding the dilated image to complete the closing simply restores the lines to their width in the original image.

2.2.6 Open-Close and Close-Open Operations

Since only binary examples have been used to demonstrate the properties of opening and closing, this subsection has been included to demonstrate a useful application of opening and closing on greyscale images. However, instead of demonstrating these separately, as was done previously, this example shows that an open-close operation, and a close-open operation, can be used to filter out noise in greyscale images. The open-close operation is simply an opening of the image followed by a closing of this result. The close-open operation is a closing of the image followed by an opening of the result. An example of these operators filtering out noise in a greyscale image that has been corrupted by salt and pepper noise is shown in Figure 2.14.

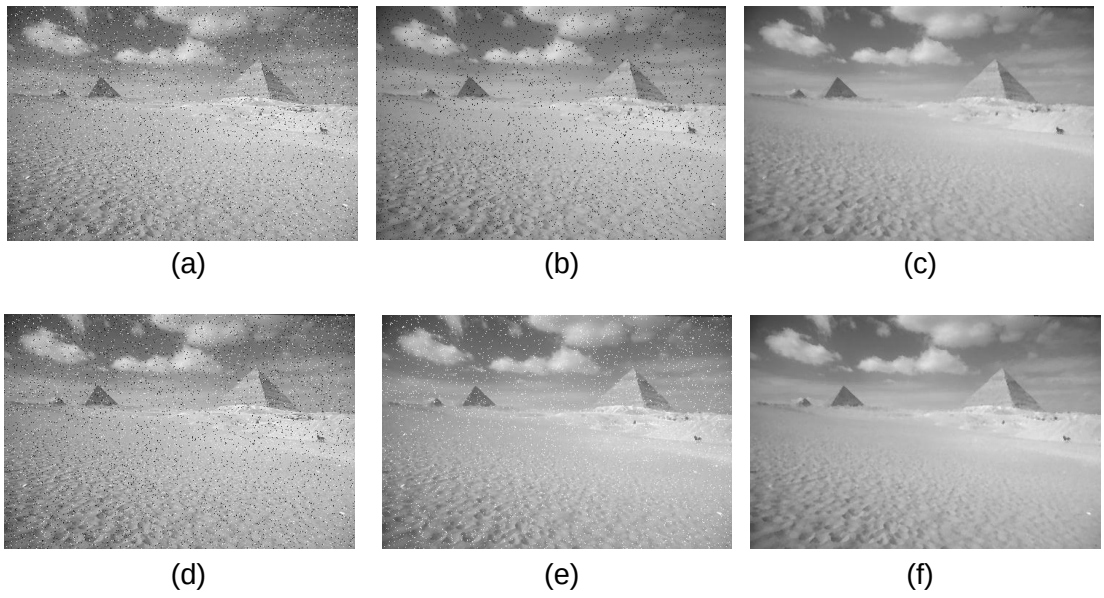


Figure 2.14 Example of using open-close and close-open operators to filter noise using the elementary diamond SE (See Figure 2.3). (a) Noisy image (321×481) (b) Opened image – salt noise suppressed. (c) Open-Close image – salt and pepper noise has been suppressed by first opening and then closing the image. (d) Noisy image (321×481). (e) Closed image – pepper noise suppressed. (f) Close-Open image, salt and pepper noise removed.

Clearly both of these methods perform well in reducing the noise. The elementary 5 point diamond SE (resembling a cross as shown in Figure 2.3) was used here. This allowed the impulsive noise points to be attenuated without overly affecting the

detail and texture that is contained in the image itself. Filtering the image in this way exploits the fact that opening and closing can be used to remove respectively bright and dark points while attempting to restore the original content of the image based on the neighbourhood of pixels that is defined by the SE.

2.2.7 Properties of the Fundamental Morphological Operators

In this section, some of the properties of the four morphological transforms that have been discussed to this point are detailed.

Erosion and dilation are dual operators with respect to complementation and reflection. That is,

$$(I \ominus B)^c = I^c \oplus \hat{B}. \tag{2.24}$$

It follows that,

$$(I \oplus B)^c = I^c \ominus \hat{B}. \tag{2.25}$$

Further to these being dual transforms, both erosion and dilation are translation invariant, increasing transforms. The obvious difference between the two is that dilation expands image features while erosion shrinks them. However dilation is not an extensive transform and erosion is not anti-extensive. Consider the case that an image is eroded by a SE which does not contain the origin. In this case, points which are not present in the original image may be introduced the eroded result.

Opening and closing are dual operators with respect to complementation and reflection. That is,

$$\left(I \circ B\right)^c = \left(I^c \bullet \hat{B}\right), \tag{2.26}$$

and

$$\left(I \bullet B\right)^c = \left(I^c \circ \hat{B}\right). \tag{2.27}$$

Like erosion and dilation, opening and closing are translation invariant, increasing transforms. Closing is extensive and opening is anti-extensive. A further property of opening and closing is idempotence, where applying an opening or closing more than once has no further affect on the result.

The final observation of these four morphological operators involves the order relation that exists between the transforms. The following order relation of the four morphological operators that have been discussed here is shown below with respect to image *I*,

$$I \ominus B \;\leq\; I \circ B \;\leq\; I \;\leq\; I \bullet B \;\leq\; I \oplus B. \tag{2.28}$$

where $F \leq G$ means that for every point in *G*, the corresponding pixel in *F* is always less than or equal to the intensity of the same point in *G*.

2.3 The Hit-or-Miss Transform

The Hit-or-Miss Transform is defined in this section for application to binary images. Extending the binary HMT for processing greyscale images is not a trivial task, and as a result, it has been defined by a number of researchers in recent years. These definitions all differ from each other in their implementation; however, they all return equivalent results. Since a new conceptual greyscale HMT is introduced in this thesis, it is appropriate that a discussion of the greyscale HMTs and the provision of their various definitions is deferred until Chapter 4. The new conceptual HMT, presented in this thesis is defined in Chapter 4 and hence for now only the standard binary HMT, [2] is described.

The HMT is a morphological transform that, among other things, can be used for template matching. Until this point, the morphological operators and transforms that have been discussed have used only one SE. The HMT uses two SEs in order to match features in a given image. One SE, denoted B_{FG}, is known as the foreground SE which searches the image for places where it can be entirely contained in the image foreground. A second SE, denoted B_{BG}, is known as the background SE and it searches for places in the image where it can fit around features to match their background neighbourhood. The HMT then marks the locations where both the foreground SE *and* the background SE simultaneously fit the image to find a match.

Mathematically, the HMT of a binary image X is the intersection of an erosion of X and an erosion of the complement of X by a complementary pair of SEs, B_{FG} and B_{BG} respectively, where X, B_{FG} and B_{BG} are sets in 2D space, $E = \mathbb{Z}^2$. B_{FG} and B_{BG} are defined relative to a common origin in E where the composite SE $B = B_{FG} \cup B_{BG}$ and $B_{FG} \cap B_{BG} = \varnothing$. That is,

$$HMT_B(X) = (X \ominus B_{FG}) \cap (X^c \ominus B_{BG}) \tag{2.29}$$

A feature is detected by the HMT if there is at least one point $x \in E$ such that the foreground SE $(B_{FG})_x$ is included in X whilst the background SE $(B_{BG})_x$ is simultaneously included in its complement, where $(B)_x = \{b + x \mid b \in B\}$, see [2], [12], [17], [18]. A diagrammatic example of the HMT locating features in a binary image is shown in Figure 2.15. Note that this is simply an illustration to demonstrate the operation of HMT and that in this example the foreground pixels are grey on a white background for convenience when printing.

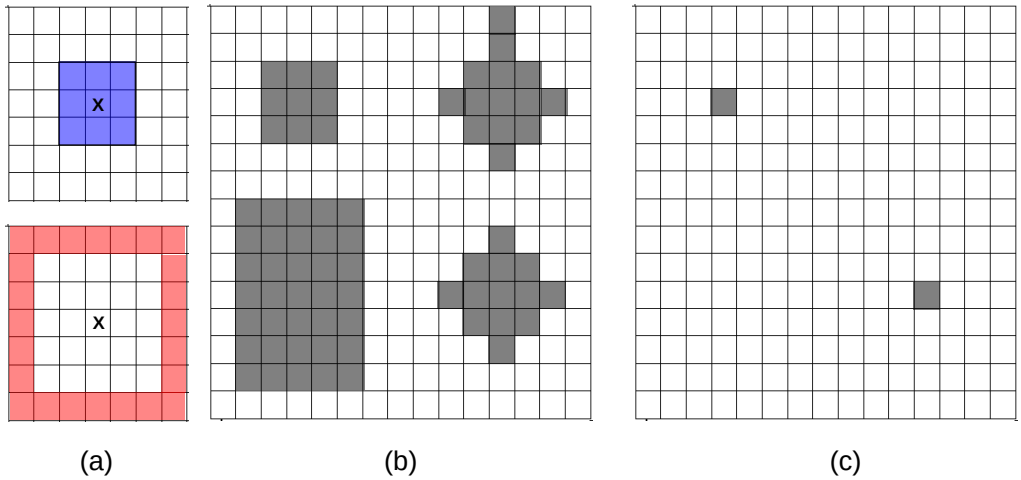


Figure 2.15 Illustrative example of the HMT. (a) Complementary SE, B_{FG} (top, blue) and B_{BG} (bottom, red) are defined relative to a common origin in the centre of B_{FG}. (b) Image that is being processed. (c) Marker produced by the HMT that contains the locations where both B_{FG} and B_{BG} simultaneously fit the image.

The HMT returns a "marker" (See Figure 2.15(c)) consisting of single pixels or groups of pixels indicating the presence and locations of the objects that have the features specified by *B*. Any feature whose geometry matches that of both SEs will be marked in the output of the HMT. If the SEs are thought of as a combined, composite entity, then any solid feature whose boundary lies between that of the foreground and the background SE will be detected by the HMT. This idea is demonstrated in Figure 2.16.

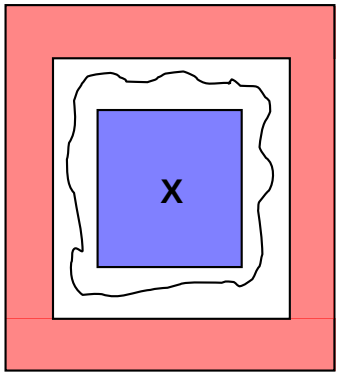


Figure 2.16 Composite SE created by combing the foreground and background SE to detect features of interest. Any solid feature, like the arbitrarily shaped object here, whose boundary lies between that of the foreground SE and background SE will be detected by the HMT.

The HMT and its extension for the processing of greyscale data is treated thoroughly in Chapter 4 where a novel extension of this transform is presented.

2.4 Morphological Reconstruction

Strictly speaking, this branch of morphology is generally addressed under the heading of geodesic operations [12]. Unlike the morphological operations that have been covered to this point, geodesic operators use two input images. Morphological operations are applied to one of the images, while the other is used to constrain the transform that is being applied to the first image. An extensive discussion of geodesic morphology is beyond the scope of this thesis, however, it is required that a definition of reconstruction by dilation is provided here. This is essential as reconstruction by dilation is used to reconstruct image features throughout this thesis, particularly in Chapter 4 where it is used to accurately reconstruct features that are detected by the extension of the HMT that is discussed there.

To facilitate the explanation of reconstruction by dilation, geodesic erosion and dilation are introduced and demonstrated by example. Reconstruction by dilation is then defined and explained by example. Although the definition of reconstruction by erosion is provided for completeness, it is not exemplified or used again in this work. For a thorough explanation of this branch of morphology see [12], [16].

2.4.1 Geodesic Erosion

Geodesic erosion requires two input images. Generally, the image to which morphological operations are iteratively applied is called the marker, while the other image, known as the mask, constrains the transform that is being applied iteratively to the marker. Let F denote the marker and G denote the mask. It is required that the support of $F, \text{supp}(F)$, is equal to the support of G, $\text{supp}(G)$ and that $\forall x \in F$ and $\forall x \in G$, $F \geq G$. Geodesic erosion of size u applied to image F, denoted $E_G^u(F)$, may now be defined,

$$E_G^{(u)}(F) = E_G^{(1)}\left[E_G^{(u-1)}(F)\right] \tag{2.30}$$

where u denotes the iteration step and,

$$E_G^{(1)}(F) = (F \ominus B) \vee G \tag{2.31}$$

Note that $E_G^{(0)}(F) = F$. An example of the technique for computing $E_G^{(1)}(F)$ is shown in Figure 2.17.

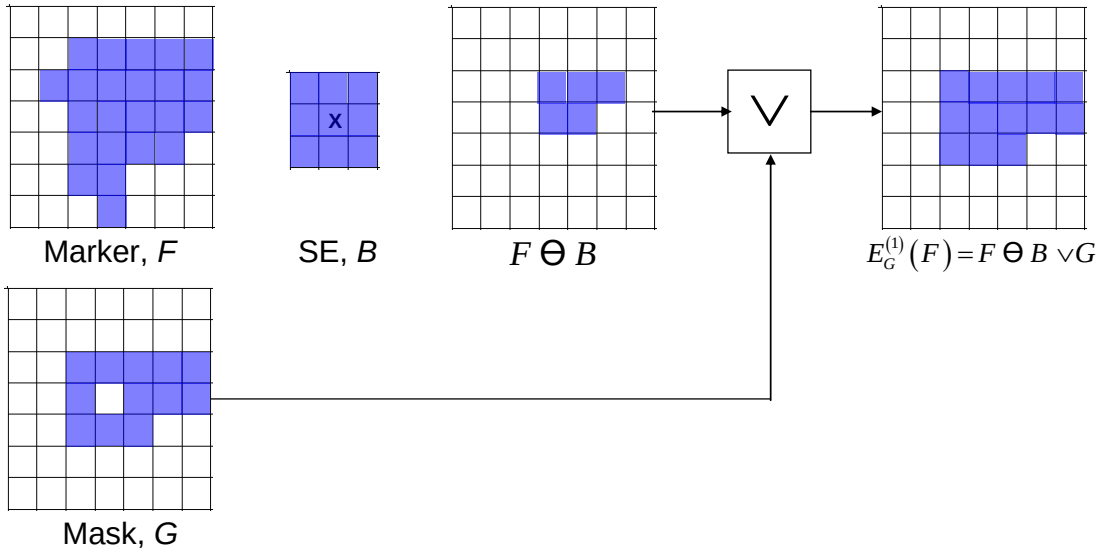


Figure 2.17 Illustration of computing a geodesic erosion

It is clear that the mask used in the geodesic erosion restricts the erosion in such a way that it is not possible to erode any pixels that are present in the mask. No matter how many times the geodesic erosion is performed, the result will always be greater than or equal to the mask. This is clear by interpretation of Figure 2.17.

2.4.2 Geodesic Dilation

Like geodesic erosion, geodesic dilation requires a marker, *F*, and mask, *G*. It is still required that the support of *F*, $\text{supp}(F)$, is equal to the support of *G*, $\text{supp}(G)$. However, for geodesic dilation, it is required that $\forall x \in F$ and $\forall x \in G$, $F \leq G$. Geodesic dilation of size *u* of image *F*, denoted $D_G^u(F)$, may now be defined,

$$D_G^{(u)}(F) = D_G^{(1)}\left[D_G^{(u-1)}(F)\right], \tag{2.32}$$

where *u* denotes the iteration step, and

$$D_G^{(1)}(F) = (F \oplus B) \wedge G. \tag{2.33}$$

Note that $D_G^0(F) = F$. The technique for computing $D_G^{(1)}(F)$ is illustrated in Figure 2.18.

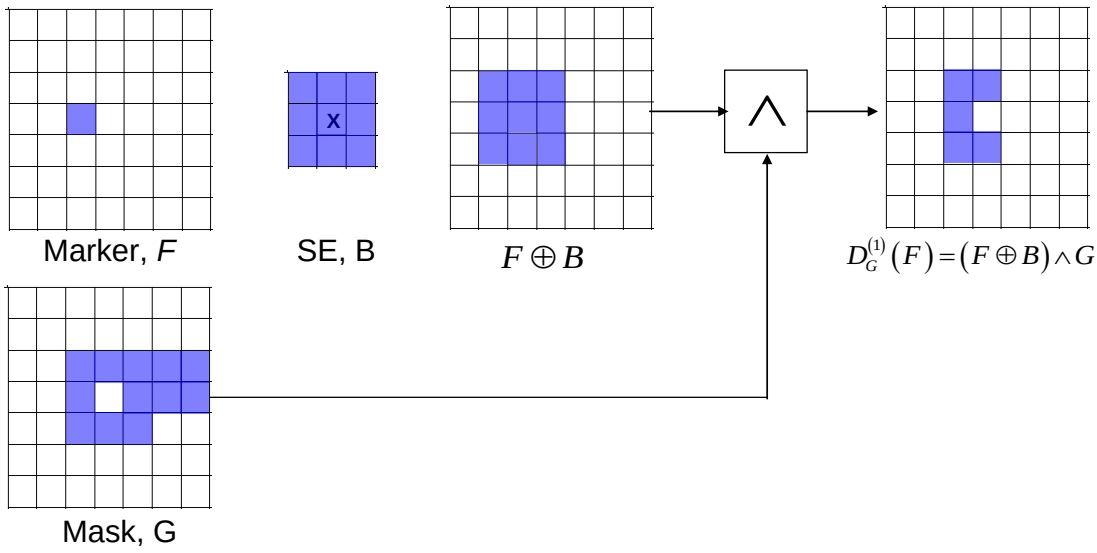


Figure 2.18 Illustration of computing a geodesic dilation

The extensivity of the elementary dilation is restricted by the mask when geodesic dilation is performed. No matter how many times the geodesic dilation is performed using the same marker and mask, there is no point at which the result will ever be greater than the mask. This is clear if the illustration shown in Figure 2.18 and the definitions of geodesic dilation given in Equation (2.32) and Equation (2.33) are fully understood.

2.4.3 Reconstruction by Dilation

Reconstruction by dilation, denoted $R_G^D(F)$, iteratively applies geodesic dilation to the marker image until its application has no further effect on the result. Reconstruction by dilation, by this definition, is idempotent. Mathematically, the reconstruction by dilation may be written,

$$R_G^D(F) = D_G^{(u)}(F), \tag{2.34}$$

where $D_G^{(u)}(F)$ denotes a geodesic dilation which is iterated until $D_G^{(u)}(F) = D_G^{(u-1)}(F)$.

In words, the first step for computing a reconstruction by dilation is to compute $D_G^{(1)}(F)$ using Equation (2.33). To compute $D_G^{(2)}(F)$, Equation (2.32) should be used, where the marker is now $D_G^{(1)}(F)$ that was computed in the previous step, see Figure 2.19. Next, $D_G^{(3)}(F)$ may be computed using Equation (2.32), where the marker is now the image $D_G^{(2)}(F)$, that was computed in the previous step. This process continues by iteratively incrementing *u* until $D_G^{(u)}(F) = D_G^{(u-1)}(F)$. In order to determine this convergence point, a comparison of the current result, and the previous one must be performed. This comparison may be implemented by computing the difference image, $D = D_G^{(u)}(F) - D_G^{(u-1)}(F)$. If $D = \varnothing$ then $D_G^{(u)}(F) = D_G^{(u-1)}(F)$. An illustrative example of performing reconstruction by dilation to fully reconstruct the feature that was shown in the mask of Figure 2.18 is shown in Figure 2.19.

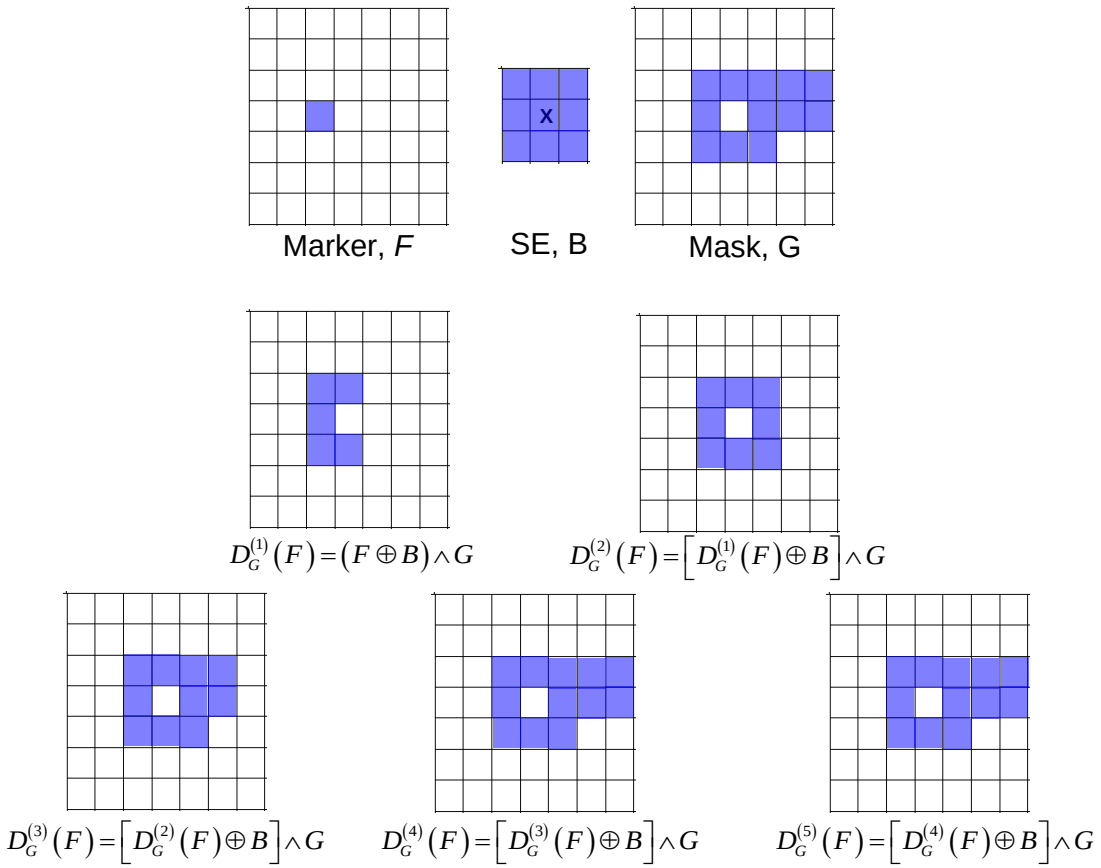


Figure 2.19 Illustration of reconstruction by dilation

The size of the reconstruction by dilation that is illustrated in Figure 2.19 is 4, as it requires four geodesic dilations before convergence. The convergence is detected when $D_G^{(5)}(F)=D_G^{(4)}(F)$ and hence there is no point in applying the transform any further. It is clear by reference of Figure 2.19 that this is the case.

2.4.4 Reconstruction by Erosion

Reconstruction by erosion, denoted $R_G^E(F)$, iteratively applies geodesic erosion to the marker image until its application has no further effect on the result. Reconstruction by erosion, like reconstruction by dilation, is idempotent. Mathematically, the reconstruction by erosion may be written,

$$R_G^E(F) = E_G^{(u)}(F), \tag{2.35}$$

where $E_G^{(u)}(F)$ is a geodesic erosion iterated until $E_G^{(u)}(F) = E_G^{(u-1)}(F)$.

Since this transform is not used in this thesis, and since it is clear from the definition of reconstruction by dilation how this transform may be implemented, a further discussion has been neglected here. The interested reader is referred to [12] for a more detailed explanation of this transform.

2.5 Rank Order Filters and Mathematical Morphology

A number of approaches that aim to extend the HMT in order to make it more robust to noise use rank order filters in place of traditional morphological operations. The extension of the HMT that is presented in this thesis may be implemented using rank order filters. In fact, the fast algorithm that is presented in Chapter 6 directly exploits a number of properties of rank order filters in order to make the routine more efficient. For this reason, a brief overview of rank order filters is provided here. A formal definition that can be used to compute the output of any rank order filter is then given before the relationship that exists between rank order filters and mathematical morphology is discussed and demonstrated.

Rank order filters are a set of non-linear filters that are commonly used to pre-process image data, although they can be designed to perform more complex tasks. The output of a rank order filter, of rank *k*, at a point *x* in an image, may be computed in two steps. First, it is necessary to sort into ascending order, the image pixels that are coincident with a window, *W*, when it is centred on a point, *x*, as it scans the image. The value assigned to point *x* in the output image is then the value of the *k*th order statistic of the image pixels that are coincident with *W* when it is centred on *x*. For example, let $p_1, p_2 \ldots p_m$ represent a set of arbitrary pixel intensities that are coincident with some window, *W*, where $m = Card(W)$ i.e. the cardinality of the set *W*. These values must first be sorted into ascending order such that,

$$p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}\,, \tag{2.36}$$

then $p_{(k)}$ represents the *k*th order statistic [19]. The output of, $\zeta_{W,k}$ a rank order filter, of rank *k* with window *W*, when centred at a point *x*, is the value $p_{(k)}$. For a point *x*, in an image *I*, the output of the rank order filter is given by,

$$\left[\zeta_{W,k}(I)\right](x) = k\text{th order}\underset{w\in W}{\text{statistic}}\left\{I(x+w)\right\}. \tag{2.37}$$

Perhaps the best known and most commonly used rank order filter is the median filter [19] which is often used as a pre-processing step in image analysis to remove/reduce noise whilst preserving edges. In this case the output of the rank order filter is simply the value that lies in the median (or centre) rank.

The relationship between morphological operators and rank order filters is well documented in the literature, [20], [21], [22]. By recognising that the window, *W*, in Equation (2.37) is equivalent to a morphological SE, *B*, i.e. $W = B$, it is clear that standard morphological erosions and dilations are in fact a special case of rank order filters. This relationship is discussed in detail in [12] where it is shown that the minimum rank filter, where *k* = 1, using a window (SE) *B*, is equivalent to an erosion, $I \ominus B$. It is also shown that the maximum rank filter, where *k* = *m*, using *B*, is equivalent to a dilation, $I \oplus B$. That is,

$$I \ominus B = \zeta_{B,1} \tag{2.38}$$

and

$$I \oplus B = \zeta_{B,m}\,. \tag{2.39}$$

This relationship allows the implementation of erosions and dilations using techniques for fast rank filtering (see Chapter 6). Further, this relationship makes it possible to relax morphological transforms by replacing erosions and dilations with rank order operators where *k* is not restricted to be either 1 (min filter) or *m* (max

filter) and can in fact take on any rank between these values, i.e. $1 \leq k \leq m$. This can be useful when image data is noisy as discussed in Chapter 4.

2.6 Summary

This chapter has laid out the notation that is used throughout this thesis and it has defined a number of elementary morphological transforms that will be used and/or developed in this work. It also provides an insight into the relationship between rank order filters and morphological operators which is critical for understanding the extensions of the HMT that are presented later in this thesis. Recognising this relationship is also critical for understanding the fast algorithm that is used to implement the extension of the HMT that is proposed in Chapter 4. Efficient techniques for computing the output of the morphological transforms that have been presented here are described briefly in the next chapter, and in Chapter 6, where an efficient technique is described that can be used to compute the output of any rank filter defined by any arbitrarily shaped window.

This chapter has also provided a number of sources where further information can be found on all aspects of morphology. The terms and techniques that have been defined here will be used to facilitate the discussion of the state-of-the-art research of the Hit-or-Miss Transform in the next Chapter.

3 The Hit-or-Miss Transform and Efficient Algorithms for Computing Morphological Operators: A Review of the Relevant Literature

This literature review is divided into two main sections. The HMT is addressed first in Section 3.1, where a number of techniques that have been used to modify the original definition of this transform are discussed and described. Section 3.1 aims to bring the reader up to speed with the state of the art research in this specific area of mathematical morphology and its applications. Then, in Section 3.2, a review of a number of methods for improving the efficiency of computing morphological transforms, and other more general rank order filters, is provided. In addition to describing a number of efficient algorithms for computing the output of these filters, this discussion introduces two specific techniques that can be combined to produce the fast algorithm that is described in Chapter 6. Further, it should be noted that the techniques which are summarised in Section 3.2 can be used to improve the efficiency of each of the morphological transforms that were discussed in the previous chapter.

The intention of this literature review is to place the contributions of the next three chapters of this thesis in context with the leading research in this area. Although a large number of contributions and methods are summarised here, the specific transforms which are closely related to work that is presented in this thesis will be described in more detail at the appropriate places in the subsequent chapters.

3.1 The Hit-or-Miss Transform

The HMT has been around for more than 40 years and in this time a great number of researchers have used and modified its original definition, in various ways, for their own specific purposes. This chapter takes the reader through a number of significant developments of the HMT. The discussion begins with the original definition of the HMT for application to binary images. Its extension for processing greyscale images

is then discussed, before a number of generalisations which have been proposed in order to make this transform more robust in the presence of noise are reviewed. Throughout the discussion in this section, it becomes clear that there exists a conceptual gap in the literature where authors have neglected to provide suitable methods that can be used to accurately determine optimal values for the parameters that are introduced to generalise the HMT.

3.1.1 The Binary HMT

Since erosion, and hence dilation, can be implemented as a special case of the HMT, this transform is described as "the universal morphological transform upon which the theory of Mathematical Morphology is based" in [23]. As such, for the processing of binary images, the HMT is well defined. It involves searching an image for locations where a pair of predefined templates simultaneously fit the image foreground, and its surrounding background, as defined by the templates. These templates, known as structuring elements in morphology, are designed to match the shape and size of features of interest in the foreground and background of the image (See Chapter 2, Section 2.3). If the SEs are designed to closely match the geometry of the image features that are of interest, just one noisy pixel in either the foreground or background, can cause this transform to fail, since the SEs will no longer fit as a result of the noise.

As will be explained in Section 3.1.3, a number of researchers have proposed techniques that aim to circumvent this issue.

3.1.2 The Greyscale HMT

Unlike most morphological transforms, extending the HMT for processing greyscale images is not straightforward. The reason for this, is that the HMT is not an increasing transform [4], [8], [12], [24]. As a result, a number of researchers have made this extension independently, and hence a number of different definitions of the

greyscale HMT exist in the literature. These are described briefly in this section, however, a much more thorough discussion of these different HMTs, and their equivalences, can be found in [8], [24], and [25], and in Chapter 4 of this thesis.

In 1996, two a greyscale HMTs were presented in the literature [26], [27]. The first of these was presented in June of that year by Khosravi and Schafer [26]. In [26], the authors implement their greyscale HMT using a single SE with which they erode the image and its negative (by the negative SE) before computing the superposition of these two results. Points in the resulting image, whose value is 0, become the points of this greyscale HMT.

The second greyscale HMT, published in September of that year, was defined by Ronse [27]. This greyscale HMT uses a foreground and background SE. It marks points in an image where the foreground SE can be raised higher than the level to which the background SE can be lowered when the origins of the SEs are centred at some point in the image. The HMT proposed by Ronse is possibly easier to understand - given an understanding of the binary HMT – than the one presented in [26], since it resembles the binary HMT to some extent, and it is therefore more intuitive to visualise. From now on, Ronse's HMT will be denoted RHMT and the greyscale HMT proposed by Khosravi and Shafer will be written KHMT.

In 2002, two greyscale HMTs were proposed by Soille in [28], and these were also explained in [12]. The greyscale HMTs were named the unconstrained HMT (UHMT) and the constrained HMT (CHMT) in [28]. Intuitively, the latter is a constrained version of the former, where the CHMT is more restrictive than the UHMT requiring that a specific fitting criterion be met by whichever SE contains the origin. For generality, and since the CHMT is not directly related to the work presented in this thesis, only the UHMT is considered from this point forward. The UHMT is very similar to the RHMT in that it searches for places where the foreground SE can be raised to a level which is higher than the lowest point that can be reached by the background SE. However, the transforms differ in the value which is assigned to points which record a "hit".

In 2003 Barrat *et al*. presented a technique which they call morphological probing [5] - this will be referred to as BHMT in this thesis. The authors state that the technique they propose for processing image data is based on the principle of mechanical profiling which is described in their paper as a technique for measuring surface height variations, using a probe. They describe this mechanical profiling as a process of passing a probe across a surface, while measuring and recording the movement of the probe as it traces the profile of the surface that is under inspection. Essentially, the technique proposed for pattern matching in images in [5] is a greyscale HMT which uses two SEs (probes as they are called in [5]) to inspect an image from above and below its surface in order to locate features of interest. Places are marked in the image when the probes come into contact with each other from above and below the topographic surface of the image. It is shown in [8], and in Chapter 4 of this thesis, that the BHMT is equivalent to the KHMT.

A unified theory for computing the output of greyscale HMTs, based mainly on the greyscale HMTs proposed by Ronse and Soille, is given in [24]. In this paper [24], the authors describe the HMTs proposed in [27] (RHMT) and [28] (UHMT), before extending the method proposed in [28] (UHMT) such that it can use greyscale SEs as opposed to flat ones. A unified theory for computing greyscale HMTs is then presented, and it is demonstrated that the value of pixels in the output image may be assigned in a number of ways.

It is shown in [8] and [25], and in Chapter 4 of this thesis, that the RHMT and the UHMT resemble each other since pixels which are marked in the output of both transforms correspond to locations where the result of processing the image with the foreground SE is greater than the result of processing the data with the background SE. It is also shown that the KHMT and the BHMT are similar since both of these transforms evaluate the distance between the SEs in order to determine whether or not a pixel should be marked in the output of the transform. It is pointed out in [8], [25], and Chapter 4, that although each of these HMTs are defined in a slightly different way, the output of each transform is equivalent. A summary of each of these HMTs is provided in Table 3.1.

Transform	Author	Description
KHMT	Khosravi and Schafer [26]	Uses a single SE to erode the image and its negative (by the negative SE). Points in the result of adding these two eroded images whose value is 0 are points of the KHMT.
RHMT	Ronse [27]	Uses a foreground SE and a background SE. Returns the result of a foreground erosion where both SEs fit the image.
UHMT	Soille [12], [28]	Uses a foreground SE and a background SE. Returns the number of levels for which both SEs simultaneously fit the image.
BHMT	Barrat *et al.* [5]	Uses a foreground SE and a background SE - called probes in [5]. When the probes come into contact with each other (from above and below the surface) at some point in the image, this point is marked in the output of the HMT.

Table 3.1 Summary of existing greyscale HMTs

Despite many definitions of the HMT that have been proposed in [5], [24], [26], [27], [28], the issues which cause the binary HMT to fail in the presence of noise, have the same effect on the greyscale HMT. If just one pixel in the foreground of the feature of interest drops below the level of the background neighbourhood (as defined by the background SE), then the SEs can be prevented from fitting. Similarly, if the intensity of just one pixel in the background surrounding a feature of interest reaches the level of the foreground, the transform will fail. As a result, a number of researchers have presented techniques that aim to extend the greyscale HMT in order to improve its robustness to noise.

In Section 3.1.3, a number of techniques that have aimed to make the HMT more robust in the presence of noise are reviewed.

3.1.3 Extensions of the HMT for Better Accuracy and Improved Robustness when Processing Noisy Data

Various researchers have introduced generalisations and extensions that make the standard HMT more robust when processing noisy data. At a high level, these extensions can be summarised as follows:

- Pre-processing is used to aid the fitting of the SEs
- The SEs are modified in some way to improve robustness to noise
- The definition of the transform is modified by relaxing the strict fitting conditions of the HMT

Some authors apply one of the aforementioned techniques to improve noise robustness, others try a few of these individually, or in some cases, authors present some combination of these techniques, in order to try and overcome the difficulties associated with using the HMT to process noisy data. Some authors consider only the binary HMT, whilst others consider only the greyscale HMT, and some authors provide methods that can be used to generalise both.

In this section a number of existing techniques will be described. The extensions will be grouped together in sections based on the methods that have been used (see above) to improve the robustness of the transform. In cases where authors have employed a combination of the aforementioned techniques, the contribution that is claimed by author (or is otherwise considered) to be the major contribution of the work will be used to determine the heading under which these extensions are described. In other cases, where authors have perhaps considered various approaches to extending the HMT, and have paid equal attention to each method, then the appropriate section of these papers is referred to under different headings in the following section. It is also noted throughout the following discussion whether or not the authors presented their extension for binary data, greyscale data, or both.

In [4], Naegel *et al*. briefly discuss some techniques that are commonly used to make the HMT perform more accurately in the presence of noise. However, the authors of [4] focus on describing algorithms that have been developed in [29], [30] and [31] which address a different problem. Therefore, further discussion of the methods described there [4], are deemed to be beyond the scope of this thesis.

3.1.3.1 Pre-processing the Image Data

Bloomberg and Maragos present a Blur Hit-Miss transform in [32] in order to help make the binary HMT more robust in the presence of noise. However, the main focus of this paper, [32], lies in using the HMT to present generalised openings and closings and, as such, many of the contributions and discussions in that paper are beyond the scope of this thesis. Only the contributions in [32] that are relevant to this body of work are presented in this thesis. It should be noted that the definitions given in [32] are for binary morphology. However, in their conclusions the authors explain how their techniques may be applied to greyscale images.

The authors propose in [32], that in contrast to the exact match between the SEs and a sought pattern that is required by the standard HMT, that their proposed blur HMT, requires two slightly less restrictive conditions to be satisfied for successful detection. The first is that there must be an ON pixel within some radius, r_1 of each "hit", and that there must be an OFF pixel within some radius r_2 of each "miss". The authors explain that this improves the robustness of the HMT when noisy pixels appear close to the border of sought features in the image.

The authors go on to explain that this idea of a blur HMT can be implemented quite easily using morphological operators. The first step is to dilate the image using an SE of radius r_1 before applying the foreground erosion of the HMT. The second step is to dilate the complement of the image using an SE of radius r_2 before applying the background erosion of the HMT. These pre-processing steps help to remove noise and make it easier for the SEs to match patterns by slightly modifying the geometry of features in a given image. While it is obvious that this method can make the

standard HMT less sensitive to noise, particularly when the borders of a sought feature are distorted, no examples or experimental results are provided to explain or support this claim. Also, no method for accurately selecting appropriate values for r_1 and r_2 are provided in [32].

More recently, Bloomberg and Vincent proposed a blur Hit-or-Miss transform and applied it to pattern detection in document images [6]. This transform was initially presented in [32] (described above), however, as the previous paragraph explained, only the definition and a very brief description of the transform was provided there. The authors point out that although rank order filters can be used to improve the performance of the HMT, the blur HMT offers a more efficient alternative as erosions and dilations can be computed using efficient Boolean operations when processing binary data. The authors also point out that a large number of researchers have focused on modifying the templates that are used in the detection process [33], [34], [35] and [36]. However, while Bloomberg and Vincent agree that modifying the templates is useful for improving the performance of the HMT in noisy images, they argue that techniques which modify the image have been overlooked.

The authors of [6] begin by defining the binary HMT before describing the blur HMT and introducing the concept of blur SEs. It is explained that the so called blur SEs can be used in one of two ways. They are either used to dilate the image, as was the case in [32] or, alternatively, they can be used to erode the templates that are used to match features in the image data. While it is explained that either approach can be taken, the authors conclude that since there is generally more salt and pepper noise in the image data than on the templates, that it is best to use the blur SEs to dilate the image. The blur HMT is therefore defined in the same way as in [32], where the first blur SE is used to dilate the image foreground, and a second blur SE is used to dilate image complement. Taking this approach provides a number of advantages: dilating the foreground using the first blur SE suppresses impulsive noise while, at the same time, it expands the foreground of the image to allow better fitting of the foreground template. Similarly, dilating the background by the second blur SE removes impulsive noise, while expanding the background regions to allow better fitting of

the background template. It will be explained in Section 3.1.3.2, how sub-sampled versions of the foreground and background templates can be used in order to increase the efficiency of the blur HMT.

Perret *et al*. in [8] use pre-processing techniques to estimate one of the parameters of their Fuzzy HMT (FHMT) and to reduce the noise in the image data which is to be processed. This allows them to estimate an appropriate level at which they should apply their SEs in order to locate the best match. In this paper [8], the authors also point out that when images are distorted by impulsive noise the performance of the HMT can often be improved by applying a rank order filter, such as the median filter [16], to clean up the image before applying the standard HMT. However, this approach is only recommended for the case of impulsive noise. Further, it requires additional computation prior to performing template matching using the HMT.

3.1.3.2 Modifying the SEs

In [33], Zhao and Daut present a technique for the detection of imperfect shapes using the HMT. The authors focus on applications of the binary HMT and state that the majority of the noise and distortion encountered in their work is introduced by the thresholding process that is used to obtain a binary image from greyscale data. The authors point out that examining the problem closely reveals that most distortions of the sought features actually occur at their boundaries.

In order to improve the performance of the HMT when matching such imperfect shapes, the authors design a set of templates which resemble the geometry of the feature that they wish to detect. They include in this set a number of variations of the sought feature which represent the distorted, imperfect shapes that they aim to match and locate in the image. These templates are then smoothed using a morphological opening, before the boundaries of these smoothed features are used to locate imperfect shapes in the image, using the HMT. It is also explained in [33] that lower and upper bounds can be determined for shape variation and hence the boundary of SEs, representing these extreme shape variations, can be used with the HMT.

The authors provide an example of their improved HMT when applied to the problem of character recognition and achieve promising results. They also point out in this paper [33] that the idea of using the boundary of the HMT templates was originally proposed in their earlier work which was presented in [37]. Using the boundary of the templates for robustness to noise has the added advantage that it significantly reduces the number of computations at every pixel to which the HMT is applied [33]. This makes the transform more efficient when compared with using the entire SE.

In [38], the same authors discuss the use of the binary HMT for detecting what they refer to as both perfect and imperfect shapes. The paper begins by thoroughly describing the technique of using the boundary of the HMT templates as an alternative to using the entire template with this transformation (see above). The authors explain that using a subset of the template allows accurate detection of the sought pattern while eliminating a large number of redundant computations that are performed when using the entire template to achieve the same goal. The problem of detecting imperfect shapes is then addressed, and the authors provide a more thorough description of the technique that is presented in [33] (and summarised above). The authors point out that using a large set of SEs which contains the possible variations in appearance of the sought template would be computationally expensive. They also explain that lower and upper bounds for the possible variation of the sought features can be determined. The boundary of SEs which represent these lower and upper bounds of shape variation can therefore be used as SEs in the HMT. This allows better matching of the templates with imperfect features, while keeping computational costs to a minimum.

Zhao and Daut demonstrate, using a number of SE pairs, that by increasing the spatial distance between the foreground and background templates, the permitted variation in the shape of the sought feature increases. This increase in the permitted shape variation improves the likelihood of successful detection when sought features have been distorted. However, this also increases the likelihood of erroneously detecting features which are not of interest (see Chapter 2, Section 2.3). The authors

also demonstrate in [38], that it is possible to use the skeleton of the sought feature, and the skeleton of its complement, as respectively the foreground, and background SEs, when applying the HMT. The authors demonstrate that this approach allows not only imperfect shapes, which have been distorted by noise to be detected, but also, that it is possible to detect features whose orientation is slightly different to that of the templates.

In [26], after describing their greyscale HMT (KHMT), Khosravi and Schafer present two techniques in an attempt to generalise this transform for improved performance in noise. One proposal is that the SEs that are used for template matching are decomposed into a number of sub-templates. These sub-templates are then applied separately to the data before the resulting images are used to compute the output of the HMT. The authors also test the performance of their greyscale HMT by replacing the strict morphological operators with rank order operators as will be discussed in Section 3.1.3.3, however, for now, only modifications to the SEs are discussed.

The authors state in [26], that their greyscale HMT can be made more robust in the presence of impulsive noise by reducing the size of the templates used for matching features in the image. They state that this fact motivated their efforts in testing the performance of the HMT using a series of sub-templates which are obtained from decomposing the original template. The authors demonstrate that decomposing the templates, and applying these independently to the data in the presence of impulsive noise before combining the results, provides a significant improvement in accuracy when compared with using the original template in a single pass of the image. They point out, that when applying this technique to data which is corrupted by impulsive noise, the greater the number of partitions of the template, the more robust is the routine.

The authors demonstrate in [26], that partitioning the template into a small number of sub-templates also provides improved performance in the presence of Gaussian noise. However, increasing the number of partitions in this case, does not necessarily mean improved performance. It is clear from the experimental results presented in

[26], that as the number of partitions increases, the results can in fact be worse (contain more errors) than when using a standard greyscale HMT with the original template and no partitioning. The authors also demonstrate, by experimentation, that when decomposing the SEs into sub-templates, the points in each sub-template should be sampled from the entire set of points in the SE. That is, the decomposition process should not be restricted in such a way that samples for each template are selected from distinct, non-overlapping regions of the template itself.

It was mentioned briefly in Section 3.1.3.1 that Bloomberg and Vincent, [6], use sub-sampled versions of their templates in order to improve the efficiency of the blur HMT. In their paper they explain that blur SEs can be applied either to the image itself, or to the templates which are used to match features in the image. In the previous discussion of the blur HMT in Section 3.1.3.1, it was noted that the authors chose to apply the blur SEs to the image. However, when considering fast implementations, the authors point out that they can actually subsample the template to reduce the number of computations per image pixel, and that as a direct result of sub-sampling the templates, they essentially implement their blur HMT. That is, sub-sampling the templates is deemed to be similar to eroding the them by the appropriate blur SEs. The authors use a number of sub-sampling techniques and conclude that sub-sampling the templates following a regular grid provides the best results.

In [39], Doh *et al*. propose an algorithm which uses a so called "synthetic SE" in order to allow the HMT to detect features in clutter. The authors point out that the performance of morphological operators when applied to object recognition tasks is critically dependent on the SEs that are used. They therefore suggest that an optimal SE be designed which is capable of improving the detection rate of the HMT while reducing the number of false alarms. In addition to this, the authors relax the strict fitting criteria of the standard HMT by implementing the erosions using thresholded correlation operators. The authors explain that by varying the threshold they are essentially introducing rank order filters in order to reduce the likelihood of the transform failing in the presence of noise, distortion or clutter.

Doh *et al*. point out that when using the HMT, the foreground SE is often designed to match the feature of interest while the background SE is designed as the complement of the foreground template. The authors then present what they call synthetic SEs and demonstrate that the synthetic foreground SE may be computed by calculating the intersection of all "true-class" foreground SEs. By a similar technique, the synthetic background SE is computed by calculating the union of the foreground SEs and taking the complement of the result. These SEs are defined in terms of set theory and hence it is assumed that the authors use only flat SEs in their synthetic HMT (SHMT). By this approach, the spatial separation between the borders of the foreground SE and background SE will be maximised to allow all features of interest to be matched by the synthetic SE in a single pass of the image.

The authors use real and synthetic image data to test their synthetic SE when using their SHMT. They explain that the SHMT implements two thresholded correlations, one for the synthetic foreground SE and one using the synthetic background SE. The result of the SHMT is then computed by calculating the intersection of these thresholded results. Doh *et al*. create their synthetic SEs by cropping examples of the features that they wish to detect, from the image, and they use these to compute the foreground and background synthetic SEs as described earlier. In the case that greyscale images are to be processed, these images are thresholded before the SEs are selected and used to generate the synthetic SEs. These synthetic binary SEs are then applied to the thresholded binary image.

The authors point out that the threshold value that is applied after correlation affects the output of the transform. It appears that when the image data is not distorted by noise, this parameter is set such that the standard HMT is implemented. It also explained that after performing the correlation on the real experimental data, the results contain a large amount of distortion. The authors therefore modify the threshold parameter arbitrarily such that different values are used for the foreground and background operations.

While the authors present a set of results which confirm that it is possible to use their synthetic SEs and a rank order HMT to detect features in the presence of noise, distortion and clutter, no robust method is provided for setting the threshold/rank parameter in their routine. Instead, the authors appear to set this critical parameter based on the mean and variance value of the image. They then "tweak" this parameter after the correlation has been applied in order to optimise their results by reducing the number of false alarms. A robust method for setting this parameter is clearly required.

It should also be noted that it is obvious that increasing the spatial separation between the foreground SE and background SE will allow greater variation of features to be detected by the standard HMT. However, the suggestion of computing the synthetic SEs using the method proposed in [39] does provide a robust method for setting this increased spatial separation. The only drawback is that in [39], the authors appear to generate these synthetic templates by cropping features of interest from the test image itself. They then create the synthetic SEs before applying them to the same image from which they were generated. It is no surprise therefore that the features are located in the image by the HMT when using this approach. It would be more appropriate to select features of interest from a set of training data and use these to generate the synthetic SEs. Then the SHMT could be applied to a set of unseen validation/test images, which contain variations of the features of interest, and these features could be located using the synthetic SEs. In fact, Harvey *et al*. in [11] adopt this approach when using the HMT for ship detection in satellite imagery. In [11], the authors extend the method proposed in [39] for computing synthetic SEs such that it can be used for greyscale templates. Further, Harvey *et al*. design their synthetic SEs using a set of training data and then apply their extended transform [39] to both the training set and an unseen test set for validation.

3.1.3.3 Modifying the Transform

In [40], Maragos discusses the problem of image matching and template matching and presents a number of techniques that can be applied to solve problems in this area. The paper is mainly theoretical and no experimental results are provided. Since

the focus of this thesis is template matching using the HMT, only the relevant sections of [40] are discussed here.

In [40], Maragos describes the idea of template matching and defines the binary HMT which can be used for this purpose. The author explains that the HMT is directly related to linear correlation, provided that the output of this correlation is thresholded appropriately. This was also demonstrated in [39] where the authors varied the threshold, following linear correlation, to implement the standard HMT and an extension of this which relaxes the strictness of the transform. Maragos points out in [40], that the HMT is sensitive to noise and that small amounts of this can cause the transform to fail. He states that in [41] Crimmins and Brown proposed that a large number of templates, which include possible variations of the spatial appearance of the sought feature due to noise, can be used to search for that feature in noisy data. Consequently, Maragos points out that in using this approach, it would be difficult to account for all possible distortions of a feature without having to apply the transform an unrealistically large number of times with an extensive set of SEs.

As an alternative, Maragos proposes in [40], that in place of erosions and dilations, a measure of how well the SEs fit the image at each point be calculated to allow partial fitting of the templates. He points out that a measure of fitness may be calculated by computing the sum of pixels which coincide with the SEs at each point and dividing the result of this summation by the number of points in the corresponding SE (note: this only applies to binary case). In this way, the measure of fitness can be calculated for each point in the image and its value may be allocated to each pixel of the output image. Maragos points out that this method allows the HMT to perform better in noise. However, the author also states that an appropriate threshold value must be identified when using this modified HMT. Although this point is mentioned in [40], the issue of determining a suitable value for this parameter is not formally addressed, and hence no method is provided for its robust estimation.

Maragos, [40], goes on to explain a relationship that exists between a statistical hypothesis testing approach to template matching and binary correlation. The author

uses a binary image containing the sought template which is corrupted by synthetic impulsive noise and proves that applying the statistical approach is equivalent to performing a linear, binary, cross correlation of an image with a template and thresholding the result. Since the probability distribution of the noise model that is used is known, a value for the threshold can be computed in terms of the mean value of the signal and the probability of the noise. However, no method for setting this value in the more general case, where, for example, the distribution of the noise in the image is not known, is provided in [40]. Often, in real applications the exact probability distribution of noise is not available, and hence alternative techniques for determining a suitable value for this rank parameter must be investigated.

Maragos, [40], also talks about relaxing the strictness of morphological operators by using rank order filters, and states that this is theoretically equivalent to applying linear correlation and varying the threshold that is used. The author points out that the rank filter can be applied to greyscale images, and states that the rank reflects the area/portion of the SE which fits the signal at any given pixel. However, no robust technique for setting this threshold or rank parameter is provided in [40].

In [32], and in addition to the blur HMT that was proposed in this paper as described in Section 3.1.3.1, Bloomberg and Maragos provide a definition for a Rank Hit-Miss Transform to make the transform more robust to noise. The authors first demonstrate that rank order filtering is equivalent to thresholded correlation, as was described in [40], and demonstrated in [39]. They then use rank order filters in place of erosions to improve the performance of the HMT when processing noisy data. The authors explain that the use of rank filters imposes looser fitting conditions than the erosion when searching for a feature using a pre-defined template. They also state that these ideas have been applied to several problems where template matching has been used locate features in binary data [42], [43], [44].

The authors [32] explain the steps that may be taken to relax the conditions of the standard HMT using rank filters before stating that this can be used to implement a generalised rank opening which consists of applying a rank HMT and dilating the

result by the foreground SE. However, no method for accurately determining an appropriate rank for the rank HMT (or any variation of this), or an appropriate threshold for using thresholded correlation, is provided in this paper. In fact, as was pointed out in Section 3.1.3.1, the authors do not provide any experimental results which demonstrate applications of the blur HMT, rank HMT, or the generalised rank order blur opening.

It was explained in Section 3.1.3.2, that Khosravi and Schafer, [26], investigated the possibility of sub-sampling the SEs to improve the performance of the HMT when operating in noisy data. In addition to this, they investigate the possibility of using rank order filters in place of the traditional morphological operations in order to assess the robustness of this extension when the transform is presented with noise.

The authors test the performance of their rank order HMT in the presence of Gaussian noise and impulsive noise. Empirical techniques are used to determine lower and upper limits for the value of the rank parameter and the authors conclude that these limits are determined by impulsive noise and Gaussian noise, respectively. However, no formula or robust method (other than the exhaustive empirical one) is provided for computing these limits. Further, no method for determining the appropriate value that should be used for the rank parameter in order to locate a feature of interest in any specific image data is provided in [26].

In [21], Soille presents a number of rank HMTs which have been developed for the processing of binary images, [32], [45], [46], [47], [48]. However, no greyscale rank HMTs appear to be presented in [21], and no robust techniques that can be used to set parameters for these rank HMTs are provided. Later, in [12], Soille presents a rank HMT which can be applied to greyscale data. The author explains that this is a generalisation of the UHMT which imposes only partial fitting constraints on the SEs. Soille suggests that rank HMTs perform better than the standard HMT when the geometry of sought patterns is perhaps unknown or distorted in some way. The author does not provide any technique that can be used to determine the rank

parameter, and no examples of the rank HMT being applied to any problems are given in [12].

It should be noted, that in [47], Cassasent *et al.* actually perform a series of experiments in order to examine the effect of varying the threshold of their foreground and background correlation operations when detecting binary rectangles of different sizes in noise. The authors use a series of synthetic images for this purpose, and they apply their transform a large number of times while varying the threshold in order to analyse its effect. Using their set of synthetic data, the authors demonstrate that it is possible to determine theoretical limits for each threshold parameter (one for the foreground and the other for the background), and they show how these limits can be used to detect rectangles of various sizes in noisy data using their rank HMT. However, when applying their rank HMT to real data, the authors arbitrarily set their rank parameters (based on heuristics) to be lower than their pre computed theoretical limits. While this further relaxes the strictness of their rank HMT and allows successful detection of their features of interest, the authors do not fully exploit the approach which was used to asses the effect of varying the threshold in the first place.

More recently, in [8], Perret *et al*. present a Fuzzy Hit-or-Miss transform which they use to detect so called Low Surface Brightness Galaxies (LSBs) in very noisy astronomical images. The method in [8] appears to be an extension and more thorough description of the method described in [49]. The authors, [8], begin by introducing the problem and they state that LSBs are not yet fully understood by astronomers. They also point out that due to the low SNR in image data (as a result of long exposure times) no method for automatically segmenting these features currently exists (until the methods presented in [8]). Later in this paper, the authors state that a tool for characterising astronomical objects was developed in [50], and that this has been tested for detecting LSBs using a segmentation map generated by a Bayesian approach which is described in [51]. The authors point out that the method proposed in [51] does not deal with the shape knowledge that is available.

Perret *et al*. then explain the binary HMT by example, before reviewing a number of greyscale HMTs that have been presented over the years. Further, the authors provide a comprehensive review of techniques that have been proposed in the literature which aim to make the HMT more robust in the presence of noise. These papers, and a number of others, have been reviewed in this thesis. The authors also describe some generic approaches which can be used to improve the robustness of the UHMT, the RHMT, the BHMT and the KHMT, in the presence of noise. These are transform dependent and involve either: increasing the distance between the SEs used by the HMT, or, modifying the threshold that is used at the output of the transform. These techniques will be addressed in more detail in Chapter 5.

Having reviewed existing techniques, the authors point out that in [40] Maragos defined an extended HMT for the binary case which allowed the extent to which the SEs matched features in the image to be computed. By measuring how well the SEs fit the image at each point, it is possible to relax the strict conditions of the standard HMT by allowing partial fitting of the SEs. This is the basis for the FHMT which is defined in [8]. It is also the basis of the extended HMT that is presented in Chapter 4.

After providing the required background material, Perret *et al*. define the FHMT which measures how well a pair of greyscale SEs fit each point in a given image when raised through all possible greylevels. The authors then discuss implementation issues and point out that it is possible to restrict the application of the SEs at each point in the image to only a small number of greylevels that are close to the intensity of the signal of interest. They also point out that it is possible to estimate a suitable level at which to apply the SEs from an average filtered version of the original image. The resultant value of each pixel of the average filtered image may be used as an estimate of the intensity level at which to apply the SEs when they coincide with the corresponding pixel in the image under study. The authors also discuss the issue of setting a suitable distance between the SEs and conclude, that where available, local noise estimates can be used for this purpose.

The authors go on to talk about the specific application of their FHMT to the problem of LSB detection. They explain a number of difficulties in the detection process posed by low SNR and the low surface intensity of the LSBs themselves. Further, the LSBs can appear in any orientation and their geometry can vary significantly. The authors then describe an algorithm based on the FHMT and they explain that the first step in the process is to design a large set of SEs that can be used as templates to locate the LSBs in the image. The authors also point out that a background map is required since the intensity of the LSBs can be very similar to that of the noisy background. Also the original image is pre-processed using a median filter to reduce noise before the set of templates, the background map, and the median filtered image, are passed to the FHMT for processing. The output of the FHMT is thresholded in order to locate the position of the LSBs in the image.

The authors are able to describe the LSBs using a sophisticated mathematical model which incorporates a number of parameters including scale length, orientation, brightness and elongation. This is used to generate a set of SEs which contains a total of 640 templates. The authors then explain that the background map is computed by windowing (partitioning) the image to be processed and using the sigma clipping method. This is a common technique used in astronomical image analysis, and, as such, it is not referenced in [8]. However, the authors describe this process by explaining that it is computed in two steps. In the first pass, the mean value, μ, and the standard deviation, σ, of all pixels in each window is computed. Then, the method masks all pixels whose value is greater than $\mu + k\sigma$ or lower than $\mu - k\sigma$, and step one is repeated iteratively until convergence. The authors point out that a typical value for *k* is 3, and the reader is referred to [52] for further information.

Having generated the set of templates and described the process by which they obtain the background map, the authors perform median filtering on the original image and use this to estimate an appropriate level, $t \in T$, at which to apply the SEs in order to perform template matching using their FHMT. For each pixel in the median filtered image, the authors transpose their greyscale SEs to the estimated intensity for this pixel in the image. The distance between the SEs is then fixed by using the standard

deviation of the local noise estimates, σ, from the background model. That is, the maximum value of the foreground SE is raised to the level $t-\sigma$, and the background SE is raised to the level $t+\sigma$. When the appropriate level has been determined, and a suitable distance has been fixed between the SEs, each SE in the large pre-computed set is then applied to this pixel at level *t*. The FHMT is used to compute a score for each pixel based on how well each of the SEs fit the image when fixed at level *t*. The highest score for each pixel, which is computed for the best fitting SE, is allocated to the corresponding pixel in the output image to produce a so called "score map".

When the score map has been computed using the FHMT, the final output of the algorithm is computed by thresholding the map in order to locate the position of the LSBs. Perret *et al*. state that using a threshold of 80% provides a good result, which would imply that this parameter has been determined empirically. By thresholding the score map, the authors obtain a binary image from which they wish to reconstruct the final LSB segmentation map. Although the authors point out that they can achieve this by geodesic operations, they chose to simply dilate the markers in the output image by the foreground SE which provided the best score. The reason, is that the software which is used to document the results obtained in the segmentation map (created by dilating the markers in the thresholded score map) performs better when the edges of features are well defined.

The authors state that processing a 512x512 image using their algorithm, and the entire set of SEs, takes around 20 mins to complete. Therefore, a few approaches are used to improve the efficiency of this routine. These are:

1. Sub-sampling of the SEs – as suggested in [6]
2. Sub-sampling of the image data at a scale of 1:2, also suggested in [6] but not implemented there
3. Abandoning the computation for pixels when it is clear that the threshold cannot be reached

Perret *et al*. point out that the last optimisation technique is heuristic and is based on processing the data in two steps. In the first step, possible points of interest are identified by applying a small subset of the entire set of templates to the image and using a low threshold. Points remaining after the first pass are then tested using the full algorithm. Using the described approach, the authors reduce the computation time from 20 minutes to around 2 minutes per image. The authors concede that using this heuristic approach could potentially lead to LSBs being missed in the detection process.

Perret *et al*. go on to present an impressive set of results when locating the LSBs in image data with low SNR. They compare their method to an approach based on traditional correlation, [16], and conclude that while this performs well, their method outperforms this technique for LSB detection. However, although the authors state that they vary the threshold at the output of the correlation, they do not state exactly how they determine an optimal value for this parameter. They state only that they adjust this threshold (presumably by experimentation) in order to ensure the best results. The authors also compare their results to a set of ground truth data in consultation with an astronomer and demonstrate that their technique performs extremely well for this particular application.

In [11], Harvey *et al*. use rank order filters in place of traditional erosions and dilations in order to make the greyscale HMT more robust in the presence of noise. The authors apply their extended HMT to the problem of ship detection in satellite imagery, where they aim to improve upon the techniques which are currently used for this purpose. The authors point out that existing techniques for ship detection rely on the fact that ships are often brighter than their surrounding background in the image. As a result, these techniques exploit local contrast properties which make them sensitive to variations in intensity. As a result a high number of false alarms are raised when using these existing routines.

In [11], the authors justify the use of the greyscale HMT for template matching by pointing out that it is insensitive to overall greyscale variations of the sought

templates in the image data. They also state that rank order filters are generally faster to compute than linear correlation techniques. As with most recent papers on the HMT, the authors first present the binary HMT before pointing out that a number of greyscale HMTs have been presented in the literature. Harvey *et al*. choose to build upon the BHMT, and hence they explain this in detail before discussing their proposed extension.

The authors explain that the BHMT involves evaluating the distance between the two probes, where the distance metric that is returned at each point, is allocated to the corresponding pixels of the output image. Harvey *et al*. explain that the output image must therefore be thresholded such that only the points of interest are marked in the output image, and they state that in [5] Barat *et al*. choose this threshold to be the difference between the values in the centre points of their two greyscale probes. As an alternative, and what would appear to be a more robust approach, Harvey *et al*. propose that the output is thresholded at a number of levels, and that the detection rate, and a false alarm rate, is recorded after each threshold is applied. Using this approach allows a Receiver-Operator Characteristic (ROC) curve, [53], to be generated in order to evaluate the performance of the threshold that is used. This allows the optimal threshold to be determined for any application, and hence the most accurate results may be obtained.

The authors then extend the BHMT by replacing the erosion and dilation with rank order operators. They point out that applying rank order filters relaxes the strict fitting criteria of the standard transform that is imposed by using erosions and dilations, as this allows some tolerance in the fitting of the SEs in the image. The authors also state that the value of the rank parameter affects the robustness of the transform, and that relaxing the strictness of the transform in this way, may lead to an increase in the number of false alarms. This point is addressed in their experimental results section, where the authors use ROC curves to evaluate the affect that varying the rank parameter has on the extended transform and they compare their rank HMT with the standard BHMT.

The authors use a set of satellite images to test their method. The data set contains a total of 40 ships that they wish to detect, as well as a number of features, including different ships, which are not of interest. The authors partition this set of images into a training set and a validation set. Each set contains 20 of the 40 ships that they wish to detect. From the training set, the 20 ships of interest are manually selected from the image and are used to generate the so called "hit template" and "miss template". These terms are synonymous to respectively the foreground and background SEs/probes that have been discussed so far in this thesis.

Having selected 20 example ships to use as templates, the authors use a technique, similar to the one presented by Doh *et al*. in [39] for designing binary SEs, to generate the templates that are to be used for their rank HMT. That is, the "hit template" is computed by registering all 20 of the manually selected targets (from the training data) and computing the point-wise minimum. The "miss template" is computed, by a similar process, except that the point-wise maximum is computed for all points of the 20 targets. These templates are then used to compute the rank HMT that is proposed in [11]. The SEs are actually applied to the image, by the rank HMT, at 64 different angles in order that the orientation of the ship, as it appears in the image, does not cause ships to be missed. When all 64 images have been computed, the final output image that is to be thresholded is calculated by computing the point-wise minimum of all pixels, in each of the 64 images. This ensures that the value for the best fitting template pair is used in the final detection process - thresholding.

The authors evaluate the effect that the rank parameter has on the output of the extended HMT that is presented in [11] using ROC curves. In the first instance, the rank parameter is set to 1 such that the standard BHMT is implemented. Then, the authors apply their extended HMT, a number of times, and each time they increase the rank parameter in order to reduce the strictness of the transform. Each resultant image is thresholded at a number of levels and the detection rate and the number of false alarms are computed for each output when varying the rank and applying the transform. This information is used to generate ROC curves for both the training data and the test data for each rank that has been used. Using the ROC curves allows the

best threshold to be selected for the optimal rank parameter after having applied the transform a number of times while varying the rank. The results obtained when applying this rank HMT to the training data are better than the results when applying the same to the test data, however, this is to be expected since the templates are essentially matching/locating themselves in the training data. However, it is clear in both cases, that increasing the rank parameter to reduce the strictness of the transform does improve the performance of the HMT to a point. A direct comparison of the proposed method and the standard BHMT is given in the ROC curves as the BHMT is implemented when the rank parameter is 1.

The authors conclude by pointing out that they have demonstrated that using rank order operations in place of erosions and dilations can significantly improve the performance of the HMT when applied to the ship detection problem. This is clear by interpreting the ROC plots that are provided in the paper. The authors also point out that they have performed a rather exhaustive search in an attempt to find the optimal rank parameter that should be used. Indeed this is true, they have applied their extended HMT a number of times while varying the rank parameter, and then they only assess the performance of each rank value, after the results have been obtained. This means that to find the optimal rank parameter, the entire transform must be computed multiple times for a large number of different rank values. Then, the best performing rank value is found (after applying a large number of thresholds and generating ROC curves), before every result, except the optimal one, is discarded. This is rather time consuming and wasteful of resources.

Ideally the rank parameter should be determined prior to executing the transform. In this way the parameter can be optimised and the transform need only run once. While it is recognised that ROC curves provide a good measure for validating performance, more efficient methods should be investigated for optimising this rank parameter. Harvey *et al*. concede this last point in [11]. Also, determining the rank parameter using the method proposed in [11] requires that training data, and reliable ground truth data, is available. Although a data set can normally be split up in order to allow the use of some images for training, accurate ground truth data cannot always be

obtained. Also, as the number of features in the training data increases, the computation and quantification of ground truth data becomes an onerous task. Finally, Harvey *et al*. point out that the algorithm is computationally expensive, and that fast algorithms for computing the output of this type of transform would be beneficial.

3.1.3.4 Summary of Extensions of the HMT

Each of the aforementioned methods and extensions aim to improve the performance of the HMT such that it is more robust for object recognition and feature extraction in noisy images. All of the methods that have been discussed provide valid solutions using a wide variety of extensions and techniques. Rank order operations feature heavily in this work, however, there appears to be a conceptual gap in that most of the authors often fail to provide a robust and general method by which it is possible to select the appropriate rank or threshold parameter. In fact, the only authors who appear to give any real attention to robustly setting this parameter are Harvey *et al*. in [11]. However, in their paper, published in 2010, the authors of [11] point out that their approach is extremely exhaustive, and that optimal methods for setting this parameter should be investigated.

This thesis presents a novel Percentage Occupancy Hit-or-Miss transform which allows partial fitting of SEs in a similar fashion to the partial fitting allowed by rank order filters. The difference here however, is that a robust and efficient design tool is provided for accurately determining the rank, or percentage of the filter that must be occupied, for accurate detection of features of interest. In addition, the design tool may be used to robustly determine the optimum parameters for the many methods, developed by other authors, which have been described so far.

3.2 Efficient Techniques for Computing the Output of Morphological Operators and Rank Order Filters

Over the last forty years, there has been significant interest in developing efficient methods for computing the output of morphological operators, and more generally, the output of rank order filters (the relationship between morphological operators and rank order filters was explained in Chapter 2, Section 2.5). Due to the computational complexity of using a direct implementation to compute the output of the generalised HMT that is presented in this thesis, a number of techniques for reducing the execution time of this transform have been investigated. It is explained in Chapter 6 that the extended transform presented in Chapter 4 of this thesis can in fact be implemented using rank order filters and, as such, an efficient method for computing this class of filter has been developed. It is shown in Chapter 6, that this fast algorithm provides a significant reduction in the time taken to compute the output of the extended HMT (Chapter 4) when compared with a direct implementation.

It is required therefore, that a short overview of existing techniques that can be used to compute the output of various morphological operators, and other specific rank order filters, is provided here. The intention of this summary of the relevant literature is to place the contribution of Chapter 6 in context with the state of the art techniques for optimising the efficiency of morphological operators, and other rank order filters. In Chapter 6, two of the methods ([54] and [55]) that are discussed here in brief will be described in much greater detail. Following a detailed explanation of these techniques, it is then shown in Chapter 6, how these methods may be combined in such a way that it is possible to compute the output of any arbitrarily shaped rank order filter in minimal time.

3.2.1 Structuring Element Decomposition and Efficient Techniques for Computing Erosions and Dilations with Linear SEs

Computing the output of an erosion or dilation by a direct implementation can be computationally intensive and time consuming. One way to increase the efficiency of

computing these operators, as described in [12], is to decompose the SE into a set of smaller linear segments which can be applied successively to the image in order to determine the result. It is more efficient to compute the morphological transform in this way than by performing it a single pass of the image using the larger SE. In fact, it is stated in [12], that whenever a SE can be decomposed, this method should be considered for improved efficiency.

An efficient method for computing morphological operators using linear SEs was presented by Van Herk [56] in July of 1992. Less than one year later, in May of 1993, the same method was published independently by Gil and Werman [57]. As a result, the method is sometimes referred to as the van Herk/Gil Werman algorithm (vHGW). Line segment SEs may be used to erode or dilate an image if they are suited to a particular application, or, more likely, when they are obtained by SE decomposition. If the vHGW method is used to implement erosion or dilation, and it is coupled with the SE decomposition method, then significant gains in speed can be achieved. In the following discussion, only the implementation of dilation will be discussed, however, erosion can easily be described by replacing references to computation of the maximum with that of computation of the minimum.

The execution time of the vHGW described in [56] and [57] is independent of the length of the SE, and it requires no more than three comparisons to compute the dilation for a pixel in any given image. Each 1D line of the image is divided into segments, where the length of each segment is determined by the length of the SE. Then, two temporary buffers are filled (one in the forward direction and one in the backward direction) by recursively computing the maximum values in each segment. When this is completed, the buffers contain the only possible maximum values for any pixel in this 1D line when it is coincident with the origin of the SE. The output for any pixel may then be computed by correctly indexing these buffers and calculating the maximum of the two values (one from each buffer) that correspond to the pixel in question. This method was later extended by Breen and Soille in [58], and by Soille *et al.* in [59], for use with line segments at arbitrary angles. A further

extension to the vHGW was introduced in [60] where the efficiency of this algorithm was further improved by Gil and Kimmel.

3.2.2 Moving Histogram Approach

An alternative approach to efficiently computing the output of morphological operators, which has the added benefit that it is compatible with other rank filters, is to use the moving histogram technique. This technique is based on the work of [54] and [61] for efficient computation of 2D median filters which were first applied to image data by Tukey in [62], [63]. Taking the method proposed by Huang *et. al* [54] as an example, the moving histogram technique can be described. This idea (as with other moving histogram techniques) exploits the fact that as a sliding window traverses the image, only a small number of pixels that coincide with the window actually change as it moves from one pixel to the next. Instead of resorting all of the values in order to compute the next median, it is possible to deal with only those pixels that change as the window moves from one position to its neighbour. Dealing with fewer pixels directly leads to an increase in speed, and this is further enhanced by using a histogram to store the frequency of the intensity of the pixels that coincide with the window.

Since the values in the histogram are sorted, it is easy to locate the value of the median from the pixels coinciding with the window at each point in the image. Each time the window moves from a pixel to its neighbour, the histogram is updated by removing from the histogram the pixels that leave the window, and adding to it, the pixels that enter during translation. Further, in the method proposed by Huang *et al.*, the search for the new median does not restart each time the window is translated. Instead, it is identified from the position of the previous median by keeping a count of the pixels in the window whose value is less than the previous median. The value of the count is used to decide how far and in which direction the histogram should be searched.

The technique proposed by Huang *et al.* [54] was extended for computing the output of more general rank filters by Chaudhuri in [64]. While these methods ([54], [61] and [64]) are useful, and prove to be extremely efficient, the major drawback is that they only provide solutions when using square and rectangular windows to filter the image data. When considering morphological operators, and a number of other filters, the use of arbitrarily shaped windows is essential in particular applications. As a result, the methods presented in [54], [61] and [64] are restrictive as only square and rectangular windows can be used.

The moving histogram technique was extended by Van Droogenbroeck and Talbot in [55] to make it possible to efficiently compute the output of erosions and dilations within any arbitrarily shaped windows. The authors achieve the extension by introducing the concept of critical points – these are the points that must be used to update the histogram as the arbitrarily shaped SE is translated by a single pixel. The authors show by example how these points can be found. They also introduce an alternative scanning technique which further optimises the routine, and they gain additional reductions in computation time by consulting the histogram much less frequently than for the method proposed by Huang *et al*. [54]. The authors of [55] achieve this by maintaining a record of the minimum (erosion)/maximum (dilation) value in the window and continually updating this value as new pixels enter it. The histogram is then only searched when the bin corresponding to the current minimum/maximum becomes zero. This final optimisation technique, which allows less frequent consultation of the histogram, can only be applied when computing max/min filters.

Van Droogenbroeck and Talbot compare their efficient implementation to three others in [55], and show that while their method outperforms two of the competing techniques, that a linear decomposition technique (See Section 3.2.1), using a square window, is slightly faster than their proposed method when using the same SE. However, the difference in execution times is marginal, and while the linear decomposition is slightly faster when using square SEs, the proposed method has the

significant advantage that it can be used to compute morphological operations within any arbitrarily shaped window.

In [55], Van Droogenbroeck and Talbot state that their method can be extended for computing the output of any rank, however, they only describe an algorithm that can be used to compute the specific case of minimum and maximum filters (erosions and dilations). The extension to more general ranks is more complicated, and more costly than the authors of [55] imply, since it requires that the histogram is consulted for every translation of the window in the image. Further, a count must be kept (similar to the one used by Huang *et al*.[54]) in order to correctly direct the search for computing the next output. Although the authors refer to the possibility of computing any rank by extending their method, the exact steps that must be taken to implement this are not provided.

In [65], Breare and Lehmann exploit the properties of the sliding window for improved efficiency when computing: morphological transforms; rank order filters; and other local linear statistics within arbitrary windows. Only the morphological operators and more general rank order filters fall within the scope of this thesis, and as such, discussion of the other techniques which are presented in [65] is omitted. The authors in [65] describe the Van Droogenbroeck and Talbot method [55] for computing erosions and dilations within arbitrary windows. They then state that they have implemented general rank filters, including the median filter, by using the techniques proposed in [55], and they point out that this method was first proposed in [54]. However, the authors do not provide specific details of the algorithm they use in [65]. Instead, they simply state that this has been done.

Breare and Lehmann also propose that although rank filters are not separable, there is an advantage in estimating the output of these filters using what they refer to as “line versions” of the kernels (windows). They point out that this was initially proposed in [66] and explain that due to the significant gains in speed that can be achieved by taking this approach of decomposing the window, it is worth testing in the case that a median filter is used to remove noise. While this appears sensible for the case of

noise removal, the technique of approximating the window, and hence the output of the filter, cannot be used to speed up the rank filters presented in this thesis. The reason is, that the transform which is presented in this work aims to detect features of interest based on their morphological properties. As a result, the shape of the window and the value of each rank in that window, are critical for computing the output of the proposed transform.

In the experimental results section of [65], the authors provide timings for the specific case of computing the output of a median filter using what they refer to as Huang's median filter. The shape of the window that is used for median filtering is not specified in the paper. However, the authors go on to compare the efficiency of this algorithm to: a direct implementation of the median filter; an implementation of their separable direct algorithm for computing the median; and an implementation which they call a separable Huang algorithm. Since it was explained that the separable filters are not compatible with the techniques that are proposed in this work, it would appear from this publication [65], that the only comparable technique to the optimised arbitrarily shaped rank order filtering algorithm would be a direct implementation of the algorithm itself. This point is strengthened by the fact that when the authors compute the timings for other specific rank filters such as the minimum and maximum filter (erosion and dilation), a number of fast implementations including the vHGW algorithm, and what are referred to only as separable algorithms are also compared to the moving histogram technique. As with the median filter experiments, the shape of the SE that is used to erode and dilate the image is not specified in [65].

The authors of [67] and [68] also make reference to the methods presented in [55]. These authors, [67] and [68], exploit the properties of the sliding window to improve the efficiency of their algorithms by only updating the points that change as the window is translated across the image. The applications in these papers are geared more toward the computation of local linear statistics within the moving window. As a result, further discussion of these methods is beyond the scope of this thesis.

A number of researchers, including Van Droogenbroeck and Talbot, state that the method presented in [55] can be extended and used to compute the output of any general rank order filter within arbitrarily shaped windows. However, it appears that no paper or other publication actually describes or documents exactly how this may be done. Even in [65], where the authors state that this extension has been made, no explanation of exactly how this extension can be implemented is given. Further, no results which support this claim are provided. The authors of [65] provide timing results for the specific case of computing the output of the max, min and median filter without mentioning the shape of the window that has been used. In fact, to the author's knowledge, there is no paper which uses this method and presents timings for computing the output of rank filers for arbitrary ranks.

In [69], Perreault *et al.* present a method which can be used to perform median filtering in constant time. That is, for a given image, the time taken to compute the filter output is not affected by the size of window that is used. The idea is similar to the method proposed by Huang *et al*., and the authors begin by describing the method proposed in [54]. Interestingly, Perreault *et al*. point out in their introduction that when using the moving histogram approach, it is possible to compute the output of any rank order filter (not just the median) by simply "changing the stopping value". However, no examples of computing ranks other than the median are provided, and no technique for computing more general ranks within arbitrarily shaped windows is described. Instead, the authors describe their method for efficient computation of the median filter defined by square windows, before showing that octagonal windows (approximating a disk) can also be used with their routine.

Before explaining their routine for square kernels, Perreault *et al.* point out an inefficiency which exists in Huang's method. That is, no information about the image pixels is retained between rows. The authors of [69] exploit this redundancy in order to realise their constant time median filter. Instead of using what they refer to as a kernel histogram, i.e. a single histogram which contains the intensity of each image pixel which coincides with the window, Perreault *et al*. store an entire row of column histograms. The height of each column histogram is determined by the height

of the kernel. Since histograms are distributive, it is possible to obtain the kernel histogram at any point in the image by computing the sum of the appropriate column histograms.

To compute the output of the median filter at any given image pixel, the kernel histogram is computed by adding together the appropriate column histograms and locating the median value. Computing the output at the next pixel is achieved in two steps. First, the column histogram immediately to the right of the window is updated. This is achieved by removing the topmost pixel from the appropriate column histogram and adding to it the intensity of the pixel which is directly below its current position. The new kernel histogram may then be computed by subtracting the leftmost column histogram from the kernel histogram and adding the updated column histogram (computed as described above) located one pixel to the right of the current window position. The new median may then be calculated, and this process continues until the entire image is processed. For each translation of the window, only two pixels are processed (regardless of the window size): the topmost pixel in the column histogram is removed; and the pixel which is directly below its current position is added.

The authors point out that when processing the first row, *every* point in each column histogram needs to be computed from the image. This means that the complexity of processing the first row is equivalent to the complexity of Huang's method. However, the authors point out that this additional complexity is negligible since it is only required on the first row of any image. The authors also provide a number of suggestions which can be used to minimise the constant execution time when implementing their proposed routine, and they demonstrate that their method is significantly faster than the one proposed in [54]- particularly when the window size is large. Perreault *et al*. also provide some suggestions for applying their method to higher precision data and to data of higher dimensions.

The authors also present a method for implementing a constant time median filter using an octagonal kernel (which is used to approximate a disk). This is realised by

retaining five histograms (one for each side of the octagon) for each translation of the window as opposed to the single column histogram which was required for square windows. While this approach seems sensible, the authors point out that they would expect the execution time for octagonal filters to be around 5 times slower than for a square window. Clearly, extending this method for use with arbitrarily shaped windows is not trivial, and in some cases it could require an unrealistically large number of histograms to be maintained and updated for each translation of the window. Dependent on the shape of the window, this may lead to very slow execution times when compared to alternative techniques.

3.2.3 Look Up Table Methods

In 2008, Urbach and Wilkinson, [70], introduced an extremely efficient technique which allows erosions and dilations to be computed within arbitrarily shaped windows in such a way that it always outperforms the method proposed in [55]. In fact, the authors state in the abstract of [70], that their method outperforms the method proposed by Van Droogenbroeck and Talbot, by a factor which ranges between 3.5 and 35.1. The variation in the improvement can be attributed to a number of factors including the type of image that is processed and the SE that is used.

The method proposed in [70] is actually an improvement of an earlier algorithm which was described in [71]. The technique uses a decomposition of the SE (which can be applied to any arbitrarily shaped SE) and a look-up table (LUT) to efficiently compute erosions and dilations. In [70], the method that is described requires that the SE is first decomposed into a series of linear chords (see [70]) in a fashion similar to the one used in [72]. These chords are essentially a set of 1D, horizontal SEs, which can be found for any arbitrarily shaped SE by reading lines/runs of connected pixels starting from the top of the SE and scanning from left to right until the bottom of the SE is reached. It is shown in [70], that computing the output of an erosion (dilation can be achieved by swapping minimum for maximum) for any image pixel using the

entire SE can be achieved by computing the minimum value within each of the chords before computing the minimum of all of these results.

When the chords for the SE have been determined, the full LUT is generated for the first row in the image, and the erosion/dilation of this row may be calculated. The output of the morphological operator is then calculated one row at a time, such that the LUT is computed only for the number of lines that are actually required to compute the full erosion/dilation for the current row. To compute the output value of an entire row of pixels, the values in the LUT are quickly located by a novel indexing technique, and the minimum value of the appropriate points in the LUT is computed to find the result of the erosion at each pixel. When an entire row has been processed, the LUT is updated recursively requiring minimal computation and the process continues until the entire image has been processed.

Urbach and Wilkinson present an impressive set of results when they compare the efficiency of their algorithm with what they claim is the only comparable method for computing erosions and dilations within arbitrarily shaped windows. This method is of course the Van Droogenbroeck and Talbot algorithm, as described in [55]. The routines are compared using a number of differently shaped SEs and image types, and the Urbach and Wilkinson method is shown to always outperform the one proposed by Van Droogenbroeck and Talbot.

The authors of [70] also compare their implementation with a number of efficient techniques for computing the output of morphological operators when specific SEs are used. For example, the authors discuss a technique presented by Soille *et al*. [59] and [73], which can be used for the efficient computation of morphological operations using polygonal approximations to disk SEs. It is shown in [70] that the method presented there outperforms the one proposed by Soille *et al*. even when the disk SE is approaching a 100 pixel diameter. Beyond this 100 pixel diameter, the algorithm proposed by Soille *et al*., [59] and [73], outperforms the Urbach and Wilkinson method [70]. However, the performance of the method reported in [70] remains particularly impressive when the authors point out that their technique

accurately computes the output of the transform using the exact digital disk SE. On the contrary, the method proposed by Soille *et al*. uses an approximation to the disk SE, and it can therefore be assumed, that this technique [59] and [73] is less accurate than the Urbach and Wilkinson method [70].

The only method which is truly able to outperform the one proposed in [70] is a method proposed by Gil and Kimmel, [60]. This routine is an improvement of the vHGW algorithm that was proposed in [56] and [57], and Urbach and Wilkinson, [70], credit this as being the fastest known method for computing the output of linear horizontal SEs. Although the improved vHGW algorithm presented in [60] is capable of outperforming the Urbach and Wilkinson method, this only occurs when the horizontal and vertical SEs are greater than a particular length. From the paper [70], these lengths are respectively, 31 pixels and 63 pixels. However, unlike the Urbach and Wilkinson method, the algorithm proposed in [60] is limited in that it cannot be used to compute the output morphological operations using arbitrarily shaped SEs.

3.2.4 Further Reading

There are a large number of efficient methods for computing morphological operators which have not been discussed here. Since the moving histogram method is the one that is used in this thesis, extensive coverage of alternative, unrelated methods, is beyond the scope of this thesis. Soille provides an excellent review of some efficient techniques in [12]. In [70], the authors point out that an overview of efficient algorithms for computing morphological operators using linear SEs and decompositions of SEs can be found in [74]. The interested reader is also referred to [55], [70], [75], [76], [72], [77], [60] for further reading in the area of fast and efficient morphological erosions and dilations.

3.2.5 Summary of Efficient Algorithms

The review presented in Section 3.2 indicates that the fastest known method for calculating morphological operations, using arbitrary SEs, is the one presented in [70]. However, the authors of [70] state explicitly that their method cannot be extended to allow the computation of more general rank order filters. As a result, the method proposed in [55] is taken to be the optimal routine that can be generalised for this purpose and hence it is used to implement an efficient algorithm for computing the output of the transform that is proposed in this thesis.

In Chapter 6, the techniques described in [54] and [55] are extended and combined to make it possible to compute any rank within any arbitrarily shaped window. Although the extension is straightforward, to the author's knowledge, this is the first time that both routines ([54] and [55]) are thoroughly described, by example, in a single publication. Further, although a number of authors point out that this generalisation can be realised [55], [65], and [68], Chapter 6 of this thesis (to the author's knowledge) will provide the first description of the full algorithm for computing arbitrary ranks within arbitrary windows. The algorithm is actually a synthesis of the two methods that are proposed in [54] and [55] and it is not a direct extension of one or the other. A thorough description of the techniques that have been presented here regarding the methods presented in [54] and [55], and the extensions that are made and documented in this thesis, are provided in Chapter 6.

Finally, it is worth mentioning at this stage, that it is thought - given the extent to which it is capable of outperforming the method proposed in [55] - that there may be a way to extend the method described in [70] such that it is possible to use this to compute the output of more general rank filters. If this were achieved, it is believed that this would become the fastest known algorithm for computing the output of arbitrarily shaped rank order filters. This idea is discussed briefly in Chapter 7, where it is indicated that further work will include investigating this possibility.

3.3 Discussion

This chapter has presented an extensive review of the literature that is related to the techniques which are presented in the next three chapters of this thesis. The first section of this chapter dealt with the HMT and began by discussing the binary HMT before explaining that a number of researchers have extended this independently. A large number of techniques which aim to extend the standard HMT in some way or other were then discussed and it was shown that rank order filters play a large role in extending the HMT for improved performance in noisy data. However, it has also been identified that robust techniques for determining suitable values for the rank parameter that is introduced are often not provided. Even in cases where values are provided for these parameters, they appear to be determined subjectively by experimentation, and hence no robust method for setting these is provided. In one case, [11], the authors use an exhaustive and time consuming technique relying on ROC curves to determine the affect that the value of the rank parameter has on their rank HMT. The authors themselves concede that better techniques are required.

In Section 3.2 of this chapter, a number of techniques for improving the efficiency of computing the output of morphological filters and other rank order filters were discussed. It was shown that there is a need for efficient computation of the output of such filters, and that although a number of techniques have been described, they each have their own merits and demerits. Some methods can only be used to compute the output of specific rank filters such as the minimum, maximum or median filter, while others can compute all of the ranks, but are restricted to specific window shapes i.e. square/rectangular. A number of authors have described techniques which can be adapted to allow the output of any arbitrarily shaped rank order filter to be computed efficiently using the moving histogram approach. However, it has been identified that the specific steps that must be taken to realise this algorithm are not actually documented anywhere in the literature. Currently, successful implementation of this algorithm is non-trivial and requires locating and understanding a number of publications before realising the link that exists between these methods and extending these appropriately.

Each of these points is addressed in the remainder of this thesis. In the following chapter, the Percentage Occupancy Hit-or-Miss transform will be introduced which allows relaxed fitting of the SEs in a fashion similar to rank order filters.

4 The Percentage Occupancy Hit-or-Miss Transform

The previous chapter reviewed the work carried out by other researchers in extending the original binary HMT, as defined by Serra, such that it is capable of reliably processing: binary data; greyscale data; and noise corrupted or otherwise distorted data of this nature. The background theory that is required to build upon the existing definitions of the HMT to make it more robust in the presence of noise in both binary, and greyscale images, has been set out in Chapter 2.

This chapter begins by recalling the definition of the binary HMT as detailed in Chapter 2 before providing definitions of the greyscale HMTs that have been proposed by various researchers working in this area, as discussed in the previous chapter. A novel, conceptual definition of the HMT in terms of SE occupancy is then presented, and this is used to explain the inability of the HMT to function in the presence of noise. Finally, this conceptual description of the greyscale HMT is used to explain how this transform may be extended to function effectively on noisy data, and its usefulness is demonstrated by example.

As was shown in Chapter 3, one of the major gaps in the literature is the lack of analytically robust methods for setting various parameters when relaxing the conditions of the HMT such that it can cope with noisy data. This chapter sets out a novel extension of the HMT for reliable functionality in noise, and it is shown here how parameters may be determined empirically. A new design tool for estimating parameters for this extension of the greyscale HMT, and for alternative greyscale HMTs, will be developed in Chapter 5. For the most part, a discussion of this new design tool will be deferred until the next chapter.

4.1 The Hit-or-Miss-Transform

As was explained in Chapter 2, the HMT of a binary image X is the intersection of an erosion of X and an erosion of the complement of X by a complementary pair of SEs B_{FG} and B_{BG} respectively. That is,

$$HMT_B(X) = (X \ominus B_{FG}) \cap (X^c \ominus B_{BG}) \tag{4.1}$$

where $(B)_x = \{b + x \mid b \in B\}$ and $B_{FG} \cap B_{BG} = \varnothing$.

Since the HMT is not an increasing transform, its extension for processing greyscale images is not a trivial task. For this reason (as was explained in Chapter 2), a number of greyscale HMTs have been defined by various researchers. These greyscale HMTs, which were discussed briefly in Chapter 3, will now be defined and compared in order to demonstrate their similarities. Defining these greyscale HMTs at this stage also facilitates the introduction of the conceptual greyscale HMT which will be presented in Section 4.2 and places this contribution in context.

The first greyscale HMT that will be discussed in more detail is defined by Ronse in [27], (denoted RHMT in [8]) which, using notation that is consistent with the notation laid out in Chapter 2 and following [24] may be written as,

$$RHMT_{[B_{FG}, B_{BG}]}(I)(x) = \begin{cases} (I \ominus B_{FG})(x) \textbf{ if } (I \ominus B_{FG})(x) \geq (I \oplus B^*_{BG})(x) \neq +\infty \\ -\infty \qquad \textbf{otherwise} \end{cases}. \tag{4.2}$$

where $B^*_{BG} : E \mapsto T$ and $B^*_{BG}(b) \to -B_{BG}(-b)$, i.e. the dual of B_{BG}.

In [12] and [28], Soille defines an unconstrained HMT using flat SEs, which $\forall x \in E$, returns the number of cross sections of a greyscale image, *I*, where $(B_{FG})_x$ fits the cross section, $CS_t(I)$, and $(B_{BG})_x$ simultaneously fits the complement of this cross section, $\complement CS_t(I)$,

$$UHMT_{[B_{FG},B_{BG}]}(I)(x) = card\left\{ t \middle| (B_{FG})_x \subseteq CS_t(I) \ , \ (B_{BG})_x \subseteq \complement CS_t(I) \right\}. \tag{4.3}$$

In [24], the authors extend the UHMT as written in Equation (4.3), to allow greyscale SEs, such that,

$$UHMT_{[B_{FG},B_{BG}]}(I)(x) = \max\left\{ (I \ominus B_{FG})(x) - (I \oplus B_{BG}^{*})(x), 0 \right\}. \tag{4.4}$$

By comparing Equation (4.2) and Equation (4.4), the similarity between the RHMT and the UHMT is clear. Both of these extensions mark features in a greyscale image when the result of the foreground erosion is greater than that of dilating the image by the dual of the background SE. The output of the RHMT is an image containing the result of foreground erosions when this condition is satisfied. In contrast, Soille's UHMT produces a greylevel image where the intensity of each pixel indicates the number of cross sections where both SEs fit the image i.e. the difference between the foreground and background erosions.

In [8], the similarity between Soille's and Barrat's greyscale HMT defined in [5] is shown, where Barrat *et al.*'s HMT (denoted BHMT in [8]) is written,

$$BHMT_{[B_{FG},B_{BG}]}(I)(x) = (I \oplus B_{BG}^{*})(x) - (I \ominus B_{FG})(x). \tag{4.5}$$

Clearly, this greyscale HMT closely resembles that of Soille's given in [24]. The two differ however in that Soille's HMT returns the difference between the *foreground operation* and the *background operation* where Barrat *et al*. return the difference between the *background operation* and the *foreground operation*. In Soille's HMT, the higher the output value, the better the fit of the SEs. The opposite is true for the

BHMT, and the equivalence of these two operators is shown in [8] as,

$$UHMT_{[B_{FG},B_{BG}]}(I)(x) = -\min\left\{BHMT_{[B_{FG},B_{BG}]}(I)(x), 0\right\} \tag{4.6}$$

Khosravi and Schafer, in [26], present their greyscale HMT which requires only one SE, B_{FG}, as,

$$KHMT_{[B_{FG}]}(I)(x) = (I \ominus B_{FG})(x) + \left(-I \ominus (-B_{FG})\right)(x). \tag{4.7}$$

This definition is discussed in [8], where it is shown that the KHMT is in fact equivalent to the BHMT as shown in Equation (4.8) and Equation (4.9).

$$KHMT_{[B_{FG}]}(I)(x) = (I \ominus B_{FG})(x) - \left(I \oplus \left(B_{FG}^{*}\right)\right)(x), \tag{4.8}$$

$$= -BHMT_{[B_{FG},B_{FG}]}(I)(x). \tag{4.9}$$

Each of the proposed methods can be used to extract the features of a greyscale scale image which match the geometry of both B_{FG} and B_{BG}. However, all of these techniques fail in the presence of noise unless further modifications, like those described in Chapter 3, are made. All of these techniques are discussed further in [8] and a thorough review of the greyscale HMTs proposed by Ronse and Soille is given in [24]. In this thesis, the definition of the greyscale HMT given by Soille in [12], [28] is extended since it is closest to the conceptual description of the operation of the HMT presented in this work.

4.2 A New Conceptual View of the Hit-or-Miss-Transform

Interpreting a greyscale image as a topographic surface allows the HMT to be considered as a translation of two SEs in this 3D image space searching for places where they simultaneously fit the image to detect objects. In the various definitions of the greyscale HMTs given in Section 4.1, the SEs are translated in the 2D space,

and standard morphological operations are used to probe the image where these operators interact with the greylevel at each pixel as shown in the greyscale definitions of erosion and dilation, as explained in Section 2.2.2 and Section 2.2.3 of this thesis.

In the 3D space, the SEs must still be translated to all points $x \in E$ in the 2D space, however, the translation of SEs in the vertical direction may also be considered. From an implementation perspective the SE may first be translated by a vector x such that the origin of the SE is coincident with an image pixel $x, \forall x \in E$. Then, at each point $x \in E$, the concept of a vertical translation of the SE may be implemented by interrogating in some way, (dependent on the operation) the image pixels that are coincident with the elements of the SE, $\forall t \in T$.

The greyscale erosion of the image foreground can be described conceptually as a process of translating B_{FG} to a point $x \in E$ and raising the SE to the highest level t for which it is entirely beneath or fully (i.e. 100%) occupied by the signal. By this interpretation, the erosion of an image at any point $x \in E$ is equivalent to calculating the maximum level, t, for which the foreground SE, B_{FG}, is 100% occupied in the image. For the foreground erosion, B_{FG} is 100% occupied, if the intensity of all image pixels that are coincident with the elements of B_{FG} is greater than or equal to t. Let $l_{FG}(x)$ denote the maximum level, t, for which the SE is 100% occupied when its origin is at any point $x \in E$,

$$l_{FG}(x) = \max_{b_{FG} \in B_{FG}} \left\{ t \middle| \forall b_{FG} \in B_{FG},\ I(x + b_{FG}) \geq t \right\}. \qquad \textbf{(4.10)}$$

By definition, the HMT uses a foreground erosion to match patterns from below the topographic surface and a background erosion to match patterns from above. The background erosion can be described by a similar process of translating B_{BG} to a point $x \in E$ and lowering the SE to the lowest level t for which it is entirely above or fully occupied in the image. The background erosion at a point $x \in E$ is therefore defined as the minimum level, t, for which the background SE, B_{BG}, is 100% occupied in the image. For the background erosion, the SE is 100% occupied at some

$x \in E$ if the intensity of all the image pixels that are coincident with the elements of B_{BG} is strictly less than t. Let $l_{BG}(x)$ denote the minimum level, t, for which the background SE is 100% occupied when its origin is centred at a point $x \in E$.

$$l_{BG}(x) = \min_{b_{BG} \in B_{BG}} \left\{ t \middle| \forall b_{BG} \in B_{BG},\ I(x + b_{BG}) < t \right\} \tag{4.11}$$

This description of the HMT resembles the one given by Soille in [12] where he states that the UHMT is equivalent to the number of intersections of the intervals [0, t_{FG}] and [t_{BG}, t_{max}]. Soille defines that t_{FG} is the highest level where B_{FG} fits the foreground, t_{BG}, is the lowest level at which B_{BG} fits the background, and t_{max}, is the highest intensity in the image as determined by the bitdepth. In the definition of the HMT that is introduced in this work, l_{FG} and l_{BG} are equivalent to t_{FG} and t_{BG}, however, the HMT introduced in this thesis differs significantly in that it may be calculated in a single pass of the image with erosion and dilation implemented simultaneously using a composite SE. Further, an occupancy calculation is performed to ensure that the SEs are simultaneously 100% occupied in the image instead of using traditional morphological operations. In Equation (4.4)(4.4), the definition of the UHMT states that for a point to be marked by this greyscale HMT, the result of I eroded by B_{FG} must be greater than or equal to that of I dilated by B_{BG}^{*}. By analogy, the proposed HMT will mark an object as a "hit", iff $l_{FG}(x) \geq l_{BG}(x)$. The greyscale HMT that is presented in this thesis may now be written in terms of $l_{FG}(x)$ and $l_{BG}(x)$, $\forall x \in E$,

$$\left[HMT_B(I) \right](x) = \begin{cases} 2^n\text{-}1 & \text{if } l_{FG}(x) \geq l_{BG}(x) \\ 0 & \text{otherwise} \end{cases}. \tag{4.12}$$

The concept of this HMT is consistent with the greyscale HMTs defined by Ronse and Soille, however, in this new definition, the result is a binary marker containing the location of any objects that have been detected. This concept and process is illustrated in Figure 4.1.

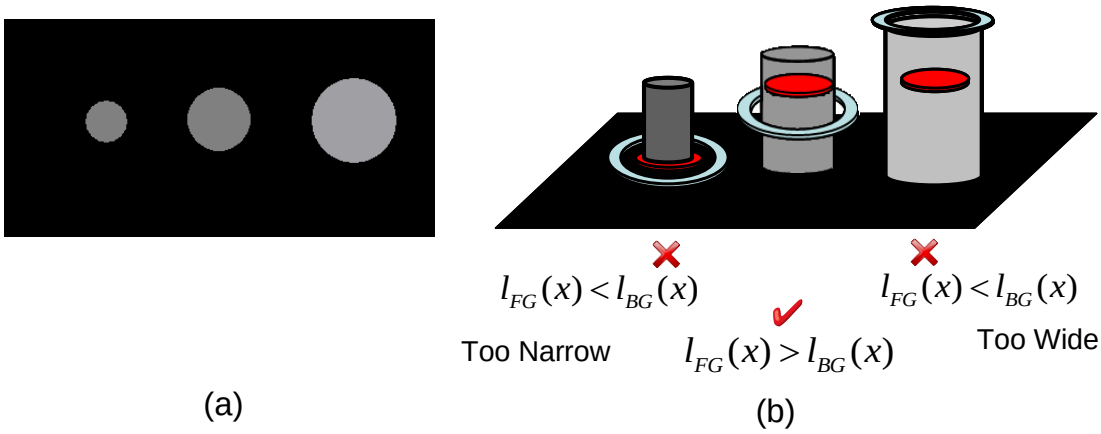


Figure 4.1 Greyscale HMT operating on a topographic surface. (a) A synthetic greyscale image. (b) Topographic representation of (a) with the HMT detecting the middle feature when $l_{FG}(x) \geq l_{BG}(x)$ which satisfies the condition in (4.12) that $l_{FG}(x) \geq l_{BG}(x)$ (ticks represent objects that are detected and crosses denote points that have not been marked).

By considering the HMT in terms of SE occupancy, the traditional implementation of the HMT (which requires two erosions) may be simplified by combining B_{FG} and B_{BG} into a unified, composite, SE (B, shown in Figure 4.2(a)). This composite SE, B, is then translated to each point x in the image. A point $x \in E$ is marked in the result if there exists a level, $t \in T$, which for all of the elements $b_{FG} \in (B_{FG})_x$, $t \leq I(b_{FG})$, while simultaneously, for all of the elements $b_{BG} \in (B_{BG})_x$, $t > I(b_{BG})$, i.e.

$$\left[HMT_B(I)\right](x) = \begin{cases} 2^n\text{-}1 & \text{if } \exists t \in T,\ \left[\forall b_{FG} \in B_{FG} \mid t \leq I(x + b_{FG}) \text{ and } \forall b_{BG} \in B_{BG} \mid t > I(x + b_{BG})\right] \\ 0 & \text{otherwise} \end{cases}. \quad \textbf{(4.13)}$$

This technique is illustrated in Figure 4.2 where a combined, composite SE is shown in Figure 4.2 (a) and an example of this SE discriminating between similar objects is shown in Figure 4.2 (b).

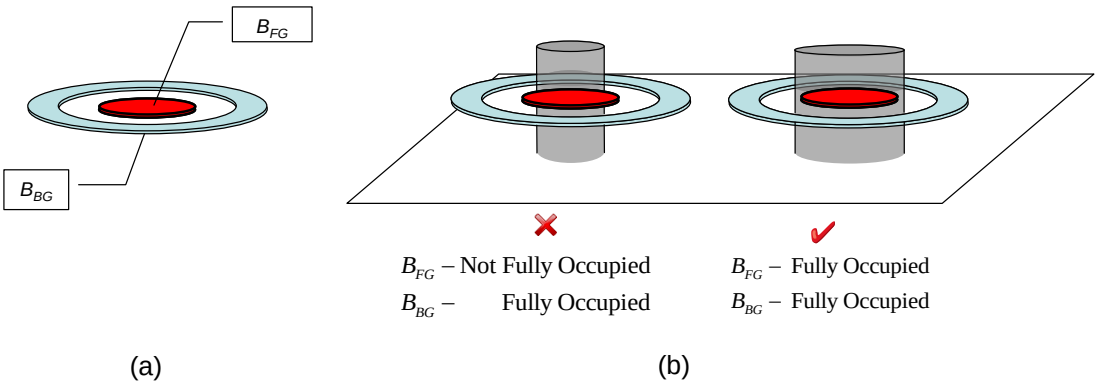


Figure 4.2 The HMT implemented using a composite SE. (a) The composite SE where elements of B_{FG} are shown in red and elements of B_{BG} shown in blue. (b) The SE searching for places where it is fully occupied in the image.

The definition of the greyscale HMT presented in this thesis allows this operator to be calculated in one pass of the image instead of the common two pass method followed by an intersection, summation or comparison of the two resultant images. As a result, the transform is simpler and more intuitive than the standard method.

4.3 The Hit-or-Miss-Transform in Noise

A "hit", i.e. an object which is detected and marked by the HMT, is one which satisfies the conditions outlined in Section 4.2. This strict definition of the HMT requires that the composite SE must be 100% occupied in both the foreground and background of the image for successful detection of an image feature. Often, when features are distorted by noise, or if image features contain a large amount of internal texture, it is not possible for B to be 100% occupied, even if its underlying geometry matches that of the feature. This causes the HMT to miss objects that should be detected as illustrated in Figure 4.3.

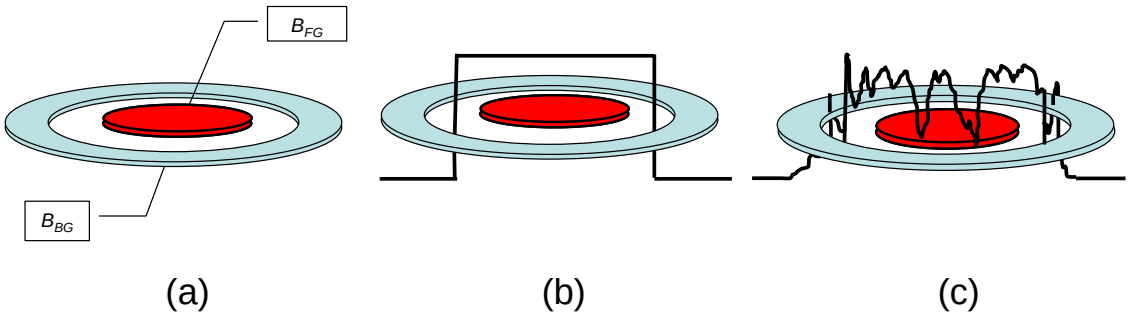


Figure 4.3 Operation of the HMT in noise (a) Composite SE that can be used to detect a circle (b) 100% occupied composite SE detecting the object of interest (c) Composite SE cannot be 100% occupied due to noise.

Figure 4.3(a) shows a composite SE which can be used to detect circular objects, provided that they lie within the boundary as defined by the disks, using the standard HMT. In the example, B_{FG} is a solid disk and B_{BG} is a surrounding solid ring. The black line in Figure 4.3 (b) represents a noise free shape which is to be detected using the SE shown in Figure 4.3 (a). In this case, the elements of B corresponding to B_{FG} are 100% occupied by the shape and the elements corresponding to B_{BG} are simultaneously 100% occupied by its background at all levels t, until t is greater than the intensity of the shape. This feature and any feature that has not been corrupted by noise and whose dimensions are greater than B_{FG} and less than that of B_{BG} will be detected by the HMT when using this SE.

In the case that an object of interest, its edges, or both are corrupted by noise, it is likely that the elements of B corresponding to B_{FG} and B_{BG} may never be simultaneously 100% occupied by the object. An example of this situation is illustrated in Figure 4.3(c) where both the foreground and background regions of the object shown in Figure 4.3(b) have been perturbed by noise. Since some of the foreground pixels within the object are at a level t that is lower than the level of its noisy background, there is no level t at which B_{FG} and B_{BG} can be simultaneously 100% occupied. As a result, the HMT will fail to recognise this feature as an object of interest. For the same reasons stated here, objects which have internal texture, such as biological cells, may also fail to be detected by the standard HMT.

4.4 A Percentage Occupancy Hit-or-Miss-Transform

In the standard HMT the foreground structuring element, B_{FG}, must fit entirely within the foreground of the object and the background SE, B_{BG}, must fit entirely within the background surrounding the object. In other words, they must be 100% occupied by the foreground and background respectively. Any noise, even just one pixel, in either the foreground or background of the object can prevent an otherwise legitimate hit occurring.

The idea behind the transform proposed in this thesis is to make the detection process of standard HMTs less sensitive to moderate amounts of noise (or texture) in the image. The idea proposed here is to relax the constraint that the SEs must be 100% occupied by allowing them to be only partially occupied and still record a "hit". Attempts at relaxing these strict constraints have been proposed in, [8], [11], [12], [20], [21], [25]. In this thesis a new concept of partial occupancy is introduced in the form of a Percentage Occupancy Hit-or-Miss Transform. The POHMT, which was first proposed in [78] and further developed in [25], allows a percentage of the SE to be "punctured" by noise or texture in a signal and still detect a "hit".

In this section, a method that can be used to calculate the extent to which B_{FG} and B_{BG} are occupied by a signal for all levels $t \in T$ when their origin is coincident with any point $x \in E$ is presented. The POHMT is then defined using this approach, to allow objects to be detected in places where the SE is only partially occupied by the signal.

4.4.1 Calculating the Occupancy of Structuring Elements

It has been shown that a greyscale HMT can be implemented with a single, composite SE, which searches the image to identify places where its foreground and background elements are simultaneously 100% occupied. By designing a SE to match the geometry of an object in both the foreground and background and, instead of using traditional morphological operations, the extent to which the object occupies the SE when it is coincident with an image feature is measured, it is possible to relax the 100% occupancy requirement of the HMT. This can be done in order to reliably

detect this object in the presence of noise provided that the measured percentage occupancy is greater than or equal to some threshold, *P*. To facilitate this explanation and its comprehension, B_{FG} and B_{BG} are considered separately before it is shown that the two can be combined (as in Section 4.2 of this Chapter) into a single operator, capable of processing an image in one pass.

The number of elements of B_{FG} that are occupied by a signal in the foreground can be calculated by translating B_{FG} to a point *x* in the image, and $\forall t \in T$, calculating the cardinality (*Card*) of the set, $\left(B_{FG}\right)_{x,t} = \left\{b_{FG} \in B_{FG} \mid I\left(x + b_{FG}\right) \geq t\right\}$, of image pixels which are coincident with B_{FG} and have intensity greater than or equal to *t*. For all $t \in T$, the foreground occupancy, $O_{FG_{x,t}}$, may be calculated using,

$$O_{FG_{x,t}} = Card\left\{\left(B_{FG}\right)_{x,t}\right\}. \tag{4.14}$$

By an equivalent technique, it is possible to measure the extent to which a feature occupies the background SE, B_{BG}, by calculating the background occupancy, $O_{BG_{x,t}}$,

$$O_{BG_{x,t}} = Card\left\{\left(B_{BG}\right)_{x,t}\right\}. \tag{4.15}$$

In this case B_{BG} is translated to a point $x \in E$, and $\forall t \in T$, the background occupancy, $O_{BG_{x,t}}$, is calculated by finding the cardinality of the set, $\left(B_{BG}\right)_{x,t} = \left\{b_{BG} \in B_{BG} \mid I\left(x + b_{BG}\right) < t\right\}$, i.e. the number of image pixels, coincident with B_{BG}, that have intensity less than *t*.

Using Equation (4.14) and Equation (4.15) produces two one dimensional arrays, $O_{FG_{x,t}}$ and $O_{BG_{x,t}}$, of length 2^n which contain the number of elements that are occupied by the signal in B_{FG} and B_{BG} respectively at each level *t* when centred on some $x \in E$. The elements of both arrays can be easily converted to percentages to produce the percentage occupancy of B_{FG} and B_{BG} for all $t \in T$ when their origin is

coincident with a point $x \in E$. The percentage occupancy of the foreground and background SEs are denoted $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ respectively where, $\forall t \in T$,

$$PO_{FG_{x,t}} = \frac{O_{FG_{x,t}}}{Card\left(B_{FG}\right)} \times 100 \tag{4.16}$$

and

$$PO_{BG_{x,t}} = \frac{O_{BG_{x,t}}}{Card\left(B_{BG}\right)} \times 100. \tag{4.17}$$

Since the cardinality of B_{FG} and B_{BG} is generally known, $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ may be calculated directly, $\forall t \in T$, using,

$$PO_{FG_{x,t}} = \left[\frac{Card\left\{b_{FG} \in B_{FG} \mid I\left(x + b_{FG}\right) \geq t\right\}}{Card\left(B_{FG}\right)} \right] \times 100 \tag{4.18}$$

$$PO_{BG_{x,t}} = \left[\frac{Card\left\{b_{BG} \in B_{BG} \mid I\left(x + b_{BG}\right) < t\right\}}{Card\left(B_{BG}\right)} \right] \times 100. \tag{4.19}$$

The advantage of working in a relative measure such as percentages is that when calculating the POHMT, being able to specify a minimum percentage of the SE that must be occupied for successful detection rather than the actual number of SE elements makes the transform more general.

Consider the following example showing the way in which the 100% occupancy requirement of the HMT can be relaxed using the quantities $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$. Figure 4.4(a) shows a noise free, synthetic, 8 bit greyscale image, containing a homogeneous circle on a uniform, dark background. By designing B_{FG} such that it can be contained entirely within this circle and the elements of B_{BG} to form a ring to encompass the disk, $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ may be calculated using Equation (4.18) and Equation (4.19). $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ have been plotted against intensity for the noise free feature in Figure 4.4(b) to illustrate how these quantities vary with *t*. By

observation of Figure 4.4(b), it is clear that B_{FG} is 100% occupied until $t = 150$ i.e. until B_{FG} is above the signal and B_{FG} is 0% occupied for $t > 150$. It is also clear that B_{BG} is 0% occupied until $t = 50$ i.e. it is completely below the background and is 100% occupied for $t > 50$ when the SE is completely above the background level of the image.

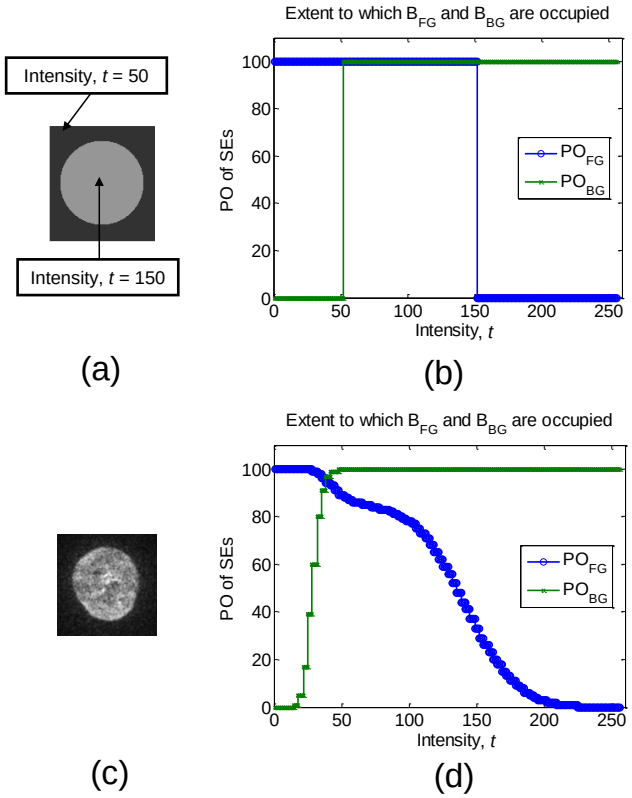


Figure 4.4 Images and plots of $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against intensity (a) Synthetic image (b) Plot of $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against intensity, *t*, for (a). (c) Noisy cell image (d) Plot of $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against intensity, *t*, for (c).

The image shown in Figure 4.4(a) is not perturbed by noise and the feature of interest does not exhibit internal texture and hence the standard HMT, using the SEs described, could be used to detect this object. Although this is obvious by simply observing the image, this point is further exemplified in the plot of the quantities $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ in Figure 4.4(b). By analysis of the plot, it is clear that there are a number of levels, *t*, 99 to be exact, for which the conditions in Equation (4.13) that there must exist some *t* for which both B_{FG} and B_{BG} are simultaneously 100% occupied are satisfied.

Calculating $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ and plotting these quantities against intensity is demonstrated further using the image of a noisy cell shown in Figure 4.4(c). Again, $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ have been plotted against intensity, and this is shown in Figure 4.4(d). Although it may not be obvious by visual inspection of this image, by examining Figure 4.4(d), it is clear that there is no level t for which $PO_{FG_{x,t}} = PO_{BG_{x,t}} = 100$ and hence the standard HMT will fail to detect this feature using the SEs described. It is in such cases where relaxing the 100% occupancy requirement of the HMT is of use. Allowing a transform to detect a feature if there exists some level t for which both B_{FG} and B_{BG} are occupied by some percentage $P < 100$ would allow this feature to be successfully detected despite moderate amounts of noise or texture that is present in the image. In the case of the noisy cell that is shown in Figure 4.4(c), it is clear by reference of the plot shown in Figure 4.4(d) of occupancy versus intensity, that setting $P < 100$ would allow this feature to be detected.

It is possible for B_{FG} and B_{BG} to be combined to form a composite SE, B, as in Section 4.2. This allows the extent to which the elements of B corresponding to the foreground, $b_{FG} \in B$, and background, $b_{BG} \in B$, of an image to be calculated simultaneously for all $t \in T$, in a single pass of the image. The highest percentage of B, denoted $PO_B(x)$ that is occupied by a signal for all $t \in T$, when its origin is coincident with a point $x \in E$ may be calculated by finding the maximum of the minimum, of all points plotted in Figure 4.4. This can be achieved by,

$$PO_B(x) = \max_{t \in T} \left[\min \left\{ PO_{FG_{x,t}},\ PO_{BG_{x,t}} \right\} \right]. \qquad \textbf{(4.20)}$$

This concept is illustrated in Figure 4.5 where Figure 4.5(a) shows the original plot of the quantities $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ when B is centred on the noisy cell shown in Figure 4.4(c). Figure 4.5(b) contains a plot of $\min\left\{PO_{FG_{x,t}}, PO_{BG_{x,t}}\right\}$ for these quantities and also points to the value that would be calculated using Equation (4.20) for these particular percentage occupancy measurements.

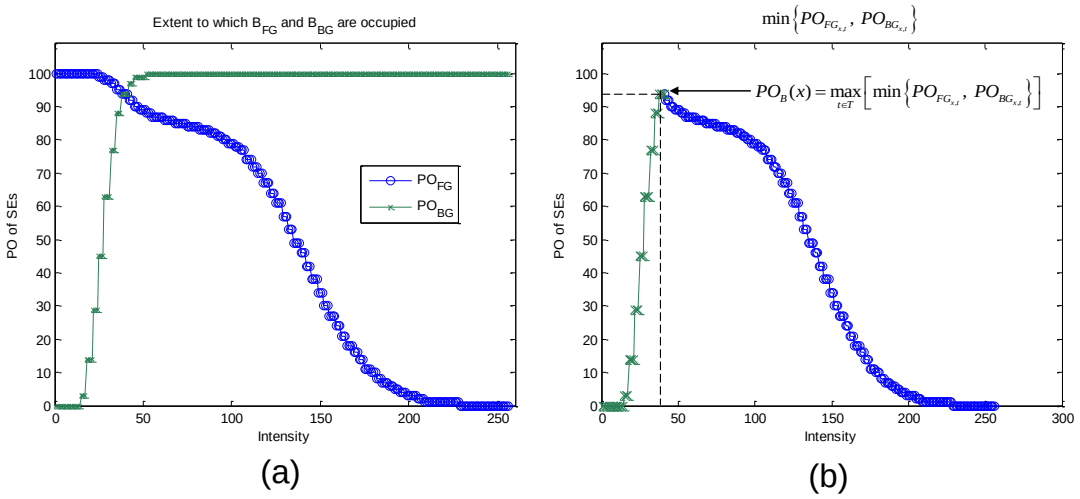


Figure 4.5 Illustration of calculation in Equation (4.20). (a) Plot of $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against intensity, *t*, when the SEs are centred on the noisy cell shown in Figure 4.4(c). (b) Plot showing the minimum of these quantities which is calculated in Equation (4.20) before taking the maximum point in this signal.

In Equation (4.20) the quantities $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ may be calculated $\forall x \in E$ using Equation (4.18) and Equation (4.19). $PO_B(x)$, which is calculated using Equation (4.20) returns the highest simultaneous percentage occupancy of both SEs at some level t when the SEs are centred on some $x \in E$. The POHMT, which is introduced in the next section, uses PO_B to determine whether or not a pixel at position $x \in E$ should be marked as a hit in the output of this transform.

4.4.2 The POHMT

The POHMT uses a composite SE to detect features in noisy images by allowing objects that occupy only a percentage, less than or equal to 100% of the SE, to be marked in the output of this transform. The POHMT can be calculated, $\forall x \in E$, using

$$\left[POHMT_{[B \in B_{FG} \cup B_{BG}]}(I) \right](x) = \begin{cases} 2^n\text{-}1 & \text{if } PO_B(x) \geq P \\ 0 & \text{otherwise} \end{cases}, \qquad \textbf{(4.21)}$$

where PO_B is calculated using Equation (4.20) and P is the minimum percentage of the SE that must be occupied for successful detection of an image feature. The value of P can be set by trial and error. Alternatively, if the power and distribution of noise

that corrupts a signal is known, an accurate value for *P* may be calculated using noise models, as will be demonstrated later in this chapter. A third technique, which is proposed in Chapter 5, is to measure an appropriate value for *P* using a novel design tool known as a PO plot and a set of training images that is representative of the real data. In this section, the effectiveness of setting *P* for the POHMT, by a process of trial and error is investigated. This value is then fixed and the POHMT is used to detect features in the image shown in Figure 4.6(a) which contains four cancer cells. From this point forward it can be assumed that in cases where images in figures have foreground intensity lower than the level of the background, that the images have been inverted for convenience when printing. Also note that the marker images in Figure 4.6 have been dilated by a 5x5 square such that the markers are visible.

By observation of Figure 4.6(a), it is clear that there are four cells in the image. Each cell has different characteristics in terms of shape, intensity and noise. To detect these cells using the POHMT, the geometry of *B* was set using prior knowledge of the shapes and sizes of the cells. The elements of *B* used to probe the foreground of the image were designed as a flat disk measuring 90 pixels in diameter such that it could fit inside that smallest cell in the image (bottom left cell in Figure 4.6(a)).

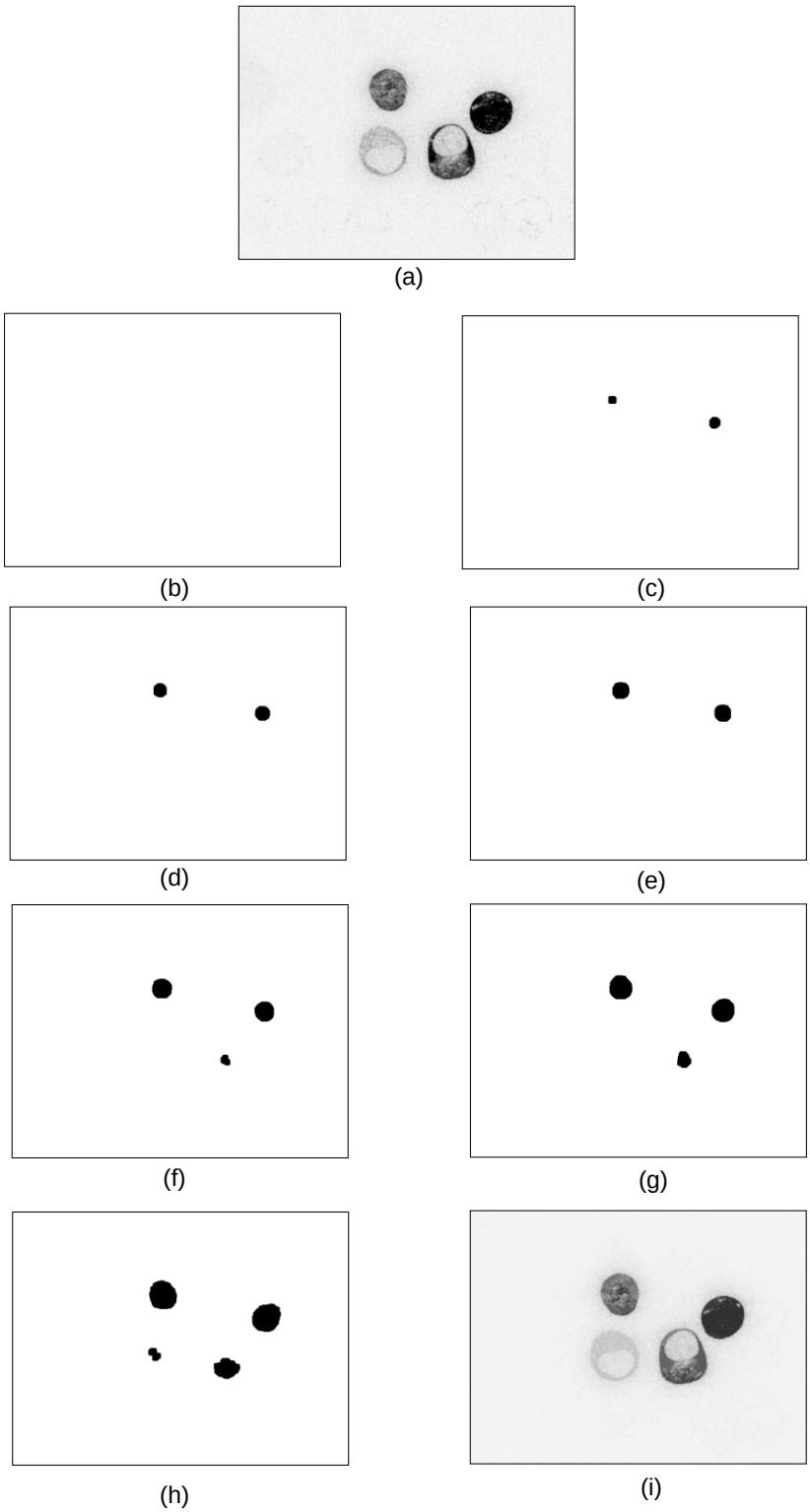


Figure 4.6 Example of POHMT operating on a noisy image where *P* is set by trial and error. (a) Original, noisy image. (b) Binary marker produced by the POHMT when *P* = 100 (c) Binary marker produced by the POHMT when *P* = 95 (d) Binary marker produced by the POHMT when *P* = 90 (e) Binary marker produced by the POHMT when *P* = 85 (f) Binary marker produced by the POHMT when *P* = 80 (g) Binary marker produced by the POHMT when *P* = 75 (h) Binary marker produced by the POHMT when *P* = 70 (i) Result of performing reconstruction by dilation using (a) as mask and (h) as marker. All images are 520 x 692 pixels.

The elements of *B* used to probe the background of the image formed a ring with an inner diameter of 110 pixels which was designed to encompass the largest cell (bottom right of Figure 4.6(a)). The POHMT was calculated $\forall x \in E$ with *P* set initially to 100%. *P* was then decremented in steps of 5 until four distinct groups of marker pixels – one for each cell – were present in the marker image that was produced by the POHMT at some value of *P*. The binary marker images that were produced by the POHMT after running it using a lower and lower value for *P* are shown in Figure 4.6.

Initially, when *P*=100, the POHMT detects no features in the image and resultantly produces an empty marker image as shown in Figure 4.6(b). When *P* = 95 the POHMT detects two of the cells in the image. When *P* = 90 the POHMT still only detects two of the cells in the image, however, the groups of marker pixels present in the image are larger than when *P* = 95. This is because more points in each cell occupy 90% of the SEs than 95%. When *P* = 80% the marker image produced by the POHMT contains three groups of marker pixel indicating that three of the four cells have been successfully detected when using this threshold. Finally, when *P* is set to 70%, the POHMT produces the binary marker shown in Figure 4.6(h). This marker image contains four groups of marker pixels in the same locations as each of the four cells in Figure 4.6(a). By this process of trial and error it has been identified that setting *P* to 70% allows successful detection of all four cells in this image and this information could perhaps be used to set the threshold *P* for other images in the same data set that exhibit similar levels of noise and texture. However, this process is simple only because there are only a small number of objects that are to be detected in the image. In an image that contains more features, with different levels of noise, this search would be exhaustive and inefficient.

Performing a reconstruction by dilation using the original image as the “mask” and the result of the POHMT as the “marker” produced the image shown in Figure 4.6(i). By reference of Figure 4.6(b) it is clear that unlike the POHMT, the standard HMT did not detect any of the cells as shown in Figure 4.6(b). It should be noted, that the POHMT is an extension of the HMT and hence the standard HMT can be

implemented as a special case of the POHMT. Setting *P* to 100% in Equation (4.21) and calculating the POHMT of an image will detect the same features as any of the greyscale HMTs discussed in Section 4.1 of this Chapter.

4.5 Setting an Appropriate Value for P Using Noise Models

As was mentioned earlier in this chapter, it is possible to estimate an appropriate value for *P* using noise models if they are available. Of course noise models can be obtained by manually selecting and analysing appropriate regions of the foreground and background from an image where the selected regions are assumed to be homogeneous. Here, it is implied that these regions should be homogeneous from an image analysis perspective. In most cases, what is considered to be noise by an image processing expert is the result of some expected process. In the context of this thesis, noise is any attribute that distorts the signal such that it is difficult or impossible to detect a feature using the standard HMT.

In this section, noise of increasing power is added to a synthetic image and this is used to illustrate the way that noise models can be used to estimate *P*. It is then demonstrated that by using selected regions of the image presented in Figure 4.6(a) to provide the noise models, it is possible to estimate the level of *P* that can be used such that the POHMT is able to detect all four cells in this image. This technique provides a robust alternative to the trial and error method that was used in the previous section to determine an appropriate level for *P*.

Figure 4.7 shows a synthetic image which contains a circle centred on a flat background as well as a 1D profile of the greylevels in this image taken from the centre row. Further, the normalized histogram that has been plotted using data that has been manually selected from a region of the foreground and a region of the background is shown. The regions that have been selected are shown in the figure.

There is no noise present in the image shown in Figure 4.7(a). This is confirmed in the profile plot shown in Figure 4.7(b) and in the histogram shown in Figure 4.7(c) which exhibits two distinct peaks. One peak in the normalized histogram is at

intensity 50 which represents the background and a second peak is at intensity 75 which represents the homogeneous circle in the foreground. Note that the histograms of each region have been normalised separately such that the sum of pixels for each region equals 1.

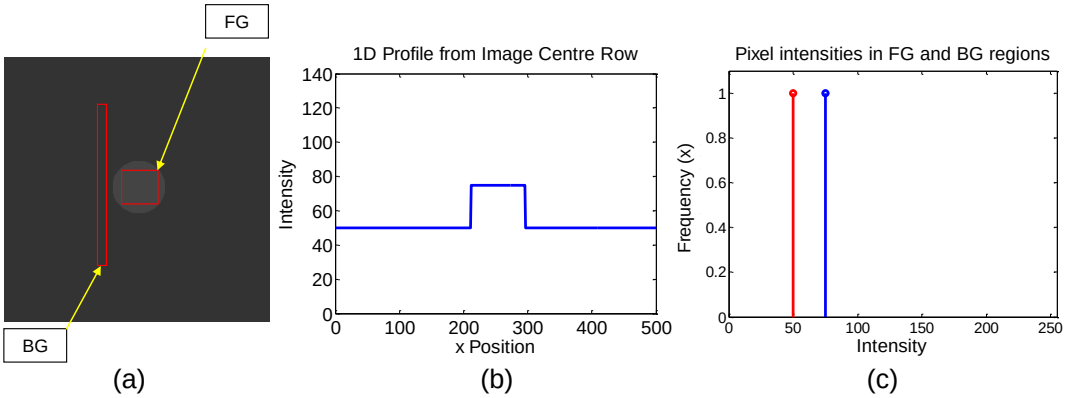


Figure 4.7 Noise free synthetic image (500 x 500) with 1D profile from the centre row and normalised histograms of FG and BG regions. (a) Synthetic, noise free image with BG at level 50 and a homogeneous disk at level 75. (b) 1D intensity profile plotted from the centre row of the image in (a). (c) Normalised histogram of manually selected regions of the FG (blue) and BG (red).

The circle that is shown in Figure 4.7(a) could easily be detected by the standard HMT provided that the SEs are appropriately designed such that B_{FG} is 100% occupied by the circle and B_{BG} is 100% occupied in its background.

If a small amount of noise is added to this image, it will still be possible to detect this feature without modifying the SEs or setting P to a value less than 100. Indeed P only ever needs to be lowered when the noise power is sufficiently large that it causes the intensity of at least one pixel of the foreground to fall below the level of just one pixel in the background. An example of adding a relatively small amount of noise to the image, that will not cause the standard HMT to fail, is shown in Figure 4.8. It is clear by reference of Figure 4.8(b) that although the signal is distorted, every point in the circle is strictly above the noisy background. The line that has been drawn in Figure 4.8(b) indicates one of a number of levels where both SEs could be 100% occupied by this image feature. This point is further exemplified in the normalized histogram of the foreground and background regions shown in Figure 4.8(c) since the distribution of the foreground and background pixel intensities do not intersect each other on the x axis.

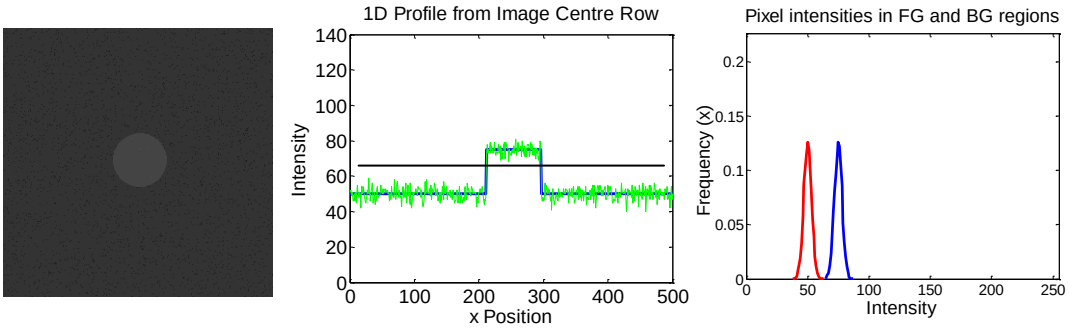


Figure 4.8 Synthetic image (500 x 500) corrupted by zero mean AWGN where, $\sigma^2 = 10$, with 1D profile from centre row and normalised histograms of FG and BG regions. (a) Synthetic, noisy image with BG average intensity of 50 and a homogeneous disk of average intensity 75. (b) 1D intensity profile of (a). (c) Normalised histogram of manually selected FG (blue) and BG(red) regions.

The information obtained from plotting the frequency of pixel intensity values in each region allows normalized histograms to be used as noise models. It is clear by reference of Figure 4.8(c) and Figure 4.9(c), (f), (i), (l), (o), that as noise power increases, the two distinct peaks that were presented in Figure 4.7(c) now resemble an almost Gaussian like distribution. As the noise power is increased, these two noise distributions move closer together on the x-axis. In cases where these two distributions overlap, the standard HMT would fail to detect a feature. The area of the overlap is shaded in purple in each of the aforementioned figures. As the noise power increases, so does the area of overlap between these two distributions of foreground and background pixels. Using a technique which somewhat resembles Maximum Likelihood Estimation (MLE), [79], allows the extent of this overlap to be used to estimate an appropriate value for *P*.

P may be estimated by first, manually selecting from the image, a region that is assumed to be homogeneous foreground, and a second region that is assumed to be homogeneous background. Then, using the following,

$$P = 100 - \frac{\left(\sum_{x=0}^{2^n - 1} \min\left\{ p_{FG}(x), p_{BG}(x) \right\} \right)}{2} \times 100, \tag{4.22}$$

where $p_{FG}(x)$ is the probability of a pixel in the foreground region taking the intensity *x*, and $p_{BG}(x)$ is the probability that a pixel in the background will take on intensity *x*. These probabilities are available in the normalised histograms which have been generated using the data that has been cropped from the image data and used as noise models. The quantity that is subtracted from 100 in Equation (4.22) is divided by two since the histogram of the foreground region and background region have been normalised separately.

Using Equation (4.22) to estimate a suitable value for *P* for each of the noisy images shown in Figure 4.9(a), (d), (g), (j), and (m) gives the value denoted in the plots that are shown in Figure 4.9(c), (f), (i), (l), and (o) respectively.

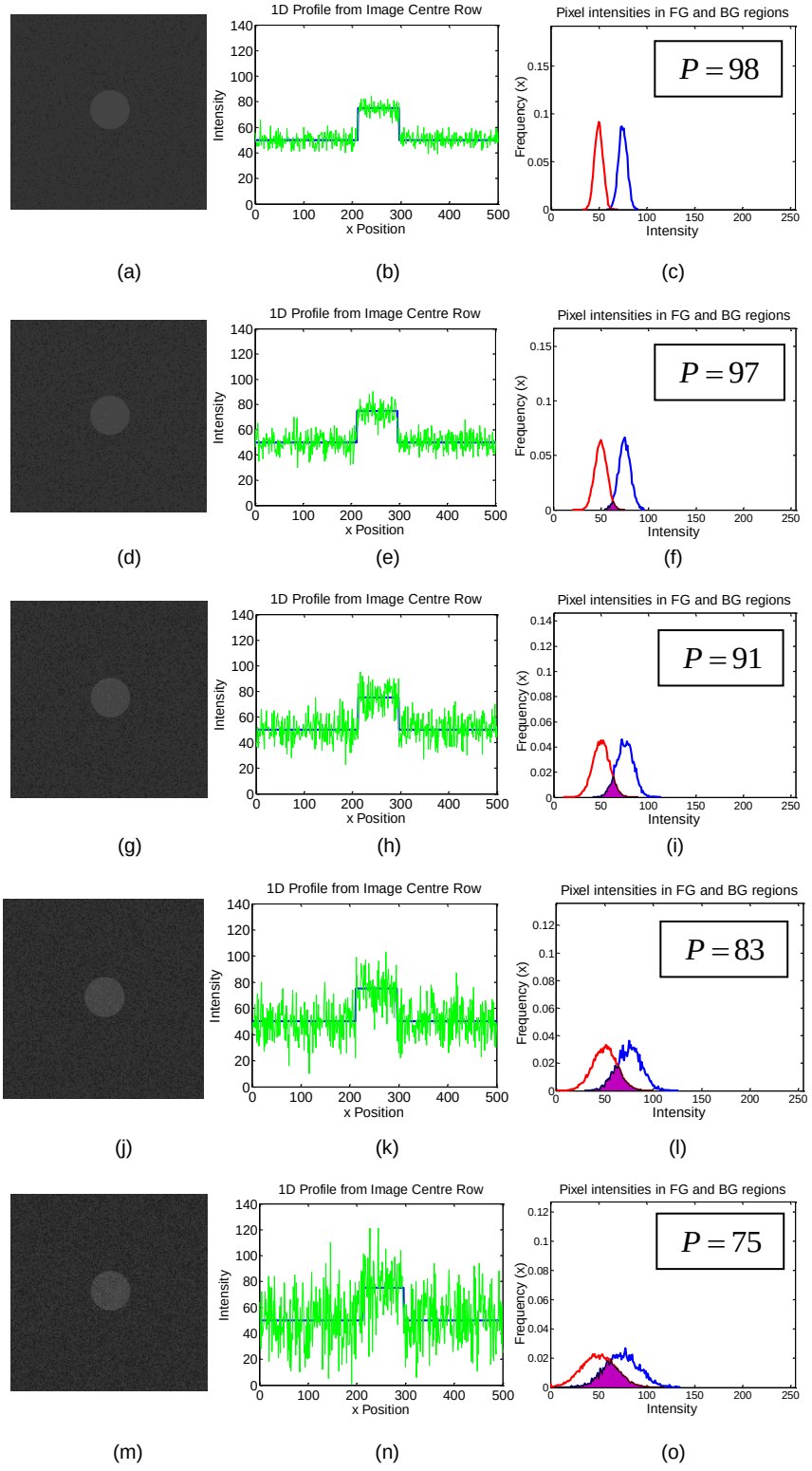


Figure 4.9 The effect which adding noise of increasing power has on the image, its profile and the normalised histograms of FG and BG regions in the image. Left – right respectively show the noisy synthetic image (500 x 500), its 1D profile and normalized histograms of the FG and BG regions. Top to bottom, examples of images that have been corrupted by noise of increasing power. (a) – (c) $\sigma^2 = 20, P = 98$, **(d) – (f)** $\sigma^2 = 40, P = 97$, **(g)-(i)** $\sigma^2 = 80, P = 91$, **(j) – (l)** $\sigma^2 = 160, P = 83$, **(m) – (o)** $\sigma^2 = 320, P = 75$.

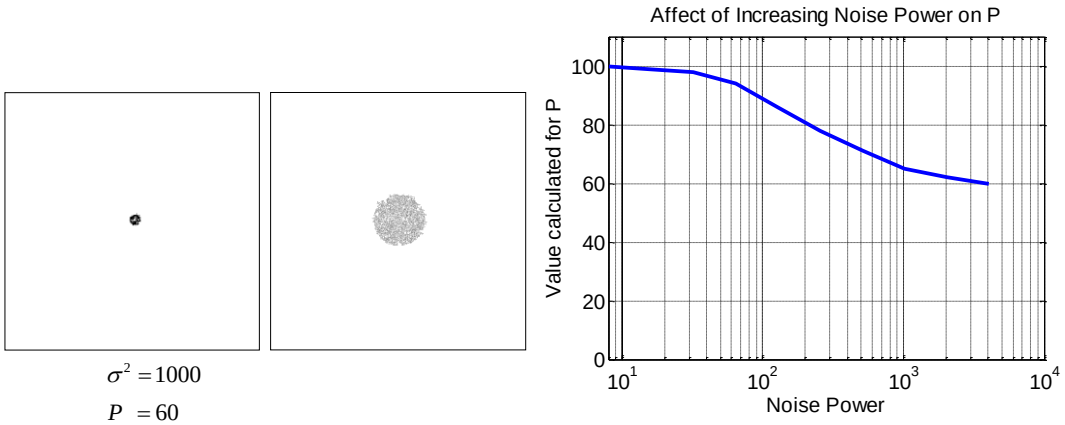


Figure 4.10 Result of using the POHMT to detect the circle in the noise corrupted images when using the noise models to estimate *P* for increasing amounts of noise. (a) One example of the output of the POHMT (500 x 500) for the case when the variance of the noise added to the image is 1000. (b) The result of applying a reconstruction by dilation using the marker and the original noise image (500 x 500). (c) The value calculated for *P* for increasing amounts of noise. N.B. The value that is calculated using this method becomes unreliable as noise power gets larger than 1000.

Setting *P* to the estimated values shown in Figure 4.9 and running the POHMT makes it possible to detect the noisy circle in each of the images. An example of the typical result of setting *P* to the values calculated using the noise models and executing the POHMT is shown in Figure 4.10(a). The result of applying a reconstruction by dilation using this marker and the original noisy image is shown in Figure 4.10(b). Figure 4.10(c) shows a plot of the values of *P* that have been calculated using Equation (4.22) when adding increasing amounts of noise to the synthetic image. As expected, the level of *P* that is estimated using the technique proposed in this section decreases as the power of the noise that is corrupting the image increases. It should be noted that when the noise power is greater than $\sigma^2 = 4016$ (SNR = -30dB), this method becomes unreliable for setting *P*.

It is now shown that it is possible to use the method proposed in this section to determine a suitable value for *P* when processing the cell image that was shown in Figure 4.6(a). Regions of the image foreground (cells) and regions of the background have been selected and used as noise models so that it is possible to estimate a value for *P* using Equation (4.22). The region selection is repeated for each cell in turn such that it is possible to obtain an estimate of *P* for all of the features that are

contained in the image. To save cross referencing, the image containing the four cells that are to be detected is shown in Figure 4.11(a).

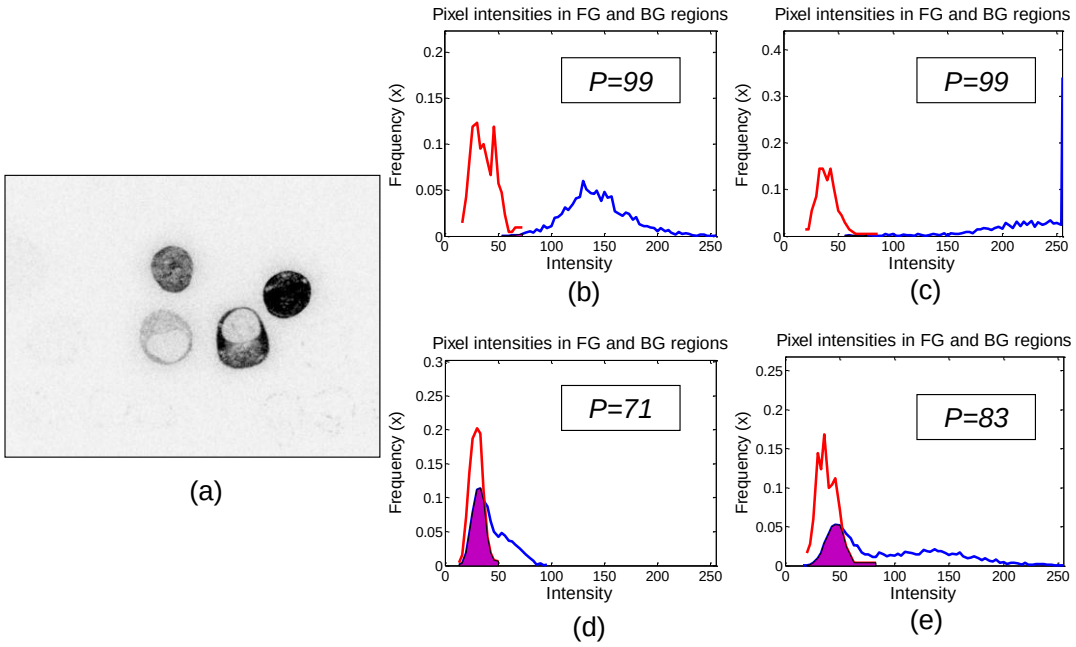


Figure 4.11 Using noise models obtained by manually selecting image regions and plotting their histograms to set *P*. (a) Noisy image containing 4 cells (520 x 692). (b) Noise models for the cell in the top left of (a) and the value of *P* calculated using Equation (4.22) and these noise models. (c) Noise models for the cell in the top right of (a) and the value of *P* calculated using Equation (4.22) and these noise models. (d) Noise models for the cell in the bottom left of (a) and the value of *P* calculated using Equation (4.22) and these noise models. (e) Noise models for the cell in the bottom right of (a) and the value of *P* calculated using Equation (4.22) and these noise models.

Figure 4.11(b) - (e) show the noise models that have been obtained from the image and used to estimate *P*, where each estimate of *P* is displayed in the corresponding plot. The minimum value that has been calculated for *P* is 71 (Figure 4.11(d)), and this is calculated when using the noise models obtained by selecting a region inside the cell in the bottom left of the image and a region of the noisy background surrounding this cell. Setting $P = 71$ and running the POHMT produces the marker image (dilated by a 5x5 square to improve visibility of the markers) that is shown in Figure 4.12 where a marker has been obtained for each cell that is present in the original image.

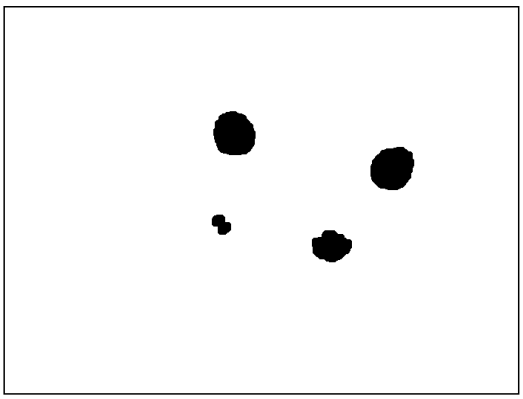


Figure 4.12 Marker image (dilated by 5x5 square) produced by POHMT when processing the image shown in Figure 4.11(a) and $P = 71$. The image dimensions are 520 x 692.

It is interesting and noteworthy, that all of the values that have been calculated for *P* using the noise models, lie within the intervals of *P* that allowed successful detection of these cells in the previous section (See Figure 4.6 and the discussion that follows it). That is, for the two cells in the top right of the image, $95 \le P \le 100$. The calculation of *P* from the noise models as shown in Figure 4.11(e) indicate that the cell in the bottom right of the image will be detected when $P \le 83$ i.e. $80 \le P \le 85$ which corresponds to the interval in which a marker for this cell appeared when setting *P* by trial and error in the previous section. Finally, Figure 4.11(d) shows that the cell in the bottom left of the image will be detected when $P \le 71$. This too correlates with the trial and error method where a marker for this cell appeared when $70 \le P \le 75$.

For a real world application, these methods would be used on training data, and when a suitable number of examples had been used to estimate *P*, this estimate would be used to set this parameter of the POHMT. The algorithm can then be executed on a separate set of test data to detect this feature, or features that are similar to those seen in the training data.

Although both the trial and error method for setting *P* and the technique using noise models have been validated here, they are not always applicable in practice. Setting *P* by trial and error becomes time consuming and onerous when the size of the training set or the number of different features increases. The noise models may also

be inaccurate since regions that are extracted/cropped from the image do not necessarily represent what is presented to the SE when it is coincident with a feature that is to be detected. In the next chapter, a novel design tool which uses training data in a similar fashion to the techniques discussed in this chapter, is introduced. This tool uses the structuring elements that are designed for feature detection in combination with the training data to estimate an appropriate value for *P*. The tool provides a number of advantages in addition to being more robust than the techniques suggested in this chapter for setting the parameter *P*.

4.6 Summary

In this chapter the standard HMT has been described and has been extended for accurately processing greyscale images. The extension that is presented here has been placed in the context of a number of existing greyscale HMTs which have been proposed in the literature.

Issues that affect the HMT when operating on noisy data have been highlighted. It has been shown that these difficulties may be overcome by the novel Percentage Occupancy Hit-or-Miss Transform that is presented here. A novel method for calculating the occupancy of SEs as they traverse a greyscale image has been described. It has also been shown both mathematically, and by example, how these occupancy calculations can be used to implement the POHMT.

Finally, two methods have been provided that can be used to set the single parameter *P* of the POHMT and the equivalences of the methods presented in this chapter have been demonstrated.

5 The PO plot

The previous chapter introduced a generalisation of the standard Hit-or-Miss Transform in the form of a Percentage Occupancy Hit-or-Miss Transform. It was shown, that so long as an appropriate value for its only parameter *P* is chosen, the POHMT offers improved performance over the standard HMT when using templates to match features in noisy data. Two methods for reliably setting this parameter *P* were demonstrated in Chapter 4, and these methods proved to work effectively on the example image. However, the empirical method of setting *P* iteratively until a suitable value was found proved to be exhaustive. Further, if the iteration step is too large, it is possible that this method could become unreliable as the complexity of the problem increases. Using noise models to estimate a suitable value for *P* is slightly more robust. However, despite the fact that these were demonstrated to work in the previous chapter, reliable noise models are not always readily available. Even in cases where noise models can be obtained by manual extraction of image regions, the regions that are used as models do not necessarily represent the noise that impacts on the SEs when attempting to match a sought feature.

This chapter describes a novel design tool that makes use of the SEs in order to accurately determine a suitable value for *P*. The tool is called a Percentage Occupancy Plot (PO plot) and it shows the extent to which a complementary pair of SEs is occupied by a signal and its background when centred on a feature in a given image. The profile of the plot differs with various noise distributions and different noise powers. This plot can be used to accurately estimate a suitable value for *P* for the POHMT. It can also be used to compute similar parameters for other generalizations of the HMT such as Soille's UHMT [12] or the BHMT [5] proposed by Barrat *et al*. Further, the PO plot can also be used to allow the POHMT to operate as a discriminatory filter which allows the transform to selectively mark and discard features based on the extent to which the SEs are occupied.

This chapter begins by describing how a PO plot can be generated and used to set parameters for the POHMT. It is then shown by example that the plot can be used to estimate similar parameters for other variations of the HMT that have been proposed.

Examples of how the PO plot can be used to make the POHMT operate in a discriminatory fashion are also provided. Each of the aforementioned uses/applications of the PO plot are exemplified in the final section of this chapter which provides experimental results that validate the theory.

5.1 The PO Plot

A PO plot can be generated using any complementary pair of SEs. To generate the plot, the SEs must be centred at some point $x \in E$ that corresponds to a feature of interest. When the SEs have been centred on some feature of interest, it is then required to compute $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ to obtain two, 1D arrays, of length 2^n. These arrays contain the percentage of the foreground and background SEs that are occupied at each discrete greylevel $t \in T$. When these calculations have been made for some feature of interest in the image, a PO plot can be obtained by simply plotting the discrete points in $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against each other, $\forall t \in T$, and interpolating. An example of generating a PO plot, as described, is shown in Figure 5.1.

The images, and the corresponding plots of $PO_{FG_{x,t}}$ and $PO_{BG_{x,t}}$ against intensity shown in Figure 5.1, are the same as those shown in Figure 4.4 of the previous chapter. They have been shown again here for consistency and to facilitate the understanding of exactly how the PO plot relates to the data that was computed in the previous chapter.

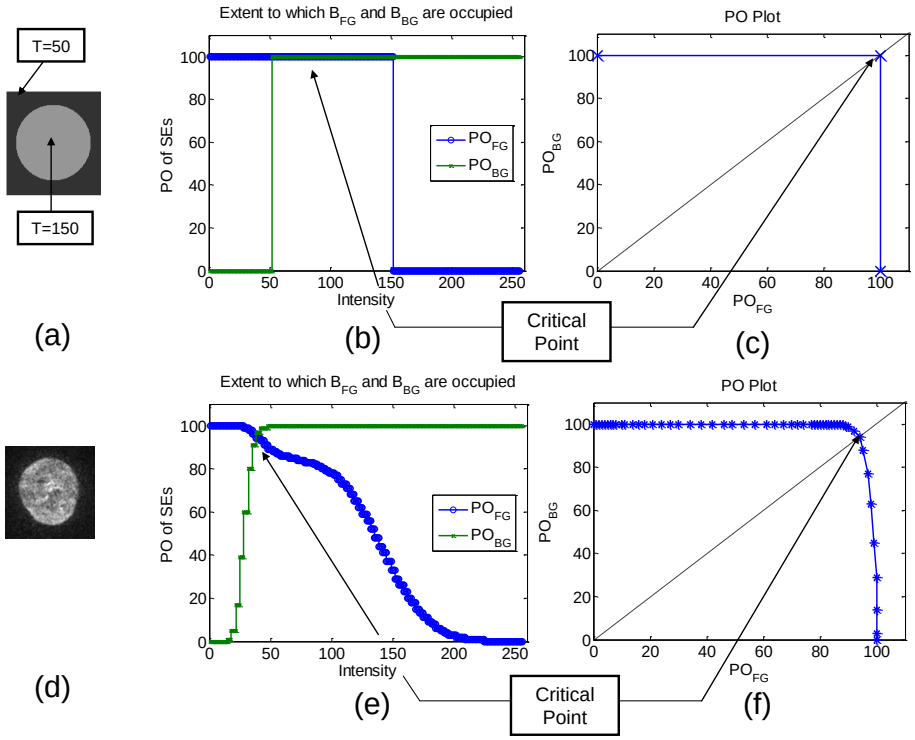


Figure 5.1 Images and their PO plot (a) Synthetic image (b) Plot of PO_{FG} and PO_{BG} against intensity for (a). (c) PO plot that is generated by plotting PO_{FG} and PO_{BG} against each other and interpolating. This PO plot indicates that the standard HMT will detect the noise free object. (d) Noisy cell image (e) Plot of PO_{FG} and PO_{BG} against intensity for (d). (f) PO plot indicating the HMT will not detect the cell. N.B. If the HMT will not be affected by noise ((a), (b) and (c)), the critical point (where curve intersects the diagonal in the PO plot) may be a set of points, the cardinality of which gives the number of times that the SEs fit the feature.

As was stated in Chapter 4, the image shown in Figure 5.1(a) is not perturbed by noise and the feature of interest does not exhibit internal texture, and hence the standard HMT, using the SEs described, could be used to detect this object. This is reflected in the PO plot since it shows that there is at least one level, *t*, such that when B_{FG} and B_{BG} are centred at $x \in E$, B_{FG} and B_{BG} are simultaneously 100% occupied. This is indicated in the PO plot shown in Figure 5.1(c) by the line forming a right angle which intersects the point on the 45° diagonal where, $PO_{FG} = PO_{BG} = 100$. However, if the image is corrupted by noise, the PO plot will not form the ideal right angle but will instead tend towards a curve. This is demonstrated using the image of a noisy cell shown in Figure 5.1(d). Again, PO_{FG} and PO_{BG} are plotted against intensity in Figure 5.1(e), and the corresponding PO plot, generated by plotting PO_{FG} *vs.* PO_{BG} and interpolating, is shown in Figure 5.1(f). In this case, by examining Figure 5.1(e), it is clear that there is no level *t* for which PO_{FG} = PO_{BG} = 100% and hence the standard HMT will fail to detect this

feature using the SEs described. This is reflected in the PO plot shown in Figure 5.1(f) since instead of forming the right angled profile shown for the noise free shape (Figure 5.1(c)), the PO plot tends towards a curve which crosses the $45°$ line where ($PO_{FG} = PO_{BG}$) < 100.

It should be noted that the so called critical point on the PO profile (Figure 5.1(c) and Figure 5.1(f)) is the point where the curve crosses the $45°$ line. This point is equivalent to the point at which PO_{FG} and PO_{BG} intersect in Figure 5.1(b) and Figure 5.1(e). The critical points and their equivalences in the plots are highlighted in Figure 5.1. It should also be noted that if the profile in the PO plot takes on the form of a right angle, then the critical point may in fact correspond to a set of points. The cardinality of this set is equal to the number of levels $t \in T$ for which both SEs can be 100% occupied when centred at this point in the image. This concept is clear by reference of Figure 5.1(b).

As the noise and texture increases, the distance between the curve and this ideal right angle increases. This property of the PO plot is demonstrated in Figure 5.2 where zero mean, Additive White Gaussian Noise (AWGN) of increasing power has been added to a synthetic image, similar to the one shown in Figure 5.1(a). It is clear by observation, that as the noise power increases to further corrupt the image, the profile tends more towards a curve and deviates further and further from the ideal right angle. More importantly, the value of the critical point (where the curve crosses the diagonal) decreases as noise power increases.

The PO plot shown in Figure 5.2 can be used to set the minimum percentage occupancy requirement, *P*, such that the circle may be detected using the POHMT and these SEs, even in very noisy conditions.

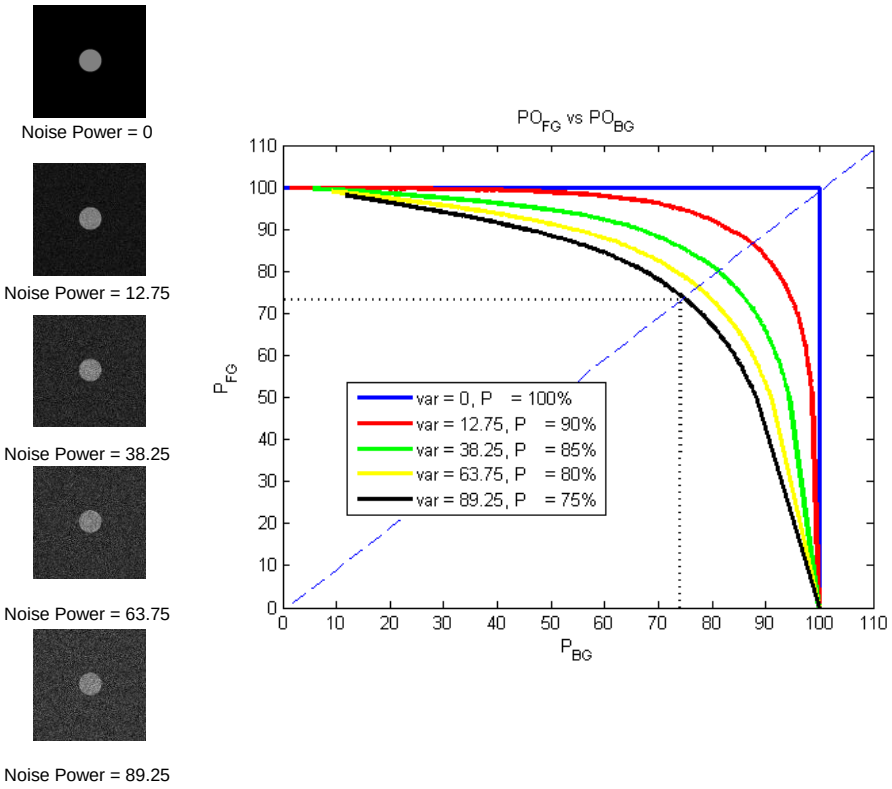


Figure 5.2 The affect that noise has on the PO plot. (left) Images (500 x 500) corrupted by AWGN of zero mean and increasing power. (right) Corresponding PO plots for the object in each image with increasing noise power.

Since there is only one object (the grey circle) in each image on the left of Figure 5.2, *P* can be set to the value of the critical point that is calculated for the feature in the noisiest image. That is, by reference of the PO plot in Figure 5.2, setting *P* to 75% guarantees that the grey circle will be detected by the POHMT in all five images. This is clear from the PO plot shown in Figure 5.2 which indicates that 75% is the lowest occupancy of the SE in all of the images. If there were other objects in the image, setting *P* so low may invoke erroneous hits in the images that are distorted by noise of lower power. Increasing *P* appropriately, in accordance with the critical points for each profile in the PO plot as noise power decreases, will reduce the likelihood of erroneous detection.

An example of generating a PO plot, using it to set *P*, and the result of applying the POHMT to a noisy image is presented in Figure 5.3. For consistency, the example in Figure 5.3 uses the image of cancer cells that was used in the previous chapter to demonstrate the alternative methods for setting *P*.

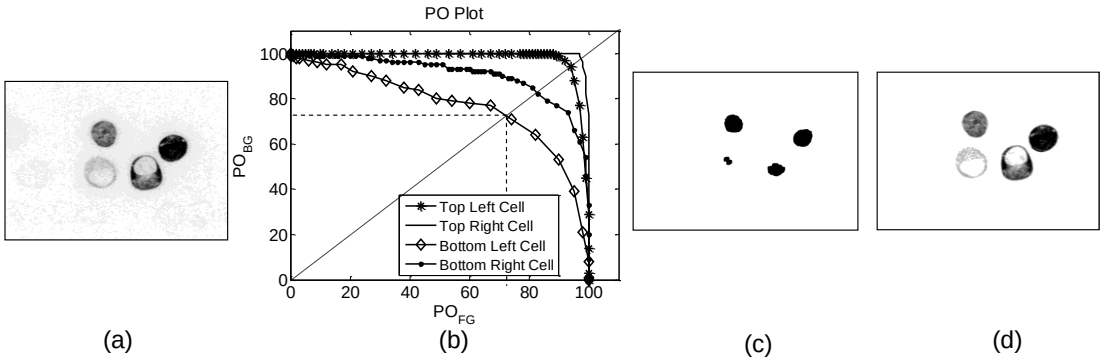


Figure 5.3 Example of applying the POHMT when using a PO plot to set *P*. (a) Original, noisy image. (b) PO plot showing the PO profile for each cell using the same composite SE. (c) Binary marker produced by the POHMT when *P* = 71%. (d) Result of applying the POHMT and performing a reconstruction by dilation. All images are 520 x 690.

By designing a complementary pair of SEs to best match the geometry of image features that are to be detected, the method described in the beginning of this section can be used to generate a PO plot for each object of interest in the image i.e. each of the four cells. The PO plot for each object can then be used to find a minimum value for *P* such that all features of interest will be detected in the image using the POHMT. This can be done by observation and manual interpretation of the PO plot, or alternatively, *P* can be determined automatically by calculating the critical points of each profile using Equation (4.20) and finding the minimum of these.

The SEs that were used to generate the PO plot shown in Figure 5.3(b) are the same as those described in Section 4.6 of the previous chapter. That is, B_{FG} is a solid disk measuring 90 pixels in diameter, and B_{BG} is a ring with an inner diameter of 110 pixels. The extent to which the cells occupied *B* was measured by centering *B* on each cell and in turn and using Equation (4.18) and Equation (4.19) to calculate PO_{FG} and PO_{BG}. The PO plot shown in Figure 5.3(b) was then generated where each profile in the PO plot corresponds to one cell in the image.

By calculating the minimum critical point, it is clear that setting *P* to any value less than or equal to 71% is sufficient to ensure that all four cells will be detected by the POHMT using *B*. It is also clear by observation of the PO plot, that the critical point for the detection of each cell falls within the intervals that were found using both the iterative method and the noise model technique for setting *P*. These intervals were

discussed and defined previously in Section 4.7 of Chapter 4.

Having generated a PO plot for the cells in the image and estimated an appropriate value for *P*, the POHMT was calculated $\forall x \in E$ with *P* set to 71%. The POHMT produced a binary marker (dilated by a 5x5 square), as shown in Figure 5.3(c) which contains four groups of marker pixels in the same locations as each of the four cells in Figure 5.3(a). Performing a reconstruction by dilation using the original image (Figure 5.3(a)) as the "mask" and the result of the POHMT (Figure 5.3(c)) as the "marker" produced the image shown in Figure 5.3(d).

Whilst this example has demonstrated the value of the POHMT for detecting features in varying amounts of noise, the PO plot was drawn from the data that was used to test the transform. In practice, the PO plot and the value of *P* would be estimated from a representative training set. Examples of this approach will be shown in Section 5.4.

5.2 A Design Tool for Existing Greyscale HMTs in Noise

In addition to the greyscale extensions of the HMT that have been presented by various researchers (as discussed in Chapters 3 and 4), numerous methods have been proposed which aim to generalize the HMT to make it more robust to noise. A few of these are discussed in detail here and a method which exploits the properties of the PO plot in order to set parameters for these methods is presented.

In [8], the authors, in reference to greyscale HMTs in general, present a "Generic solution to improve noise robustness". They state that the greyscale HMTs proposed by Ronse (RHMT) and Soille (UHMT) can be made more robust to noise if the distance between the two SEs, B_{FG} and B_{BG}, is increased. In [8], an example of how to modify this distance is given as, $B'_{FG} = B_{FG} - l$ and $B'_{BG} = B_{BG}$, or, $B'_{FG} = B_{FG}$ and $B'_{BG} = B_{BG} + l$, where *l* denotes some constant greylevel $t \in T$. However, no formula or method is provided that can be used to calculate an appropriate value for *l*.

It is possible to use the PO plot to determine an appropriate value for this parameter, *l*, by forcing the PO plot to form the ideal right angle. By calculating a distance, *d*, from the *PO* arrays, and shifting the elements of either PO_{FG} or PO_{BG}, by this distance, the PO plot is forced to form a right angle despite any noise or texture in the image. Provided that the critical point is less than 100 (i.e. the standard HMT will not work), the value *d* may be calculated as the difference between the lowest level, *t*, for which B_{BG} is 100% occupied, and the highest level *t*, for which B_{FG} is 100% occupied. More formally,

$$d = \min_{t \in T}\left(t \mid PO_{BG_{x,t}} = 100\right) - \max_{t \in T}\left(t \mid PO_{FG_{x,t}} = 100\right). \qquad \textbf{(5.1)}$$

The PO plot can then be forced to form a right angle by shifting the elements of either PO_{FG} or PO_{BG} by this distance *d*, to the right or left respectively, to obtain either PO'_{FG} or PO'_{BG}. By plotting PO'_{FG} vs. PO_{BG} or PO_{FG} vs. PO'_{BG}, the right angled plot is obtained which implies that by setting $l = d$, it is possible to accurately set the distance between the SEs to improve the noise robustness of the RHMT or the UHMT as suggested in [8]. That is, the level *d* that is calculated in order to force the plot to form the right angle, is equivalent to the minimum distance that must be fixed between B_{FG} and B_{BG} such that the feature of interest may be detected by either of these HMTs in the presence of noise. To demonstrate this technique, the synthetic image shown in Figure 5.4(a) has been used. This is a low contrast image that has been corrupted by zero mean, AWGN.

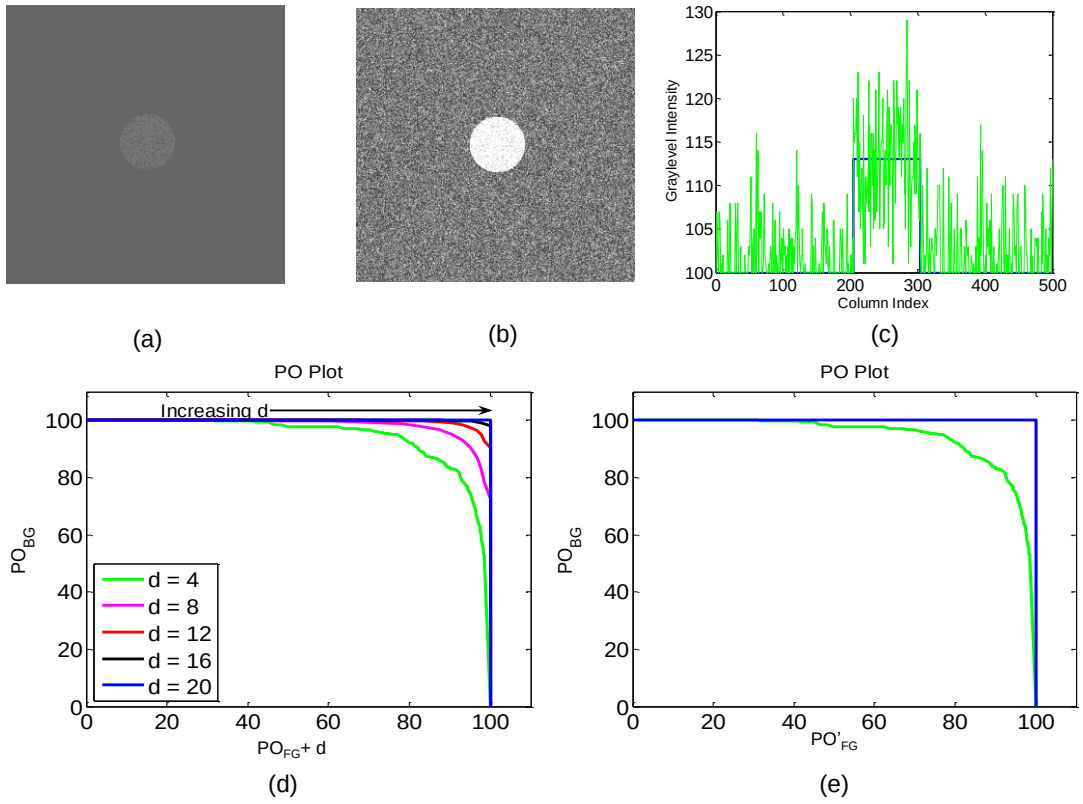


Figure 5.4 Setting the SE separation for Ronse's and Soille's HMTs. (a) Noisy synthetic image (500 x 500). (b) Histogram equalization of (a) for visualization of noise (500 x 500). (c) Intensity profile of image centre row, no noise (blue), noise corrupted signal (green). (d) PO plot obtained before (green) and after (blue) shifting the elements of PO_{FG} by d as well as intermediate plots for increasing d. (e) PO Plots obtained before (green) and after (blue) setting the SE separation to d and calculating PO_{FG} and PO_{BG}.

For the purpose of illustrating the effect that the noise has on this signal, the image after histogram equalization has been shown in Figure 5.4(b), and in Figure 5.4(c), a 1D intensity profile taken from the centre row of the image before and after the noise has been added is plotted. By generating the PO plot for the case where the distance between the SEs is initially zero, Equation (5.1) can be used to calculate the distance d that should be set between the SEs to allow this feature to be detected. By shifting the elements of PO_{FG} to the right by d and plotting PO'_{FG} vs. PO_{BG}, the result is the right angle as shown in Figure 5.4(d). To demonstrate the way in which the plot is forced to form the right angle, three additional curves are present in the PO plot in Figure 5.4(d). These curves have been generated purely for example by setting d to values that lie between zero and the critical distance of 20 greylevels that was calculated using Equation (5.1).

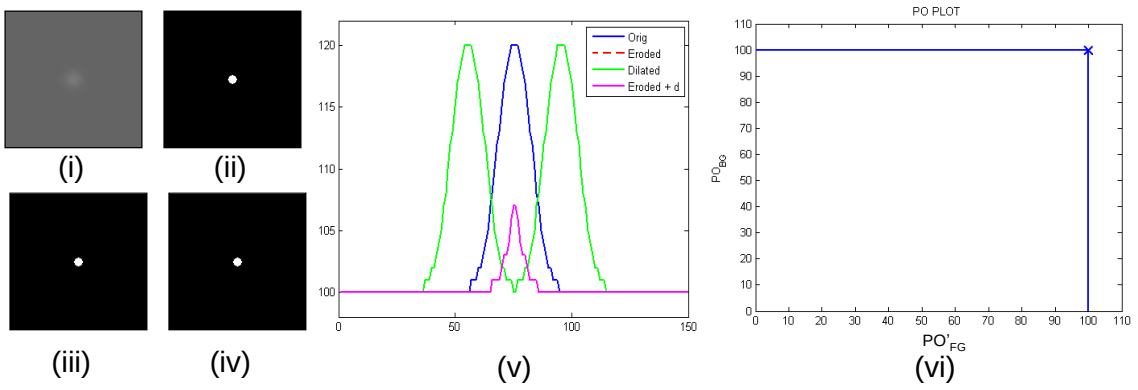
By reference of Figure 5.4(d), it is clear that as *d* increases, the plot gradually approaches the desired right angle before attaining this profile when *d = 20*. If the distance, *d*, between the SEs is sufficient to allow the HMTs defined by Ronse or Soille to detect the noisy feature, the PO plot that is generated after fixing this distance between the SEs and recalculating PO_{FG} and PO_{BG}, also forms the right angle. This case is shown in Figure 5.4(e) where the distance between the SEs was set to 20 greylevels before calculating PO_{FG} and PO_{BG}.

The authors of [8] discuss the performance of the HMTs proposed by Barrat *et al*. (BHMT) and Khosravi and Schafer (KHMT) when images are corrupted by noise. They conclude that since these HMTs already evaluate a distance between the SEs, the problem of finding a suitable distance is transformed into a problem of thresholding the result of their HMTs. It is possible to use the PO plot using the technique demonstrated in Figure 5.4 to determine this threshold, where the threshold is equivalent to the distance *d* which was previously calculated using Equation (5.1).

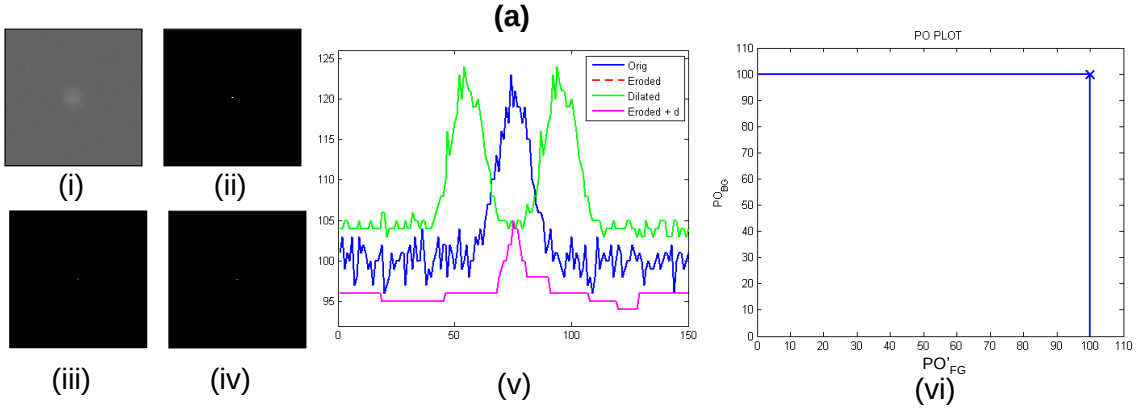
By performing the BHMT and the KHMT and thresholding the results at *d=20*, the disk in the centre of the image is successfully detected. However, a large number of erroneous detections appear in the result when using this technique. Thresholding the result of these HMTs at a level less than *d* does not allow successful detection of the circle in noise, however, the result still contains a high number of false positives. The same is true for the RHMT and the UHMT, where setting the distance between the SEs to be less than 20 greylevels results in erroneous “hits” while the feature of interest is not detected.

In [8], it is stated that the RHMT and the UHMT may be modified in the same way as each other for improved noise robustness – by increasing the distance between the SEs. The authors also state that the BHMT and the KHMT may be modified in the same way as each other – by determining the same suitable threshold. For these reasons and to simplify the following explanation, only the UHMT and the BHMT are discussed in more detail.

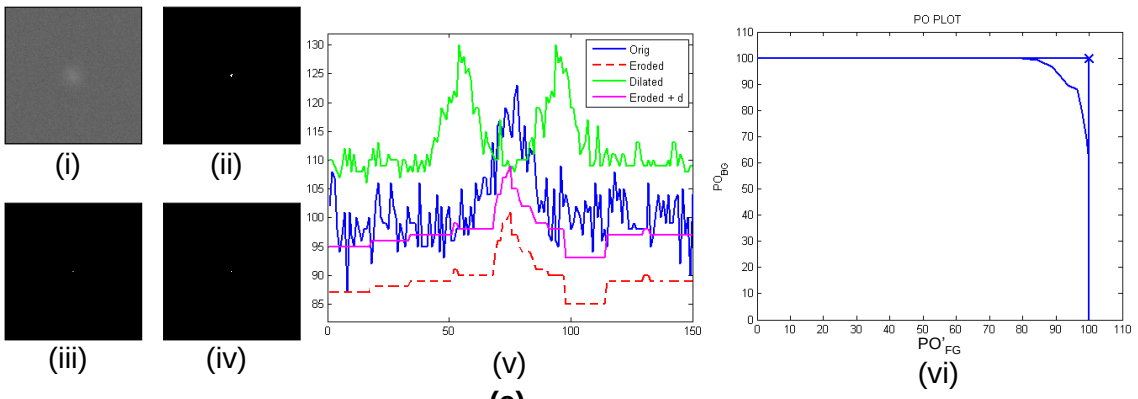
Figure 5.5(a)(i) shows a low contrast, noise free, synthetic greyscale image, that has a feature in its centre. The feature is a peak that is slightly brighter than the background and exhibits a Gaussian intensity profile. Increasing amounts of noise have been added to the images labelled (i) in Figure 5.5(b) – (e) in order to reduce the SNR and make it more difficult for the UHMT, the BHMT and the POHMT to detect this feature. In Figure 5.5(a)(vi) – (e)(vi) is a PO plot that has been generated using the SEs that have been designed to match the feature in (i) for each synthetic image.


(i)
(ii)
(iii)
(iv)
Orig
Eroded
Dilated
Eroded + d
(v)
PO PLOT
PO_BG
PO'_FG
(vi)

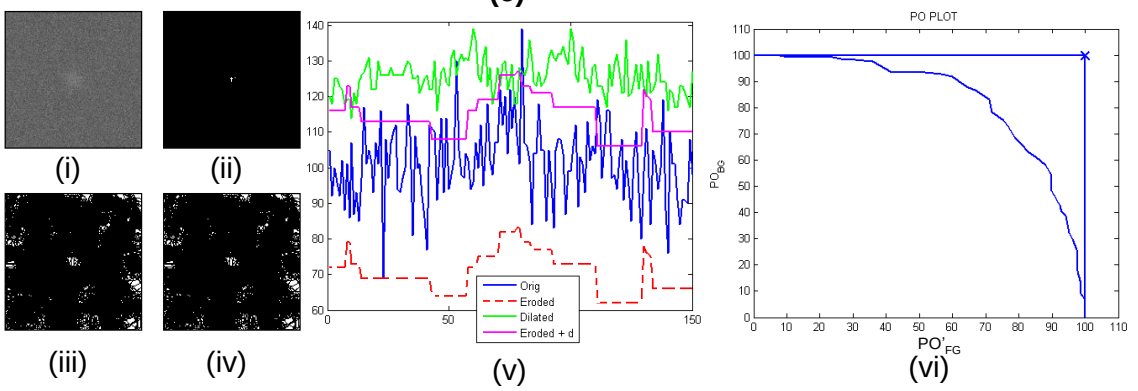
(a)


(i)
(ii)
(iii)
(iv)
Orig
Eroded
Dilated
Eroded + d
(v)
PO PLOT
PO_BG
PO'_FG
(vi)

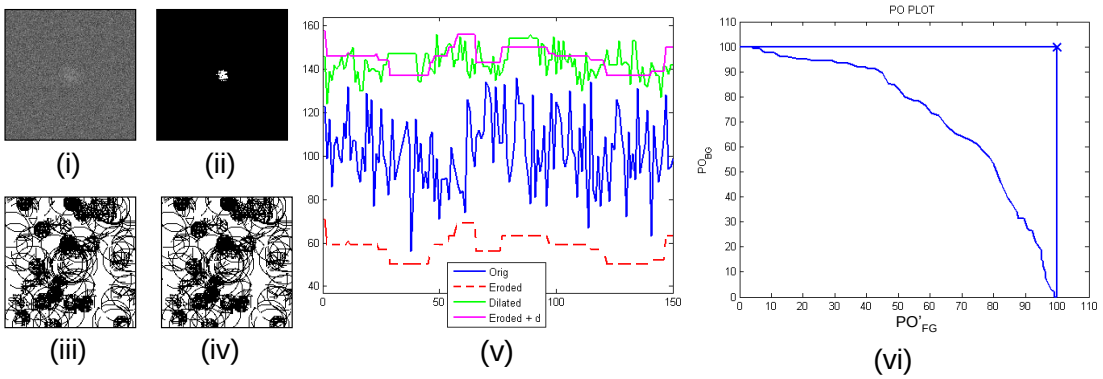
(b)


(i)
(ii)
(iii)
(iv)
Orig
Eroded
Dilated
Eroded + d
(v)
PO PLOT
PO_BG
PO'_FG
(vi)

(c)


(i)
(ii)
(iii)
(iv)
Orig
Eroded
Dilated
Eroded + d
(v)
PO PLOT
PO_BG
PO'_FG
(vi)

(d)



(e)

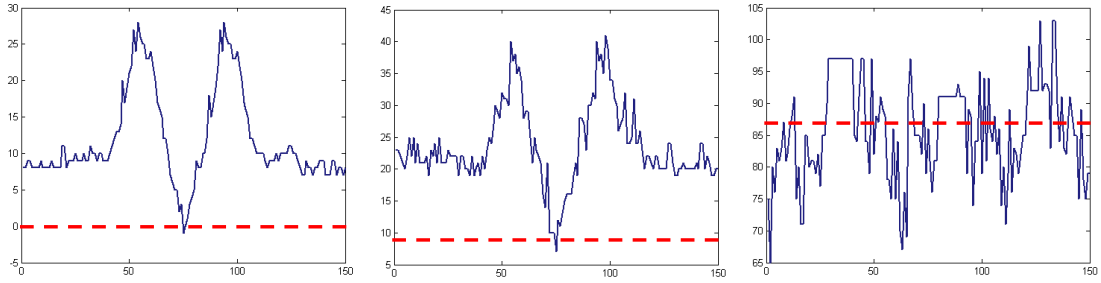
Figure 5.5 Example of using a PO plot to set parameters for the POHMT, the UHMT (Soille) and the BHMT (Barrat et. al) for improved performance in noise. (a) – (e) show a number of images (all 150 x 150) and plots when processing a synthetic image that contains a bright Gaussian profile where the SNR is gradually decreased. (a) SNR = ∞ dB; (b) SNR = 0dB; (c) SNR = -8dB; (d) SNR = -15dB; (e) SNR = -20dB. In each subfigure: (a) – (e), the image being processed is shown in (i), the result of the POHMT is shown in (ii), the result of the UHMT is shown in (iii) and the result of the BHMT is shown in (iv). A 1D intensity profile of the original image, *I*, (blue), $I \ominus B_{FG}$ (red), $I \oplus B_{BG}$ (green) and $I \ominus B_{FG} + d$ (purple) is shown in (v); and a PO plot is shown in (vi); Note that the result of the UHMT is equivalent to the that of the BHMT and that the POHMT outperforms the two modified alternatives when the SNR drops to -15dB.

It is clear that as the SNR decreases, the profile in each of the PO plots shown in Figure 5.5(a)(vi) – (e)(vi) tends more towards a curve and that the critical point becomes lower. For the POHMT, this means that the parameter *P* should be reduced in accordance with the value of the critical point to improve its performance and allow successful detection of the feature despite the noise. For the UHMT to be capable of locating the feature in noise, the distance between the SEs should be increased [8]. It should be noted that increasing the distance between the SEs is equivalent to vertically translating the eroded image by adding a greylevel, equal to the distance between the SEs, to the result of $I \ominus B_{FG}$. This method of fixing a distance between the SEs has been applied such that *d*, as computed using Equation (5.1), is added to the result of $I \ominus B_{FG}$. The right angle profile that is shown in each PO plot (Figure 5.5(a)(vi) – (e)(vi)) is the result of plotting PO'_{FG} vs. PO_{BG} where PO'_{FG} was obtained by shifting the elements of PO_{FG} to the right by the value of *d* which is calculated for each profile. For the BHMT, the threshold should be increased as the SNR decreases in order to improve its robustness to noise [8]. As has been explained, this threshold is equivalent to the distance which is allowed between the SEs for the UHMT and hence it too may be calculated using Equation (5.1).

Figure 5.5(a)(v) – (e)(v) shows a one dimensional cut (taken from the centre row) of the image labelled (i) in Figure 5.5(a) – (e) in blue. In addition to the profile of the image being processed, the plot labelled (v) in Figure 5.5(a) – (e) shows the intensity of the result of eroding this image by B_{FG}, the result of dilating by B_{BG}, and the result of eroding the image and adding d to the result (increasing the distance between the SEs). Labels (ii), (iii) and (iv) in Figure 5.5(a) – (e), show respectively, the result obtained by the POHMT, the UHMT and the BHMT when attempting to locate the feature in noise when using the PO plot to set their parameters.

It is clear that as the SNR drops to -15dB (Figure 5.5(d)) the UHMT and the BHMT begin to mark places in the image incorrectly where the POHMT detects only the feature of interest. The reason for these erroneous detections is evident when observing the one dimensional cuts shown in Figure 5.5(d)(v). Take for example the UHMT which marks places in the image when the result of $I \ominus B_{FG}$ (red profile) is greater than $I \oplus B_{BG}$ (green). The only time the standard transform is capable of detecting the sought feature is when the SNR ≥ 0, see Figure 5.5(a)(v) and Figure 5.5(b)(v). In Figure 5.5(a) and Figure 5.5(b) the PO plot exhibits a right angle and hence the standard HMT is capable of detecting the feature, and $d = 0$. That is, the SEs are at the same level as each other, hence, $I \ominus B_{FG} = I \ominus B_{FG} + d$. For this reason the red profile is not visible in Figure 5.5(a) and Figure 5.5(b). When the SNR falls to -8dB (Figure 5.5(c)) it is clear that the traditional implementation of the UHMT will not detect the feature in the image since the result of $I \ominus B_{FG}$ (red) is always lower than $I \oplus B_{BG}$ (green). However, increasing the distance between the SEs as suggested in [8] allows the UHMT to detect the feature. By observing Figure 5.5(c)(v) it is clear that the result of $I \ominus B_{FG} + d$ (purple profile) is greater than the result of $I \oplus B_{BG}$ only when the SE is coincident with the feature in the centre of the image. This is a positive result which indicates that the method of increasing the distance between the SEs can help the UHMT to perform more accurately in noise. However, when the SNR falls as low as -15dB, the distance that must be allowed between the SEs is so large that the transform introduces a number of erroneous markers. By observing Figure 5.5(d)(v) the reason for this is obvious – the result of

eroding the image by B_{FG} is so noisy that it is not possible to offset this result without inevitably creating a large number of points for which $I \ominus B_{FG} + d$ (purple profile) is greater than $I \oplus B_{BG}$ (green profile). When the SNR is as low as -20dB, it is evident that both the UHMT and the BHMT produce so many false positives that it is not practical to use either of these in such noisy conditions. This is clear by observation of the green and purple profiles shown in Figure 5.5(e)(v) where it is obvious that there exists a large number of points for which $I \ominus B_{FG} + d > I \oplus B_{BG}$. Each of these points produces an erroneous positive result in the output of the UHMT. It should be noted that although it is possible to reduce, *d*, and that this will reduce the number of false positives, it will also mean that the feature of interest will likely not be marked in the output of the UHMT. The discussion here has focused on the modifications that have been made to the UHMT, however, the same applies to the BHMT as these transforms are equivalent [8]. The equivalence is clear as in all cases the results of the transforms (although implemented differently) always produce identical output images. For the BHMT, as the SNR falls, the result is so noisy that it becomes impossible to set a threshold that will detect only the feature of interest. To exemplify this point, Figure 5.6 shows a 1D intensity profile of the output of the BHMT when the SNR is 0dB, -8dB and -20dB. It is clear from Figure 5.6(c) that when the SNR is -20 dB, it is impossible to accurately set a threshold that will allow only the feature of interest to be detected with no false positives.



(a) (b) (c)

Figure 5.6 1D profile of the result of BHMT and the calculated threshold, *d*, when operating on the images shown in Figure 5.5(b)(iii), Figure 5.5(c)(iii) and Figure 5.5(e)(iii). Each plot shows the result of applying the BHMT (blue) and the threshold level which was calculated using (5.1) when the SNR is (a) 0dB, (b) -8dB and (c) -20dB

It has been demonstrated using Figure 5.5 and Figure 5.6 that as the SNR falls below -15dB the modified UHMT and BHMT become unusable. The POHMT on the other hand is still capable of locating the feature of interest without any erroneous hits (see Figure 5.5(a)(ii) – (e)(ii)). These results indicate that the POHMT is better suited to making the HMT more robust in the presence of noise than the modifications suggested in [8]. However, the key point here is that until now, no one has provided a method that can be used to set the parameters that must be determined prior to executing any of the routines discussed here when the image data is very noisy. The PO plot that has been introduced in this work can be used to set these parameters.

Perret *et al*. present their solution to overcome the difficulties faced by the HMT in the presence of noise in [8]. As was explained in Chapter 3, the authors introduce a Fuzzy Hit-or-Miss Transform which they use to detect features in very noisy astronomical images and provide an impressive set of results. Their technique involves generating a large set of SEs using a mathematical model that incorporates the characteristics of the features they aim to detect for various scale lengths, orientations and elongations. A measure of fitness (in a fashion similar to the POHMT) is obtained for all patterns in the set of SEs at each point in the image, and a record of the best fitting SE at each pixel is stored as well as a measure of how well this SE fits the image. This data is used to form a so called "Score map" which is thresholded at a particular level to produce a binary marker image from which detected features can be reconstructed.

For the FHMT, the PO plot could be used to set the ideal distance between the SEs, or to provide an indication of a suitable threshold that can be used on the output of this transform. Additionally, the PO plot and a suitable set of training data could be used to set a minimum occupancy requirement for a single SE, or at least a small subset taken from the large set of SEs that are used currently. This would allow their algorithm to execute more quickly as it would no longer require the use of such a large number of SEs.

The PO plot could also be used to estimate the optimal rank parameter for the method presented by Harvey *et al*. in [11]. This would significantly reduce the time taken by their current approach which uses ROC curves to assess the effect of the rank parameter when applying their generalised HMT to detect ships in satellite imagery. The POHMT can in fact be used to estimate parameters for a large number of the modifications of the standard HMT which were discussed in Chapter 3. Examples of setting parameters for some of the routines that have been discussed in this section, using the PO plot, are provided in Section 5.4 of this chapter.

5.3 A Discriminatory Filter

Often, features that are to be detected in an image are not geometrically identical. If, therefore, it is required to extract from an image, a number of features which differ from each other in terms of shape and size, it is possible to design a number of composite SEs where each one is designed to match the geometry of each object that is sought in the image. It is then possible to perform a greyscale HMT using each of the composite SEs in turn before calculating the union of all the resulting binary images to obtain a single image that contains markers for each image feature that has been detected. That is of course assuming that the HMT will not fail to detect these features due to noise or texture in the image.

As was illustrated in Figure 5.3, the POHMT allows multiple objects which are geometrically very different to be detected using just one composite SE in a single pass of the image. This can be achieved by exploiting the information contained within the PO plot in order to determine an appropriate level for P such that it is possible to ensure the detection of all the features in this image. The PO plot provides a further advantage in that P may be set in such a way that it is possible to discriminate between image features using just one composite SE. The simplest case of discriminating between features using the POHMT is to set the value of P high enough to eliminate objects which simultaneously occupy a maximum percentage of B that is always less than P. An example of selectively detecting cells in the image by varying P, using the information contained in the PO plot, is shown in Figure 5.7.

Clearly, by reference of Figure 5.7(a), setting the level of P to any value that lies between the critical points of the curves representing the cell in the bottom right and left of the image, the cell in the bottom left can be eliminated by the POHMT while the other three cells can be detected successfully. In Figure 5.7(c) only three of the four cells shown in Figure 5.7(b) have been extracted by setting $P = 75\%$ in order to eliminate the cell in the bottom left of the image for which the maximum, simultaneous occupancy of the SE when coincident with this cell is 71%. By raising the level of P to 90% and then 96% in accordance with the PO plot shown in Figure 5.7(a), respectively two cells are extracted (Figure 5.7(d)) and then only one cell is extracted (Figure 5.7(e)). Evidently, the PO plot is an extremely powerful design tool, as it provides information that allows objects to be detected selectively using one composite SE in a single pass of the image.

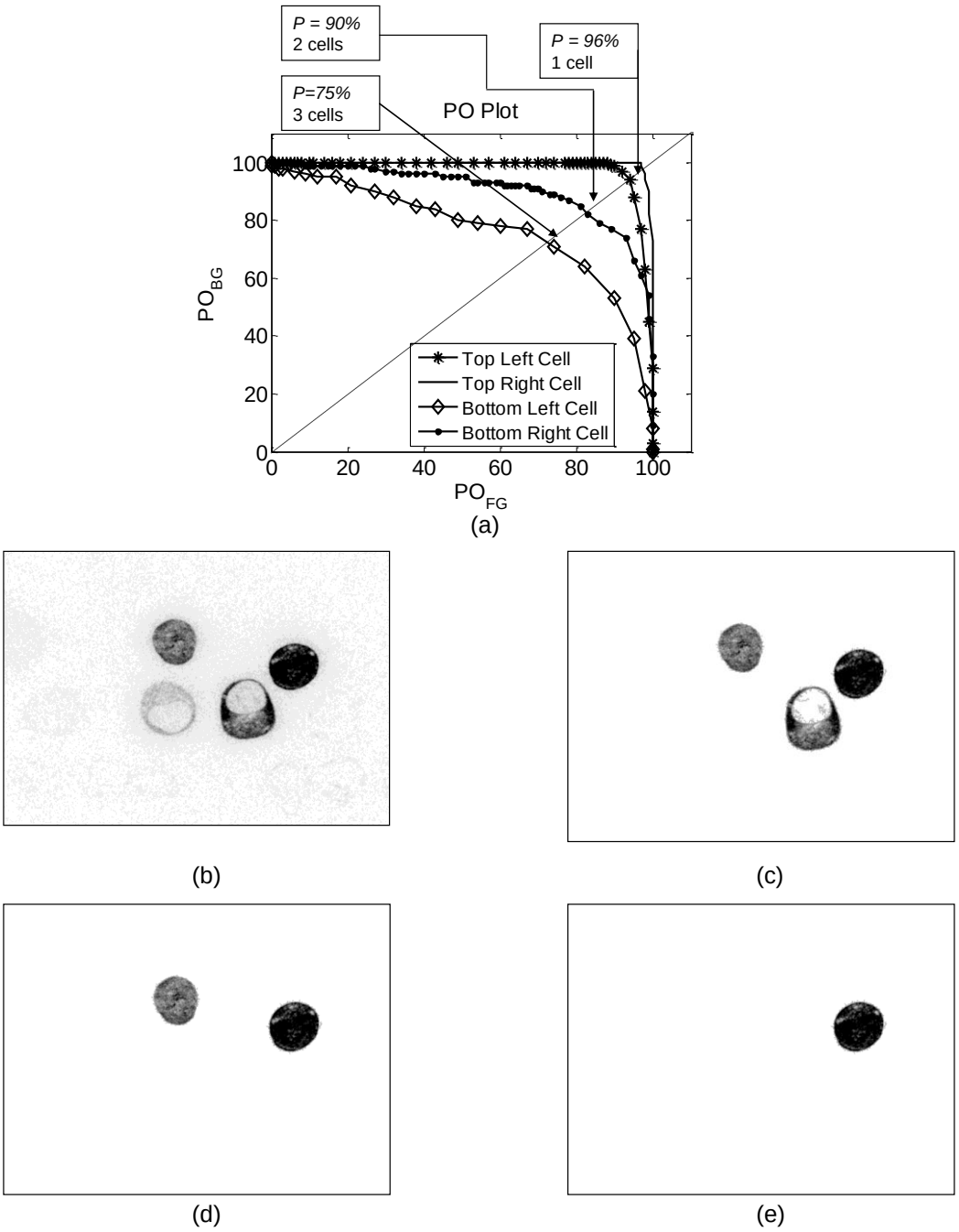


Figure 5.7 Example of POHMT operating as a discriminatory filter. (a) PO plot for the four cells in (b). (b) Noisy image containing four cells. (c) Three of the four cells detected by setting *P* = 75%. (d) Two of the cells detected by setting *P* = 90%. (e) One of the cells detected by setting *P* = 96%. All images are 520 x 692.

The case demonstrated in Figure 5.7 is a powerful yet simple one. It is obvious that increasing the level *P*, or in other words increasing the strictness of the transform, results in objects being discarded in the detection process. What is more interesting, is that by a similar technique to the one described above, it is possible to isolate any of the four cells in the image shown in Figure 5.7(b) and hence any combination of the image features can be segmented using just one composite SE. In this case, the PO plot makes it possible to use one composite SE to discriminate between objects of interest in an image and objects which may have very similar geometrical properties in the spatial domain, like the two cells at the top of the image. Figure 5.8 shows each of the cells being extracted on their own using the same composite SE and the POHMT.

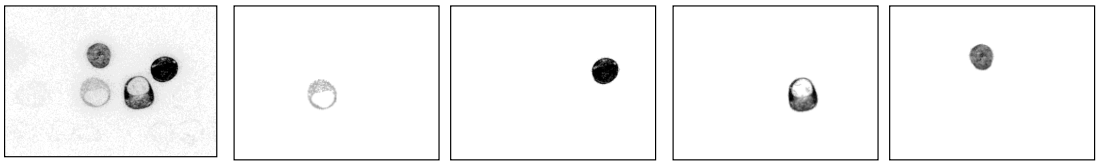


Figure 5.8 POHMT operating as a discriminatory filter (a) Image containing four cells of different shape and size. (b) Bottom left cell isolated. (c) Top right cell isolated. (d) Bottom right cell isolated. (e) Top left cell isolated. All images are 520 x 692.

The results shown in Figure 5.8 can be achieved by a few simple steps which will now be explained. Consider the result shown in Figure 5.8(d) where the cell in the bottom right of the image shown in Figure 5.8(a) has been isolated. To achieve this, the parameter, *P*, must be set low enough to detect the cell in the first place. By interpretation of the PO plot shown in Figure 5.7(a) it is clear that setting $P = 75$ allows this cell to be detected. However, setting $P = 75$ results in the POHMT also detecting the two circular cells in the top of the image. In order to isolate the cell in the bottom right, the markers that have been produced for the two cells in the top of the image must be removed.

The unwanted markers can be removed by exploiting some properties of the POHMT. The first of these is that as the value of *P* is reduced, the POHMT becomes less strict and allows more features to be detected. Further, while the POHMT is not

an increasing transform, and hence if, $\forall x \in E$, $I(x) \leq J(x)$, the following expression does not always hold,

$$POHMT_{B,P}(I) \leq POHMT_{B,P}(J), \tag{5.2}$$

the POHMT is in fact increasing in *P* with respect to the same image. This property is based on the threshold decomposition property of the POHMT where *P* is analogous to a threshold function [80]. This means that although the POHMT is not increasing in the traditional sense, the result of the POHMT is increasing with respect to *P*. The following relationship holds when the POHMT is applied to the same image *I* with different values of *P*,

$$POHMT_{B,P}(I) \leq POHMT_{B,P-1}(I), \tag{5.3}$$

This means that when a feature has been detected at some level *P*, it will always be detected, for all values between 0 and this value of *P*. This is why setting $P = 75$ results in three markers in the output of the POHMT instead of just the cell that should be isolated. It is also true, by interpretation of Equation (5.3), that the size of the markers produced when *P* is high, remain the same size, or, get larger, as the value of this parameter is reduced. This means that the markers produced when $P = 90$, are a subset of the markers that are produced when $P = 75$. It is therefore possible to use bounding boxes in order to isolate the cell in the bottom right of the image which occupies 75% of the SEs. A bounding box can be defined as the smallest rectangular area that can contain any connected image region (See Figure 5.9(c)). In this case, the connected regions are the markers produced by the POHMT.

When $P = 75$, the POHMT produces the marker image shown in Figure 5.9(a). A bounding box can be computed for each marker in the image produced by the POHMT when $P = 75$ as illustrated in Figure 5.9(c). Each bounding box in Figure

5.9(c) contains one marker for each of the three cells that have been detected. When $P = 90$, the marker for the cell of interest is *not* contained in the output of the POHMT. The output does however still contain markers for the two cells that are not of interest - see Figure 5.9(b). The bounding boxes that were computed for the image containing markers for all three cells, shown in Figure 5.9(c), can now be used to locate any features that have disappeared as a result of increasing *P*. This makes it possible to isolate the cell (or any other feature of interest) by locating any empty bounding box. This is demonstrated in Figure 5.9(d), where the empty bounding box provides the location of the desired cell. The centre of this bounding box may be marked in an output image and this marker can then be reconstructed to provide the isolated cell of interest.

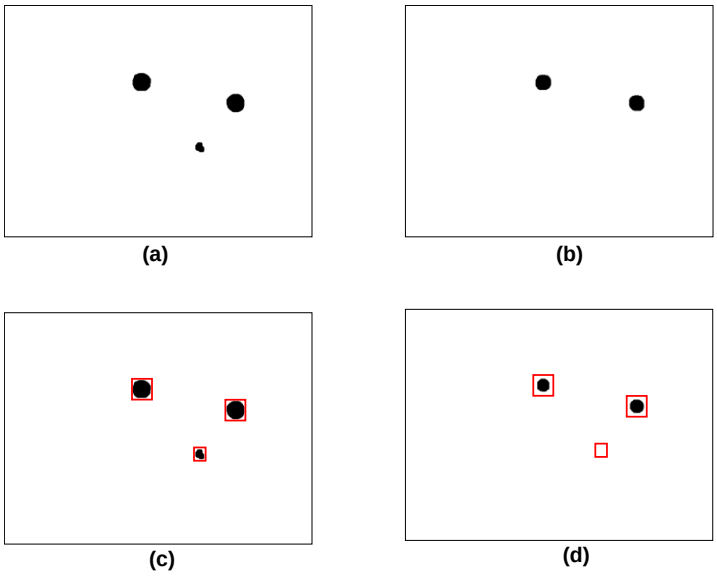


Figure 5.9 Technique used to isolate a feature of interest (a) POHMT marker image when $P = 75$ (b) POHMT marker image when $P = 90$. (c) Bounding boxes obtained for the markers shown in (a) where $P = 75$ and all of the bounding boxes contain a marker. (d) Marker obtained when $P = 90$ with bounding boxes from (c) overlaid. The empty bounding box contains the location of the cell of interest which can be marked in the output image. All images are 520 x 692.

5.4 Experimental Results

Everything that has been discussed and explained by example in this chapter is now tested on real data. First, it is shown that the PO plot may be used to set *P* such that the POHMT can be used to detect a biological cell in a series of very noisy images. The PO plot is then used to determine a suitable distance that may be fixed between the SEs such that the UHMT [12] can detect the same biological cell in the same series of noisy images using the modification suggested in [8]. Next, the PO plot is used to set a threshold for the BHMT so that the modification of this transform, as proposed in [8], can also be used to detect this cell in the same noisy data. A comparison of the output of all three techniques is provided. The method of using the PO plot to set *P* for the POHMT is also performed on the same images as Perret *et al*. used in [8]. This test has been carried out to determine if the routine proposed in this thesis is capable of detecting the LSB galaxies as was achieved by the authors of [8]. Finally, an example of the discriminatory filter differentiating between dice showing different values between one and six is provided.

5.4.1 The POHMT, the UHMT and the BHMT in Very Noisy Data

An example of eight very noisy biological images (of an immune system cell) are shown in Figure 5.10 where three of the images (Figure 5.10(a)) have been selected and used as a training set in order that *P* can be determined and used to detect the features of interest in the test set (Figure 5.10(d)). There is a small group of pixels in each image in Figure 5.10 that represents the feature of interest, while the rest of the image contains noise and other features that are not of interest. The images are extremely noisy, and, by observation of the data, it is evident that the shape and orientation of the cell changes between the images.

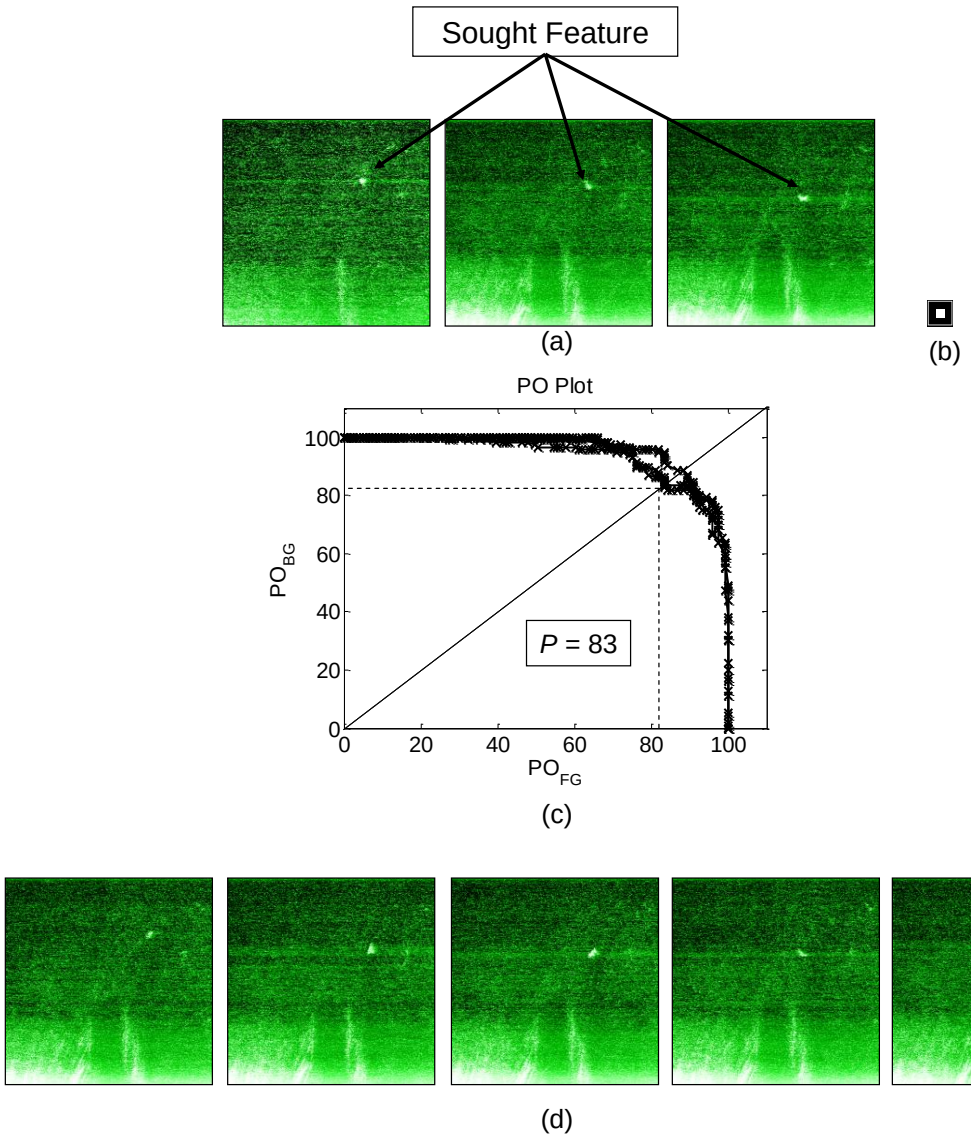


Figure 5.10 The image set containing the cell of interest and some other features where the entire image is submersed in noise. Each of these false colour images are 512 x 512. (a) Training set to determine an appropriate value for *P*. (b) SEs used to locate cell of interest (c) PO plot obtained for the training set shown in (a). (d) The set of test images in which the feature is sought after *P* has been fixed using the PO plot for the training set.

The first stage in the process is to generate a PO plot for each feature of interest in the training set in order to determine an appropriate level for P. Although the cell is not a constant shape and size in all images, B (shown in Figure 5.10(b)) was designed such that its elements corresponding to B_{FG} will fit inside it in each image. Similarly, B_{BG} was designed to encompass all of the features of interest in each image to guarantee that the cell can be detected in all possible orientations and variations of shape and size. By increasing the spatial distance between the SEs, as has been done here, it can be argued that the transform may produce erroneous hits. If a problem

occurs, this issue can be overcome by exploiting the discriminatory property of the POHMT shown in Section 5.3. Although automatic techniques are available for SE design, [11] and [39], a manual method has been used here to design the SEs. Square SEs have been used for processing simplicity in order to compare the performance of the POHMT, UHMT and BHMT, however any arbitrarily shaped SE may be used.

The composite SE *B*, shown in Figure 5.10(b), was used to generate a PO plot for each image in the training set in order to obtain a suitable level for *P*, such that the feature could be detected in the test set using the POHMT, without picking up erroneous hits. The PO plot, generated for the training set, is shown in Figure 5.10(c). Clearly, by reference of the PO plot, setting *P=83%* is sufficient to ensure that this feature may be detected using one composite SE for the entire test set. The POHMT was calculated for each image in the test set where the results of applying this transform and dilating (by a 5x5 square SE) the markers in its output, are shown in Figure 5.11. The test set of images are shown in Figure 5.11 such that it is easy to view the position of the marker with respect to the feature of interest in the original image.

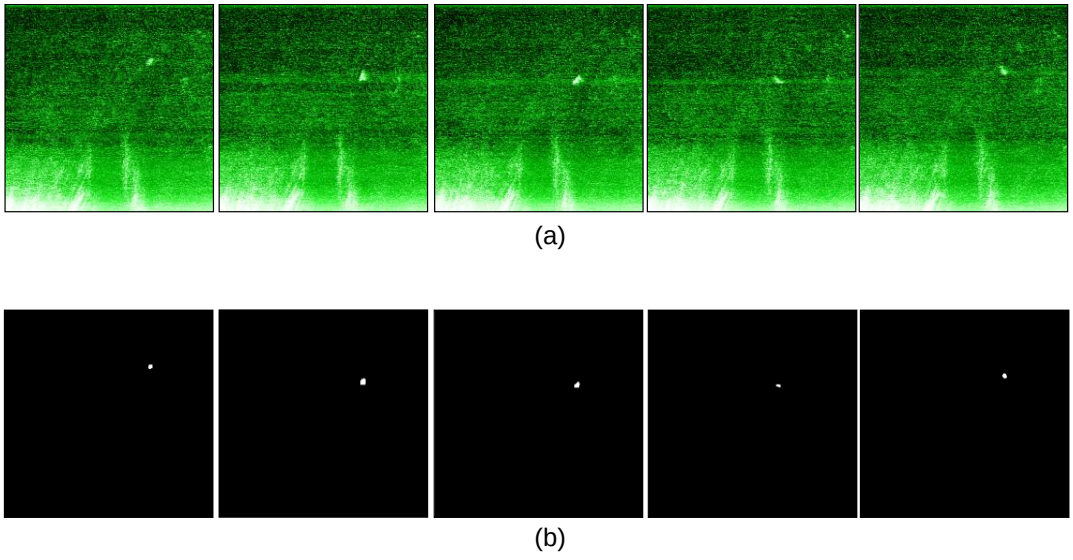


Figure 5.11 Result of applying the POHMT to each image (512 x 512) in the test set with *P* = 83. (a) Original images. (b) POHMT output for the image shown directly above it in (a).

The results shown in Figure 5.11(b) demonstrate that the feature of interest is detected in every image of the test set, and that there are no false positives. In

Section 5.2, it was shown that the PO plot could be used to determine a suitable distance that could be fixed between the SEs for the UHMT or RHMT to operate more effectively in noisy images. Using the PO plot shown in Figure 5.10(c) and Equation (5.1), the distance *d* that was calculated for the training data was 81 greylevels. This distance was fixed between the same SEs as those used by the POHMT to detect the noisy cell in the test set shown in Figure 5.10(b). The result of fixing a distance of 81 greylevels between the SEs and executing the UHMT on these images is shown in Figure 5.12(b). As in Figure 5.11, the original images have been provided here such that the location of the marker can be viewed with respect to the sought feature.

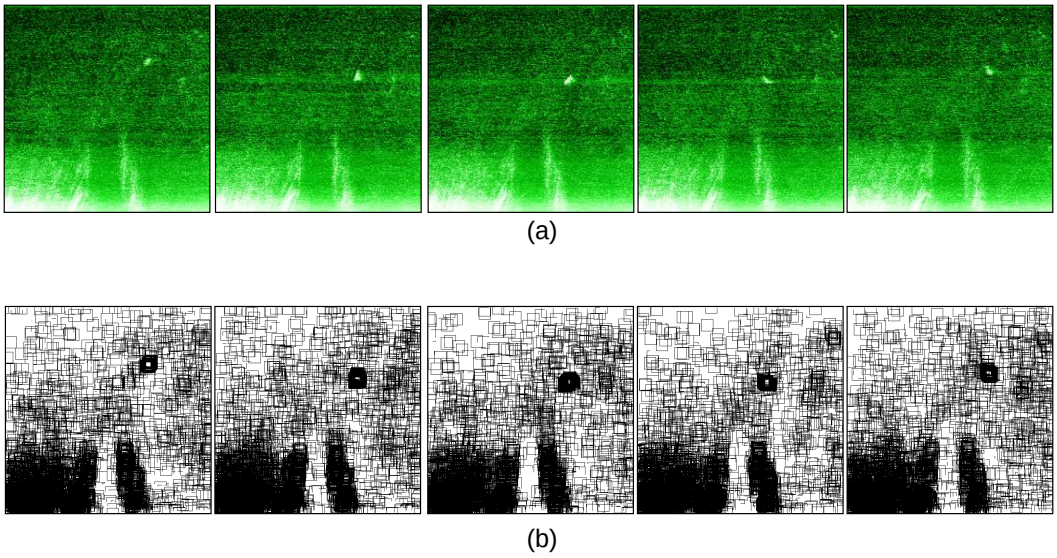


Figure 5.12 Result of applying the UHMT to each image (512 x 512) in the test set. (a) Original images. (b) UHMT output for the image shown directly above it in (a).

By reference of Figure 5.12, it is clear that although the feature of interest has been detected, there is also a large number of false positives in the result. In fact the number of false positives is so high that the UHMT, even with the suggested modifications where the parameters have been determined accurately using the PO plot and Equation (5.1), is rendered almost useless for processing images with this level of noise.

The same value of *d* was used to set the threshold for the BHMT – this technique was explained in Section 5.2. It was also explained in Section 5.2 that the BHMT and the

UHMT are equivalent and that if the distance between the SEs for the UHMT is the same as the threshold used to compute the output of the BHMT, that the two results are identical. This is of course is the case here, and hence, like the UHMT with a distance of 81 greylevels fixed between the SEs, the BHMT thresholded at a level of 81 produces a large number of false positives in addition to marking the feature of interest. The result of thresholding the output of the BHMT at a level of 81 as calculated using Equation (5.1) is shown in Figure 5.13. Clearly, like the UHMT, this transform cannot be used to reliably detect the sought feature in these images as a result of the noise.

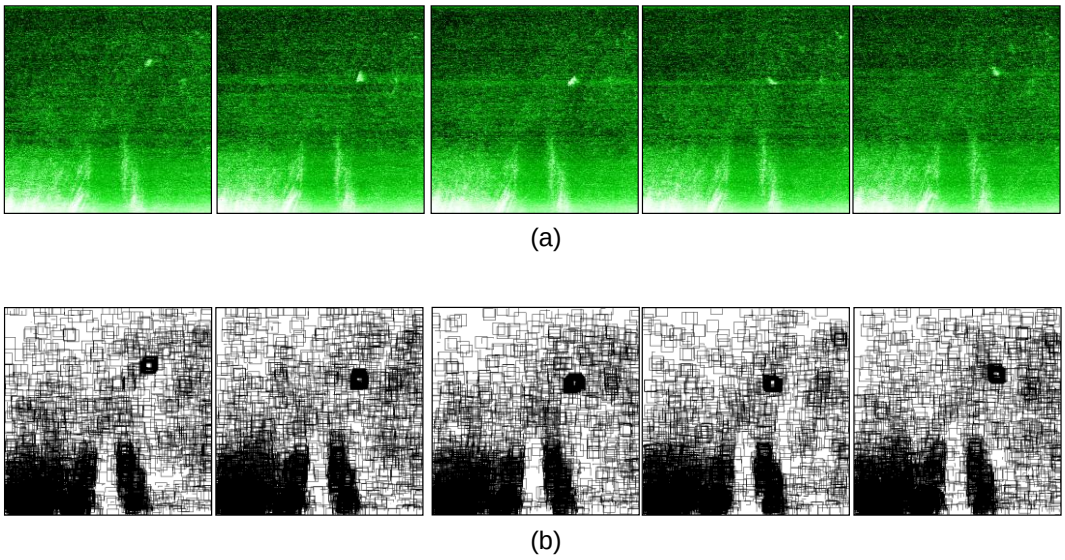


Figure 5.13 Result of applying the BHMT to each image (512 x 512) in the test set and thresholding the output using a threshold of 81. (a) Original images. (b) BHMT output for the image shown directly above it in (a).

It is clear that although the PO plot can be used to set parameters for the UHMT, the RHMT (in the same way as the UHMT), the BHMT, and the KHMT (in the same way as the BHMT), the POHMT outperforms all of these transforms when the images are as noisy as those being tested in Figure 5.10. Further, these alternative methods require unrealistically large values of d when images are as noisy as those used in this experiment.

5.4.2 The POHMT and the FHMT

The POHMT has also been tested on the images used in [8] to compare its performance with the FHMT described by Perret *et al*. An example of the POHMT detecting two LSB galaxies in two different images has been shown in Figure 5.14. The contrast of the image shown in Figure 5.14(a) has been enhanced to make the LSB clearly visible. Although the POHMT is capable of detecting the LSB in the images, as demonstrated in Figure 5.14, it should be noted that the POHMT takes longer to compute the results than the optimised FHMT proposed in [8]. The execution time of the POHMT is discussed in the next chapter where a fast implementation (which outperforms the heuristic method proposed in [8]) for computing the transform is presented.

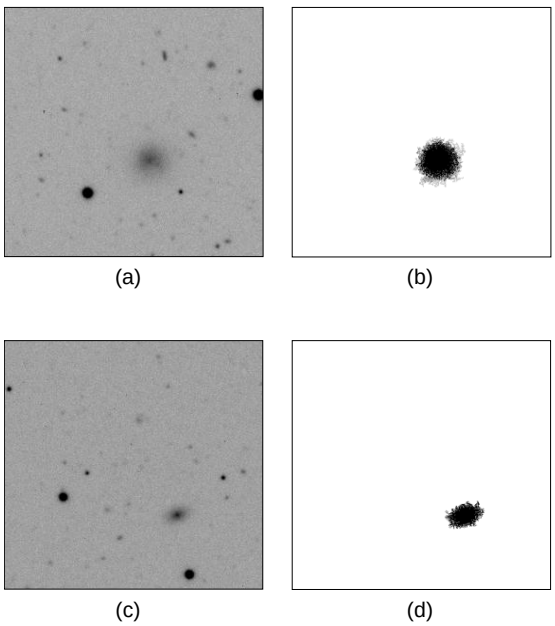


Figure 5.14 The POHMT detecting a LSB galaxy. (a) Original noisy image containing a LSB galaxy in the lower right quadrant of the image. (b) The output of the POHMT when processing the image in (a). (c) Original noisy image containing a LSB galaxy in the lower right quadrant of the image. (d) The output of the POHMT when processing the image in (c). All images are 512 x 512.

5.4.3 The POHMT as a Discriminatory Filter

The final experiment that is presented in this chapter exploits the information contained in a PO plot to make the POHMT operate as a discriminatory filter in order to selectively detect dice from an image. It should be noted that a synthetic example is used to demonstrate how the discriminatory filter may be used in practice to selectively recognise features in an image.

Training data which consists of six images, each representing one face of a die, the corresponding PO plot, and the critical point that has been calculated for each image in the training set are shown in Figure 5.15.

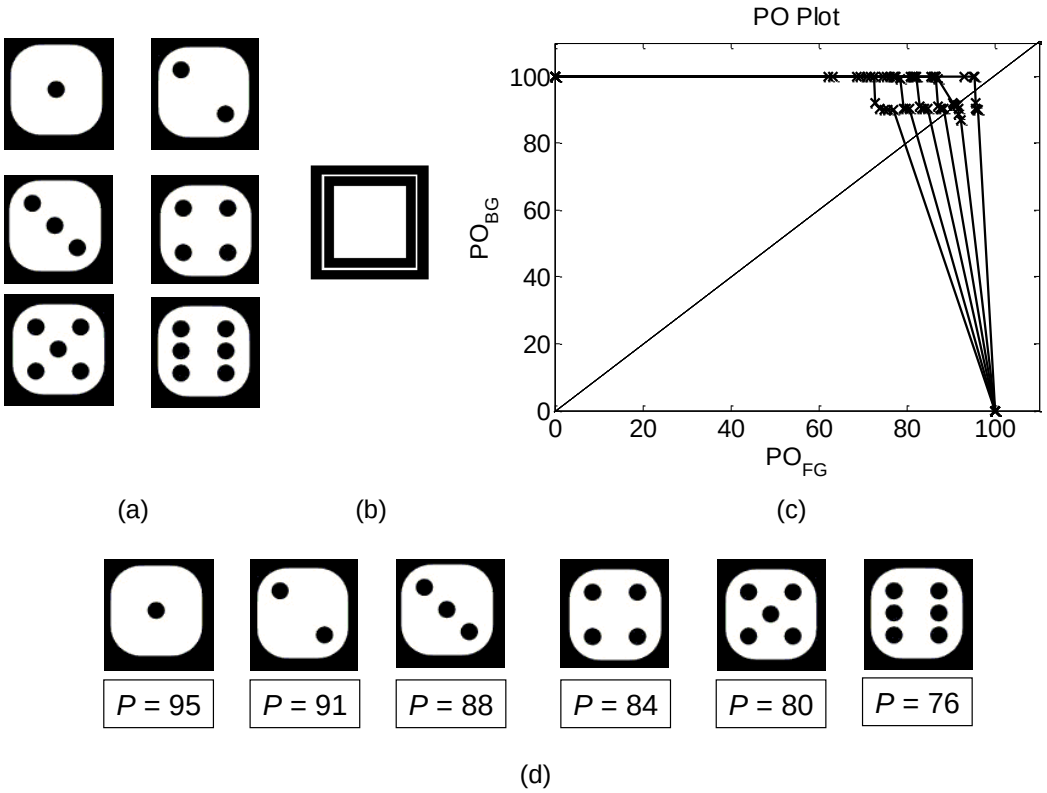
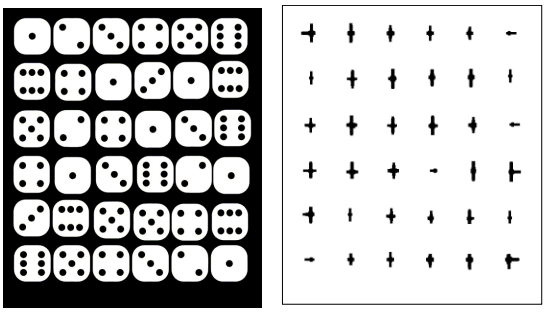


Figure 5.15 Training data for discriminating between dice. (a) 6 training images: all possible faces of the die. (b) SE used to generate PO plot and to detect the dice in this section (c) PO plot generated by centring a complementary pair of SEs on each training image. (d) Value of *P* computed by finding all critical points of the PO plot in (c).

As was discussed in Section 5.3 it is obvious that setting P to the value of the minimum critical point ($P = 76$ in the case shown in Figure 5.15) will result in all of the features of the training set being detected in the test set. This situation is shown in

Figure 5.16. It was also pointed out in Section 5.3 that it is easy, for example, to make the transform stricter in order to selectively detect only dice whose face is displaying a value of one in the image. This can be achieved by setting $P = 95$ before executing the POHMT on the image shown in Figure 5.16(a) (See Figure 5.17(a)). Setting $P = 91$ in accordance with the training data will allow all dice whose value is 1 or 2 to be detected in Figure 5.16(a). Continuing to lower *P* will allow all of the dice whose occupancy value is equal to or above the threshold *P* to be marked and those dice whose occupancy value is lower than the threshold *P* to be discarded. As was stated in Section 5.3, this is the trivial case of the POHMT operating as a discriminatory filter.



(a) (b)

Figure 5.16 Test image (609 x 529) and POHMT output with $P = 76$. (a) Test image containing 36 dice where all faces of the die are present. (b) Result of applying the POHMT to detect all dice in the test image.

Figure 5.17 shows the results of varying *P* and using the bounding box method that was described in Section 5.3 and demonstrated in Figure 5.9. In this case the PO plot has been exploited in order to isolate dice whose value is: one (Figure 5.17(a)), six (Figure 5.17(b)), three (Figure 5.17(c)), and two or five (Figure 5.17(d)). The combination of dice whose value is two or five was obtained by simply isolating the locations of dice displaying a value of two and the same for those displaying a value of five before computing the union of the two marker images to produce the result shown in Figure 5.17(d). It should be noted that while the POHMT is demonstrated to successfully discriminate between the specified features, the transform is in fact blind to the configuration and shape of the features in the image. That is, any feature, regardless of its shape, size, or appearance in the spatial domain, will be detected and marked by the POHMT if the feature occupies both SEs by some percentage which is

greater than or equal to *P*. This could lead to features being detected in error when taking this approach.

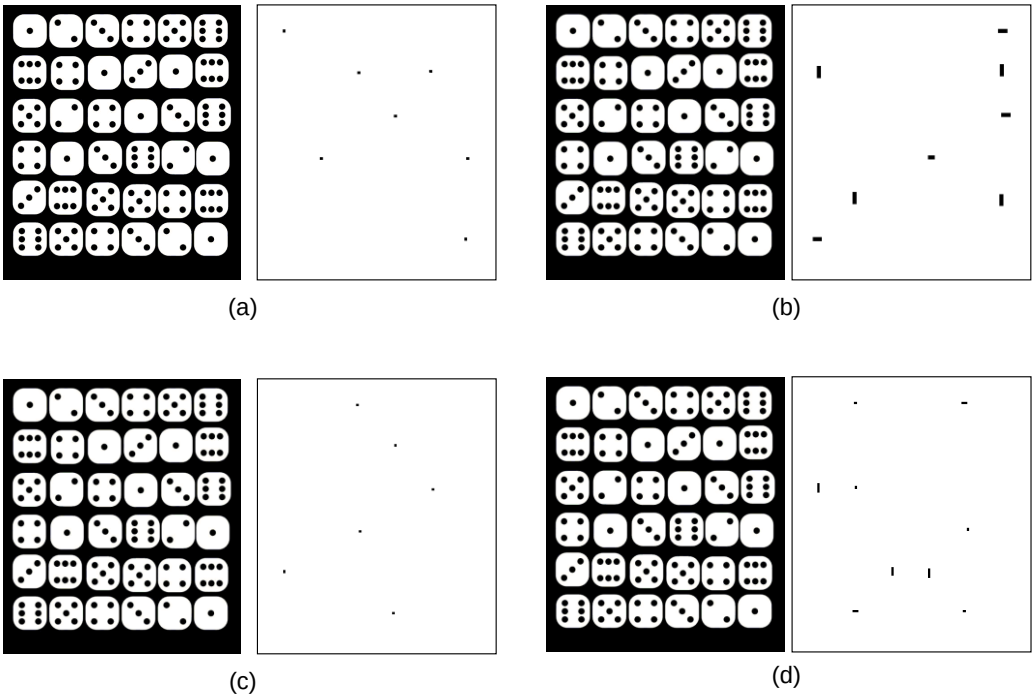


Figure 5.17 Discriminatory POHMT selectively detecting dice whose value is (a) one, (b) six, (c) three (d) two or five. All images are 609 x 529.

To test this method further, noise has been added to the image, the PO plot has been recalculated, and the dice displaying two in the image have been selectively detected. The noisy image and the result when selectively searching for dice whose value is two, is shown in Figure 5.18.

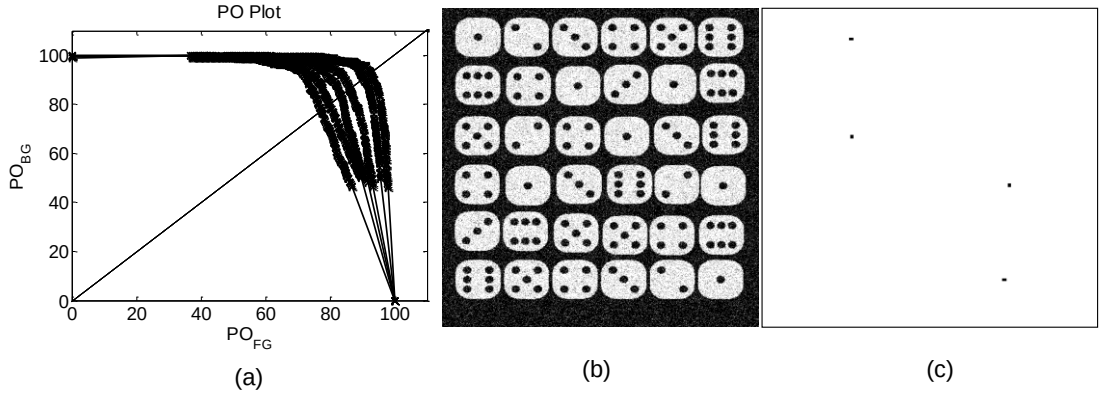


Figure 5.18 POHMT selectively detecting dice displaying two after noise has been added to the image. (a) PO plot that has been generated after noise has been added to the image. (b) Image (609 x x529) after zero mean AWGN has been added to the image. (c) Example of the POHMT successfully locating the dice that display a two in the noisy data (609 x 529).

The POHMT is still capable of selectively marking the features of interest despite the noise in the image.

5.5 Summary

In this Chapter, a novel design tool known as a PO plot has been introduced. The plot can be used to set the only parameter required by the POHMT and can be used by other researchers to set parameters for their own routines. Various examples of how this can be achieved have been presented, and the PO plot has been used to incorporate some suggested modifications by other researchers to make existing greyscale HMTs perform better in noise. When performing various HMTs and using the suggested modifications, it has been shown that image features can be detected in noise, but unlike the POHMT, there are also a large number of false positives in the result. This was demonstrated in Section 5.2 and verified in the experimental results in Section 5.4.

In addition to the PO plot being used to set parameters for greyscale HMTs in noise, it has been shown that this tool provides some additional benefits. The discriminatory filter aspect of the POHMT, which is a direct result of analysing the PO plot, allows the differentiation of objects in the image that are to be detected and others which may appear visually similar in the spatial domain but that are not of interest. An example of how this technique could be applied in practice is given in Section 5.4.3.

Section 5.4.2 demonstrated that the proposed approach achieves similar results to one of the most recent extensions of the HMT, the FHMT, presented by Perret *et al.* [8]. This has been verified using the same images (Figure 5.14) that were used in [8], however, the method presented here takes longer to execute than the optimised FHMT. The execution time of the POHMT is addressed in the next chapter where a fast algorithm that does not use heuristic techniques (unlike the method proposed in [8]) is shown to be extremely effective in making the POHMT execute in a fraction of the time taken by the optimised FHMT.

6 A Fast POHMT

This chapter presents an algorithm that can be used to efficiently compute the output of the POHMT which was introduced in Chapter 4. This Chapter begins by describing the methods provided in [54] and [55] in order to demonstrate how the techniques presented there make it possible to efficiently compute the output of respectively: square and rectangular median filters; and arbitrarily shaped, maximum and minimum filters.

The novel contribution of this Chapter lies in describing the steps that must be taken to generalise and combine these two methods to make it possible to quickly compute the output of any arbitrarily shaped rank order filter. The method that is used to achieve this is presented following the discussion of the existing techniques [54], [55]. Since the relationship between rank order filters and morphological operations has already been discussed (See Chapter 2, Section 2.5), only the relationship between rank filters and the POHMT is discussed here, and it is shown how P can be used to accurately set the rank parameter.

The chapter concludes by providing a number of speed comparisons which demonstrate a number of properties of the proposed fast routine. First, the effect that the value of P has on the execution time of the routine is evaluated. The efficiency of the proposed method is then tested using a number of different SEs of increasing dimension to determine how the size and shape of the window affect the efficiency of the routine. It is also shown that the execution time of the proposed routine is data dependent. The efficiency of the fast rank filters which are described here are then compared with an optimised routine for computing output of this class of filter using arbitrary windows. The POHMT is then implemented using the fast rank order filters and a comparison of the execution time of the fast POHMT and a direct implementation of the POHMT is provided. Finally, the efficiency of the proposed routine is compared to the method presented by Perret *et al*., [8].

6.1 A Fast 2D Median Filter

In the late seventies, Huang *et al*. presented a "Fast Two Dimensional Median Filtering Algorithm", [54], which allowed efficient computation of the median filter within square and rectangular windows. The authors exploit the fact that when calculating the output of the median filter using a sliding window, only a small number of the values that are considered in the calculation of the median actually change as the window moves from the current pixel to its neighbour. This means that instead of re-sorting every value in the window, only the pixels that exit the window, and the new pixels that enter it as it is translated from some pixel to its neighbour, need to be considered before calculating the new median. This leads to an increase in speed which is further enhanced by an efficient histogram technique that is used to sort the values that are coincident with the window and efficiently locate the median value.

The method proposed in [54] calculates a histogram of the image pixels that are coincident with the sliding window as it scans the image. It is then possible to find the median of the values in this histogram in order to calculate the output of the median filter at each image pixel. However, instead of generating a new histogram for each translation of the window in the image, the histogram is simply updated by removing the values that exit the window and adding those values that enter it as it is translated from pixel, x, to its neighbour $x+1$. Updating the histogram is achieved by decrementing the count in the bins that correspond to the values exiting the window, and incrementing the count in the bins corresponding to the values entering it. The median value is easily located using this technique since the pixels are already sorted in the histogram. All that is required is to accumulate the number of counts in each bin in the histogram until the count reaches the position of the median in this window. The intensity value at which this count is reached is the median value of the pixels that coincide with the window. The property of the sliding window that is exploited and the histogram technique for computing the output of the median filter at each point x is exemplified in Figure 6.1 where a square, 3x3 window, has been used to filter the data.

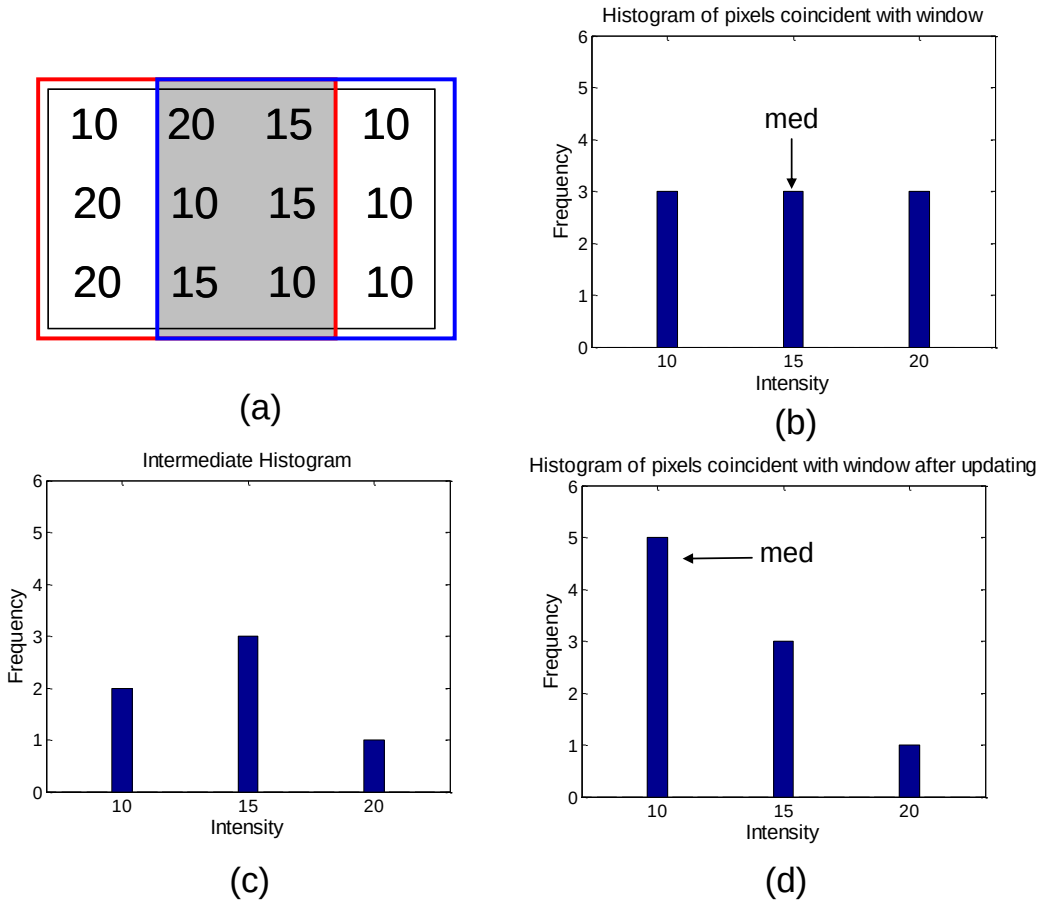


Figure 6.1 Illustration of the histogram method used to calculate the median filter. (a) Arbitrary pixel values coincident with a 3x3 window before (red) and after translation (blue). Shaded area - pixels that do not change as window is translated. (b) Histogram of pixels in red window, median = 15. (c) Histogram of pixels that remain in the window after those leaving it have been subtracted. (d) Histogram with new values in blue window added, new median = 10.

The techniques discussed so far provide a significant reduction in computation, and this is further enhanced by correctly directing the search that is performed to locate the value of the next median in the histogram when the current median has been calculated. This is achieved by keeping a count of the number of pixels in the histogram that have intensity lower than the median in the previous window. It is then possible to use this count to decide whether it is required to search up or down the histogram to find the new median. Let the count of pixels that have a grey level value less than the previous median be denoted *c*. If *c* is greater than

$$floor\left(\frac{Window\ \ Size}{2}\right), \tag{6.1}$$

values to the left (lower values within the histogram) of the current median should be interrogated in order to find the new median. If *c* is not greater than the value computed by Equation (6.1), then the new median may either: remain unchanged; or, if not, values to the right of the current median should be searched. For further understanding of this method, the interested reader is referred to [54] where Huang *et al*. provide a thorough description of their method using pseudo code.

While this technique is extremely efficient, the method presented in [54] is limited to computing the output of a median filter within square or rectangular windows.

6.2 Efficient Computation of Erosion and Dilation within Arbitrarily Shaped Windows

In [55], the authors extend the method proposed in [54] such that it is possible to compute erosions (minimum) and dilations (maximum) using arbitrarily shaped windows. Like the median filter, erosions and dilations are a special case of the more general rank order filter. As a result, the histogram technique that was employed by Huang *et al*. is used by Van Droogenbroeck and Talbot in [55] in order to find the minimum/maximum value of the pixels that are coincident with the SE *B* at each point *x* in the image. Van Droogenbroeck and Talbot also exploit the properties of the sliding window in a fashion similar to Huang. Further, the authors point out in [55], that it is not always necessary to consult the histogram when searching for the maximum or minimum rank. In the case of erosion (resp. dilation) it is only required to search for a new minimum (resp. maximum) if a value entering the window during translation is lower (resp. higher) than the current minimum (resp. dilation), or if the count in the histogram bin corresponding to the current minimum (resp. maximum) reaches zero. For this reason, the authors keep a record of the minimum (resp. maximum) value which can be easily updated as the pixels enter the histogram following a translation of the window. A new minimum (resp. maximum) is only

computed if one of the previously mentioned situations occurs. An example of this process for locating the minimum value and maximum value is shown in Figure 6.2. Note that although the arbitrary pixel values look similar between Figure 6.1(a) and Figure 6.2(a), the value in the bottom right corner of the blue window is different.

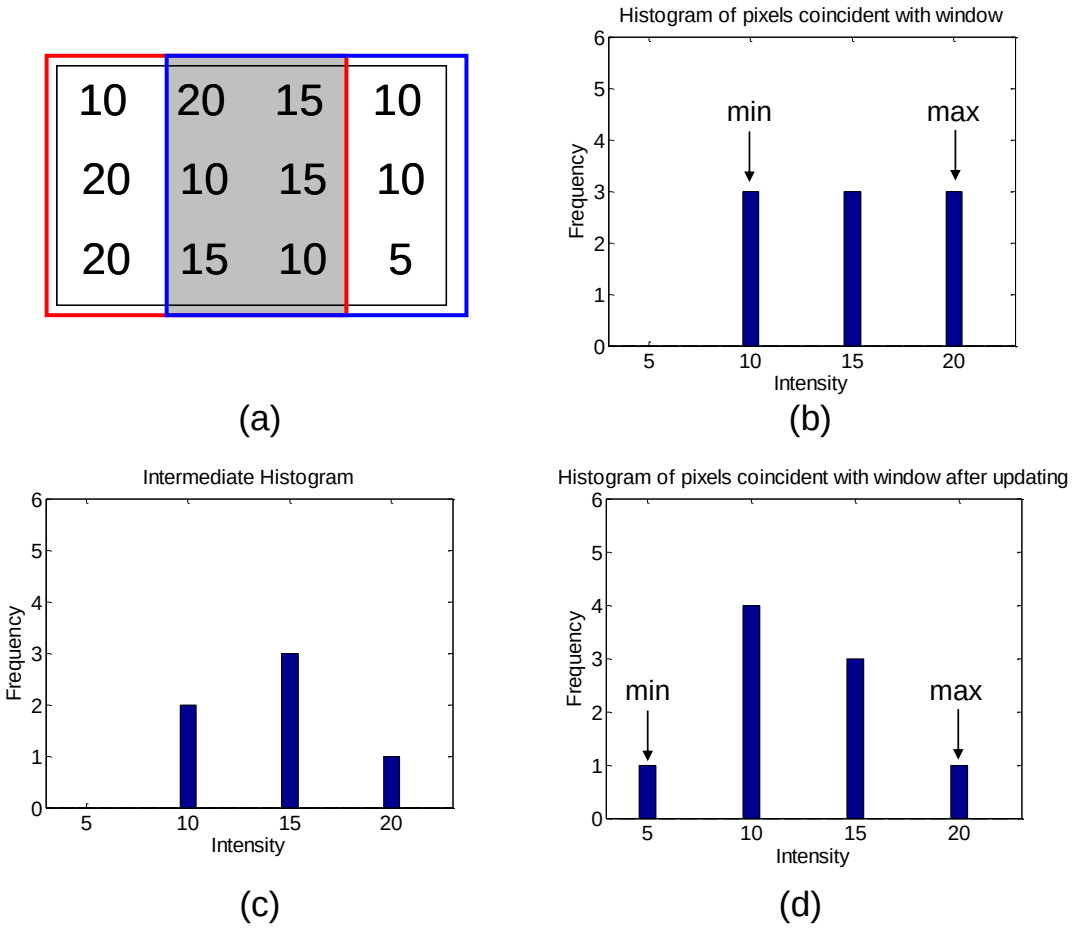


Figure 6.2 Illustration of the histogram method used to calculate erosion/dilation. (a) Arbitrary pixel values coincident with a 3x3 window before (red) and after translation (blue). Shaded area - pixels that do not change as window is translated. (b) Histogram of pixels in red window, min = 10 (erosion) and max = 20 (dilation). (c) Histogram of pixels that remain in the window after those leaving it have been subtracted. At this stage the minimum and maximum are unchanged since the count in their corresponding bin is not 0 (d) Histogram with new values in blue window added. A new minimum is instantly computed since a value lower than the previous minimum has entered the histogram, new min = 5. No value higher than the previous maximum has entered the window hence it remains unchanged.

Although the authors of [55] have made it possible to efficiently compute max and min filters, the major contribution of [55] is that the moving histogram technique is extended in such a way that it is compatible with any arbitrarily shaped window i.e. it is no longer restricted to square and rectangular windows. This is of particular importance for morphological operations, where the shape of the structuring element (window) that is used to process an image is critical. Van Droogenbroeck and Talbot extend the method proposed in [54] to include the use of arbitrary windows by identifying a set of so called "critical points" of the SE/window. These critical points are used to update the histogram as the SE traverses the image. Take for example the square window used in Figure 6.1, the only pixels which need to be updated in the histogram when this SE is translated by one pixel to the right are those pixels which coincide with the points that are not included in the intersection (shown in grey region of window in Figure 6.1) of the window and its translation by one pixel to the right. These points are called the "critical points" in [55]. This concept, and the critical points of an arbitrarily shaped window (when translating the window to the right), are shown in Figure 6.3. The pixels leaving the left side of the window are denoted B_L, and those entering from the right are denoted B_R.

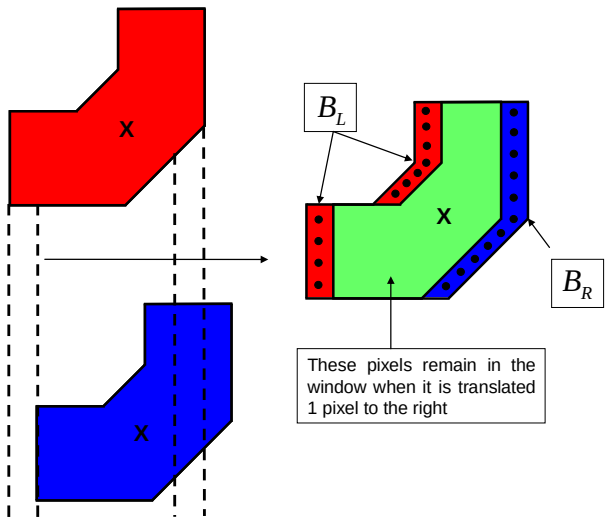


Figure 6.3 Illustration of the critical points in the window. (Top) red window centred at initial position. (Bottom) Blue window – red window translated 1 pixel to the right. (Right) Critical points of the window (red = B_L and blue = B_R) when it is translated to the right by a single pixel and the points that remain in the window (green) following this translation.

When processing an image using a sliding window, the window traditionally scans the image in raster fashion, as shown in Figure 6.4(a), to compute the output value for each pixel. A further contribution in [55] is that the authors introduce an alternative scanning method that can be used to compute the filter output. The idea is to minimise the number of times - ideally to one - that the entire histogram of pixels that coincide with the window must be computed. If the traditional raster scan is used, then a new histogram must be calculated every time a new row is to be processed. To avoid this redundancy, Van Droogenbroeck and Talbot propose, that in the first instance, the entire histogram is computed for all pixels that coincide with the window. Then, the first row is scanned from left to right, the histogram is updated using the critical points calculated for the translation by one pixel to the right, and the output value is calculated for all pixels in this row. When the end of this row is reached, the window is translated down in the vertical direction by one pixel to the next row. The histogram is updated using the critical points that can be computed for the top and bottom of the window, the output is calculated, and the window is then translated along this second row from right to left. As the origin of SE is translated to each pixel in this row, the histogram is updated using the same critical points that were stored for the first row, and the output value is calculated for all pixels on this row. When the SE reaches the leftmost pixel in this row, it is translated down by one pixel to the next row, the histogram is updated using the critical points corresponding to the top and bottom of the SE, and the process continues until the entire image is processed. This scanning order is shown in Figure 6.4(b).

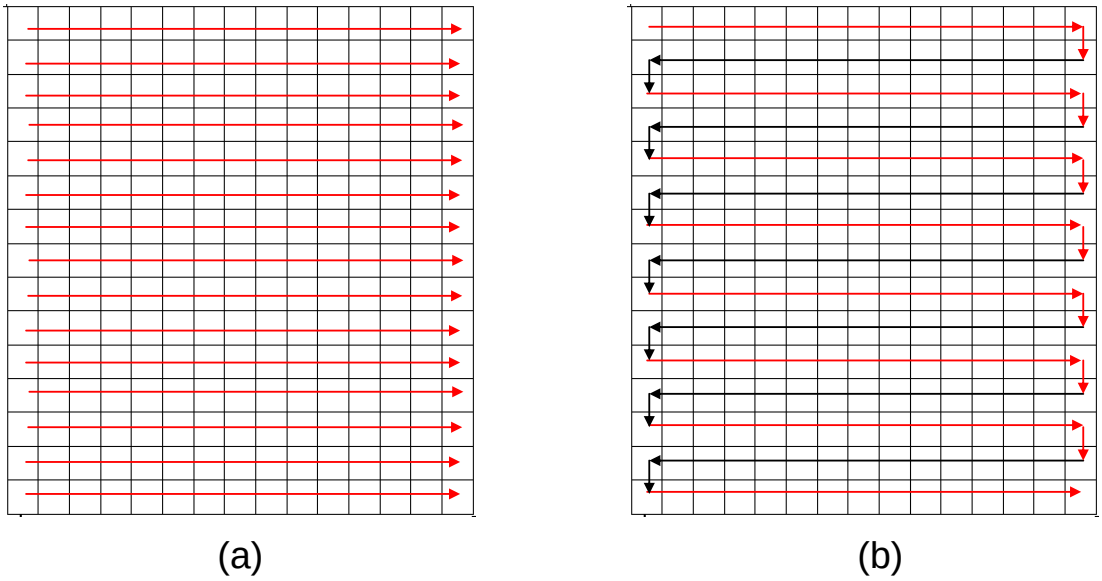


Figure 6.4 Example of scanning techniques used to process the image using a window. (a) Traditional raster scan requiring full computation of the histogram of pixels in the window for the first pixel in each row. (b) Alternative scanning method proposed in [55] to minimise the number of times the entire histogram must be computed.

The method presented in [55] is extremely effective in allowing erosions and dilations to be computed in an efficient manner. So long as the required critical points can be identified for a SE, it is possible to use these to update the histogram. The method for finding the critical points in [55] is given in the form of an intuitive description that is similar to the one that has been provided in this section so far. In [55], the authors only demonstrate the usefulness of their extension for computing erosions and dilations. They mention, although they do not discuss in detail, the extensions and generalisations that can be made to their technique such that it is possible to compute the output of any arbitrarily shaped rank order filter. In the next section, a mathematical formulation that can be used to calculate the critical points of the SE is provided. Then, a thorough explanation of how the method proposed in [55] can be generalised such that the output of any rank filter can be computed within any arbitrarily shaped window is provided.

6.3 Efficient Computation of Any Rank Order Filter within Arbitrarily Shaped Windows

The previous two sections have described the methods that have been presented by Huang et. al for the efficient computation of median filters, and by Van Droogenbroeck and Talbot for the efficient computation of erosions and dilations using arbitrarily shaped SEs. The median filter, and the erosion and dilation, are all special cases of the general rank order filter. In this section it is explained that the method presented in [55] for computing max and min filters within arbitrary windows can be coupled with a generalisation of the histogram search method used in [54] in order to allow the efficient computation of the output of any rank order filter within any arbitrarily shaped window. In addition to this, a mathematical description of the method that can be used to compute the critical points of any SE is provided. This was not given in [55] where the authors explained the method of identifying the critical points by example and discussed how they could be found using logical operations.

Table 6.6.1 highlights exactly where the contribution of this chapter fits within the context of the existing techniques.

Method	Window	Output
Huang *et al*. [54]	Square and Rectangular	Median
Van Droogenbroeck & Talbot [55]	Arbitrary	Min and Max
Murray & Marshall [77]	Arbitrary	Any Rank $1 \leq k \leq m$

Table 6.6.1 Table highlighting where the various techniques discussed in this section relate to each other.

Before explaining the method that is used to compute the output of any rank, k, of the pixels that coincide with the SE at each point in the image, a mathematical formulation that can be used to compute the critical points is provided. Based on the discussion in the previous section regarding critical points and the alternative

scanning method that can be used to minimise the number of times the entire histogram must be computed when processing the image data, it is clear that at least two sets of critical points are required in order that they may be used to update the histogram for all translations of the SE. When the SE is being translated to the right, the points B_L and B_R (See Figure 6.3) are used. When the SE is translated vertically downwards from row to row the critical points B_T (from the top of the SE) and B_B (from the bottom of the SE) are used to update the histogram. By letting (i, j) denote the origin of the SE, the sets of critical points B_L, B_R, B_T and B_B may be computed using the following,

$$B_L = B(i, j) \setminus \left(B(i, j) \cap B(i+1, j) \right) \tag{6.2}$$

$$B_R = B(i+1, j) \setminus \left(B(i, j) \cap B(i+1, j) \right) \tag{6.3}$$

$$B_T = B(i, j) \setminus \left(B(i, j) \cap B(i, j+1) \right) \tag{6.4}$$

$$B_B = B(i, j+1) \setminus \left(B(i, j) \cap B(i, j+1) \right) \tag{6.5}$$

where B_L and B_R denote respectively: the points in the SE where old pixels will leave the window; and the points where new pixels will enter it, as the SE is translated one pixel to the right. B_T and B_B denote respectively: the points in the SE where old pixels will leave the window; and the points where new pixels will enter it, as the SE is translated vertically downward by a single pixel.

Given Equation (6.2) and (6.3), and Equation (6.4) and (6.5), it is possible to compute the critical points for any arbitrarily shaped window and store these in memory. For each translation of the arbitrary window to the right, the value of those pixels which coincide with the points B_L are removed from the histogram, and the values in the image that coincide with the points of B_R are added to it. When a row

has been processed, and hence the end of the row is reached, the SE is translated down to the next row. Following this translation, the value of those pixels which coincide with the points B_T are removed from the histogram, and the values in the image that coincide with the points of B_B, are added to it. When the SE is then translated from right to left, the points coinciding with B_R are removed from the histogram, and the points that coincide with B_L are added to it. When the SE returns to the leftmost pixel in the row, the SE is translated downwards by a single pixel and the points coinciding with B_T are removed from the histogram. Those points coinciding with B_B are then added to the histogram, before the process starts over and is repeated until the output image is calculated.

By combining and generalising the techniques demonstrated in Figure 6.1 and Figure 6.3, it becomes possible to compute the output of any rank order filter within any arbitrarily shaped window. By using the method of computing the critical points of the window and using these to update the histogram, the histogram may then be searched for any rank, $1 \leq k \leq m$, using a technique similar to the one that was described for calculating the median filter as demonstrated in Figure 6.1 and discussed in Section 6.1. That is, by generating a histogram of the image pixels coincident with the window, and updating this histogram with the pixels that are coincident with the critical points as the window moves, it is possible to exploit the redundancy associated with re-sorting these values. Then, by keeping a count, *c*, of the number of points that have an intensity that is lower than the value of rank *k*-1 of the previous window, it is straightforward to compute the output of any rank order filter defined by any arbitrary window *B*. An example of this technique for computing the output of a rank order filter, $\zeta_{B,k=7}$, for a 9 point square window is shown in Figure 6.5.

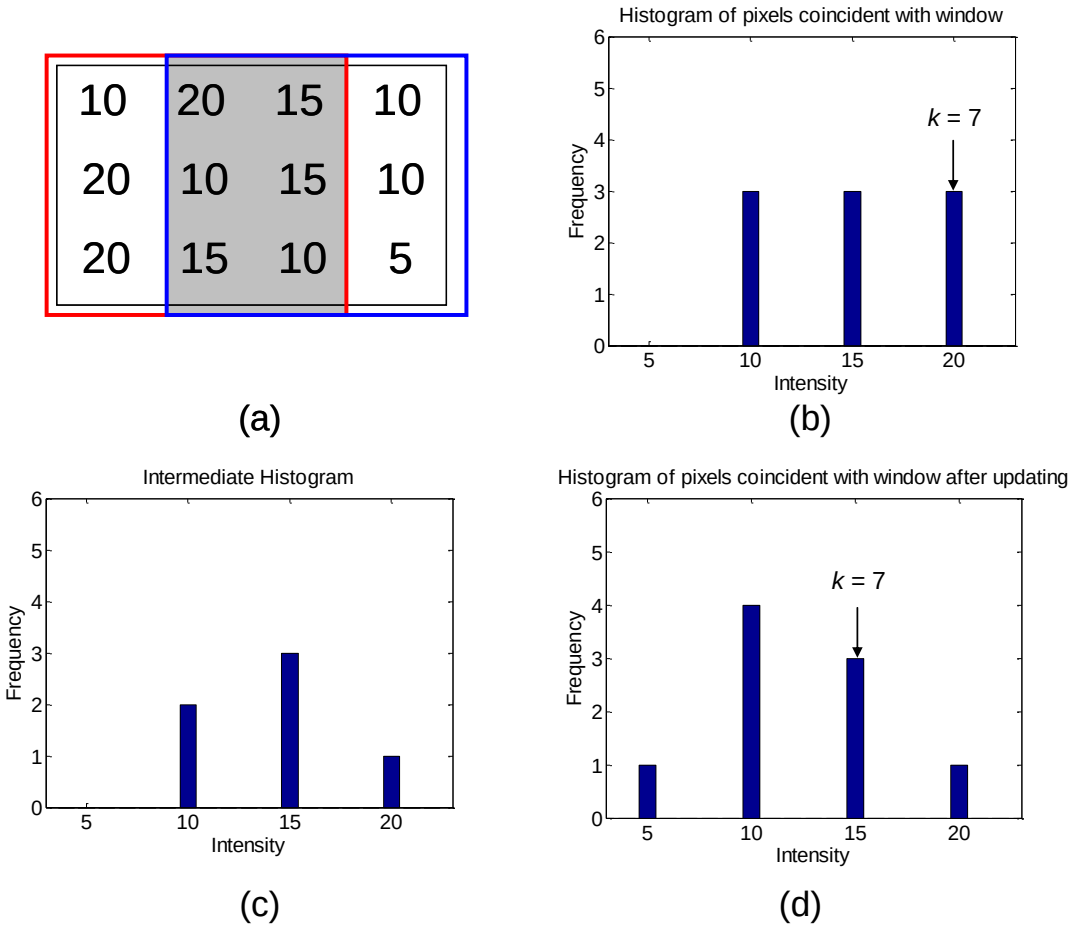


Figure 6.5 Example of computing the output of a rank order filter where *k* = 7.

6.4 Rank Order Filters and the POHMT

It is shown here that the extension of the method proposed in [55] that was described in Section 6.3 may be used to implement a fast POHMT (introduced in Chapter 4). As it was defined in Chapter 4, the POHMT generalises the HMT by allowing partial fitting of a composite SE such that objects of interest can be detected in an image despite the presence of noise. Instead of using traditional erosions and dilations which require a perfect match between image features and the SEs, the POHMT considers the extent to which a composite SE, B, "fits" the image as it is raised through all grey levels, $t \in T$, when its origin is coincident with a point $x \in E$, $\forall x \in E$. A point $x \in E$ is marked in the output of the POHMT if - when the origin of the SE is coincident with $x \in E$, P %, of the points $B_{FG} \in B$ are beneath or at the same level as the signal, while simultaneously, P % of the points in $B_{BG} \in B$ are strictly

above the signal. If, when the origin of B is coincident with a point $x \in E$, there is at least one level $t \in T$ for which this condition is satisfied, the composite SE is said to be P % occupied in the image and this point is marked in the output of the transform.

A direct implementation of the POHMT which processes the image in a single pass can be realised using Equations (4.18), (4.19), (4.20) and (4.21). The extent to which the SEs fit a feature in the image may be measured using Equation (4.18) and (4.19), and a point $x \in E$ is then marked in the output of the transform if $\exists t \in T$ such that P% of B_{FG} fits the feature while P% of B_{BG} fits its background. Implementing the POHMT in this way is effective but time consuming.

A common technique that can be used to relax the strictness of morphological operators is to implement more general rank order filters in place of traditional erosions and dilations [12] and [25]. Therefore, an equivalent and more efficient implementation of the POHMT can be achieved using the fast rank order filters described in Section 6.3. Instead of calculating PO_{FG} and PO_{BG} using (4.18) and (4.19), and subsequently the output of the POHMT using Equation (4.21) $\forall x \in E$ and $\forall t \in T$, it is equivalent to implement this transform using,

$$POHMT_{[B \in B_{FG} \cup B_{BG}]}(x) = \begin{cases} 2^n\text{-}1 & \text{if} \quad \left[\zeta_{B_{FG},100-P}(I)\right](x) > \left[\zeta_{B_{BG},P}(I)\right](x) \\ 0 & \text{otherwise} \end{cases}. \tag{6.6}$$

It is clear from Equation (6.6) that the POHMT may be calculated using rank order filters where the parameter P in (6) may be used to set the rank k for the filter. When implementing the POHMT in this way, the idea is to look for places in an image where the intensity in rank ($100 - P$) is greater for the elements of B_{FG} than the intensity in rank P for the elements of B_{BG}. The output of the POHMT contains marker pixels for all $x \in E$ for which this condition is satisfied.

An example of how this may be implemented is provided here for clarity. Say that it has been determined using the PO plot (See Chapter 5) for a particular image set, that P should equal 75% for successful detection of the features that are to be located in a

noisy image. The POHMT can be implemented using rank order filters such that any image pixel that is coincident with the origin of B is marked in the output if the rank corresponding to (100 - 75)% of B_{FG} is at an intensity level greater than the level contained in the rank corresponding to 75% in B_{BG}. This is consistent with the definition of the POHMT as given in Equation (4.21).

6.5 Experimental Results

This section tests the performance of the fast algorithm that has been introduced in this chapter under a number of different conditions and against other methods. First, the effect that the position of the rank has on the execution time of the algorithm that was introduced in Section 6.3 is investigated. Then, a number of differently shaped SEs of increasing size are used to compute the output of a rank order filter. In this experiment, the rank k is set to the most computationally expensive rank, as determined by the aforementioned experiment. In this way, the extent to which the execution time varies with the number of points in a window can be determined for the worst case. It is also shown in this section that the content of the image data can affect the execution time of the routine. The routine is also compared with the execution time of an optimised Matlab function, ordfilt2(), (using Matlab 2011b, the most recent version of Matlab released on 01/09/2011) for computing the output of rank order filters. The fast POHMT is then compared with a direct implementation of the POHMT (no optimisation has been used in the direct implementation) to ensure that the proposed method is in fact more efficient. Finally the execution time of the fast POHMT is measured when operating on the image data that was used by Perret in [8] to test the FHMT. It is shown that the proposed method is more efficient than the one that was presented there. However, in contrast, the method proposed in this thesis does not incorporate heuristic techniques in order to achieve reductions in computational complexity.

6.5.1 Varying the Rank Parameter in a Square Window

In this section a 100×100 square window is used to determine the effect that the position of the sought rank has on the execution time of the routine. Although it is well known that the median is generally more expensive to compute than other ranks, this experiment has been carried out in order to quantify this for the proposed method. A natural, noise free image, shown in Figure 6.6, has been used for this purpose, where the image dimensions are 1440×2160.

Figure 6.6 Natural scene used to measure the execution time of the proposed fast rank order filters. The image dimensions are 1440 x 2160.

The output of the 100×100 window was calculated for ranks ranging from $k=1$ (minimum filter) to $k=10000$ (maximum filter) in steps of 100. The execution time that was measured for the computation of each rank is shown in Figure 6.7. The points that correspond to ranks between those that were measured have been interpolated in the plot.

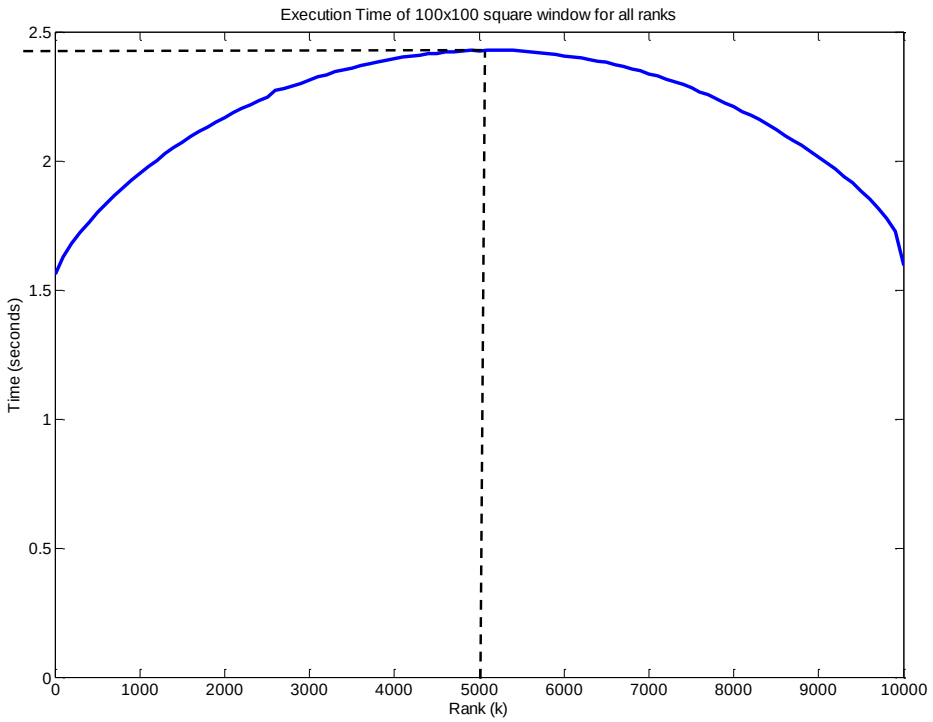


Figure 6.7 Execution time of the fast algorithm using a 100x100 square SE for all ranks between the minimum and maximum for this window. Computationally, the most expensive rank to calculate is the median.

It is clear by reference of Figure 6.7, that as expected, the most expensive rank to compute using the proposed method is the median. In this case, computing the median takes 868ms longer to compute than the minimum. That is slightly more than a 50% increase in the time taken to compute the median when compared to that of computing the output of the minimum filter. One reason for this may be that the search for the median requires a more thorough search of the histogram when compared with the search that is required for locating the minimum or maximum value. That is, when searching for the maximum or minimum in a histogram, the search only requires that the first value (starting from the left or right) is found. This implies that the search for the maximum and minimum values is less exhaustive than the search for the median. It is also possible that the minimum/maximum changes less frequently than the value of the median. If this is the case, then the histogram is searched less frequently when computing the minimum or maximum.

6.5.2 Execution Time using Different SEs

In this section, the five differently shaped SEs shown in Figure 6.8 have been used to process the image that is shown in Figure 6.6. Each of the 5 SEs has been applied to the image at a number of different scales in order to determine how the size of the window, as well as its shape, affects the execution time of the routine. In this experiment, the median rank has been calculated for all SEs, at all points in the image, since it was established in the previous section (see Figure 6.7) that the median is the most expensive of all rank filters to compute, using the proposed technique. As a result, all of the timing plots in this section provide the worst case computation time for each of the SEs of increasing size.

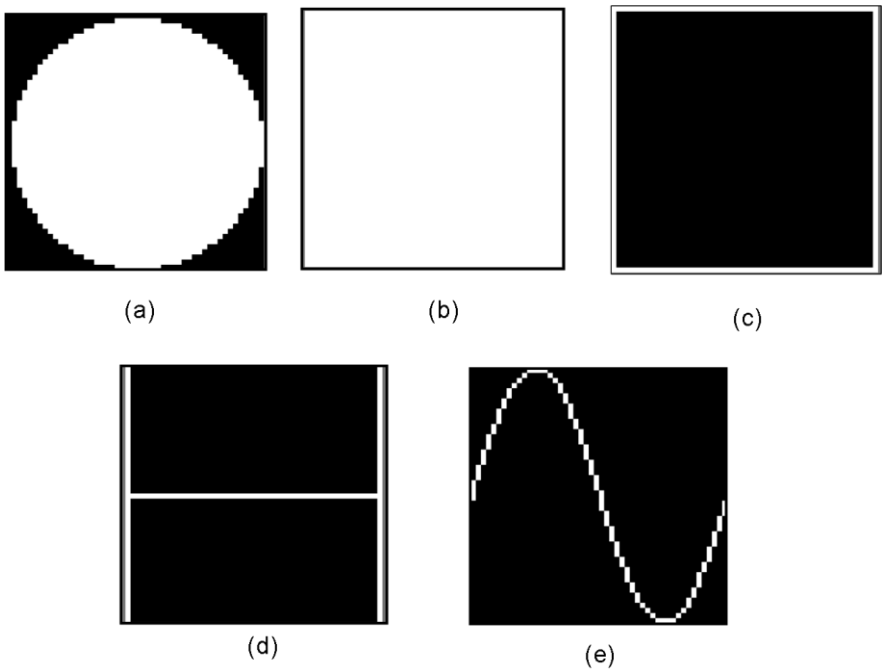


Figure 6.8 Pictorial representation of the 5 SEs that have been used to measure the execution time of the proposed method for computing the output of the median filter (most expensive rank to compute). (a) Disk SE, (b) Square SE, (c) Hollow square SE, (d) H shaped SE, (e) Sin shaped SE.

Each of the five SEs were applied to the image data in turn. The width and height of each SE was increased from 5 pixels to 50 pixels in steps of 5. The execution time that was measured for each SE of increasing size is shown in a separate plot in Figure 6.9(a)-(e). Each of these individual plots have been combined for comparison in Figure 6.9(f).

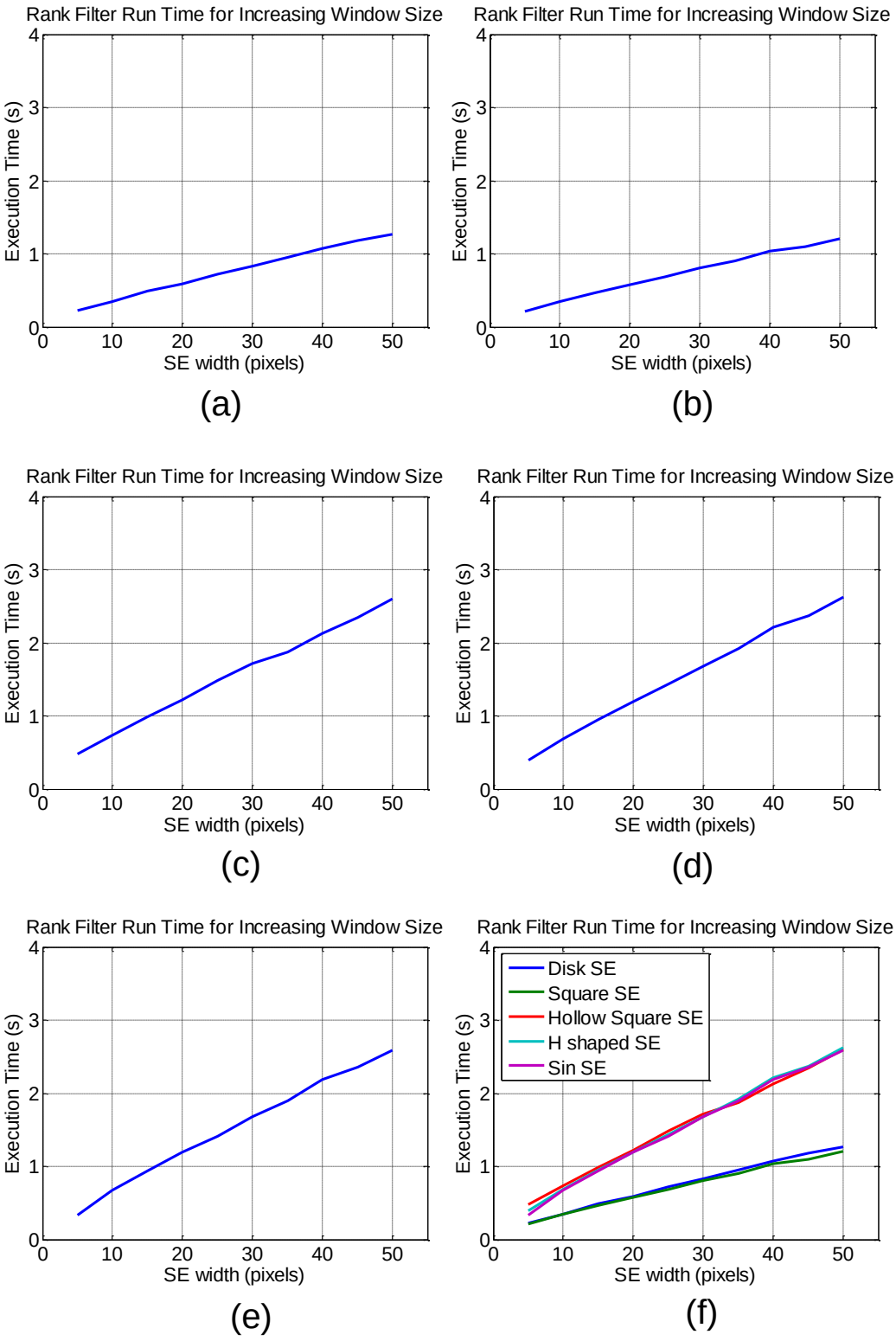


Figure 6.9 Execution time of the fast algorithm operating on the natural image shown in Figure 6.6. Each of the windows shown in Figure 6.8 have been used to compute the median at a number of increasing dimensions starting at 5 pixels wide and ending at 50 pixels wide. The execution time for the disk SE is shown in (a), the square SE in (b), the hollow square SE in (c), the H shaped SE in (d) and the Sin shaped SE in (e). (f) All plots combined for comparison.

It is clear by reference of the plots shown in Figure 6.9 that in all cases, the execution time of the algorithm increases linearly with the increasing dimensions of the SE. Further, by reference of Figure 6.9, it is apparent that the square and disk SEs compute the output in a similar time to each other, and that in all cases, these SEs take less time to compute the output than the others. The hollow square, the H shaped SE, and the sin shaped SE, all compute the output image in similar times to each other. The reason that these SEs take longer to compute the output than the disk and square SEs is related to the number of critical points that exist for the SE when compared to the total size of the window. In fact, it seems counter intuitive that the disk SE and square SE would be more efficient to compute rank outputs than the others as the number of points in these windows is significantly greater than that of say a pixel wide sinusoid or hollow square SE. However, the ratio of the points in the SE that are used as critical points for the square and disk SE, is significantly lower than the ratio of points in the other SEs that are critical points. This concept is exemplified in Figure 6.10 using a square SE and a hollow square SE. The ratio of critical points to the total number of points affects the computation time because this factor determines how many operations must be carried out to compute the output at each pixel. Further, this factor determines the extent of the histogram search that must be performed.

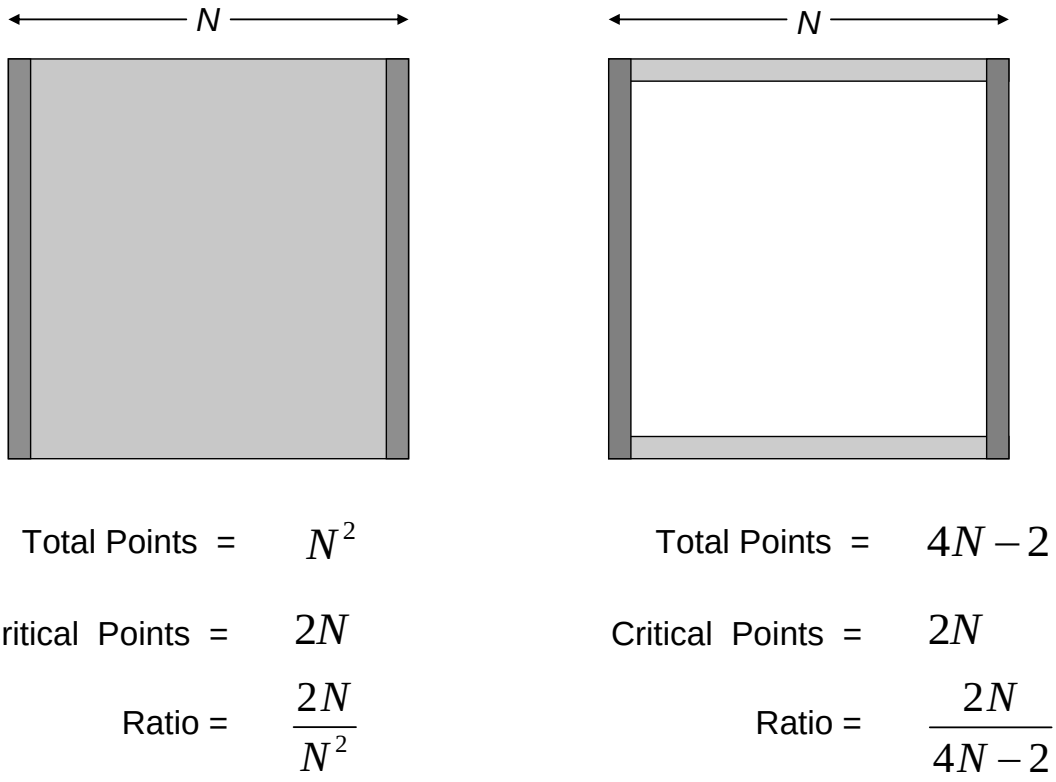


Figure 6.10 Ratio of critical points to the total window size for square and hollow square SE. Critical points are shown in dark grey and the other points in the window are shaded in light grey.

Due to the nature of the algorithm, and given the fact that the histogram must be searched in order to compute the output value of the filter for each point in the image, it is reasonable to assume that the efficiency of the algorithm will be affected by the frequency with which the output value changes. That is, the more frequently the output value changes, the more often the histogram must be searched, and further, if this value changes significantly, the histogram search becomes more exhaustive. To test this, theory, Gaussian noise has been added to the natural image such that value of the output at each pixel will change more frequently. The noise corrupted image is shown in Figure 6.11.

Figure 6.11 Natural scene corrupted by Gaussian noise that was used to measure the execution time of the proposed fast rank order filters. The image dimensions are 1440 x 2160.

The results of applying each of the SEs in turn to filter the noisy data, while increasing their dimensions as before, are shown in Figure 6.12. Further, in Figure 6.12(a) – (e), the timings that were computed for each SE when filtering the natural image have been included (dashed line) for comparison. The plots shown in Figure 6.9(f) and Figure 6.12(f) are provided to allow a comparison of the different SEs when applied to the same image data.

By comparing each of the SEs in turn, it is clear that all of the SEs take longer to compute the median of pixels that coincide with the window as it is translated to each pixel in the noise corrupted image. It is also obvious by comparing Figure 6.9(a) – (e) and Figure 6.12(a) – (e) that the increase in execution time appears to be constant for all of the SEs. This is clear as no SE becomes more efficient when compared to the others regardless of the image content.

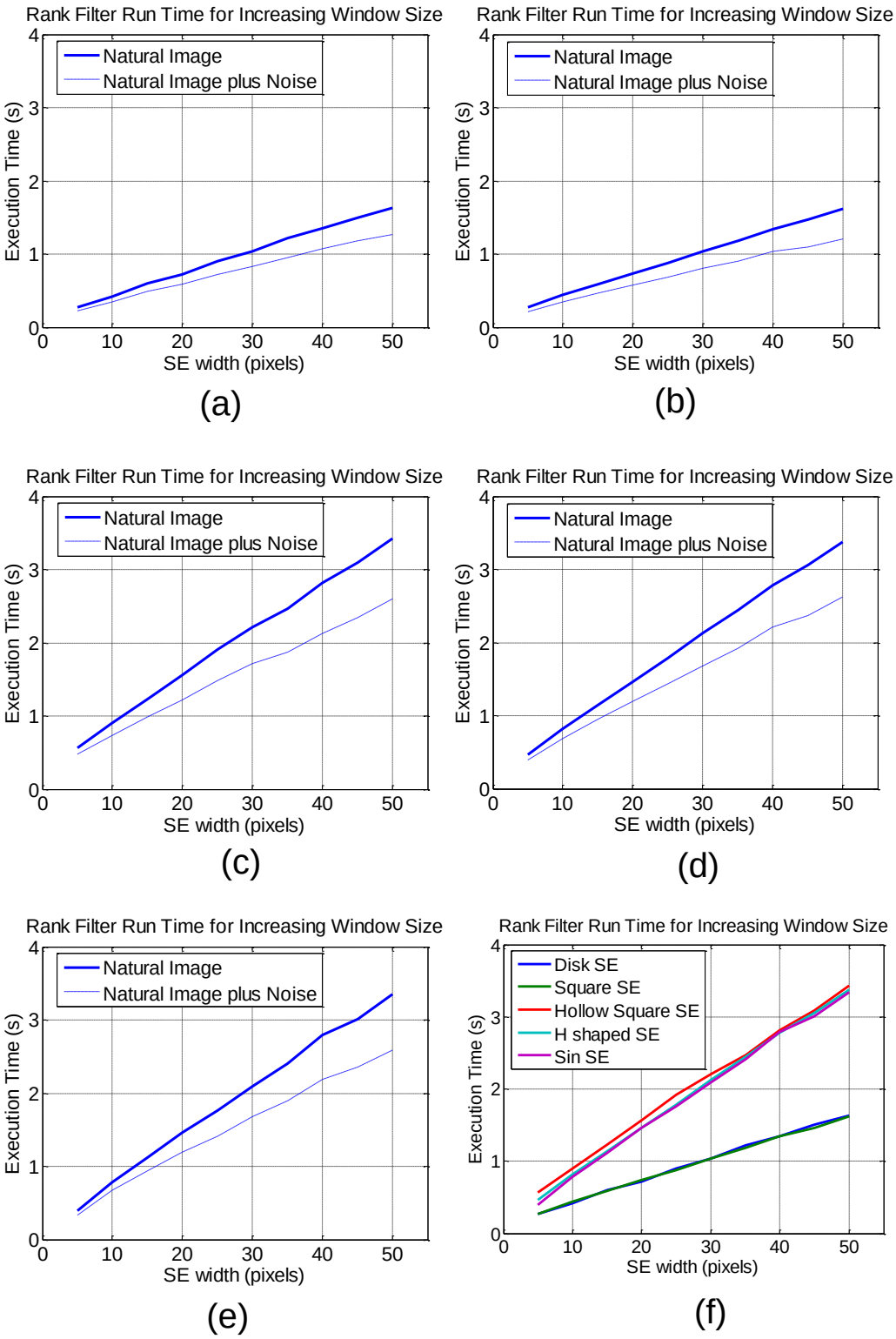


Figure 6.12 Execution time of the fast algorithm operating on the noise corrupted image shown in Figure 6.11. The plots of the timing that was computed for the natural image have been included (dashed line) for comparison. Each of the windows shown in Figure 6.8 have been used to compute the median at a number of increasing dimensions starting at 5 pixels wide and ending at 50 pixels wide. The execution time for the disk SE is shown in (a), the square SE in (b), the hollow square SE in (c), the H shaped SE in (d) and the Sin shaped SE in (e). (f) All plots combined for comparison.

6.5.3 Comparing the Proposed Method with an Optimised Matlab Routine

To the author's knowledge, as of December 2011, there exists no competing technique which is capable of efficiently computing the output of any arbitrarily shaped rank order filter in the same time as the proposed method. One reason for this claim is that no paper describing such a competing/similar method has been found. This point is supported by the fact that Urbach and Wilkinson [70] (see discussion in Chapter 3) compare their efficient method for computing erosions and dilations with the one proposed in [55]. The authors of [70] state explicitly that the method proposed in [55] is the only technique that is comparable to their routine. Since the method presented in [70] cannot be used to compute the output of general rank filters, and if it is true that the method in [70] is currently the fastest for computing the output of arbitrarily shaped minimum and maximum filters, then it is reasonable to assume that there is no comparable technique to the method that is proposed here. This point is further strengthened since in [65], Breare and Lehmann claim to have used the method proposed in [55] to implement general rank order filters where the authors compare the efficiency of this technique with a direct method for computing the output of a median filter. Although the authors also compare their method to an efficient technique which uses an approximation of the filter window to reduce the number of computations, it was explained in Chapter 3, Section 3.2.2, that this approach cannot be used here. Therefore, the only fair comparison that can be applied here is with a direct implementation of general rank filters within arbitrary windows. As such, there is no comparison that can be performed in order to determine the extent to which the algorithm proposed in this thesis improves upon existing techniques.

For the aforementioned reasons, the technique detailed in Section 6.3 has been compared to the Matlab function, ordfilt2() [81], in order to provide some comparison of the proposed method with current techniques that are used to compute the output of arbitrarily shaped rank order filters in leading software packages for signal processing. Since Matlab functions tend to use state of the art algorithms to implement their functions, and given the fact that the help document of the ordfilt2()

function [81] references the method described in [54], it would appear that the method employed there should be implemented in the same fashion as the proposed method, at least for square and rectangular windows. The SEs (see Figure 6.8) and the natural image (see Figure 6.6) that were used in the previous section to test the proposed fast rank order filter algorithm have been used here to compare the efficiency of the proposed routine when compared with that of ordfilt2.

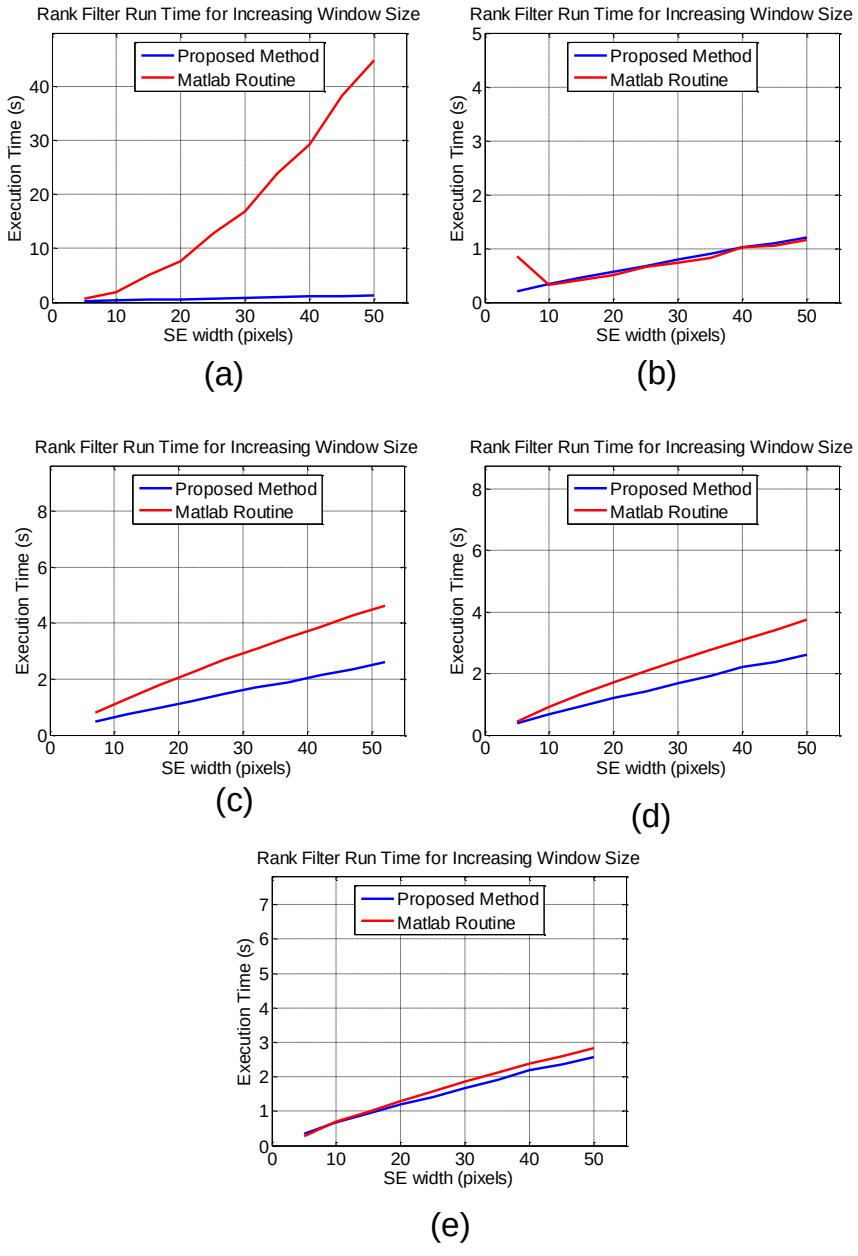


Figure 6.13 Comparison of the proposed method with an optimised Matlab routine for a number of SEs of increasing dimensions. The execution time of the Matlab routine and the proposed method using (a) Disk SE, (b) Square SE, (c) Hollow Square SE, (d) H shaped SE, (e) Sin shaped SE.

The results presented in Figure 6.13 appear to confirm that the proposed method has not been used for computing the output of arbitrarily shaped filters in ordfilt2(). Consider the execution time of the proposed routine when compared to that of ordfilt2() when using the square window of increasing size (Figure 6.13(b)). For each of the SEs that have been used, there is a fractional difference between the execution time of these routines. The fact that the difference is so small would suggest that ordfilt2() uses the same techniques proposed in [54] to compute the output of rank order filters within square and rectangular windows. This assumption is backed up by the reference to [54] in the Matlab help file for the ordfilt2() function. This explains why the method presented here executes in the same time as the optimised Matlab function for each of the SEs in the set - both methods are implemented in the same way. This also validates the comparison of the proposed method with ordfilt2() for the remaining SEs. That is, since when using the square SEs the implementations appear to be the same, it can be assumed that there is no significant difference in execution time as a result of the tools that have been used to implement the techniques. This implies that any differences in execution time come as a result of one of the algorithms being more efficient than the other.

When comparing the execution time of the proposed method and ordfilt2() using a disk SE of increasing size to filter the image, there is a significant difference (Figure 6.13(a)). The execution time of ordfilt2() when using the disk SE increases exponentially with the increase in SE dimensions. However, the increase in the execution time of the proposed method, when using the same set of SEs, is linear. When the diameter of the disk SE is 50 pixels, the proposed method is around 20 times faster than ordfilt2().

The number of points in the square SE and the disk SE is similar at each dimension that is used. This provides further confirmation that such a significant difference in execution times, when using a disk SE, is in fact due to the method that has been used to compute the filter output at each image pixel. Since the proposed method and ordfilt2() computed the output images in similar times using square windows, the difference in computing time when using disk SEs cannot be attributed to the

difference in packages used for implementation. This implies that the proposed method is superior to the current sorting techniques that are used to compute the output of arbitrarily shaped windows.

For the proposed method, only the critical points are considered in the computation of the output as the filter traverses the image. This leads to the linear increase in execution time with window size as is shown in Figure 6.13(a) – (e). It appears that ordfilt2() processes every pixel that coincides with the window as it traverses the image – except when square and rectangular windows are used. This explains why the time taken to compute the output of rank order filters using the disk SE increases exponentially. This theory is further confirmed by observation of the data shown in Figure 6.13(c), Figure 6.13(d) and Figure 6.13(e). For each of these SEs, the number of points in the window increases linearly with the width and height of the SE. This causes a linear increase in the execution time of ordfilt2(), however, this optimised Matlab function still takes longer to compute the output than the method that is proposed here.

There is a significant difference in the execution time of the proposed method when compared with ordfilt2() as the dimensions of the hollow square SE and the H shaped SE are increased. The gain in speed that is achieved when using the proposed method is different for the two SEs that have been used to obtain the results shown in Figure 6.13(c) and Figure 6.13(d). In fact, the gain in speed is proportional to the number of pixels that need to be considered in the computation of each output pixel. Since ordfilt2() appears to be processing all of the pixels in the window to compute the output at each pixel, it can be expected that the proposed method will always be faster than ordfilt2() by a factor that is equivalent to the percentage of pixels that remain unchanged as the window is translated by one pixel.

Figure 6.13(c) shows the time taken by the proposed method and ordfilt2() to compute the output image when using the hollow square SE. It is clear by observation of the plots that the proposed method always outperforms ordfilt2(). For each of the SEs of increasing size that were used, the proposed method outperforms

ordfilt2() by a factor of 2. This comes as a result of the fact that the number of critical points of the hollow square SE represents roughly half of the pixels while the other half remain in the window following translation by a single pixel. Assuming ordfilt2() uses all of the points in the window to compute the output at each pixel it seems reasonable that the proposed method takes half of the time when compared to that function.

Figure 6.13(d) shows that the time taken by the proposed method is less than that of ordfilt2() when using the H shaped SE. By reference of the plot shown in Figure 6.13(d) it is evident that the proposed method always outperforms ordfilt2(). However, the gain in speed when using the H shaped SE is not as great as when using the disk or the hollow square SE. In fact, the proposed method outperforms ordfilt2() by a factor of around 1/3 when using the H shaped SE. Intuitively, this is the expected result, since translating the H shaped SE by a single pixel to the right results in around 2/3 of the pixels in the window changing at each translation. Assuming that ordfilt2() uses all of the points in the window to compute the output at each pixel it follows that the proposed method takes 2/3 of the time to compute the output for the H shaped SE when compared to that function. Indeed it would be more efficient to rotate the image, and the SE, in order to compute the output of the H shaped SE such that the number of points remaining in the window after translation is maximised.

A comparison of the time taken for the proposed method and ordfilt2() to compute the output image when using the sin shaped SE is shown in Figure 6.13(e). This SE provides the lowest gain in speed when compared with ordfilt2(). The reason for this minimal gain in computation is that very few points in this SE overlap as it is translated by a single pixel to the right. In fact, if an ideal SE could be created for each increasing dimension of the sin shaped SE, then there would only ever be two overlapping points in the window as it is translated by one pixel to the right. As a result, the number of critical points for this SE is almost equal to the total number of pixels in the window itself. When the window is small, there is no real gain in using the proposed method when compared to ordfilt2(). This fact is confirmed in the plot

shown in Figure 6.13(e), where the execution time for both methods is almost identical. However, when the width of the SE reaches 20 pixels, the gain in speed of the proposed method appears to increase. This gain becomes greater as the width of the SE increases. The increasing gain in speed comes as a result of the fact that it is not possible to create an ideal, single pixel wide sinusoid, on such a small grid. For example, the peak and trough of the sin wave are represented by a single pixel when the window is small, however, as the window size is increased, a number of pixels are used to represent this same peak and trough. This means that as the window size increases, the percentage of points in the window that become critical points is reduced. It is this slight reduction in the percentage of critical points that results in the gain in efficiency when computing the output of the sin shaped window using the proposed technique. That is, as the number of critical points reduces with respect to the total number of pixels in the window, the execution time of the routine is reduced when compared to ordfilt2().

It is clear by reference of Figure 6.13 that the proposed method always outperforms Matlab's ordfilt2() function. When comparing the difference in execution times of these routines using a square SE of increasing size, the difference in execution time is marginal. This implies that Matlab have implemented ordfilt2() such that it exploits the property of the sliding window for square and rectangular windows. However, for all other SEs ordfilt2() appears to compute the filter output by performing some comparison of all the image pixels that coincide with each point in the window. The extent to which the proposed method outperforms ordfilt2() depends on the number of critical points in the window, however, even if there are no overlapping points in the window (e.g. a vertical line SE, ignoring the fact that it can be rotated) the proposed method is no more expensive to compute than the method used to implement ordfilt2. A number of SEs have been tested in this section, and to some extent, these SEs have been chosen to explore the limitations of this algorithm.

To the author's knowledge, it is not common practise to compute rank order filters using H shaped or sin shaped SEs. It is much more common to use solid SEs such as the disk and square SE, in which case, a large number of points remain in the

window following a translation by a single pixel. In such cases the proposed method can be used to significantly reduce the computational requirements of these filters.

6.5.4 A Fast POHMT

In order to assess the extent to which the method proposed in this chapter can be used to increase the efficiency of the POHMT, the images shown in Figure 6.14 have been used. A direct implementation of the POHMT has been used to detect the cells in these noisy images, and the execution time of this routine has been measured and compared with that of the fast POHMT. The fast POHMT has been implemented as described in Section 6.4 using the fast rank filters that have been described in this chapter.

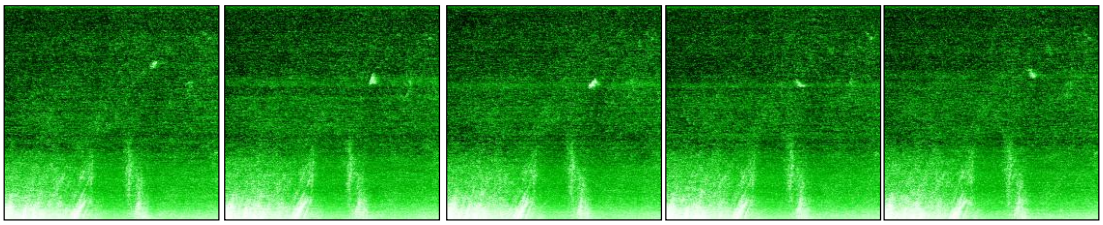


Figure 6.14 Images used to compare a direct implementation of the POHMT with a fast implementation using the fast rank order filters that have been described in this section. All images are 512 x 512.

The images shown in Figure 6.14 are the same as those used in Chapter 5 for testing the PO plot. As such, the results of applying the direct implementation and the fast implementation are not shown here since they are identical to the results shown in Chapter 5. The original images have been shown here simply to save cross referencing.

The dimensions of each of the images shown in Figure 6.14 are 512x512. Both routines have been implemented in Matlab, and these routines have been used to compute the output of the POHMT on a Dell Latitude Laptop with a 2.5Ghz, Core 2 Duo processor, with 2GB of RAM. The SEs used are the same as those describe in the experimental results section of Chapter 5, and the value of P has been set to 83, as was determined using a PO plot there. The average processing time of the direct

implementation is around 9 minutes while the fast POHMT produced the output image in an average time of 320ms. Clearly, there is a significant gain in speed when using the fast rank filters described in this section to implement the POHMT, particularly when compared with a naive direct implementation using Equation (4.20).

The POHMT, when implemented using rank order filters has also been used to detect the LSBs in the images that were used by Perret *et al.* to test their FHMT. Using their optimised FHMT, the authors in [8] state that their algorithm takes around 2 minutes to compute the output for the images shown in Figure 6.15. Again, these images are shown here to save cross referencing. The rank filter implementation of the POHMT has been used to locate the LSB in each of the images shown in Figure 6.15 where the results have been shown in Figure 6.15(b) to highlight the location of the LSB in the original image.

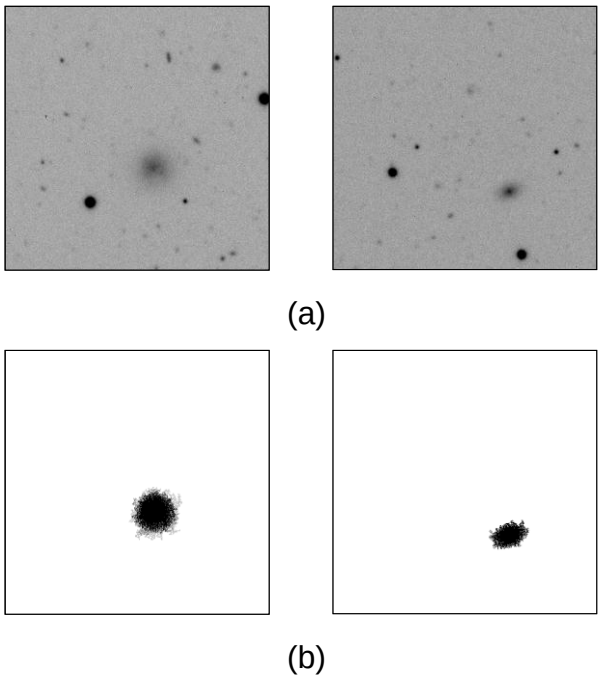


Figure 6.15 Noisy images containing LSB. (a) original images. (b) Result of applying the fast POHMT to locate the LSB and performing a reconstruction by dilation. All images are 512 x 512.

The POHMT, when implemented as described, computes the output in an average time of 15 seconds. This is a significant improvement on the time quoted for the

optimised FHMT in [8]. Further, some of the techniques used in [8] to optimise the FHMT are heuristic, and as a result, LSBs that are present may fail to be detected.

It must be noted that the POHMT only uses one pair of SEs, while the FHMT presented in [8] uses a large set of SEs, in order to automate the process of LSB detection. Each of these SEs is applied to the image and a measure of fitness is calculated and used to generate a score map from which the output of the FHMT is calculated (See Chapter 2 or [8]). The description of the FHMT that is presented in [8] is extremely well suited to the author's application, and it assumes that prior information about the sought features is available in the form of a mathematical model. This model is then used to automatically generate a set of templates that represent the likely spatial appearance of the sought feature in any image.

The POHMT on the other hand requires manual design of the SEs using prior information about the feature that is sought in the image. It tends to use one, or perhaps a small set of SEs, to locate features whose appearance is similar in all images. As a result, the two methods are not directly comparable in terms of execution times since the goal of each routine is very different. In cases where a mathematical model can be used to describe a feature whose geometry can differ significantly between images then method described in [8] is optimal, although it takes longer to compute. When the geometry of a sought pattern does not change significantly between images, the POHMT would be optimal as it executes in a fraction of the time proposed in [8] without using heuristic measures to increase efficiency. The comparison that has been performed here has been carried out in order to validate the efficiency of the fast POHMT and to demonstrate that it is capable of competing with current research in this area.

6.5.5 Summary

In this section an efficient method for computing the output of rank order filters within arbitrarily shaped windows has been presented. The algorithm is a combination of an extension of the efficient median filter presented by Huang *et al.* [54] and the method presented by Van Droogenbroeck and Talbot for computing

erosions and dilations within arbitrarily shaped windows [55]. The method that has been described in this thesis has been presented in the context of both of these methods, and the contribution of this work has been in highlighted in Table 6.6.1.

It should be noted that although Van Droogenbroeck and Talbot state in their paper that their method can be extended to compute arbitrary rank at no extra cost, they do not provide any computational results to support this. Further, they do not take into account the fact that the histogram must be consulted at each translation of the SE when computing ranks other than the maximum/minimum. This means that extra computation is required to retain a count of the pixels that are lower than the sought rank, in order to direct the search when a new value is to be located. The extension to arbitrary ranks is therefore more complex than implied in [55]. This leads to additional computational costs that can be significant when computing the output of any arbitrary rank filter as was demonstrated in Figure 6.7. Further, a mathematical formulation, which was not provided in [55], has been given in this chapter for computing the critical points of any arbitrarily shaped window. Breare and Lehmann claim to have used the method proposed in [55] to implement efficient rank order filters. However, no description of the algorithm that has been used is provided in [65]. Then, in the experimental results section of [65], the authors only show timing data when computing the output of a median filter, and they do not refer to the shape of the window which is used to filter the image.

The method that is described in this chapter has been rigorously tested, and the limitations of this algorithm have been explored. The proposed method has also been compared to an optimised Matlab function for computing the output of arbitrarily shaped rank order filters. It has been determined that when using square and rectangular windows, the Matlab routine and the proposed method execute in the same time as each other, and hence it can be assumed that Matlab have exploited the same properties of the sliding window that have been generalised in this work. However, when comparing the execution time of the other SEs, it is obvious that the proposed method always outperforms the optimised Matlab function. In some cases,

this gain in computation is extremely significant (in some cases, linear as opposed to quadratic), particularly when large, solid SEs are used.

Finally, it has been demonstrated that the POHMT can be implemented using rank order filters and the benefits of using the proposed method to implement the POHMT has been shown to significantly improve the execution time of the algorithm. The gain in efficiency is substantial when compared with that of a direct implementation of the routine. The fast POHMT has also been used to locate the LSB galaxies in the images used by Perret *et al.* to test their routine. It has been shown that the POHMT can detect these features in a fraction of the time taken by the FHMT as proposed in [8]. However, as was mentioned in Section 6.5.4, the POHMT and the FHMT both aim to solve two very different problems. While a direct comparison of the execution times for these methods does not indicate the superiority of either method, it has been included here to demonstrate that the POHMT is more than capable of competing with the state-of-the-art research in extending the Hit-or-Miss Transform for application to noisy images.

7 Conclusions and Further Work

This study initially set out to extend the standard HMT in order to make it more robust for detecting features in the presence of noise. The work that has been presented in this thesis has clearly achieved this goal and more.

Three major contributions have been presented in this body of work. The first of these is the extension of the HMT to the more general POHMT. The POHMT relaxes the strict constraints of the standard HMT by allowing features to be detected when only a percentage of the foreground and background probes are occupied. It is required that this percentage, P, is determined and set prior to applying the POHMT. Therefore, the PO plot, which is the second major contribution of this work, was developed as a design tool that can be used in order to accurately determine a suitable value for P. Further, it has been shown that the PO plot can be used to estimate similar parameters for a number of alternative HMTs that can be made more robust to noise by applying techniques similar to those used by the POHMT. Finally, due to the computational complexity of implementing the POHMT directly, a fast method for computing the transform was sought. Since the POHMT can be implemented using rank order filters, the solution was to generalise existing techniques for efficiently computing the output of this class of filter. As a result, the third contribution of this work lies in the description of an efficient method for computing the output of any rank order filter within an arbitrarily shaped window. Significant gains in speed have been demonstrated when implementing the POHMT using this technique.

This chapter draws a number of conclusions about the work that has been presented. The chapter concludes by providing a number of suggestions for future work that may be carried out as a result of the findings in this thesis.

7.1 The POHMT

Following the background notions that were presented in Chapter 2, and the literature review that was provided in Chapter 3, the discussion in Chapter 4 focused on extending the standard HMT to make it more robust in the presence of noise. At the outset, Chapter 4 discussed and defined a number of greyscale HMTs that have been presented by various researchers over the years. A new conceptual definition of the greyscale HMT was then introduced and placed in the context of these alternative greyscale HMTs. The conceptual definition of the HMT provided in Chapter 4 considers the greyscale image as a topographic surface in which a complementary pair of SEs search for, and mark, locations where they are both 100% occupied by a feature and its surrounding background. This definition of the HMT is consistent with each of the other greyscale HMTs that were discussed in Chapter 4. Given that a number of greyscale HMTs already exist, it is necessary to point out that the additional definition that has been introduced in this thesis was provided in order to facilitate the explanation of the novel techniques that are proposed here.

The extension of the HMT that was presented in Chapter 4 relaxes the strict conditions of the HMT by requiring that only a percentage of the SEs need be occupied in the image for a feature to be marked in the output. As such, the proposed extension of the HMT is known as a Percentage Occupancy Hit-or-Miss Transform. The POHMT allows partial fitting of the SEs by allowing features to be marked when the SEs used to probe the image are some percentage $P\% < 100\%$ occupied by a signal and its background. Expressions that can be used to compute the percentage occupancy of the SEs when their origin is centred at any given point in an image have been provided in Section 4.4.1. A formal definition of the POHMT, which is expressed in terms of these percentage occupancy calculations, was then provided in Section 4.4.2. It was also pointed out that the POHMT is simply a generalisation of the standard HMT and, as such, the standard HMT can be implemented as a special case of this more general transform.

Having defined the POHMT, and established that this extension of the HMT has only one parameter, *P*, two methods for setting *P* were then presented and tested. The first of these was an empirical method which involved computing the POHMT with *P* set initially to 100% before iteratively lowering the value of *P* until all the features of interest were detected. While this technique was successful, it proved to be time consuming, and, in cases where the iteration step is too large, it is clear that the optimal value for *P* may not be found.

The second technique used noise models (generated by cropping regions of the image data) to compute an appropriate value for *P* which proved slightly more robust than the empirical technique. The idea was to use probability distributions to estimate the percentage of foreground and background pixels that were likely to puncture the SEs when centred on the features of interest. This knowledge allowed the parameter *P* to be determined in less time than when using the empirical method. However, one drawback with this approach is that accurate noise models are not always readily available. Even when these can be cropped from the image, they are not always representative of the noise that is presented to the SE when it is centred on a feature of interest.

In Chapter 4, both techniques for setting *P* were tested on the same data that was processed by the POHMT. While this was useful for explaining the technique, it is not practical for real world applications. In such cases, training data should be used to determine a suitable value for *P* when using these techniques.

7.2 The PO Plot

Perhaps the most significant contribution of this thesis is the original design tool which was presented in Chapter 5. The tool, known as a PO plot, can be used to set the only parameter *P* of the POHMT, and even more interestingly, it can be used to set a number of similar parameters that make the alternative greyscale HMTs more robust to noise.

As it was explained in Chapter 5, a PO plot may be generated using any complementary pair of SEs by centring these at some point in an image, measuring the extent to which the foreground SE and the background SE are occupied at this point, and then plotting these quantities against each other. It was demonstrated that the resultant profile, exhibited in the PO plot, varies with different noise distributions and changes significantly with increasing amounts of noise. This variation in the profile is what allows the PO plot to be used for estimating the parameter *P*. Chapter 5 has explained, using a number of examples, how the PO plot can be generated.

It was also explained in Chapter 5 how the point where the profile crosses the 45° diagonal, known as the critical point, may be computed (See Chapter 5). It was demonstrated that when there are insufficient amounts of noise present in an image to prevent the standard HMT from operating successfully, that the profile in the PO plot forms an ideal right angle. The right angled profile intersects the 45° diagonal at the point indicating 100% occupancy. In such cases, *P* may be set to 100 and the standard HMT can be used to detect the feature of interest. However, in cases where noise power is increased, the profile shown in the PO plot tends more towards a curve. As a result, the point where this profile intersects the 45° line deviates further and further from the 100% case of the right angled profile. This concept has been thoroughly discussed and clearly explained by example in Chapter 5.

It was demonstrated in Chapter 5, that the PO plot could be generated using one complementary pair of SEs for a number of features in a noisy image, and that the minimum of these critical points could be calculated and used to set *P*. This concept

was tested further in the experimental results section of Chapter 5 where the PO plot proved to be a powerful tool for setting the only parameter of the POHMT.

In addition to being capable of setting *P*, the PO plot provides a number of other advantages that have been thoroughly explained in Chapter 5. It was shown that this novel design tool can be used to set an appropriate distance between the SEs for the HMTs presented by Ronse [27] and Soille [12] in order to make them more robust in the presence of noise. Additionally, it was shown that the PO plot could be used to determine an appropriate threshold that can be used to increase the noise robustness of the HMTs presented by Khosravi and Schafer [26] and Barrat *et al.*[5].

Another property of the PO plot that was described and demonstrated in Chapter 5 is that it can be interpreted in such a way that it is possible to make the POHMT operate as a discriminatory filter. This additional functionality allows the transform to selectively mark or discard image features based on the extent to which they occupy the SEs. Chapter 5 demonstrated that this information can be used to isolate features in a given image thus allowing any combination of image features to be selectively detected or discarded.

All of the properties of the PO plot that were discussed in Chapter 5 were demonstrated in the experimental results Section of that chapter. The experimental results validated all of the theory that was presented in Chapter 5, and the major benefits of using the PO plot to set parameters for the POHMT and other HMTs were made clear. The PO plot was used to set parameters for the POHMT, and for a number of alternative greyscale HMTs, in order to compare their performance in noisy images. It was demonstrated that the POHMT could cope with larger amounts of noise than the competing techniques. It was also shown that using the PO plot to set *P* and using the POHMT provided similar results to those achieved by Perret *et al.* when processing the data that they used to test the FHMT. The only drawback, which was pointed out at the end of chapter 5, is that a direct implementation of the POHMT is computationally expensive. This hurdle was overcome in Chapter 6 where a fast algorithm for computing the POHMT was introduced.

7.3 The Fast POHMT

The experiments that were carried out to produce the results that concluded Chapter 5 highlighted the fact that the POHMT was extremely inefficient when implemented directly. As a result, a fast algorithm was needed to make the POHMT useable in practice.

By recognising that it is possible to compute the output of the POHMT using rank order filters, the search for a fast POHMT, became a search for an efficient method for computing the output of rank order filters within arbitrarily shaped windows. Two potentially applicable methods were found: one presented by Hung *et al.* [54] that is capable of computing the output of a median filter using square and rectangular windows; and a method presented by Van Droogenbroeck and Talbot [55] for computing maximum and minimum filters within arbitrarily shaped windows. Both authors exploit the property of the sliding window when computing the output of these rank order filters, and both authors use histograms to sort the pixels that coincide with the "critical points" of the sliding window.

The novel contribution in this thesis lies in detailing the extension that can be made to the method proposed in [55] such that it can be used to compute the output of any rank order filter defined by any arbitrarily shaped window. It was shown that this can be easily achieved by combining a slight extension of the method presented in [54] such that instead of computing the median value, the value of any rank can be computed. In addition to this, mathematical formulations have been introduced in this thesis which can be used to calculate the set of critical points that are required for updating the histogram. Having described the extension of [55] and explained it by example in Chapter 6, the relationship between the POHMT and rank order filters was explained. Further, it was also explained how P can be used to set the rank parameter of the filters, and a description of this relationship was provided.

Chapter 6 concluded by providing a number of speed comparisons that highlighted some properties of the proposed method. It was shown that efficiency of the proposed routine is dependent on the rank that is to be computed in the window. It

was also demonstrated that the shape and size of the window affected the execution time of the routine. Further, the effect that the ratio of the critical points when compared to the total number of points the window has on the algorithm was shown to be significant. It was also shown that the fast algorithm is in fact data dependent.

It was pointed out in the experimental results section of this chapter, that to the author's knowledge when submitting this thesis, there was no method that could be compared to the method proposed in this thesis for efficiently computing the output of more general rank order filters. As a result, the proposed method was compared to an optimised Matlab function for computing the output of rank order filters in 2 dimensional windows. Interestingly, the execution times for both routines when using square windows was the same. This indicated that ordfilt2() is implemented in the same way as the proposed method for filtering with square and rectangular windows. However, when other arbitrarily shaped SEs were used, the proposed method always outperformed the optimised Matlab function. The extent to which the proposed method was faster was shown to be related to the percentage of points in the window that become critical points during translation. It was explained that as the percentage of the SE which becomes critical points decreases the efficiency of the proposed fast algorithm increases.

Finally, the fast POHMT, implemented using the proposed method for computing rank order filters, was shown to provide a significant gain in speed when compared with a direct implementation of this transform. It was also shown that the POHMT, when implemented using rank order filters, is capable of locating the LSB galaxies in the images that were used to test the FHMT which was presented in [8]. It was explained that this comparison was only used to prove that the POHMT was capable of competing with the most recent methods for making the standard HMT more robust to noise. However, it was also made clear that since both of these methods aim to achieve two completely different things, a comparison of efficiency is not a measure of superiority of one algorithm when compared to the other.

7.4 Further Work

There are a number of areas that could be developed in order to build upon and extend the ideas and techniques that have been presented in this thesis. First, given that the profile in the PO plot changes with increasing amounts noise, and given that its shape changes for different noise distributions, it is thought that the PO plot could be used to estimate noise in image data. This could provide a number of advantages in cases when the robustness of an algorithm can be improved given accurate estimates of image noise. Further, if the power and distribution of noise that is corrupting a data set can be accurately quantified, it is more likely that this noise can be suppressed, or in some cases, removed from the image.

It is thought that perhaps calculating the area under the curve, (analogous to techniques used to characterise ROC curves) may provide an insight into the power of noise that is corrupting the image. Alternatively, it may be possible to generate a large set of model curves for varying distributions of noise at a number of increasing powers. These model curves could be used as a benchmark for estimating noise where samples of interest could be compared to this model set. It has also been noticed that the profile of the curve in the PO plot is affected when there are different amounts of noise on the foreground and background regions of the image.

Perhaps one advantage of using the PO plot to estimate noise could lie in the fact that it is capable of displaying information about foreground and background noise simultaneously. A second advantage may exist as a result of the fact that a single PO plot can be used to characterise a large number of features in a given image. A common technique for obtaining noise models and estimating noise is to select what is assumed to be a homogeneous region of an image. Noise estimates can then be obtained by plotting a histogram of the selected region and analysing it. If multiple regions of the image were selected for analysis, and their histograms were plotted on the same axis, a lot of information from each region could potentially be lost. Since the PO plot can display profiles of a large number of features simultaneously, then unlike the histogram method, information about the noise corrupting one feature and its background is retained in the PO plot.

A second avenue for further work lies in implementing an even faster algorithm than the one that was presented in Chapter 6 for computing the POHMT. In [70], Urbach and Wilkinson present an extremely efficient method for computing the output of erosions and dilations within arbitrarily shaped windows. Their method is compared to the one presented by Van Droogenbroeck and Talbot in [55], and it is shown that a significant gain in speed is achieved when using the method proposed in [70].

Since the method proposed in [55] is a special case of the more general algorithm that is presented in Chapter 6 of this thesis, it is clear that the method proposed in [70] is more efficient than the method proposed here when the special case of minimum and maximum filters are used. However, the authors of [70] point out in their paper that is not possible to extend their method in order to compute the output of more general rank order filters. It is for this reason that the Van Droogenbroeck and Talbot method was extended in this thesis.

However, due to the fact that the method proposed in [70] is so efficient, preliminary findings suggest that there may be a way of extending it such that it can be used to compute the output of more general rank order filters. Further work will therefore investigate the possibility of computing the minimum value using the method proposed in [70], then by removing the minimum and rerunning the algorithm, the second rank will be found. Removing the value of the second rank and recalculating the minimum would result in the third rank being computed, and so on, until the median is reached. When computing ranks greater than the median, the algorithm should start by computing the dilation and the next maximum and so on until the sought rank is found. The execution time of this method would therefore become a function of the rank that is to be computed by the filter. The position of the rank that is desired, which will determine the number of times the routine must be reused, will determine whether it is more efficient than the one proposed in Chapter 6.

The efficiency and usability of many morphological operators has been enhanced by implementing them in hardware. It would perhaps be interesting to implement the fast POHMT in hardware in order to determine whether or not the routine is capable

of detecting features in times that would allow it to be used in an online detection system where data needs to be processed in video rates. Currently, despite the fast algorithm proposed in Chapter 6, it is unlikely that the POHMT would be used to detect features when real time online processing is required.

Finally, most applications of the POHMT have so far focused on the analysis of biological images. The standard HMT has been used in a large number of applications from optical character recognition to face detection to mention but a few. It would be interesting to investigate the improvement in the robustness that is offered in these application areas when using the fast POHMT and the PO plot that have been introduced in this thesis.

References

1. Matheron, G., *Random sets and integral geometry*. New York: Wiley, 1974.

2. Serra, J.P., *Image analysis and mathematical morphology*. London; New York: Academic Press, 1982.

3. Heijmans, H. J., *Morphological Operators*. New York: Academic Press, 1994.

4. Naegel, B., Passat, N., Ronse, C., "Grey-level hit-or-miss transforms-part II: Application to angiographic image processing." *Pattern Recognition*, 2007. 40(2): p. 648-658.

5. Barat, C., Ducottet, C., Jourlin, M., "Pattern matching using morphological probing." in *Proceedings International Conference on Image Processing, ICIP 2003*,. 2003. Barcelona, Spain, p.I - 369-372.

6. Bloomberg, D.S., Vincent, L., "Pattern matching using the blur hit-miss transform." *Journal of Electronic Imaging*, (9), 2000. p 140-150.

7. Raducanu, B., Grana, M., "A grayscale hit-or-miss transform based on level sets." in *Proc. International Conference on Image Processing, ICIP 2000 . 2000*. p. 931-933.

8. Perret, B., Lefevre, S., Collet, C., "A robust hit-or-miss transform for template matching applied to very noisy astronomical images." *Pattern Recognition*, 2009. 42 (11): p. 2470-2480.

9. Lefevre, S., Weber, J., "Automatic Building Extraction in VHR Images Using Advanced Morphological Operators." in *Urban Remote Sensing Joint Event*, 2007.

10. Puissant, A., Weber, J., Lefevre, S., "Coastline Extraction in VHR Imagery Using Mathematical Morphology with Spatial and Spectral Knowledge." *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 2008. 37(3): p. 1305-1310.

11. Harvey, N., Porter, R., Theiler, J., "Ship detection in satellite imagery using rank-order greyscale hit-or-miss transforms.", 2010, Available from: http://www.osti.gov/servlets/purl/993121-0WQKfL/.

12. Soille, P., *Morphological Image Analysis: Principles and Applications*. 2nd Edition, Springer-Verlag New York, Inc., 2003.

13. Gabbouj, M., Coyle, E.J., Gallagher, N.C., "An overview of median and stack filtering." *Circuits, Systems, and Signal Processing*, 1992. 11(1): p. 7-45.

14. Marshall, S., *Logic-based nonlinear image processing*. Bellingham, Wash, SPIE Press, 2007.

15. Harvey, N., *New Techniques for the Design of Morphological Filters Using Genetic Algorithms.* PhD Thesis, University of Strathclyde, 1997.

16. Gonzalez, R.C., Woods, R.E., *Digital image processing*. 3rd Edition, Upper Saddle River, NJ: Prentice Hall, 2007.

17. Sonka, M., Hlavac, V., Boyle., R., *Image processing, analysis, and machine vision*. London: Chapman & Hall, 1993.

18. Haralick, R.M., Shapiro, L.G., *Computer and robot vision*. Volume 1, Reading, MA.: Addison-Wesley Pub. Co., 1992.

19. Pitas, I., Venetsanopoulos, A.N., "Order statistics in digital image processing." *Proceedings of the IEEE,* 1992. 80(12): p. 1893-1921.

20. Harvey, N.R., Marshall, S., "Rank-Order Morphological Filters: A New Class of Filters." *In IEEE Workshop on nonlinear signal and image processing*, 1994: p. 975 - 978.

21. Soille, P., "On morphological operators based on rank filters." *Pattern recognition.*, 2002. 35: p. 527-536.

22. Maragos, P., Schafer, R., "Morphological filters--Part II: Their relations to median, order-statistic, and stack filters.", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 1987. 35(8): p. 1170-1184.

23. Hawkes, P.W., *Image mathematics and image processing*. Boston: Academic Press, 1992.

24. Naegel, B., Passat, N., Ronse, C., "Grey-level hit-or-miss transforms-Part I: Unified theory." *Pattern Recognition,* 2007. 40(2): p. 635-647.

25. Murray, P., Marshall, S., "A New Design Tool for Feature Extraction in Noisy Images Based on Grayscale Hit-or-Miss Transforms.", *IEEE Transactions on Image Processing,* 2011. 20(7): p. 1938-1948.

26. Khosravi, M., Schafer, R.W., "Template matching based on a grayscale hit-or-miss transform.", *IEEE Transactions on Image Processing,* 1996. 5(6): p. 1060 - 1066.

27. Ronse, C., "A Lattice-Theoretical Morphological View on Template Extraction in Images." *Journal of Visual Communication and Image Representation,* 1996. 7(3): p. 273-295.

28. Soille, P., *Advances in the Analysis of Topographic Features on Discrete Images.* Lecture notes in computer science., (2301): p. 175-186, 2002.

29. Passat, N., Ronse, C., Baruthio, J., Armspach, J.P., "Automatic Parameterization of Grey-Level Hit-or-Miss Operators for Brain Vessel Segmentation." *Proc. International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05).* Philadelphia, PA, USA, 2005, pp. 737–740.

30. Passat, N., Ronse, C., Baruthio, J., Armspach, J.P., Maillot, C., "Magnetic resonance angiography: From anatomical knowledge modeling to vessel segmentation." *Medical Image Analysis*, 2006. 10(2): p. 259-274.

31. Naegel, B., Ronse, C., Soler, L., "Using grey scale hit-or-miss transform for segmenting the portal network of the liver." *Computational Imaging and Vision*, 2004. 30: p. 429-440.

32. Bloomberg, D., Maragos, P.A., "Generalized hit-miss operations." *SPIE Conference 1350, Image Algebra and Morphological Image Processing*, San Diego, CA, 1990, p. 116–128.

33. Zhao, D., Daut., D.G., "Shape recognition using morphological transformations." in *International Conference on Acoustics, Speech, and Signal Processing,* (*ICASSP-91*). Toronto, Canada, pp. 2565–2568.

34. Kraus, E.J., Dougherty, E.R., "Segmentation-free morphological character recognition" [2181-03]. *Proceedings- SPIE The International Society for Optical Engineering*, 1994(2181): p. 14.

35. Gillies, A.M., "Automatic generation of morphological template features." in SPIE conf. Image Algebra and Morphological Image Processing. 1990: San Diago, CA. p. 252-261.

36. Wilson, S.S., "Training Structuring Elements in Morphological Networks, in Mathematical." *Morphology in Image Processing*, E.R. Dougherty, Editor, Marcel Dekker: NY.

37. Zhao, D., Daut., D.G., "An efficient approach to automatic shape recognition." in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1990, (*ICASSP-90*). p.2161 - 2164.

38. Zhao, D., Daut, D.G., "Morphological hit-or-miss transformation for shape recognition." *Journal of Visual Communication and Image Representation*, 1991. 2 (3): p. 230-243.

39. Doh, Y., Kim, J., Kim, J., Kim, S., Alam, M.S., "New morphological detection algorithm based on the hit-miss transform." *Optical engineering: the journal of the Society of Photo-optical Instrumentation Engineers*, 2002. 41(1): p. 26-31.

40. Maragos, P., "Optimal Morphological Approaches To Image Matching And Object Detection." in *Proc. Second International Conference on Computer Vision,*. 1988., Tampa, Florida, p. 655-699.

41. Crimmins, T.R., Brown, W.M., "Image Algebra and Automatic Shape Recognition." *IEEE Transactions on Aerospace and Electronic Systems*, 1985. AES-21(1): p. 60-69.

42. Duda, R.O., Hart, P.E., *Pattern classification and scene analysis*. New York: Wiley, 1973.

43. Preston, K., Duff, M.J.B., *Modern cellular automata: theory and applications*. New York: Plenum Press, 1984.

44. Rosenfeld, A., Kak, A.C., *Digital picture processing*. New York: Academic Press, 1982.

45. Wilson, S.S., "Vector morphology and iconic neural networks." *IEEE Transactions on Systems, Man, and Cybernetics*, 1989. 19:6: p. 1636-1643.

46. Agam, G., Dinstein, I., "Regulated morphological operations." *Pattern recognition*, 1999. 32(6): p. 947.

47. Casasent, D., Schaefer, R., Sturgill, R. "Optical hit-miss morphological transform" *Applied Optics*, 1992. 31(29): p. 6255-6263.

48. Gilmour, J.E., Casasent, D.P., "Split Threshold Hit/Miss Transform for Mines" [2765-15]. *Proceedings- SPIE The international Society for Optical Engineering*, 1996(2765): p. 149-156.

49. Perret B., B., Lefevre, S., Collet, C., and Vollmer, B., "Astronomical object detection with a robust hit-or-miss transform." in *European Signal Processing Conference (EUSIPCO '08)*, 2008, Lausanne, Switzerland, p. 1-5.

50. Louys, M., Perret, B., Vollmer, B., Bonnarel, F., Lefèvre, S., Collet, C., "LSB Galaxies Detection Using Markovian Segmentation on Astronomical Image." *Astronomical Society of the Pacific Conference Series*, 2008. 394: p. 125-130.

51. Collet, C., Flitti, F., "Variations on Markovian quadtree model for multiband astronomical image analysis." in *International Symposium on Applied Stochastic Models and Data Analysis*, 2005: France.

52. http://terapix.iap.fr/.

53. Lusted, L.B., "Decision-making studies in patient management." *The New England journal of medicine*, 1971. 284(8): p. 416-24.

54. Huang, T., Yang, G., Tang, G., "A fast two-dimensional median filtering algorithm" *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1979. 27(1): p. 13-18.

55. Van Droogenbroeck, M., Talbot, H., "Fast computation of morphological operations with arbitrary structuring elements." *Pattern Recognition Letters*, 1996. 17(14): p. 1451-1460.

56. van Herk, M., "A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels." *Pattern Recognition Letters*, 1992. 13(7): p. 517-521.

57. Gil, J., Werman, M., "Computing 2-D min, median, and max filters.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993. 15(5): p. 504-507.

58. Breen, E., Soille, P., "Generalization of van Herk recursive erosion/dilation algorithm to lines and arbitrary angles." in *DICTA '98: Digital Image Computing: Techniques and Applications*, K.F.a.A. Ginige, Editor. 1993, Australian Pattern Recognition Society: Sydney. p. 549-555.

59. Soille, P., Breen, E.J., Jones, R., "Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996. 18(5): p. 562-566.

60. Gil, J.Y., Kimmel, R., "Efficient dilation, erosion, opening, and closing algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 24(12): p. 1606-1617.

61. Garibotto, G., Lambarelli, L., "Fast on-line implementation of two-dimensional median filtering." *Electronics Letters*, 1979. 15(1): p. 24-25.

62. Tukey, J. "Nonlinear (nonsuperposable) methods for smoothing data", in Cong. Rec., *EASCON*. 1974, p.673.

63. Tukey, J., *Exploratory data analysis*. Reading, Mass.: Addison-Wesley Pub. Co., 1977.

64. Chaudhuri, B.B., "An efficient algorithm for running window pel gray level ranking 2-D images." *Pattern Recognition Letters.*, 1990. 11(2): p. 77-80.

65. Beare, R., Lehmann, G., "Efficient implementation of kernel filtering." in *Proceedings of ISCNAMIC Workshop on Open Science at MICCAI 2007*. p.1–18.

66. Narendra, P.M., "A separable median filter for image noise smoothing.", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1981. 3(1): p. 20-29.

67. Sun, C., Diamond, "Hexagon, and General Ploygonal Shaped Window Smoothing." in *Proc. IIVth Digital Image Computing: Techniques and Applications*. 2003: Sydney. p. 39 - 48.

68. Yaroslavsky, L., Bilevich, L., "Recursive algorithms for image local statistics in nonrectangular windows." in *Int. Workshop Spectral Methods Multirate Signal Processing.*

69. Perreault, S. and Hebert P., "Median Filtering in Constant Time", *IEEE Transactions on Image Processing,* 2007. 16 (9): p. 2389-2394.

70. Urbach, E.R., Wilkinson, M.H.F., "Efficient 2-D Gray-Scale Dilations and Erosions with Arbitrary Flat Structuring Elements." *IEEE Transactions on Image Processing*, 2008. 17(1): p. 1 - 8.

71. Urbach, E.R., Wilkinson., M.H.F., "Efficient 2-D Gray-Scale Dilations and Erosions with Arbitrary Flat Structuring Elements." in *IEEE International Conference on Image Processing*, 2006, p. 1573 – 1576.

72. Ji, L., Piper, J., Tang, J.-Y., "Erosion and dilation of binary images by arbitrary structuring elements using interval coding." *Pattern Recognition Letters*, 1989. 9(3): p. 201-209.

73. Soille, P., Talbot, H., "Directional morphological filtering.", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2001. 23(11): p. 1313-1329.

74. Van Droogenbroeck, M., Buckley, M.J., "Morphological Erosions and Openings: Fast Algorithms Based on Anchors." *Journal of Mathematical Imaging and Vision*, 2005. 22(2): p. 121-142.

75. Boomgaard, R.v.d., v. Balen, R., "Methods for fast morphological image transforms using bitmapped binary images." *CVGIP: Graph. Models Image Process.*, 1992. 54(3): p. 252-258.

76. Young, I.T., Pevereni, R., Verbeek, P., Otterloo, P., "A new implementation for the binary and Minkowski operators." *Computer Graphics and Image Processing*, 1981. 17(3): p. 189-210.

77. Murray, P., Marshall, S., "A Fast Method for Computing the Output of Rank Order Filters Within Arbitrarily Shaped Windows." *Conference Proceedings*: *19th European Signal Processing Conference (EUSIPCO 2011)*, p. 1668 – 1672.

78. Murray, P., Marshall, S., Bullinger, E., "The Percentage Occupancy Hit or Miss Transform." *Conference Proceedings: 17th European Signal Processing Conference, (EUSIPCO-2009)*, p. 253 – 257.

79. Glover, I., Grant, P.M., *Digital communications*. London; New York: Prentice Hall, 1998.

80. Wendt, P., Coyle, E., Gallagher, N. Jr., "Stack filters.", *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1986. 34(4): p. 898-911.

81. Mathworks, *Ordfilt2() Help file.* Matlab, R2011b Documentation, Image Processing Toolbox, 2011.

Publications by the Author

Below is a list of papers which have resulted from the work that has been described in this thesis:

1. Murray, P., Marshall, S., "Selectively Filtering Image Features Based on a Percentage Occupancy Hit-or-Miss Transform", *Proc. IET Conference on Image Processing 2012*. (accepted for publication)

2. Murray, P., Marshall, S., "A Fast Method for Computing the Output of Rank Order Filters within Arbitrarily Shaped Windows", *Proc. The 19th European Signal Processing Conference* (EUSIPCO 2011), pp.253-257, Aug.2011.

3. Murray, P., Marshall, S., *A New Design Tool for Feature Extraction in Noisy Images Based on Grayscale Hit-or-Miss Transforms*. Image Processing, IEEE Transactions on, 2011. 20(7): p. 1938-1948.

4. Murray, P., Garmendia-Torres, C., Michael, S. A., Marshall, S. Flann, N. S., Shmulevich, I., Dudley, A. M., "Using image processing to determine the spatial dynamics of proteins and RNAs in single cells", presented at the 2010 meeting on: Automated & High-Throughput Phenotyping, at Cold Spring Harbour Laboratory, New York.

5. Hirst, A.R., Roy, S., Arora, M., Das, A.K., Hodson, N., Murray, P., Marshall, S., Javid, N., Sefcik, J, van Esch, J. H., Santabarbara, S., Hunt, N.T., Ulijn, R.V. *Biocatalytic induction of supramolecular order*. Nature Chemistry. (2010) ISSN 1755-4330.

6. Murray, P., Marshall, S., Bullinger, E., "The Percentage Occupancy Hit-or-Miss Transform", *Proc. The 17th European Signal Processing Conference* (EUSIPCO '09), Aug.2009, pp.253-257.