UNIVERSITY of LIVERPOOL

Morphological Filtering In Signal/Image Processing

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor of Philosophy

in

Electrical Engineering and Electronics

by

Mohammad Hossein Sedaaghi , BSc, MSc

June 1998

In The Name Of The Most High

Morphological Filtering
In Signal/Image Processing

by

Mohammad Hossein Sedaaghi

Dedication

To:

My beloved parents and brothers who encouraged me, My supervisor who enabled me, My friends and staff in Liverpool University who helped me, Sahand University of Technology who sponsored me, and My wife and sons who put up with me.

♡

Acknowledgements

I would like to express my gratitude to my supervisor Professor Q.H. Wu, not only for his dedicated and thorough supervision, but also for his wise advice at moments of decision, his support, and encouragement at moments of uncertainty , and for being source of inspiration throughout my Ph.D. programme. He gave me intuition coming from the mathematical beauty and guided me to enthusiasm for imagination.

I owe a debt of thanks to Prof. P. J. G. Lisboa who gave me initial ideas about the research.

I would like to thank Dr. Goodyear who showed me the beauty of information theory, coding and signal processing, Dr. Cheetham for signal processing course, Dr. Shimmin who exposed me to Matlab-package approach.

Also, I have been very fortune to be surrounded by some very giving friends, and I would like to take this opportunity to express my appreciation.

I would like to acknowledge the financial support provided by Sahand University of Technology, Tabriz, Iran.

I would like to thank Professor Michael Weindling, Dr. Rosaline Garr and the rest of the staff in Neonatal Unit, Liverpool Women's Hospital who provided me a chance to develop the ECG algorithms for recording and analysing fetal ECG. I am deeply grateful to the Royal Liverpool University Hospital and Marquette Electronics Co. Ltd., especially Dr. Shankara Reddy for providing ECG data and guiding us to a better ECG analysis, Thorn EMI Co. for providing fingerprint samples, NIST (National Institute of Standards and Technology, Gaithersburg, MD,USA) for providing fingerprint classifier database on a CD, CIS (Center for Imaging Science, Electrical Engineering Department, Washington University) for providing valuable images, and The University of Texas Health Science Center in San Antonio for priceless images available in "Image Tool" package.

Also I would like to thank Dr. R. van den Boomgaard for providing me valuable sources related to mathematical morphology such as his PhD. thesis and also documents of ASCI course. I would like to thank Dr. H. J. A. M. Heijmans, the editor of morphology digest. I was a member of the list and found it very useful and updated.

Last, but certainly not least, my warmest appreciation goes to my wife, sons, parents and brothers for giving me the love and understanding that without their persistent encouragement, I would not have been able to complete this research.

Abstract

Morphological Filtering
In Signal/Image Processing

by

Mohammad Hossein Sedaaghi

The objective of this research is to develop a methodology which meets different specifications and representations of fast, efficient and real-time signal/image processing by means of mathematical morphology. Mathematical morphology is a relatively new non-linear theory for image processing, based on set theory. It considers images as sets which permits geometry-oriented transformations of the images. This approach seems very appropriate for dealing with objects in images, and it has gained increasing attention in recent years. It was first developed for binary images, then extended for greyscale images, and finally, generalised for sets in a generic mathematical space, called complete lattices. The practical realisation of the algorithms is one of the major motivations of this research. The differences, advantages and drawbacks between our approach and traditional methods are discussed. This thesis begins with a systematical introduction to mathematical morphology. It first explains historical background of mathematical morphology (MM). Then it tries to convince the reader with the ability of MM as an advanced nonlinear tool for image processing. This work also performs a comparison between linear and morphological representations. Various morphological operations and their definitions are presented. The profound algebraic similarity, and the qualitative differences between the approaches are presented and analysed. After the definition of the operators, a great effort is done to propose fast and time-saving algorithms. Real-time implementation of morphological operators is then introduced. A

hardware architecture for parallel computation of morphological operators is presented. Convolved morphological filtering as hybrid representations combining both linear and morphological approaches, are also considered. New morphological operators, called weighted morphological operators, are investigated which their efficiency dominate over classical morphological operators. Some applications of morphological filtering in fingerprint processing and ECG waves detection are demonstrated. In these applications, many approaches, such as directional filtering and syntactic analysis, to pattern recognition and segment detection are involved.

Our major contributions are mainly concentrated on some critiques about the conditions of defining the opening and closing operators, introducing fast algorithms for open-closing and close-opening, generalisation of real-time implementation of morphological operators in 1-D and 2-D, hybrid representation of MFs with convolution techniques, introducing new novel operators as weighted morphological filters, application of MF as a pre-processor for fingerprint processing, and as efficient technique for ECG wave filtering and analysis.

Contents

List of Figures

List of Tables

Abbreviations, acronyms and symbols

Abrev.	Description
ECG	electrocardiogram
m-D	m dimensional ,m=1,2,...
MM	mathematical morphology
MF	morphological filter
CMF	convolved morphological filter
CMO	convolved morphological operator
WMF	weighted morphological filter
WMO	weighted morphological operator
BN	binary
GS	grey-scale
GSMF	grey-scale morphological filter
SA	similar as Minkowski addition/subtraction
	for binary dilation/erosion
SE	structuring element
GSE	grey-scale structuring element
FP	function processing
SP	set processing
FSP	function-and-set processing
PAC	Premature Atrial Contraction
POSET	partly ordered set
$\emptyset$	empty set
$\hat{B}$	symmetric set of B about the origin
A/B	set difference (BN)

continued on next page

continued from previous page

Abrev.	Description
$A \oplus B$	dilation of A by B (BN)
$A \overset{M}{\oplus} B$	Minkowski addition of A by B (BN)
$A \ominus B$	erosion of A by B (BN)
$A \overset{M}{\ominus} B$	Minkowski subtraction of A by B (BN)
$A \circ B$	opening of A by B (BN)
$A \bullet B$	closing of A by B (BN)
$B \Uparrow A$	B hits A (BN)
$B \subset A^c$	B misses A (BN)
$A \circledast B$	hit-or-miss transform (BN)
$\square$	foreground in binary image
.	background in binary image
$+$	the origin of SE in binary image
$\cup$	union
$\cap$	intersection
$f \oplus g$	dilation of f by g (GS)
$f \ominus g$	erosion of f by g (GS)
$f \circ g$	opening of f by g (GS)
$f \bullet g$	closing of f by g (GS)
$F \oplus G$	dilation of F by G (2-D GS)
$F \ominus G$	erosion of F by G (2-D GS)
$F \circ G$	opening of F by G (2-D GS)
$F \bullet G$	closing of F by G (2-D GS)
$C(A)$	Convex hull
$A \otimes B$	Thinning
$A \odot B$	Thickening

continued on next page

continued from previous page

Abrev.	Description
$S(A)$	Skeleton
$g(f)$	Gradient
$h(f)$	Top-hat transform

Chapter 1

Introduction

1.1 Motivation

Computer science, mathematics and electrical engineering have very broad and rapidly expanding fields of study in many tasks of computer vision and image processing [58]. Jackway [39] addresses Levine's ([49]) description of computer vision as "Computer vision largely deals with the analysis of pictures in order to achieve results similar to those obtained by man." Human being has a visual system that enables him to see and process a large amount of visual information quickly, easily and without a great effort. However, for example, for counting and measuring particles in images iteratively, machines show better performance without getting tired [58]. The important problem is that all information is buried in a raw image implicitly, but not in a useful form, and images are ambiguous at each of many levels of a contextual hierarchy. Image analysis tries to identify specific subparts of the image and to analyse them to extract useful information, and present them in a form that emphasises the desired characteristics. However there is no general purpose automatic vision system, yet, compared to that of the human [58].

Different techniques are applied in signal processing depending on the type

of information carried by the signal. Image processing has thus developed its own set of techniques apart from those used widely in signal processing [39]. Pavlidis [61] says that the recent years of research have only had a small progress in practical applications of image analysis. A multitude of methodologies and techniques exist without a general framework indicating the scope of each. There is no coherent theory as in classical physics and mathematics, for example. A widespread attitude, used by researchers, is empiricism: try something (or even anything), and see what happens ([39]).

The important point is that analysis technique must be matched to the way in which information is being carried in the signal [6] (as Jackway [39] addresses).

Images in general are binary or multi-valued. Binary images are a mapping from 2D to 1D space, where the range is 0 or 1. There are different multi-valued image mapping as shown in table 1.1 ([28]) mapping from $mD \rightarrow nD$.

Table 1.1: Several types of multivalued images.

Mapping		$mD \rightarrow nD$	Description
$f:(x;y)$	$\rightarrow Y$	$2 \rightarrow 1$	grey-tone image
$f:(x;y;z)$	$\rightarrow Y$	$3 \rightarrow 1$	3D grey-tone image
$f:(x;y;t)$	$\rightarrow Y$	$3 \rightarrow 1$	moving grey-tone image sequence
$f:(x;y)$	$\rightarrow (R;G;B)$	$2 \rightarrow 3$	colour image
$f:(x;y;z)$	$\rightarrow (R;G;B)$	$3 \rightarrow 3$	3D colour image
$f:(x;y;t)$	$\rightarrow (R;G;B)$	$3 \rightarrow 3$	moving colour image sequence
$f:(x;y)$	$\rightarrow (Vx;Vy)$	$2 \rightarrow 2$	dense motion vector fields
$f:(x;y;t)$	$\rightarrow (Vx;Vy)$	$3 \rightarrow 2$	dense motion vector sequence
$f:(x;y;z;t)$	$\rightarrow (R;G;B)$	$4 \rightarrow 3$	4D colour image
$f:(x;y;z;t)$	$\rightarrow (\alpha,\beta,\gamma,\pi,\tau,\ \mu,\ldots)$	$4 \rightarrow ?$	real world

1.2 Mathematical morphology in image processing

1.2.1 Historical notes

Matheron [54] and Serra [78] are considered as the leaders of the field of mathematical morphology (MM). They initiated a theory for the analysis of spatial structures in image processing at the Fontainebleau research centre of the Paris School of Mines in 1964 [80], where Matheron was asked to investigate the relationship between the geometry of porous media and their permeabilities, and when J. Serra, at the same time was asked to quantify the petrography of iron ores, in order to predict their milling properties [78]. Due to their pioneering work, MM has achieved the status of a powerful tool in image processing with applications in materials science, microscopic imaging, pattern recognition, medical imaging, and computer vision. MM has become an important formalism in image processing and computer vision as a geometrical approach and is considered as a powerful tool for geometrical shape analysis ([88]). The original theoretical work of Matheron and Serra as a quantitative description of shape and size was initially used for binary images. However it can now be applied to grey-scale images and even for complete lattices [36], and can be realised by special hardware [72, 75, 44].

MM is a branch of digital image processing and analysis that uses concepts from algebra (set theory, complete lattices) and geometry (translation, distance, convexity) [37]. In set-theoretic methodology, signals are modeled as sets, systems (signal transformations) are viewed as set mappings, and translation-invariant systems are uniquely characterised by special collections of input signals [52].

Haralick [33] addresses some applications of morphological filtering in Golay logic processor [23], Diff3 [25], PICAP [47], the Leitz Texture Analysis System

TAS [43], the CLIP processor arrays [13], and the Delft image processor DIP [22].

1.2.2 Definitions

Mathematical morphology (MM) is not only a theory but also a powerful practical image analysis technique. "It is *mathematical* in the sense that the analysis is based on the set theory, topology, lattice algebra, random functions, etc. and it is called *'Morphology'* since it aims at analysing the shape and form of the objects" (Serra [80]). Appropriate MM operations, as nonlinear transformations, tend to simplify image data, preserving their essential shape characteristics, and eliminate irrelevancies, by modifying geometrical features of an image locally [33]. The language of MM is set theory. Sets in MM represent the shapes manifested on binary or grey tone images [33]. The main notion in MM is the interaction between the image under analysis and a structuring element (SE), where images and SEs are considered as sets of points and the operations come from set theory [88]. MM can estimate and measure many useful geometric features in an image such as shape, size, connectivity and so on, based on set-theoretical methodology, employing specific sequences of neighbourhood transformations. Like any theory, MM has a perspective which allows it to focus on certain phenomena within images. It views images as set theory and geometry, thus, distinguishing itself from other image processing theories, e.g., syntactic theories based upon generative grammars and signal processing theories based on Fourier analysis. MM provides an algebraic formulation to apply neighbourhood operations on images. SE, as a probe, slides through the image as a moving window, inspects its interaction with the image, and detects specific features in the neighbourhood of every point in the image. SE behaves as an elementary building block. Therefore a-priori knowledge of the information content of the image can help to choose a proper

SE [88]. As the identification of objects, object features, and assembly defects correlate directly with shape, MM is considered as a priceless tool for machine vision and recognition processes for robots [33]. In practice we consider morphological algorithms for extracting boundaries, connected components, the convex hull, and the skeleton of a region. Several methods, however, have been developed for region filling, thinning, thickening, and pruning that are often used in conjunction with these algorithms as pre- or post-processing steps. Also we have applied MM in ECG waves detection and fingerprint processing.

1.3 MF versus non-MF

Serra in [79] says: "When one scrutinises the behaviour of the scientists who access morphological filtering for the first time, a number of their reactions, more or less implicitly, refer to linear processing. They tend to extrapolate, if not the results, at least the style and the a-priori of the linear approach. Progressively then, they usually set themselves free from these crutches." Boomgaard [88] discusses the differences between two techniques as: " In linear image processing the basic underlying assumption is the superposition of visual stimuli. The visual signal is thought of as the weighted addition of basic signals. This assumption of linearity is questionable in case the image is formed by projection, where one object completely hides another object behind it. Due to using non-linear transformations, it is impossible to reconstruct the original image."

Some major differences between the two approaches are listed in Table 1.2 based on [79].

Table 1.2: Morphological filtering versus convolution.

Features	MM	Convolution
Reversibility	No	Yes
Unique algebraic structure	No	Yes
Idempotence and loss of Information	Yes	No
Defined in complete lattice	Yes	No
Iteration	Yes	No
Flatness ($\psi(\log(f)) = \log(\psi(f))$)	Yes	No

1.4 Citations

MM operations have been applied successfully to a broad variety of image processing/analysis tasks (including noise suppression, image enhancement, coding, feature extraction, thinning, texture analysis, and shape recognition) encountered in diverse areas such as biomedical image processing, cellular automata, electron microscopy, astronomy, and automated industrial visual inspection [52]. We have also applied MF as a pre-processor in fingerprint processing and as an efficient technique in ECG wave analysis.

A lot of scientists and research students are now working on the theory and application area of mathematical morphology. Appendix A addresses some of important English citations, categorised separately in tables.

1.5 Overview of the thesis

In the subsequent chapters we will undertake the following tasks:

- *Chapter 2* introduces a theoretical background on mathematical morphology. It also explains the existing confusion in literature about the definition of the operators. Binary and grey-scale operators are defined.

Properties of morphological operators are also discussed. Our critique about extensivity of closing and anti-extensivity of opening is also included.

- *Chapter 3* presents fast algorithms in 1-D and 2-D for morphological erosion, dilation, opening, closing, open-closing and close-opening. It covers our contributions to the development of fast algorithms for open-closing and close-opening. The algorithms applied in this chapter are direct implementation of the morphological operators, by single one-pass procedures. They are computationally simple and very efficient with a 18-20 % reduction in computational effort for morphological operations compared with the fastest alternative method, and an order of magnitude improvements over naive implementations for large structuring elements. All the algorithms mentioned above, have been realised by a user-friendly package designed by author for 1-D and 2-D morphological systems design.

- *Chapter 4* improves and generalises real-time implementation of 1-D and 2-D GS morphological operators, letting SE have its origin at any point of its domain. It shows the superb efficiency of the proposed methods. Schematic diagrams of hardware implementations are presented. Our hardware architecture for real-time MF computation avoids unnecessary operations and doesn't keep intermediate results. Therefore it is much faster than the classical cascade methods for opening, closing, open-closing and close-opening operations in which the intermediate results are needed. It leads to demanding less memory, comparisons, additions/subtractions and less computational time.

- *Chapter 5* proposes a technique to combine MF with convolution. It emphasizes the resultant operators' efficiency both in time and frequency

domain. It suggests the corresponding hardware architecture. The CMF performs based on morphological operations and their convolution with selected transfer functions and offers highly accurate characteristics of high-, low- and band-pass filtering. Designing a suitable structuring element based on the characteristics of the input signal is discussed and hardware implementation of CMF is also investigated. We have discussed the design of CMF and shown its super performance in time domain and accurate characteristics in frequency domain when applied for signal/image filtering. Its outstanding performance can be seen in both time and frequency domain.

- *Chapter 6* introduces novel operators called weighted morphological operators (WMOs). Using WMOs, weighted morphological filters (WMFs) are designed which illustrate their superb performance compared to classical MFs. The newly introduced operators employ a weighted structuring element and apply multiplication or division in place of addition and subtraction in classical morphological operations. Experimental results prove that the new operators' performance dominate over classical ones for signals/images buried in salt&pepper, speckle and Gaussian noises.

- *Chapter 7* shows how MF can be applied for fingerprint classification as a pre-processor. It also introduces a syntactic approach to fingerprint classification including the details of pattern classification and string analysis. The application of MF speeds up pre-processing with a reliable output compared to the existing methods. The proposed algorithm for fingerprint classification extends and improves the existing approaches in literature. The algorithm, presented in this chapter, shows an improvement of fingerprint processing and classification, compared to the existing methods, owing to a powerful morphological pre-processor and robust technique for smoothing and an efficient grammar. It can be im-

plemented to process and match fingerprints with reliability.

- *Chapter 8* is about ECG waves detection using MF in different stages. Following an introduction and definition of the terms in ECG, a pre-processor based on MF is introduced. It emphasises the efficiency of the applied pre-processor in removing noise and baseline drift without losing necessary details of the signal. The existing QRS detectors are reviewed and our contribution is included. The proposed method for $ST-T$ detection is then followed. A residual signal is produced, resulting from interpolation of QRS and subtracting the median(s) of $ST-T$ segments from the corresponding parts. Later step is about analysing the residue and detecting P waves, even when buried on T waves.

- *Chapter 9* concludes the thesis and discusses about the limitations of morphological approach and future work.

- *Appendix A* addresses some of important English citations categorised separately in tables.

- *Appendix B* presents the existing relations and proof of some properties.

- *Appendix C* introduces some of arrhythmia types adopted from literature.

1.6 Major contributions

1. **Fast algorithms for 1-D and 2-D GS morphological operators ([69, 70]).**

2. **Real-time implementation of GS morphological filtering ([72, 75, 73]).**

3. **Convolved morphological filters ([74, 77]).**

4. **Linearly-weighted morphological operators ([76]).**

5. **Morphological filters in fingerprint processing ([71]).**

6. **ECG waves detection using morphological filtering.**

Chapter 2

An introduction to MM

2.1 Introduction

The previous chapter established a systematic definition of MM. This chapter focuses on the exact definition of the operators which will be used in later chapters. It explains the existing confusion in literature about the definition of the Minkowski addition/subtraction and binary dilation/erosion, and introduces a specific and unique definition for the above-mentioned operators. Complete lattice is briefly discussed highlighting its relation to the definition of the grey-scale morphological filter (GSMF). Different approaches for GSMF are explained. Binary operators are discussed as the original MF operators. Our critique about the definition of opening and closing is also highlighted.

Morphological operators are classified into three groups [53]: (1) set processing (SP) systems: in which both the input and output are binary, (2) function processing (FP) systems: in which both the input and output are grey-scale, and (3) function-and-set processing (FSP) systems: which are subclasses of FP systems and can produce binary outputs whenever the input is also binary. These systems will be defined in the following sections.

2.2 Literature confusion in definition of binary operators

There exists a confusion in the literature about the definitions of binary dilation and erosion and also Minkowski addition and subtraction. Table 2.1 compares the important existing definitions. The rest of the definitions by other authors are mainly similar to those of Table 2.1. We use our own notations that are rather similar to the next definitions for grey-scale ones, whereas we use a reflected SE (structuring element) for dilation, and a non-reflected SE for erosion. Therefore the duality will exist between Minkowski addition and subtraction in one hand and dilation and erosion in another. When SE is symmetric regarding to its origin, the definitions of the Minkowski addition and subtraction will be similar to those of dilation and erosion. In Table 2.1, A and B denote the input image and the structuring element, respectively, and $\check{B}$ denotes the reflection of B around its origin which will later be defined in this chapter. Figure 2.1 shows the result of the Minkowski addition and subtraction, dilation and erosion of the test image A (Fig. 2.1-a) with different structuring elements whose origins are in different positions. The sign + denotes the origin of SE in parts g, h, m, n, s and t of Figure 2.1. Foreground and background are denoted "□" and "." respectively. Figure 2.1-b shows the complement of the image A. The second row of Figure 2.1 (i.e. c, d, e, f) is the shifted case of the third row (i.e. i, j, k, l) as the SEs applied (i.e. B_1 and B_2 shown in part g and m) differ only on the position of their origins. The complete details in section 2.5 illustrate the effect of the shift in the operations. B_3 is symmetric (B_3 (Figure 2.1-s) = $\check{B}_3$ (Figure 2.1-t)), therefore the result of the Minkowski addition ($A \overset{M}{\oplus} B_3$ as shown in Figure 2.1-o) is the same as of the dilation ($A \oplus B_3$ as shown in Figure 2.1-p), and the result of the Minkowski subtraction ($A \overset{M}{\ominus} B_3$ as shown in Figure 2.1-q) is the same as

of the erosion ($A \ominus B_3$ as shown in Figure 2.1-r).

Table 2.1: Dilation, erosion, Minkowski addition and subtraction.

Source	Minkowski addition	Minkowski subtraction	Binary dilation	Binary erosion
Hadwiger [30]	$\{a+b : a \in A, b \in B\}$	$\{x : x+b \in A, b \in B\}$		
Matheron [54]	$A \oplus B =$ $\{a+b : a \in A, b \in B\}$	$A \ominus B = (A^c \oplus B)^c =$ $\{x : \check{B}+x \subseteq A\}$	$A \oplus \check{B} = \{x : A \cap B_x \neq \emptyset\}$ $= \{a-b : a \in A, b \in B\}$	$A \ominus \check{B} = \{x : B_x \subseteq A\}$ $= \{x : x+b \in A, b \in B\}$
Sternberg [85]	$A \oplus B =$ $\{a+b : a \in A, b \in B\}$	$A \ominus B = \{x : (B)_x \subseteq A\}$ $= \{x : x+b \in A, b \in B\}$	$A \oplus B =$ $\{a+b : a \in A, b \in B\}$	$A \ominus B = \{x : (B)_x \subseteq A\}$ $= \{x : x+b \in A, b \in B\}$
Haralick [33]			$A \oplus B = \bigcup_{b \in B} (A)_b =$ $\{a+b : a \in A, b \in B\}$	$A \ominus B = \bigcap_{b \in B} (A)_{-b} =$ $\{x : B_x \subseteq A\}$
Our notations:	$A \overset{M}{\oplus} B = \bigcup_{b \in B} (A)_b =$ $\{a+b : a \in A, b \in B\}$	$A \overset{M}{\ominus} B = (A^c \overset{M}{\oplus} B)^c =$ $\{x : (\check{B})_x \subseteq A$	$A \oplus B = \bigcup_{b \in \check{B}} (A)_b =$ $\{a+b : a \in A, b \in \check{B}\}$	$A \ominus B = (A^c \oplus B)^c =$ $\{x : (B)_x \subseteq A\}$

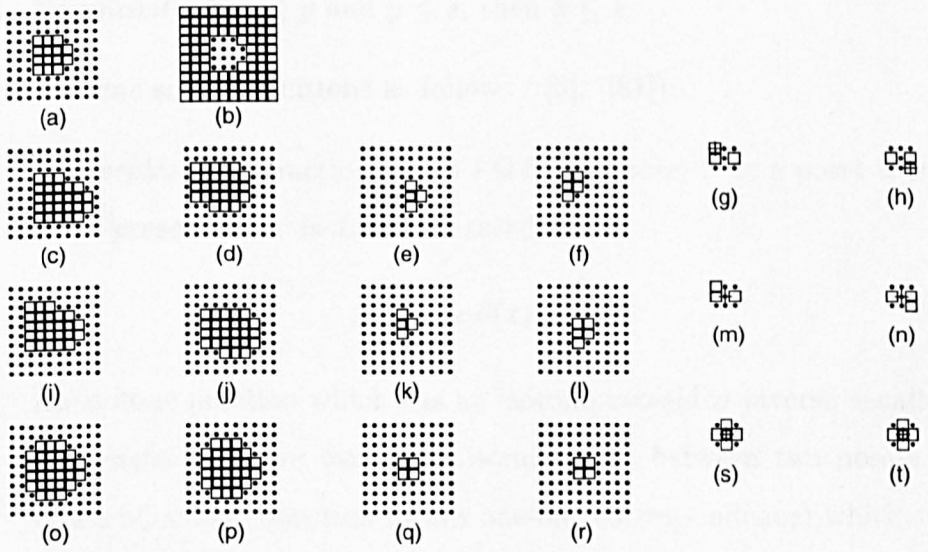


Figure 2.1: Binary operators with different SEs: (a):A, (b):A^c, (c):$A \overset{M}{\oplus} B_1$, (d):$A \oplus B_1$, (e):$A \overset{M}{\ominus} B_1$, (f):$A \ominus B_1$, (g):$B_1$, (h):$\check{B}_1$, (i):$A \overset{M}{\oplus} B_2$, (j):$A \oplus B_2$, (k):$A \overset{M}{\ominus} B_2$, (l):$A \ominus B_2$, (m):$B_2$, (n):$\check{B}_2$, (o):$A \overset{M}{\oplus} B_3$, (p):$A \oplus B_3$, (q):$A \overset{M}{\ominus} B_3$, (r):$A \ominus B_3$, (s):$B_3$, (t):$\check{B}_3$.

2.3 Notion of a complete lattice

MM is the application of the lattice theory to the spatial structures. The properties of a single binary relation "$\leq$" concerns the pure lattice theory [8]. This relation is assumed to have certain properties, the most basic of which lead to the following concept of a "*partly ordered set*" or "*poset*". A poset is a set in which a binary relation $x \leq y$ is defined, which satisfies for all x, y, z the following conditions: ([81])

1. *Reflexive:* For all x, $x \leq x$.
2. *Antisymmetry:* If $x \leq y$ and $y \leq x$, then $x = y$.
3. *Transitivity:* If $x \leq y$ and $y \leq z$, then $x \leq z$.

We will define some definitions as follows ([8], [81]):

1. *Isomorphism:* A function $\theta : \mathcal{P} \rightarrow \mathcal{Q}$ from a poset $\mathcal{P}$ to a poset $\mathcal{Q}$ is called *order-preserving* or *isotone* if it satisfies

$$x \leq y \Rightarrow \theta(x) \leq \theta(y). \tag{2.3.1}$$

   An isotone function which has an isotone two-sided inverse is called *isomorphism.* In other words, an isomorphism between two posets $\mathcal{P}$ and $\mathcal{Q}$ is a bijection (bijection means one-one correspondence) which satisfies Eq. 2.3.1 and also

$$\theta(x) \leq \theta(y) \Rightarrow x \leq y. \tag{2.3.2}$$

   An isomorphism from a poset $\mathcal{P}$ to itself is called an *automorphism.*

2. *Duality:* The converse of any partial ordering is itself a partial ordering.
3. *Antitone:* A function $\theta : \mathcal{P} \rightarrow \mathcal{Q}$ is antitone if and only if

$$x \leq y \Rightarrow \theta(x) \geq \theta(y), \tag{2.3.3}$$

$$\theta(x) \leq \theta(y) \Rightarrow x \geq y. \tag{2.3.4}$$

A bijection θ which satisfies Eqs. 2.3.3- 2.3.4 is called a dual isomorphism.

4. *Lattice:* A lattice $\mathcal{T}$ is a poset $\mathcal{P}$ if any two elements of it (x and y) have a greatest lower bound (g.l.b. or inf) denoted by $x \wedge y$, and a least upper bound (l.u.b. or sup) denoted $x \vee y$.

5. *Complete lattice:* A lattice $\mathcal{T}$ is complete when each of its subsets $\mathcal{H}$ has a l.u.b. and a g.l.b. in $\mathcal{T}$. In other words, a complete lattice (either on the Euclidean space $\mathbb{R}^n$ or on digital space $\mathbb{Z}^n$) is a partially ordered set $(\mathcal{T}, \leq)$ in which every subset $\mathcal{H} \subseteq \mathcal{T}$ has a supremum and infimum denoted $\bigvee \mathcal{H}$ and $\bigwedge \mathcal{H}$ respectively [4]. Any finite lattice or lattice of finite length is complete. Not every lattice is complete: thus the rational numbers are not complete, and the real numbers (in their natural order) are not complete unless $\pm\infty$ are adjoined as universal bounds.

In a lattice, any logical consequence of a choice of ordering remains true when we commute the symbols $\vee$ and $\wedge$, and $\leq$ and $\geq$. This is called the principle of the duality with respect to the order ([81]). For complete lattice we can write ([46]):

1. Commutativity:

$$X \vee Y = Y \vee X \quad , \quad X \wedge Y = Y \wedge X. \tag{2.3.5}$$

2. Associativity:

$$(X \vee Y) \vee Z = X \vee (Y \vee Z), (X \wedge Y) \wedge Z = X \wedge (Y \wedge Z). \tag{2.3.6}$$

3. The law of absorption:

$$X \wedge (X \vee Y) = X \quad , \quad X \vee (X \wedge Y) = X. \tag{2.3.7}$$

The following section explains function lattice briefly. Other kinds of lattices may be found in literature as [8].

2.4 Function lattice

Let E be an arbitrary space. The class $\mathcal{F}$ of the extended (including $\pm\infty$) real-valued functions $u : E \rightarrow \mathbb{R}$ is obviously ordered by the relation $u \leq v$, if, for each $x \in E$, $u(x) \leq v(x)$ and constitutes a complete lattice.

2.4.1 Functions and umbrae

An extension of morphological operators to functions is due to Sternberg [85], who uses the representation of a n-D function $f(x)$ by a n+1-D set, its umbra. The umbra of a function f, denoted $\mathfrak{U}(f)$, is the set of points below the surface represented by $f(x)$:

$$\mathfrak{U}(f) = \{(x, y) : y \leq f(x)\}. \tag{2.4.1}$$

Figure 2.2 shows, as an example, the umbra of a sinus function. After getting the umbra, the binary morphological operators can be applied. The function f can be reconstructed from its umbra as ([53]):

$$f(x) = \max\{y : (x, y) \in \mathfrak{U}(f)\}. \tag{2.4.2}$$

We can easily show that $f \leq g \Leftrightarrow \mathfrak{U}(f) \subseteq \mathfrak{U}(g)$. Some definitions for grey-value operations based on sets are defined as follows ([88]):

1. *Grey-value union:* The union of two functions f and g denoted as $f \vee g$ is defined as:

$$(f \vee g)(x) = f(x) \vee g(x). \tag{2.4.3}$$

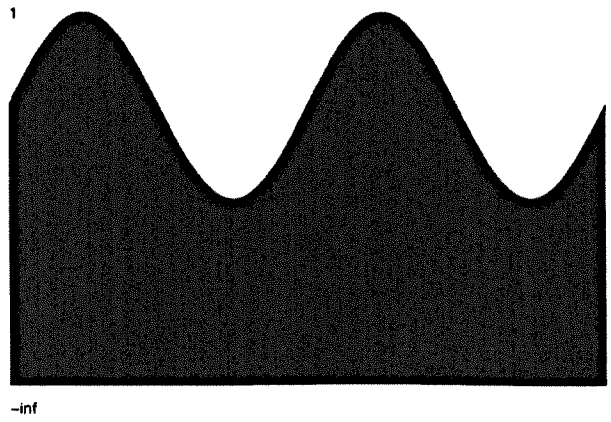


Figure 2.2: Umbrae ($\mathfrak{U}$) of a sinus.

There will be a one-to-one correspondence between the union of functions and the set union:

$$\mathfrak{U}(f \vee g) = \mathfrak{U}(f) \cup \mathfrak{U}(g). \tag{2.4.4}$$

2. *Grey-value Intersection:* The intersection of two functions f and g denoted as $f \wedge g$ is defined as:

$$(f \wedge g)(x) = f(x) \wedge g(x). \tag{2.4.5}$$

The same one-to-one correspondence exists for the function and the set intersection:

$$\mathfrak{U}(f \wedge g) = \mathfrak{U}(f) \cap \mathfrak{U}(g). \tag{2.4.6}$$

3. *Grey-value transpose:* The transpose $\check{f}$ of a function f is defined as:

$$\check{f}(x) = f(-x). \tag{2.4.7}$$

4. *Grey-value complement:* The complement f^c of a function f is defined as:

$$f^c(x) = -f(x). \tag{2.4.8}$$

We notice that $f \vee f^c = |f|$ and $f \wedge f^c = -|f|$. However for set domain we have $A \cup A^c = \mathbb{E}$ and $A \cap A^c = \emptyset$.

Figure 2.3 shows the above properties.

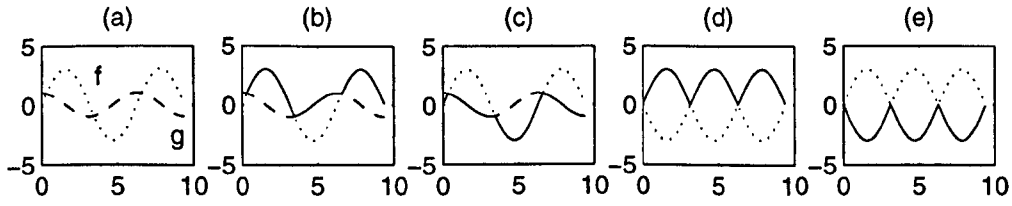


Figure 2.3: Grey-value operations: (a): two functions f and g plotted by ".." and "- -" respectively, (b): $f \vee g$ (solid), (c): $f \wedge g$ (solid), (d): $f \vee f^c$ (solid), (e): $f \wedge f^c$ (solid).

2.4.2 Functions and stacks of sets

Another extension of morphological operators to functions is presenting them as stacks of sets. Instead of associating with function $f : E \rightarrow \mathbb{R}$ its set-oriented umbra, we can alternatively consider the stack of its horizontal sections $T_z(f)$ ([78]):

$$T_z(f) = \{x, f(x) \geq z\}. \tag{2.4.9}$$

As threshold level z increases, $T_z(f)$ decreases continuously, i.e.

$$T_z(f) = \bigcap_{z'<z} T_{z'}(f). \tag{2.4.10}$$

Any set operations can be done over $T_Z(f)$. Conversely, it is easy to see that every continuously decreasing family $\{T_z\} \mid z \in \mathbb{Z}$ of sets generates a unique function f, by the algorithm

$$f(x) = \sup\{z, x \in T_z(f)\}, \tag{2.4.11}$$

where sup is equal to max for grey-scale functions. Table 2.2 shows an example about the above relations. The symbol $\square$ indicates 1 for set operations. The reconstructed function is $\hat{f}(x)$.

Table 2.2: Threshold representation of a discrete function.

$f(x)$	3	1	1	0	4	2	3	4	3	2	1	0
$T_0(f)$	□	□	□	□	□	□	□	□	□	□	□	□
$T_1(f)$	□	□	□		□	□	□	□	□	□	□	
$T_2(f)$	□				□	□	□	□	□	□		
$T_3(f)$	□				□		□	□	□			
$T_4(f)$					□			□				
$\hat{f}(x)$	3	1	1	0	4	2	3	4	3	2	1	0

2.5 Definitions for binary operations

The structure of a Boolean algebra provides the general framework on which we shall perform binary morphological operations. The sets in binary images are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector with (x, y) as co-ordinates and 0 or 1 as value of each pixel. Let A and B be sets in Z^2, with components $a = (a_1, a_2)$ and $b = (b_1, b_2)$ respectively. We will assume that A and B represent the binary input image and the binary structuring element respectively (the size of B is considerably smaller than that of A) . The following definitions can be developed:

1. *Translation:* The translation of A by $x = (x_1, x_2)$, denoted $(A)_x$ is defined as $(A)_x = \{c : c = a + x, a \in A\}$.

2. *Reflection:* The reflection of B around its origin, denoted $\check{B}$, is defined as $\check{B} = \{x : x = -b, b \in B\}$. Based on the above two equations, we can easily extract the following equation:

$$(B)_x \subset A \Rightarrow B \subset (A)_{-x}. \tag{2.5.1}$$

3. *Complement:* The complement of the set A, denoted A^c, is

$$A^c = \{x : x \notin A\}. \tag{2.5.2}$$

4. *Set difference:* The set difference of two sets A and B, denoted as A/B, is defined as

$$A/B = \{x : x \in A, x \notin B\} = A \cap B^c. \tag{2.5.3}$$

5. *Incidence:* Two subsets $X, Y \subset \mathbb{R}^n$ are incident if $X \cap Y \neq 0$.

2.5.1 Minkowski set addition and subtraction

[54]: The Minkowski addition denoted $\overset{M}{\oplus}$ is defined as:

$$A \overset{M}{\oplus} B = \{a + b : a \in A, b \in B\} = \bigcup_{b \in B} (A)_b. \tag{2.5.4}$$

It is assumed that $A \overset{M}{\oplus} \{o\} = A$ and $A \overset{M}{\oplus} \emptyset = \emptyset$. The dual operation is called Minkowski subtraction, denoted $\overset{M}{\ominus}$, and defined as

$$A \overset{M}{\ominus} B = (A^c \overset{M}{\oplus} B)^c = \{x : (\check{B})_x \subseteq A\}. \tag{2.5.5}$$

The following relation will be true for Minkowski subtraction:

$$A \overset{M}{\ominus} B = \bigcap_{b \in B} (A)_b. \tag{2.5.6}$$

2.5.2 Binary dilation and erosion

Dilation by disk SE corresponds to an isotropic swelling or expansion algorithms [33]. Our definition for dilation and erosion is similar to Minkowski addition and subtraction except that we use the reflected SE ($\check{B}$). We define dilation as :

$$A \oplus B = \{a + b : a \in A, b \in \check{B}\} = \bigcup_{b \in \check{B}} (A)_b. \tag{2.5.7}$$

Erosion is the morphological dual to dilation. The structuring element (B) slides as a probe across the image (A), testing the spatial nature of the image

at every point [33]. Where B_x (B translated to x) can be contained in A (by placing the origin of B at x), then x belongs to the erosion $A \ominus B$. We define erosion as

$$A \ominus B = (A^c \oplus B)^c = \{x : (B)_x \subseteq A\}. \tag{2.5.8}$$

Similar to Minkowski subtraction we can say

$$A \ominus B = \bigcap_{b \in \check{B}} (A)_b. \tag{2.5.9}$$

Erosion is conceived of as a shrinking of the original image.

2.5.3 Binary opening and closing

Any set transform $\mathcal{O}$ in algebra satisfying the following three conditions is called opening:

1. *Anti extensivity:* $\mathcal{O}(A) \subset A$.
2. *Increasing:* $A_1 \subset A_2 \Rightarrow \mathcal{O}(A_1) \subset \mathcal{O}(A_2)$.
3. *Idempotency:* $\mathcal{O}(\mathcal{O}(A)) = \mathcal{O}(A)$.

Similarly a set $\mathcal{C}$ is called algebraic closing if it satisfies the following conditions:

1. *Extensivity:* $A \subset \mathcal{C}(A)$.
2. *Increasing:* $A_1 \subset A_2 \Rightarrow \mathcal{C}(A_1) \subset \mathcal{C}(A_2)$.
3. *Idempotency:* $\mathcal{C}(\mathcal{C}(A)) = \mathcal{C}(A)$.

Matheron [54] defines algebraic opening as :

$$\mathcal{O}(A) = \bigcup_{B_i \in B} A \circ B_i. \tag{2.5.10}$$

and for algebraic closing he has:

$$\mathcal{C}(A) = \bigcap_{B_i \in B} A \bullet B_i. \tag{2.5.11}$$

where B_i is structuring element belonging to the structuring element B.

The morphological opening and closing can also be defined in terms of erosion and dilation. Opening and closing are defined, respectively, as:

$$A \circ B = (A \ominus B) \oplus B. \tag{2.5.12}$$

$$A \bullet B = (A \oplus B) \ominus B. \tag{2.5.13}$$

Whenever we use opening and closing in this thesis, it is assumed that they are morphological (binary or grey-level) unless specified. Closing tends to smooth sections contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

As in the case of dilation and erosion, opening and closing are dual with respect to set complementation and reflection. That is

$$A \circ B = (A^c \bullet \check{B})^c. \tag{2.5.14}$$

The important features of opening and closing are that in either case the result is an elimination of specific image detail, smaller than the structuring element, without the global geometric distortion of unsuppressed features. Haralick et. al. [33] say: "For example, opening an image with a disk SE smoothes the contour, breaks narrow isthmuses, and eliminates small islands and sharp peaks or capes. Closing an image with a disk SE smoothes the contour, fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps on the contours".

2.5.4 Is (opening) closing always (anti-)extensive?

In this subsection we want to have a critique about a definition and prove that opening and closing are in some circumstances anti-extensive and extensive, respectively. Kresch [46] says "Opening is always anti-extensive and

closing is extensive, regardless to whether the origin is or is not contained in the structuring element". However Figure 2.4 shows that it is not correct. Binary opening/closing is anti-extensive/extensive only when the structuring element is symmetric. Table 2.3 shows the extensive and anti-extensive re-

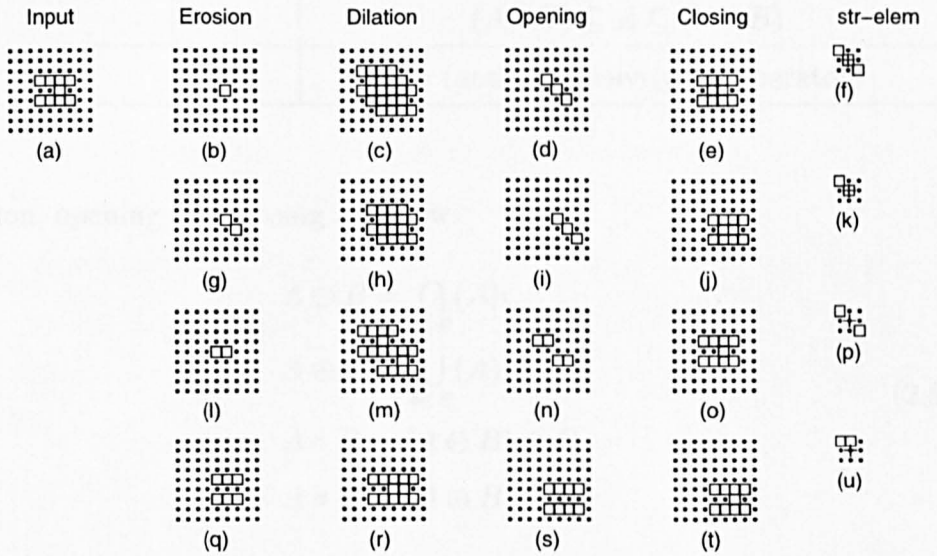


Figure 2.4: Extensivity and anti-extensivity.

lations based on the symmetry and the origin of the structuring element. If only the structuring element contains the origin, erosion is anti-extensive (Figs. 2.4-b and g) and dilation is extensive (Figs. 2.4-c and h) and the symmetry of the SE is not important. If only the SE is symmetric, opening is anti-extensive (Figs. 2.4-d and n), and closing is extensive (Figs. 2.4-e and o), whether (or not) the SE contains the origin. A morphological filter should be increasing, idempotent, and extensive/or anti-extensive. Table 2.4 shows that opening and closing can be used as filters. However dilation and erosion can not be considered as filters alone because they are not idempotent. It is assumed that SE is symmetric and contains its origin. However if we define binary erosion,

Table 2.3: Extensivity and anti-extensivity in binary MM.

Origin in SE	$\check{B} = B$	Relation
+	+	$(A \ominus B) \subseteq (A \circ B) \subseteq A \subseteq (A \bullet B) \subseteq (A \oplus B)$
+	-	$(A \ominus B) \subseteq A \subseteq (A \oplus B)$
-	+	$(A \circ B) \subseteq A \subseteq (A \bullet B)$
-	-	No (anti-)extensivity for operators

dilation, opening and closing as below:

$$\begin{aligned} A \ominus B &= \bigcap_{b \in B} (A)_b, \\ A \oplus B &= \bigcup_{b \in B} (A)_b, \\ A \circ B &= (A \ominus B) \oplus B, \\ A \bullet B &= (A \oplus B) \ominus B, \end{aligned} \tag{2.5.15}$$

then we will have opening as anti-extensive and closing as extensive. In this regard, we will lose duality for defining dilation based on erosion, or closing based on opening, and vice versa. Therefore we prefer to follow our previous definitions in Table 2.1.

Table 2.4: Properties of morphological operators based on the definitions of Table 2.3.

Property	Erosion	Dilation	Opening	Closing
Idempotent	-	-	+	+
Increasing	+	+	+	+
Extensive	-	+	-	+
Anti-extensive	+	-	+	-

2.6 Grey-scale morphological filtering

Section 2.3 introduced two techniques available for GS morphological operators as umbra and decomposition approaches. In this section we will develop direct relations for GS operators. Grey-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the co-ordinates of a pixel, and the third corresponds to its discrete intensity value. Sets in higher dimensional space can contain other image attributes, such as color and time varying components. In this section the operators are assumed to be grey-scale unless specified, and to generalise the idea, we will assume grey-value is not restricted to the range $0-255$, but any integer number unless specified. Let $f(n)$ and $g(n)$ denote respectively a 1-D GS signal and a 1-D GSE of length L, D_f and D_g denote their domains. 1-D Grey-scale erosion, dilation, opening and closing, denoted respectively by $(f \ominus g)(n)$, $(f \oplus g)(n)$, $((f \ominus g) \oplus g)(n)$ and $((f \oplus g) \ominus g)(n)$, are defined as follows.

$$(f \ominus g)(n) = \min_v\{f(n+v) - g(v)\}. \tag{2.6.1}$$

$$(f \oplus g)(n) = \max_v\{f(n-v) + g(v)\}. \tag{2.6.2}$$

$$((f \ominus g) \oplus g)(n) = \max_v\{\min_u\{f(n+u-v) - g(u) + g(v)\}\}. \tag{2.6.3}$$

$$((f \oplus g) \ominus g)(n) = \min_v\{\max_u\{f(n-u+v) + g(u) - g(v)\}\}. \tag{2.6.4}$$

where $u, v \in D_g$ and $f(\alpha), \alpha \in D_f$. If the structuring element is flat (i.e. $g(k) = 0, \quad \forall k \in D_g$) , then the above equations will be simplified to:

$$(f \ominus g)(n) = \min_v\{f(n+v)\}. \tag{2.6.5}$$

$$(f \oplus g)(n) = \max_v\{f(n-v)\}. \tag{2.6.6}$$

$$((f \ominus g) \oplus g)(n) = \max_{v}\{\min_{u}\{f(n+u-v)\}\}. \tag{2.6.7}$$

$$((f \oplus g) \ominus g)(n) = \min_{v}\{\max_{u}\{f(n-u+v)\}\}. \tag{2.6.8}$$

The above equations can easily be extended to 2-D. We let $F(m,n)$ and $G(m,n)$ denote respectively a 2-D GS image and a 2-D $(L \times K)$ GSE, D_F and D_G denote their domains. 2-D Grey-scale erosion, dilation, opening and closing, denoted respectively by $(F \ominus G)(m,n)$, $(F \oplus G)(m,n)$, $((F \ominus G) \oplus G)(m,n)$ and $((F \oplus G) \ominus G)(m,n)$, are defined as follows.

$$(F \ominus G)(m,n) = \min_{v,w}\{F(m+v,n+w) - G(v,w)\}. \tag{2.6.9}$$

$$(F \oplus G)(m,n) = \max_{v,w}\{F(m-v,n-w) + G(v,w)\}. \tag{2.6.10}$$

$$((F \ominus G) \oplus G)(m,n) = \max_{v,w}\{\min_{t,u}\{F(m+t-v,n+u-w) - G(t,u) + G(v,w)\}\}. \tag{2.6.11}$$

$$((F \oplus G) \ominus G)(m,n) = \min_{v,w}\{\max_{t,u}\{F(m-t+v,n-u+w) + G(t,u) - G(v,w)\}\}. \tag{2.6.12}$$

where $(t,u),(v,w) \in D_G$ and $F(\alpha,\beta),(\alpha,\beta) \in D_F$. If the structuring element is flat (i.e. $G(r,s) = 0, \quad \forall (r,s) \in D_G$) , then the above equations will be simplified to:

$$(F \ominus G)(m,n) = \min_{v,w}\{F(m+v,n+w)\}. \tag{2.6.13}$$

$$(F \oplus G)(m,n) = \max_{v,w}\{F(m-v,n-w)\}. \tag{2.6.14}$$

$$((F \ominus G) \oplus G)(m,n) = \max_{v,w}\{\min_{t,u}\{F(m+t-v,n+u-w)\}\}. \tag{2.6.15}$$

$$((F \oplus G) \ominus G)(m,n) = \min_{v,w}\{\max_{t,u}\{F(m-t+v,n-u+w)\}\}. \tag{2.6.16}$$

We can see that the above equations can easily be developed in n-D. As an example the following equation shows a 3-D grey-scale erosion of a 3-D function $\mathfrak{F}(l,m,n)$, with a 3-D flat SE $\mathfrak{G}(l,m,n)$:

$$(\mathfrak{F} \ominus \mathfrak{G})(l,m,n) = \min_{u,v,w}\{\mathfrak{F}(l+u,m+v,n+w)\}. \tag{2.6.17}$$

where $(u,v,w) \in D_{\mathfrak{G}}$ and $F(\alpha,\beta,\gamma),(\alpha,\beta,\gamma) \in D_{\mathfrak{F}}$. We will define the following GS relations similar to binary operations for translation, transpose and complement as:

1. **Grey-scale translation:** The translation of a function f by k is defined as:

$$f(n)_k = f(n+k). \tag{2.6.18}$$

2. **Grey-scale transpose:** The transpose $\check{f}$ of a function f is defined as

$$\check{f}(n) = f(-n). \tag{2.6.19}$$

3. **Grey-scale complement:** The complement f^c of a function f is:

$$f^c(n) = -f(n). \tag{2.6.20}$$

4. **Grey-scale (anti-)extensivity:** Grey-scale dilation (erosion) is extensive (anti-extensive) if the value of the origin of SE is non-negative:

$$g(0) \geq 0 \Rightarrow f \ominus g \leq f \leq f \oplus g. \tag{2.6.21}$$

GS opening is always anti-extensive $f \circ g \leq f$ and GS closing is always extensive $f \leq f \bullet g$, and

$$g(0) \geq 0 \Rightarrow f \ominus g \leq f \circ g \leq f \leq f \bullet g \leq f \oplus g. \tag{2.6.22}$$

2.7 Remarks and summary

In this chapter we have defined the basic operators engaged in MM, mainly as binary or grey-scale. We have realised that there are only two basic operators: erosion and dilation, and the rest of the operators can be obtained by proper combination of them. More relations and some proofs of morphological operators have been given in Chapter B. We have also adopted a unique definition for all operators, and have shown the conditions for opening and closing to be anti-extensive and extensive, respectively. Also we have shown that MF can be easily developed in higher dimensions too.

Chapter 3

Fast algorithms for GSMFs

3.1 Introduction

We have, so far, described MM and the important existing operators. This chapter covers our contributions to the development of fast algorithms for open-closing and close-opening. Fast algorithms for erosion, dilation, opening and closing are also presented to facilitate our contribution for fast open-closing and close-opening.

Binary morphological filtering (MF) has deeply been investigated by a lot of authors in every aspect. However grey-scale MF (GSMF) has been noticed a little, and most of the efforts were to convert GSMF to binary equivalence using the thresholding and umbra approaches discussed in chapter 2. In this chapter we will deal with GSMF directly proposing fast MF techniques [69, 70]. It is assumed that all SEs are flat, otherwise the classical algorithms of chapter 2 should be applied.

3.2 Fast GS erosion and dilation

Erosion and dilation are basic morphological operators, and the remaining operators can be made from proper combination of them. Therefore it is important to find their fast implementations. In this section, the algorithms for fast 1-D and 2-D erosion will be discussed.

3.2.1 1-D GS erosion

Let the input signal $f(n)$ and the flat SE $g(w)$ have N and W samples respectively (flat means $g(w) = 0, \quad w \in D_g$). The origin of g is assumed to be the first sample, otherwise the algorithms should slightly be modified. At first we consider a simple example to illustrate how fast the proposed method works. As an example let

$$\begin{aligned} f = &\ \{9,8,7,6,9,7,2,3,6,5,5,5,7,8,8,7,9,8\}, \\ g = &\ \{\underline{0},0,0,0,0\}. \end{aligned}$$

The underlined sample in g shows its origin. Based on Eq. 2.6.5, as the SE is flat, the first two samples of erosion are calculated as

$$\begin{aligned} er(1) = &\ \min\{f(1), f(2), f(3), \underline{f(4)}, f(5)\} = f(4) = 6, \\ er(2) = &\ \min\{\underbrace{f(2)}_{*}, \underbrace{f(3)}_{*}, \underline{f(4)}, \underbrace{f(5)}_{*}, f(6)\} = f(4) = 6. \end{aligned}$$

The underlined sample indicates the minimum within the search area. The symbol * under some samples means that they are not required for the current operation, as they have been controlled before. We realise that there are unnecessary comparisons for $er(2)$ when repeating to find the minimum among the samples $f(2), f(3), f(4), f(5)$ for $er(2)$. The minimum occurs at sample 4 as $f(4)$ for $er(1)$. Instead if we remember the location of $f(4)$ as the last minimum (*lastmin*), we may calculate $er(2)$ as

$$er(2) = \min\{lastmin, f(6)\},$$

because *lastmin* from the operations of $er(1)$ is in the search area of the operations for $er(2)$. $f(6)$ is the $W-1$.th sample regarding $er(2)$ ($W = 5, 2+W-1 = 6$). Again $f(4)$ is a minimum for $er(2)$, and its value and location will be remembered as *lastmin* and *lastminloc* for the next operation. For $er(3)$, *lastminloc* is still in search area, i.e.

$$er(3) = \min\{\underbrace{f(3)}_{*}, f(4), \underbrace{f(5)}_{*}, \underbrace{f(6)}_{*}, \underline{f(7)}\} = \min\{lastmin, \underline{f(7)}\} = f(7) = 2.$$

The new minimum location needs to be updated to 7 as $f(7) = 2$ is the minimum. Therefore at this stage $lastmin = 2$ and $lastminloc = 7$. From the above simplifications, we have reduced the amount of the required comparisons for every output sample from $W - 1 = 4$ to 1 comparison. For large-size input signals it considerably reduces the computation time. $er(4) : er(7)$ can be calculated the same way as *lastminloc* is inside the search area and they will all be equal to 2. However for $er(8)$, the location of the last minimum ($lastmin = 2 = f(7), lastminloc = 7$) is not in the search area and a complete search is required as

$$er(8) = \min\{\underline{f(8)}, f(9), f(10), f(11), f(12)\} = f(8) = 3.$$

Therefore we conclude that if *lastminloc* for an operation is leftward of the search area (e.g. $f(8) = lastmin$ is the leftward sample in operations for $er(8)$), we should have a complete search for the next operation.

$$er(9) = \min\{f(9), \underline{f(10)}, \underline{f(11)}, \underline{f(12)}, f(13)\} = f(12) = 5.$$

For $er(9)$ there are more than one minimum (i.e. $f(10) = f(11) = f(12) = 5$). We candidate the rightmost sample as *lastmin* (i.e. $lastmin = f(12)$) to have *lastmin* inside the rightmost part of the search area for the next operation. $er(10) : er(12) = 5$ as *lastminloc* is not leftward and *lastmin* remains minimum until $er(13)$.

$$er(13) = \min\{\underline{f(13)}, f(14), f(15), \underline{f(16)}, f(17)\} = f(16) = 7.$$

$f(13) = f(16) = 7$ and we choose the rightmost one $f(16)$ as *lastmin* for next operation.

$$er(14) = \min\{\underline{lastmin}, f(18)\} = lastmin = 7.$$

When approaching to the last samples we need to have extra input samples for comparisons as

$$er(15) = \{\underline{lastmin}, \underbrace{f(19)}_{?}\}.$$

Noticing that $N = 18$, there is no real $f(19)$. Therefore early/late samples in dilation/erosion are not defined. There are three choices for such samples:

1. the extra samples required are equal to 0.
2. the extra samples required are equal to the last real sample.
3. their values are transparent for operation (big enough not to be considered as minima for erosion and small enough for dilation).

We choose the third option for instance. In this case

$$\begin{aligned} er(15) &= er(16) = \min\{lastmin\} = lastmin = 7, \\ er(17) &= \min\{f(17), \underline{f(18)}\} = f(18) = 8, \\ er(18) &= f(18) = 8. \end{aligned}$$

Now we develop the fast algorithm:

1. For $er(1)$ make the search area from 1 to W and find the minimum (as *lastmin*) and remember its location (as *lastminloc*). If there is more than one minimum, remember the rightmost location. Assign $ptr = 1$ corresponding to the first sample of the output $er(1)$.
2. Increase the sample pointer (i.e. $ptr \leftarrow ptr + 1$). If $ptr > N$ then stop, otherwise do the following instructions: If the operation has reached to

the last samples (i.e. $N - W < ptr < N$), then assume extra null samples as necessary. If *lastminloc* is inside the current search area, then compare only *lastmin* with $f(ptr + W - 1)$, else make a complete search area. Update the new *lastmin* and *lastminloc*. Repeat step 2.

Assuming full operations for late samples, the given example needs $N(W - 1) = 18 \times 4 = 72$ comparisons for erosion with classical method while only 28 comparisons are required for fast proposed method. The speed-up ratio is therefore $72/28 = 2.57$.

3.2.2 2-D GS erosion

We can expect similar fast algorithm for 2-D erosion based on the fast algorithm for 1-D GS erosion. The difference is only in the input signal which is 2-D (an $M \times N$ image F(m,n)) and SE is also a 2-D $(2L + 1 \times 2L + 1)$ flat square $G(k, l)$. As an alternative, let SE be a flat square with its origin at its centre $(G(L, L))$. Therefore the width and height of G is assumed to be odd $(2L + 1)$. Let $y = 1 \rightarrow N$ and $x = 1 \rightarrow M$ be respectively column and row pointer sweeping all samples of the input image row-wise. The proposed algorithm is as follows: [70]

1. Start from top left pixel and find the minimum inside a search area of length $2L+1$ and width $2L+1$ with the current pixel at its origin. Assign the minimum to the relevant pixel, and increase the column pointer by one.

2. If the column location of the last minimum at the same row is inside the current search area, reduce the search area to the rightmost column of square. It means that if the current pixel position is (x, y), limit the search area to $2L + 1$ points from $F(x + L, y - L)$ up to $F(x + L, y + L)$, because the remaining area has been searched for finding last

minimum of the previous operation. Then compare those pixels with the last minimum and find new minimum. increase the column pointer by one and go to step 5.

3. If step 2 is not correct, consider whether the row and column location of the minimum from the previous row is inside the current search area. If so, reduce the search area to $2L + 1$ points (last row of the search area from $F(x - L, y + L)$ up to $F(x + L, y + L)$. Then compare those pixels with the minimum from corresponding previous row, and find new minimum. increase the column pointer by one and go to step 5.

4. If neither step 2 nor step 3 is correct, make a complete search area of $2L + 1$ by $2L + 1$ with the current pixel position at origin and find the minimum. increase the column pointer by one and go to step 5.

5. If last column, then increase the row counter by one and initialise the column counter to the first column. If the last row, then stop, otherwise go to step 2.

Figure 3.1 shows the Nassi-Shneidermann chart of 2-D GS erosion. The description of the chart follows as below:

Line 1: *Min_Loc_X* and *Min_Loc_Y* denote respectively two arrays required for tracking the column and row locations of the minima obtained when processing a row of data. For beginning of the operations they need to be initialised to a negative value for the first row's comparisons indicating that there is no minima beforehand to compare their locations. The length of the arrays are equal to the size of the columns of the image.

Lines 2 and 5: Initialise row counter (denoted with *Row_Cntr*) and column counter (denoted with *Col_Cntr*) for tracking the rows and columns of the input image to 0 to start from the first row and column.

Line 3: Row_Offset_Down and Col_Offset_Right, denoting the offset from the current pixel's row downward and the current pixel's column rightward respectively, when structuring element slides forward, are initialised to L (half size of the length of the structuring element noticing that the origin of the structuring element is at its origin).

Lines 4-41 and 6-39: shows the loop when the row counter (Row_Cntr) and column counter (Col_Cntr) scan respectively from the first up to the last row and from the first up to the last column.

Line 7: evaluates a condition. If the column location of the minimum from the previous column (denoted by $Min\_Loc\_X[\max(0, Col\_Cntr - 1)]$) is inside the search area (i.e. greater than $\max(Col\_Cntr - L, 1)$ then execute lines 8-12, otherwise run lines 13-17. Don't worry about the row. $\max(0, Col\_Cntr - 1)$ prevents column counter subtracted from 1 to be negative and $\max(Col\_Cntr - L, 1)$ prevents the execution of the "if-clause" for the first column, because there is no column before it.

Lines 8,9: Upper row offset limit (denoted with Row_Offset_Up) is assigned to $-L$ (all rows of the window for sliding the structuring element), and lefthand limit for the column offset (denoted with Col_Offset_Left) is L (last column of the window). These two assignments limit the search area to the rightmost column of the window.

Lines 10-12: The value and location of the minimum from the previous column is assigned as a possible candidate for the minimum of the current operations.

Line 13: gives permission for full column operations.

Line 14: If the row and column location of the minimum from the operations of the previous row and the same column is inside the search area, then run **line 15**, otherwise run **lines 16-17**. Notice that the location and value of every minimum found for the operations of a pixel are kept inside the relevant arrays, and before writing any new data inside them, their values represent the

previous information.

Lines 15,13: limits the search area to the last row of the window.

Lines 20,23: shows the initialisation of the row offset counter (denoted by Row_Offset_Cntr) and column offset counter (denoted by Col_Offset_Cntr) to start from the upper and leftmost limits defined in the previous lines.

Lines 21-36 and 24-33: show respectively a loop for the row offset and column offset counter to slide the structuring element from top to bottom and leftmost to rightmost limit of the window.

Lines 22 and 25: cares about the row and column pointer not to exceed the real borders.

Lines 27-29: finds the new minimum if the value of the processed pixel is less than or equal to the current minimum. The evaluation is done in **Line 25**.

Line 37: assigns the minimum found as the corresponding output pixel of the operation.

To find out how the proposed algorithm is efficient, we consider an example below. Table 3.1 shows an 8×8 image F, a flat 3×3 square SE denoted G, and the erosion of F by G denoted $F \ominus G$. Table 3.2 presents the details of the operation. It has been organised in 8 rows by 8 columns composed of a 3×3 squares. We label each square as $\mathrm{sqr}(i,j)$ required as the search area for the pixel (i,j) of the image, i standing for i-th. row and j for j-th. column. The location of each pixel is shown as $(i = 1 \ldots 8, j = 1 \ldots 8)$ above each square of Table 3.2. There are 4 different types of squares. We label them respectively type 1 up to type 4.

For **type 1**, the location of the last minimum obtained from the previous pixel and at the same row is inside the search area of the current pixel. Therefore we only need to make comparisons between the last minimum and the pixels in the last column of the square, i.e. $(i-1, j+1), (i, j+1), (i+1, j+1)$. One example for type 1 is $\mathrm{sqr}(2,4)$ at row 2, column 4 of Table 3.2. The

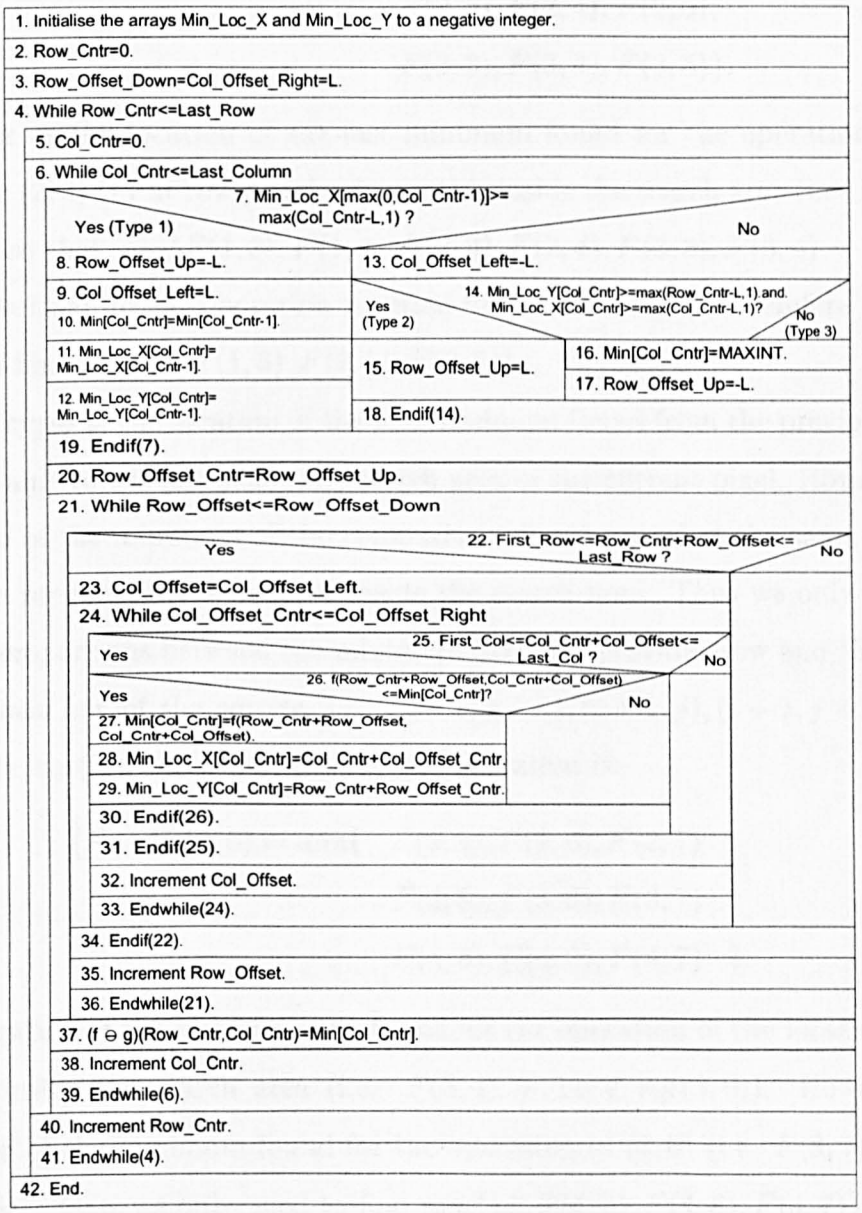


Figure 3.1: Nassi-Shneidermann chart for 2-D erosion.

complete operation is to find

$$(F \ominus G)(2,4) = \min\{ \begin{array}{l} F(1,3), F(1,4), F(1,5), \\ F(2,3), F(2,4), F(2,5), \\ F(3,3), F(3,4), F(3,5)\}. \end{array}$$

However as the location of the last minimum found for the operation of the location (2, 3) (11 at row 3 and column 4) is inside the search area (i.e. F(3,4)), we realise that $\min\{F(1,3), F(1,4), F(2,3), F(2,4), F(3,3), F(3,4) = 11\}$ has been searched for the operation required for location (2, 3). Therefore we only need to find $\min\{11, F(1,5), F(2,5), F(3,5)\}$.

For **type 2**, the location of the last minimum found from the previous pixel at the same row is not inside the search area of the current pixel. However the location of the minimum of the comparisons for the pixel at the same column and the previous row is instead inside the search area. Thus we only need to make comparisons between the minimum from the previous row and the pixels of the last row of the square, i.e. $(i+1, j-1), (i+1, j), (i+1, j+1)$. For example, sqr(3, 6) is of type 2. The full operation is:

$$(F \ominus G)(3,6) = \min\{ \begin{array}{ll} F(2,5), F(2,6), F(2,7) & , \\ F(3,5), F(3,6), F(3,7) & , \\ F(4,5), F(4,6), F(4,7) & \}. \end{array}$$

The location of the last minimum found for the operation of the location (3, 5) is not inside the search area (i.e. $F(3,4) = 11 \notin$ sqr(3, 6)). However the location of the minimum found for the operation of (2, 6) (i.e. $F(3,7) = 12 \in$ sqr(3, 6)). Thus we only need to find $\min\{12, F(4,5), F(4,6), F(4,7)\}$.

For **type 3** we need a full comparison, because none of the minima of type 1 and type 2 is not inside the search area. An example for type 3 is sqr(5, 5).

Type 4 is for some of the pixels of the last row or last column. They do not need any comparison. The minimum from the previous pixel either at the same row and the previous column, or the previous row and the same column is valid

for this pixel, and there are no new pixels to be compared. An example for type 4 is sqr(4, 8). Also we notice that the indexes for $i-1$ and $j-1$, required respectively for the comparisons of the first row and column are negative and we assign the value of the adjacent pixels. In other words they are considered as "don't care". Similar discussion is valid for the pixels of the last row and column. The values of the pixels not required for comparison are presented with the symbol x standing for "don't care". The symbol . means that the corresponding input signal does not exist. The underlined values inside each square shows the minimum found as the result of erosion for the corresponding output pixel. In classical method, 8 comparisons need to be done for every

Table 3.1: An example of 2-D GS Erosion.

F							
59	61	55	53	53	66	65	55
58	57	52	51	17	16	28	10
16	14	18	11	21	22	12	17
17	20	10	20	33	17	10	32
16	11	28	24	22	26	29	21
17	24	28	40	29	20	27	25
25	23	30	39	58	59	52	56
31	30	29	24	26	23	26	27

$\ominus$

G		
0	0	0
0	0	0
0	0	0

$=$

$F \ominus G$							
57	52	51	17	16	16	10	10
14	14	11	11	11	12	10	10
14	10	10	10	11	10	10	10
11	10	10	10	11	10	10	10
11	10	10	10	17	10	10	10
11	11	11	22	20	20	20	21
17	17	23	24	20	20	20	25
23	23	23	24	23	23	23	26

pixel of the image which will be $8 \times 8 \times 8 = 512$ for the whole image. However we only need 136 comparisons with our fast algorithm (3.77 times faster for an 8×8 image). For a large-size image (e.g. 512×512 which is considered as a normal size), it will be a great reduction in computation.

Table 3.3 shows the types of the required comparisons for every pixel of the given example (four types defined in the previous paragraph).

There is another fast algorithm for 2-D GS erosion which is less efficient than the proposed method. Regarding Eq. B.1.25 we can decompose the flat GSE based on Table 3.4 into two 1-D flat GSE. Therefore with applying two fast 1-D GS erosions, we find, at first, the minima row-wise (across each row) and then apply another erosion, column-wise, upon the result of the previous

Table 3.2: Details of the required operations for output pixels of Table 3.1.

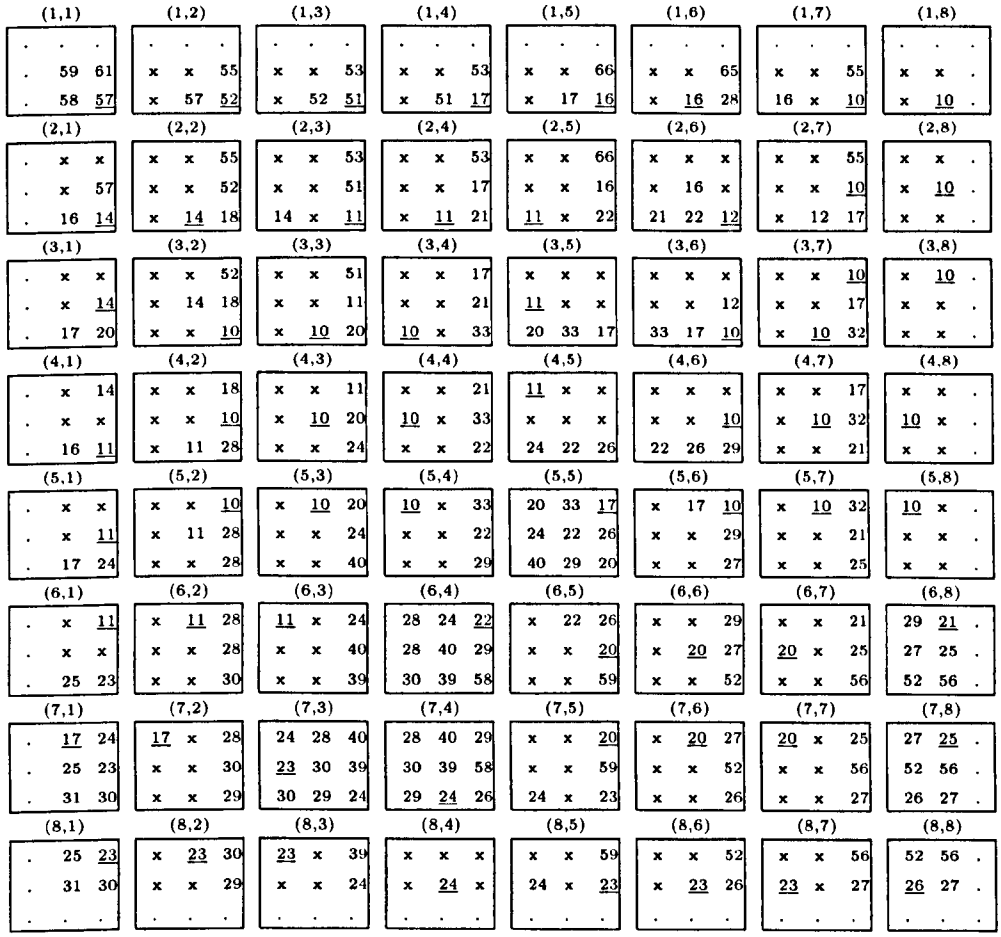


Table 3.3: Types of comparison required for the example of 2-D GS erosion.

	col. 1	col. 2	col. 3	col. 4	col. 5	col. 6	col. 7	col. 8
row 1	3	1	1	1	1	1	1	4
row 2	2	1	1	1	1	2	1	4
row 3	2	1	1	1	2	2	1	4
row 4	2	1	1	1	2	2	1	4
row 5	2	1	1	1	3	1	1	4
row 6	2	1	1	3	1	1	1	3
row 7	3	1	3	3	1	1	1	3
row 8	3	1	1	4	1	1	1	3

one.

Table 3.4: Decomposing a flat square GS

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & \underline{0} & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & \underline{0} & 0 \\ \hline \end{array} \oplus \begin{array}{|c|} \hline 0 \\ \hline \underline{0} \\ \hline 0 \\ \hline \end{array}$$

3.2.3 Fast 1-D and 2-D dilation

Based on Eqs. 2.6.6 and 2.6.14, we can apply similar fast algorithms for 1-D and 2-D GS dilation as we developed for erosion. The modifications will mainly concentrate on changing the terms minimum to maximum, and sweeping the signal/image from end to start due to the existing reflection of SE in dilation. The rest of the structure of the algorithm will remain the same.

3.3 Fast GS opening and closing

In this section the fast algorithms for 1-D and 2-D GS opening are presented.

3.3.1 Fast 1-D GS opening

Applying a combination of fast erosion followed by fast dilation, will definitely be faster than finding the classical opening. This procedure needs to be completed in two steps. Developing an algorithm to obtain opening in single-pass will definitely speed up the operation. The main idea is adopted from Wang et. al. [89]. We will use the same notations for the input signal and SE as before. We assume that the origin of g is its first sample. The algorithm applied is as follows:

1. For the first point $n = 1$, the smallest samples (*lastmin*) of the search area (i.e. samples 1:W) are found by a full comparison of all samples. If there are more than one smallest sample, remember the location of the rightmost one as *lastminloc*. Assign *lastmin* from *lastminloc* for the output samples with the locations smaller than, or equal to *lastminloc*.

2. After the above step, if *lastminloc* is the leftmost sample of the search area and it is the only smallest sample, it will not be included in the next sample's search area. Therefore we need to make a full comparison. Otherwise we only need to compare *lastmin* with the rightmost sample of the search area and define the smallest value (This part is similar to what is mentioned for step one of the fast 1-D GS erosion). If the recent smallest sample is less than *lastmin*, it means there is a new smallest value. If there is a new smallest value go to step 3 otherwise repeat step 2. If the whole samples have been scanned, then go to step 4.

3. If there is one or more new smallest value(s), we assign the corresponding output sample(s), from leftmost to rightmost sample, the same value as the new smallest value. The output samples between the leftmost new smallest location and *lastminloc* are assigned with

$$\max\{lastmin, new\ smallest\ value\}.$$

After assigning all output samples with correct values, we update the *lastmin* and *lastminloc* with the new smallest value, and then go to step 2.

4. The value of *lastmin* is assigned to the output samples after *lastminloc* and the algorithm ends up with this step.

Table 3.5 illustrates, as an example, the fast single-pass algorithm for 1-D GS opening.

Table 3.5: Illustration of the fast 1-D GS opening ($W = 5$).

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
f(n)	9	8	7	6	9	7	2	3	6	8	9	5	7	8	8	7	9	8
- Samples in n+L	9	8	7	6	9													
		*	*	6	*	7												
-The smallest points are			*	6	*	*	2											
underlined.				*	*	*	2	3										
					*	*	2	*	6									
-The symbol * means the						*	2	*	*	8								
related point is not							2	*	*	*	9							
required for comparison.								3	6	8	9	5						
									6	8	9	5	7					
										*	*	5	*	8				
											*	5	*	*	8			
												5	*	*	*	7		
													7	8	8	7	9	
														*	*	7	*	8
															*	7	*	*
																7	*	*
																	9	8
(f o g)(n)	6	6	6	6	6	6	2	3	5	5	5	5	7	7	7	7	8	8

Only 28 comparisons are required for the given example while the classical method needs $2N(W-1) = 144$ comparisons. The speed-up ratio is about 5.14.

Table 3.6 shows the performance of the proposed method in [89] for fast 1-D GS opening over the classical opening for a radar signal.

Table 3.6: The efficiency of fast 1-D GS opening over classical method adopted from [89]

W (size of SE)	3	7	11	15	19	23	27	31
The proposed method over classical one	2.09	4.00	6.00	8.00	10.00	12.17	14.08	16.08

3.3.2 Fast 2-D GS opening

We have experienced that the best fast technique for 2-D GS opening is applying fast 2-D GS erosion followed by dilation, otherwise single-pass fast 2-D opening is rather difficult to be analysed.

3.3.3 Fast 1-D and 2-D closing

For fast GS closing, similar dual algorithm regarding the start point and modifying the minima term with maxima can be applied.

3.4 Fast GS open-closing and close-opening

3.4.1 Fast 1-D GS open-closing

The fast technique for single-pass 1-D GS open-closing [69] is presented in this part. The current disscussion extends the method described for fast opening in [89] to combined open-closing operators within a single procedure for 1-D gray-scale signals. The structuring element is assumed to be flat. Open-closing, and close-opening of a 1-D signal with a line segment W of length $L(W = \{0, 1, \cdots, L-1\})$ as SE may be done by translating W point by point from left to right, looking for invariant points and assigning appropriate values to other points. The algorithm for open-closing is described below. A corresponding dual algorithm applies for close-opening.

1. For the starting point n=0, the smallest value(s) of the region $W + n$ is (are) found by comparing all the samples in the region. There may be several equally small sample values, and they will all be invariant points. This value is assigned to all samples up to the rightmost smallest sample.

2. Moving the comparison pointer to the right, new smallest samples are sought by two different methods. If there is only one smallest sample in the region $W + n - 1$ and it is the left-most sample of the region, the smallest sample will not be included in the region $W + n$, and the new smallest values are found by the comparison of all the samples in the region $W + n$. Otherwise the region $W + n$ will only have a new sample to be compared at the location of $W + n - 1$, which is not contained in the region $W + n - 1$. Then if $f(n + L - 1)$ is equal to, or smaller than the last smallest value of the region $W + n - 1$, it is a new location with the possibility of being an invariant sample. Otherwise there is no new smallest sample in the region $W + n$. We translate W and repeat the second part. If a new smallest value(s) is (are) found, we go to step 3.

3. If the location of the smallest value(s) is smaller than the size of the window (L), or if the new smallest value(s) is equal to, or greater than the last smallest, or the distance between the current smallest position and the first existing rejected sample between the last invariant sample and the current point is equal to, or greater than a window size, it is a new invariant point and go to step 4 for assigning data. Otherwise investigate other smallest points by sliding the window forward in the region n+W and put an index to the sought area for next comparisons. If the new smallest value is equal to, or greater than the smallest of the searched area, it will be an invariant point and go to step 4. Otherwise declare this point(s) as rejected points to be invariant, and the next time start data tracking from the sample after the rejected point. With this movement,

we will avoid the unnecessary comparisons between the current and the rejected point, and it will make the proposed algorithm more efficient than [89]. Of course a similar technique to step 2 is used for searching backward (or forward) to avoid seeking an area twice.

4. Assign the new smallest value to the locations between the first new smallest location and the last new smallest location (if there is more than one smallest). Then assign the locations between the last smallest location and the new left-most location with two different ways. If there is a rejected invariant point between them, fill the locations between the last smallest, and the rejected invariant location with the minimum of the last smallest, and the new smallest value including the rejected points as well. Otherwise for the rest of the points left between them, choose the maximum of the last and the new smallest. Candidate the right-most smallest location as last smallest location for the next operations, and use index to find whether there is a chance of moving forward the start location of the next tracking (another advantage of the proposed algorithm for speeding up compared to [89], and go to step 2.

5. When the last invariant sample of a signal is found, its value is assigned to the remaining output samples after it.

The proposed algorithm is illustrated by an example shown in Table 3.7. The window size of the structuring element (L) is 5 (i.e. $g = \{0, 0, 0, 0, 0\}$). The first row of the table shows the location of 18 samples ($0 - 17$). The second row shows how the smallest points are found. The underlined points are the smallest samples within W+n. The symbol $*$ indicates that the related points are not required for comparison. The symbol # shows the efficiency of the algorithm in order to avoid unnecessary comparisons, and moves forward the comparison pointer. The third row evaluates whether (or not) the smallest samples are invariant. The underlined points are invariant samples. The rejected points

are shown by the symbol x. The last row shows how the output values are found.

Table 3.7: Illustration of fast 1-D open-closing ($W = 5$)

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
f(n)	9	8	7	6	9	7	2	3	6	8	9	5	7	8	8	7	9	8
- Samples in n+L	9	8	7	6	9													
		#	#	#	#	#												
-The smallest points are			*	6	*	*	2											
underlined.				#	#	#	#	#										
					#	#	#	#	#									
-The symbol * means the						#	#	#	#	#								
related point is not							#	#	#	#	#							
required for comparison.								3	6	8	9	5						
									6	8	9	5	7					
-The symbol # indicates										#	#	#	#	#				
that the relevant set of											#	#	#	#	#			
data is not required to												#	#	#	#	#		
be compared.													7	8	8	7	9	
														#	#	#	#	#
															#	#	#	#
																#	#	#
																	9	8
-Accept 6 as invariant.				6														
-Reject 2 and 3 to be							x											
invariant points.								x										
-Accept 5,7,7,8 as												5						
invariant points.													7			7		
																		8
	6	6	6		6	6			5	5	5			7	7		8	
opcl(f)(n)				6			x	x				5	7			7		8
							5	5										

Legend for Table 3.7

1. Let the comparison pointer be identical to 0, to compare the set {9,8,7,6,9} and choose the smallest sample (6 in position 3). As the position is less than the window size, it will be an invariant point. Assign the same value to the positions less than 3 , i.e. assign 6 to the positions 0,1,2. Since the next point value after the searched area is greater than 6 (the point 5 value is $7 > 6$), we know that for the next comparison set (from point 1 to 5) 6 will again be the smallest. So to increase the efficiency, the

comparison pointer is incremented by 1 and starts from 2 for the next comparison set.

2. Now comparison pointer is at 2 to compare the set {*,6,*,*,2}. The symbol * means that there is no need to consider the relevant samples. Choose the smallest point (2 in position 6). The distance between this point and the last smallest point (6 in position 2) is less than the window size. Therefore we should evaluate whether it is greater than the next interval's smallest value, and whether this is true for more than a window size. We find that it is not true (due to 3 in location 7). Thus it will be declared as a rejected point. We may move the comparison pointer up to the next point for next operations (position 7).

3. The pointer is now at 7. Compare the set {3,6,8,9,5}. Choose the smallest point (3 in position 7). To the same reason as in part b , define this sample as another rejected point. The pointer after this step will automatically be at position 8.

4. With the pointer at 8, compare the set {6,8,9,5,7}. Choose the point (5 in position 11) as the smallest. The distance between this point and the last smallest point is equal to the window size. So it will be an invariant point. The last invariant point is 6 in position 3. Therefore put the minimum of the two invariant samples in the rejected points (i.e. put min(5,6)=5 in the locations 6 and 7). Then fill the locations between the last invariant point and the rejected points (i.e. the locations 4 and 5) with the maximum of the two values (max(5,6)=6). Then fill the rest of the empty points between the rejected points and the current invariant point (i.e. locations 8,9, and 10) with 5 (=max(5,5)). Then increase the comparison pointer to 12, because 5 in location 11 is less than 7,8,8,7 in locations 12 up to 15, and it will be again the smallest from pointer 9

up to 11.

5. Compare the set {7,8,8,7,9} at position 12, and choose the smallest sample (7 in positions 12 and 15). There are 2 equal smallest values in the set. Both of them are invariant, because they are greater than the last invariant point, and they will last for more than a window size. So write 7 from the position 12 up to 15, and increment the position pointer to 16.

6. When the pointer is at position 16, compare the set {9,8} and choose the value 8 in position 17 as the smallest point. It will again be an invariant point. Put 8 in position 17, and choose 8 (=max(7,8)) for the position 16. The processing in this example ends here.

For the given example, our approach needs only 18 comparisons while the classical method needs $4N(W-1) = 4 * 18 * 4 = 288$ comparisons, leading to a speed-up ratio of 16.

3.4.2 Fast 2-D GS open-closing and close-opening

For simplicity, we suggest applying proper cascades of fast 2-D GS erosion and dilation.

3.5 Efficiency and experimental results

The efficiency of the proposed algorithms arise because they avoid comparisons along the locations which have been compared previously. On the other hand, the standard methods slide the structuring element over the data, and make exhaustive comparisons at every position. Therefore there are repeated comparisons which take much longer time to complete, especially for large SEs. The method suggested by [89] for opening has enhanced the computational

Table 3.8: Relative performance of required time ratio for the open-closing for an ECG data set of 2500 samples with different algorithms

	ECG data with a baseline shift		ECG data with added Gaussian noise	
Window size (L)	Time ratio of our method over standard method	Time ratio of method [89] over standard method	Time ratio of our method over standard method	Time ratio of method [89] over standard method
3	0.606	0.860	0.627	0.886
5	0.445	0.690	0.434	0.628
10	0.270	0.471	0.218	0.390
20	0.168	0.315	0.110	0.236
30	0.116	0.260	0.079	0.177
40	0.118	0.231	0.069	0.146
50	0.102	0.215	0.058	0.127
60	0.091	0.197	0.051	0.114

Table 3.9: Relative performance of our method over [89] in Table 3.8 for fast 1-D open-closing

	ECG data with a baseline shift	ECG data with added Gaussian noise
L	our method over [89]	our method over [89]
3	1.42	1.41
5	1.55	1.45
10	1.74	1.79
20	1.88	2.15
30	2.24	2.24
40	1.98	2.12
50	2.11	2.27
60	2.16	2.24

efficiency of these operators and our method for open-closing has improved the method in [89] by both avoiding the unnecessary comparisons suggested in [89] and moreover, by putting a track point for the next required comparisons. Also the proposed algorithm can realise open-closing by a single pass when the method in [89] needs two passes.

Table 3.8 compares the relative required time with different algorithms for an ECG data set of 2500 samples with several window sizes run on a 386 PC, for three different methods.

Table tab:compopcl11 shows the results of Table tab:compopcl1 as absolute performance ratio of our method over the proposed method in [89].

The speed ratio of the proposed algorithm over the standard method for

erosion of a 512 × 512-point gray-scale image with different flat structuring elements is presented in table 3.10.

Table 3.10: Relative performance of the speed ratio of fast 2-D erosion using a flat GSE for a 512 × 512 image.

2-D flat Structuring element size	3 × 3	5 × 5	7 × 7	9 × 9
Speed ratio of our method over standard method	1.29	2.25	3.00	3.98

We can guess the amount of the efficiency for the remaining fast operators. The worst case seems to happen when the input data is sorted ascending for erosion and descending for dilation. In either case the minima/maxima happens in leftward/rightward position of the search area and therefore a full comparison will be required for next operation. However with a simple change of the algorithm we can still apply the fast algorithms by changing the direction of sweeping the input signal/image from end/start for erosion/dilation.

3.6 Remarks and conclusion

This chapter has shown our contribution about fast implementation of 1-D GS open-closing and close-opening, 2-D erosion and dilation besides another fast algorithms for the remaining operators. The algorithms applied in this chapter are direct implementation of the morphological operators, by single one-pass procedures. They are computationally simple and very efficient with a 18-20 % reduction in computational effort for morphological operations compared with the fastest alternative method, and an order of magnitude improvements over naive implementations for large structuring elements. All the algorithms mentioned above have been realised by a user-friendly package designed by author for 1-D and 2-D morphological systems design.

Chapter 4

Real-time implementation & hardware architecture of GSMF

4.1 Introduction

In previous chapter we had a look to fast algorithms in software. This chapter improves and generalises real-time implementations of GS (grey-scale) morphological operators proposed in [44] for any GSE (grey-scale structuring elements) having its centre in any position of the defined domain [72, 75, 73]. Extended algorithms are also applied in 2-D (two dimension). Schematic diagrams of hardware architecture and real-time implementations are included.

4.2 Background

As mentioned in Chapter 2, MF operators are classified into three groups as SP (set processing), FP (function processing) and FSP (function and set processing) systems. There are a variety of algorithms for real-time implementation of SP and FSP systems, however they are not easily applicable for FP systems. Some fast non-recursive methods for FP operators have been devel-

oped [89, 69]. For real-time applications, recursive algorithms are required. The proposed method in [44] is one of the best techniques for real-time FP operations. However it is applied for a specific GSE. Sedaaghi and Wu [72, 75, 73] improve and generalise the proposed algorithm in [44] and extend it to 2-D applications.

4.3 The proposed algorithms

If the centre of GSE is called $\mathbf{b(0)}$, there will be m points to the left (i.e. $b(-m), ..., b(-1)$) and n points to the right (i.e. $b(1), ..., b(n)$) of the centre point $\mathbf{b(0)}$ leading to a general GSE of the form:

$$\{b(-m), \cdots, b(-1), \underline{\mathbf{b(0)}}, b(1), \cdots, b(n)\}.$$

The total length (L) is thus $m + 1 + n$. To find out how the centre position of GSE affects the result of a morphological operator, we consider the following example with $m = n = 1$ and 7 input samples defined as $\{f(0), ..., f(6)\}$. The erosion $\{er(0), \cdots, er(6)\}$, dilation $\{di(0), \cdots, di(6)\}$, opening $\{op(0), \cdots, op(6)\}$ and closing $\{cl(0), \cdots, cl(6)\}$ are shown in Eqs. 4.3.1- 4.3.4 respectively.

$$\begin{aligned}
er(0) &= \min\{f(-1) - b(-1), f(0) - \mathbf{b(0)}, f(1) - b(1)\} \\
er(1) &= \min\{f(0) - b(-1) \ , f(1) - \mathbf{b(0)}, f(2) - b(1)\} \\
&\vdots \\
er(5) &= \min\{f(4) - b(-1) \ , f(5) - \mathbf{b(0)}, f(6) - b(1)\} \\
er(6) &= \min\{f(5) - b(-1) \ , f(6) - \mathbf{b(0)}, f(7) - b(1)\}
\end{aligned} \tag{4.3.1}$$

$$\begin{aligned}
di(0) &= \max\{f(-1) + b(1), f(0) + \mathbf{b(0)}, f(1) + b(-1)\} \\
di(1) &= \max\{f(0) + b(1) \ , f(1) + \mathbf{b(0)}, f(2) + b(-1)\} \\
&\vdots \\
di(5) &= \max\{f(4) + b(1) \ , f(5) + \mathbf{b(0)}, f(6) + b(-1)\} \\
di(6) &= \max\{f(5) + b(1) \ , f(6) + \mathbf{b(0)}, f(7) + b(-1)\}
\end{aligned} \tag{4.3.2}$$

$$
\begin{aligned}
op(0) &= \max\{er(-1)+b(1), er(0)+\mathbf{b(0)}, er(1)+b(-1)\} \\
op(1) &= \max\{er(0)+b(1) \ \ , er(1)+\mathbf{b(0)}, er(2)+b(-1)\} \\
&\vdots \\
op(5) &= \max\{er(4)+b(1) \ \ , er(5)+\mathbf{b(0)}, er(6)+b(-1)\} \\
op(6) &= \max\{er(5)+b(1) \ \ , er(6)+\mathbf{b(0)}, er(7)+b(-1)\}
\end{aligned}
\tag{4.3.3}
$$

and

$$
\begin{aligned}
cl(0) &= \min\{di(-1)-b(-1), di(0)-\mathbf{b(0)}, di(1)-b(1)\} \\
cl(1) &= \min\{di(0)-b(-1) \ \ , di(1)-\mathbf{b(0)}, di(2)-b(1)\} \\
&\vdots \\
cl(5) &= \min\{di(4)-b(-1) \ \ , di(5)-\mathbf{b(0)}, di(6)-b(1)\} \\
cl(6) &= \min\{di(5)-b(-1) \ \ , di(6)-\mathbf{b(0)}, di(7)-b(1)\}
\end{aligned}
\tag{4.3.4}
$$

The morphological operations are not defined for their first/last samples. The number of undefined samples equals to $m+n$. For example the erosion using the above samples needs m (=1) more samples before $f(0)$ ($f(-1)$) to calculate $er(0)$ and n (=1) more samples after $f(6)$ ($f(7)$) for $er(6)$. Three assumptions can be taken respectively based on what mentioned in chapter 3:

1. ignore early and late samples,
2. assume $f(-1)=f(0)$ and $f(7)=f(6)$,
3. assume $f(-1)$ and $f(7)$ are large enough to be ignored when finding minima for $er(0)$ and $er(6)$.

The dual assumption is required for dilation. We use the third assumption in Table 4.1 to illustrate how the position of the centre point $\mathbf{b(0)}$ affects the results. Except the early/late samples, the results in three columns are similar but have shifted values. This shift is very important when considering correlation between the input and output. Therefore the method proposed

Table 4.1: Results for $f(n) = \{29, 21, 18, 23, 26, 20, 15\}$ with different centre points for GSE.

g(n)	$\{\underline{2}, 5, 3\}$	$\{2, \underline{5}, 3\}$	$\{2, 5, \underline{3}\}$
$er(n)$	$\{15, 13, 16, 17, 12, 10, 13\}$	$\{18, 15, 13, 16, 17, 12, 10\}$	$\{26, 18, 15, 13, 16, 17, 12\}$
$di(n)$	$\{31, 34, 32, 25, 28, 31, 29\}$	$\{34, 32, 25, 28, 31, 29, 23\}$	$\{32, 25, 28, 31, 29, 23, 18\}$
$op(n)$	$\{17, 20, 18, 21, 22, 20, 15\}$	$\{23, 21, 18, 21, 22, 20, 15\}$	$\{29, 21, 18, 21, 22, 20, 15\}$
$cl(n)$	$\{29, 22, 20, 23, 26, 24, 27\}$	$\{29, 22, 20, 23, 26, 20, 18\}$	$\{29, 22, 20, 23, 26, 20, 15\}$
$opcl(n)$	$\{17, 20, 18, 21, 22, 20, 23\}$	$\{23, 21, 19, 21, 22, 20, 18\}$	$\{29, 21, 19, 21, 22, 20, 15\}$
$clop(n)$	$\{19, 22, 20, 23, 26, 24, 27\}$	$\{24, 22, 20, 23, 22, 20, 18\}$	$\{29, 22, 20, 23, 22, 20, 15\}$

in [44] is only correct for $g = \{b(-m), ..., b(-1), \mathbf{b(0)}\}$ (i.e. the centre is the rightmost sample in GSE).

For hardware implementation, the equation 2.6.1 can be simplified. Let $\beta_{L_i} = \mathbf{b(0)} - b(-i), \quad 0 < i \leq m$ and $\beta_{R_j} = \mathbf{b(0)} - b(j), \quad 0 < j \leq n$. The new results for the given example are:

$$\begin{aligned}
\epsilon(0) &= \min\{f(-1) + \beta_{L_1}, f(0), f(1) + \beta_{R_1}\} \\
\epsilon(1) &= \min\{f(0) + \beta_{L_1}, \ f(1), f(2) + \beta_{R_1}\} \\
&\vdots \\
\epsilon(5) &= \min\{f(4) + \beta_{L_1}, \ f(5), f(6) + \beta_{R_1}\} \\
\epsilon(6) &= \min\{f(5) + \beta_{L_1}, \ f(6), f(7) + \beta_{R_1}\}
\end{aligned} \tag{4.3.5}$$

The erosion will be:

$$er(i) = \epsilon(i) - \mathbf{b(0)}, \quad 0 \leq i \leq 6. \tag{4.3.6}$$

Similarly we can have the next simplifications for the dilation:

$$\begin{aligned}
\delta(0) &= \max\{f(-1) - \beta_{R_1}, f(0), f(1) - \beta_{L_1}\} \\
\delta(1) &= \max\{f(0) - \beta_{R_1}, \ f(1), f(2) - \beta_{L_1}\} \\
&\vdots \\
\delta(5) &= \max\{f(4) - \beta_{R_1}, \ f(5), f(6) - \beta_{L_1}\} \\
\delta(6) &= \max\{f(5) - \beta_{R_1}, \ f(6), f(7) - \beta_{L_1}\}
\end{aligned} \tag{4.3.7}$$

The dilation will be:

$$di(i) = \delta(i) + \mathbf{b(0)}, \quad 0 \leq i \leq 6. \tag{4.3.8}$$

The simplified opening will be:

$$\begin{aligned}
op(0) &= \max\{\epsilon(-1) - \beta_{R_1}, \epsilon(0), \epsilon(1) - \beta_{L_1}\} \\
op(1) &= \max\{\epsilon(0) - \beta_{R_1}, \ \epsilon(1), \epsilon(2) - \beta_{L_1}\} \\
&\vdots \\
op(5) &= \max\{\epsilon(4) - \beta_{R_1}, \ \epsilon(5), \epsilon(6) - \beta_{L_1}\} \\
op(6) &= \max\{\epsilon(5) - \beta_{R_1}, \ \epsilon(6), \epsilon(7) - \beta_{L_1}\}
\end{aligned} \tag{4.3.9}$$

The simplified closing will be:

$$\begin{aligned}
cl(0) &= \min\{\delta(-1) + \beta_{L_1}, \delta(0), \delta(1) + \beta_{R_1}\} \\
cl(1) &= \min\{\delta(0) + \beta_{L_1}, \ \delta(1), \delta(2) + \beta_{R_1}\} \\
&\vdots \\
cl(5) &= \min\{\delta(4) + \beta_{L_1}, \ \delta(5), \delta(6) + \beta_{R_1}\} \\
cl(6) &= \min\{\delta(5) + \beta_{L_1}, \ \delta(6), \delta(7) + \beta_{R_1}\}
\end{aligned} \tag{4.3.10}$$

Figures 4.1-a, 4.1-b and 4.1-c show the outputs for erosion and opening respectively when

1. *(i)* $m = 0, n = 2$ (i.e. $\mathbf{b(0)}$ in the left),
2. *(ii)* $m = n = 1$ (i.e. $\mathbf{b(0)}$ in the middle),
3. *(iii)* $m = 2, n = 0$ (i.e. $\mathbf{b(0)}$ in the right).

Figure 4.2-a presents a generalised implementation of erosion and opening. The delay lines can be initialised properly to be correct for early/late samples. Considering the duality, the generalised implementation of dilation and closing are also shown in Figure 4.2-b. The recursive implementation for *opcl* and *clop* can be achieved by proper cascade of the minima/maxima blocks of Figures 4.2-a and 4.2-b.

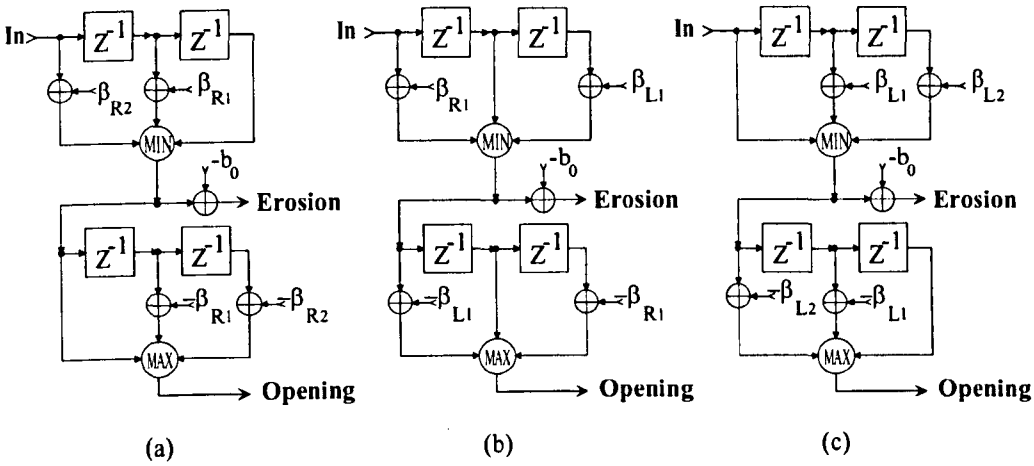


Figure 4.1: The fast recursive implementation of erosion and opening with GSE of size 3 with $\mathbf{b(0)}$ as centre: (a) $g(1) = \{\underline{\mathbf{b(0)}}, b(1), b(2)\}$, (b) $g(2) = \{b(-1), \underline{\mathbf{b(0)}}, b(1)\}$, (c) $g(3) = \{b(-2), b(-1), \underline{\mathbf{b(0)}}\}$

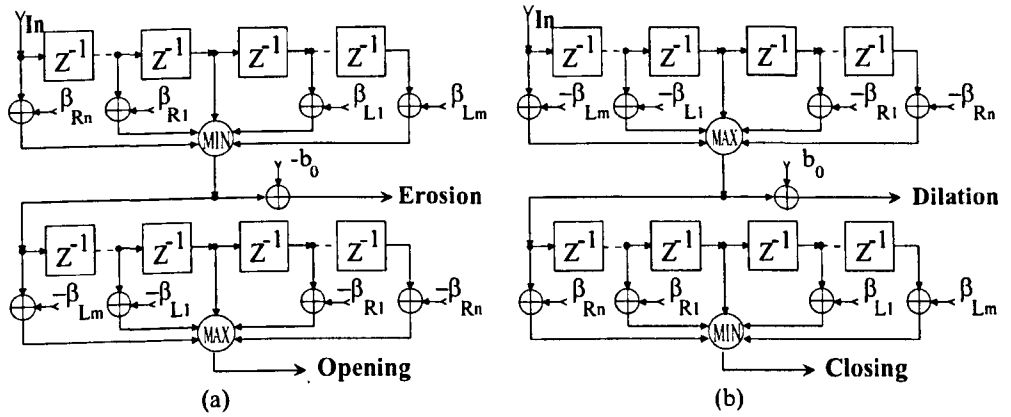


Figure 4.2: The generalised implementation of fast recursive: (a) erosion and opening (b) dilation and closing.

We now extend the proposed method in 2-D. Let $F(x,y)$ and $G(k,l)$ denote respectively a 2-D image with width Y and height X and a 2-D GSE with width L and height K, D_F and D_B denote their domains and $F(\alpha,\beta) \mid (\alpha,\beta) \in D_F$. recalling Eq. 2.6.9 for 2-D GS erosion, we know that the GSE is a matrix and is defined as $(m+n+1=L, p+q+1=K)$:

$$G = \begin{bmatrix} b(-p,-m) & \dots & b(-p,-1) & b(-p,0) & b(-p,1) & \dots & b(-p,n) \\ \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ b(-1,-m) & \dots & b(-1,-1) & b(-1,0) & b(-1,1) & \dots & b(-1,n) \\ b(0,-m) & \dots & b(0,-1) & \underline{\mathbf{b(0,0)}} & b(0,1) & \dots & b(0,n) \\ b(1,-m) & \dots & b(1,-1) & b(1,0) & b(1,1) & \dots & b(1,n) \\ \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ b(q,-m) & \dots & b(q,-1) & b(q,0) & b(q,1) & \dots & b(q,n) \end{bmatrix}$$

As an example let $m=n=p=q=1$. Then 2-D erosion will be:

$$\begin{bmatrix} ER(0,0) = \min \begin{Bmatrix} F(-1,-1)-b(-1,-1) & , & F(-1,0)-b(-1,0) & , & F(-1,1)-b(-1,1), \\ F(0,-1)-b(0,-1) & , & F(0,0)-\underline{\mathbf{b(0,0)}} & , & F(0,1)-b(0,1), \\ F(1,-1)-b(1,-1) & , & F(1,0)-b(1,0) & , & F(1,1)-b(1,1) \end{Bmatrix} \\ \dots \\ ER(0,6) = \min \begin{Bmatrix} F(-1,5)-b(-1,-1) & , & F(-1,6)-b(-1,0) & , & F(-1,7)-b(-1,1), \\ F(0,5)-b(0,-1) & , & F(0,6)-\underline{\mathbf{b(0,0)}} & , & F(0,7)-b(0,1), \\ F(1,5)-b(1,-1) & , & F(1,6)-b(1,0) & , & F(1,7)-b(1,1) \end{Bmatrix} \\ \vdots \\ ER(6,0) = \min \begin{Bmatrix} F(5,-1)-b(-1,-1) & , & F(5,0)-b(-1,0) & , & F(5,1)-b(-1,1), \\ F(6,-1)-b(0,-1) & , & F(6,0)-\underline{\mathbf{b(0,0)}} & , & F(6,1)-b(0,1), \\ F(7,-1)-b(1,-1) & , & F(7,0)-b(1,0) & , & F(7,1)-b(1,1) \end{Bmatrix} \\ \dots \\ ER(6,6) = \min \begin{Bmatrix} F(5,5)-b(-1,-1) & , & F(5,6)-b(-1,0) & , & F(5,7)-b(-1,1), \\ F(6,5)-b(0,-1) & , & F(6,6)-\underline{\mathbf{b(0,0)}} & , & F(6,7)-b(0,1), \\ F(7,5)-b(1,-1) & , & F(7,6)-b(1,0) & , & F(7,7)-b(1,1) \end{Bmatrix} \end{bmatrix} \quad (4.3.11)$$

The dilation will be:

$$\begin{bmatrix} DI(0,0) = \max \begin{Bmatrix} F(-1,-1)+b(1,1) & , & F(-1,0)+b(1,0) & , & F(-1,1)+b(1,-1), \\ F(0,-1)+b(0,1) & , & F(0,0)+\underline{\mathbf{b(0,0)}} & , & F(0,1)+b(0,-1), \\ F(1,-1)+b(-1,1) & , & F(1,0)+b(-1,0) & , & F(1,1)+b(-1,-1) \end{Bmatrix} \\ \dots \\ DI(0,6) = \max \begin{Bmatrix} F(-1,5)+b(1,1) & , & F(-1,6)+b(1,0) & , & F(-1,7)+b(1,-1), \\ F(0,5)+b(0,1) & , & F(0,6)+\underline{\mathbf{b(0,0)}} & , & F(0,7)+b(0,-1), \\ F(1,5)+b(-1,1) & , & F(1,6)+b(-1,0) & , & F(1,7)+b(-1,-1) \end{Bmatrix} \\ \vdots \\ DI_{6,0} = \max \begin{Bmatrix} F(5,-1)+b(1,1) & , & F(5,0)+b(1,0) & , & F(5,1)+b(1,-1), \\ F(6,-1)+b(0,1) & , & F(6,0)+\underline{\mathbf{b(0,0)}} & , & F(6,1)+b(0,-1), \\ F(7,-1)+b(-1,1) & , & F(7,0)+b(-1,0) & , & F(7,1)+b(-1,-1) \end{Bmatrix} \\ \dots \\ DI(6,6) = \max \begin{Bmatrix} F(5,5)+b(1,1) & , & F(5,6)+b(1,0) & , & F(5,7)+b(1,-1), \\ F(6,5)+b(0,1) & , & F(6,6)+\underline{\mathbf{b(0,0)}} & , & F(6,7)+b(0,-1), \\ F(7,5)+b(-1,1) & , & F(7,6)+b(-1,0) & , & F(7,7)+b(-1,-1) \end{Bmatrix} \end{bmatrix} \quad (4.3.12)$$

The opening will be:

$$\left[\begin{array}{ll}
OP(0,0) = \max & \left\{\begin{array}{lll} ER(-1,-1)+b(1,1) & , & ER(-1,0)+b(1,0) \quad , \quad ER(-1,1)+b(1,-1), \\ ER(0,-1)+b(0,1) & , & ER(0,0)+\underline{\mathbf{b(0,0)}} \quad , \quad ER(0,1)+b(0,-1), \\ ER(1,-1)+b(-1,1) & , & ER(1,0)+b(-1,0) \quad , \quad ER(1,1)+b(-1,-1) \end{array}\right\} \\
\cdots & \\
OP(0,6) = \max & \left\{\begin{array}{lll} ER(-1,5)+b(1,1) & , & ER(-1,6)+b(1,0) \quad , \quad ER(-1,7)+b(1,-1), \\ ER(0,5)+b(0,1) & , & ER(0,6)+\underline{\mathbf{b(0,0)}} \quad , \quad ER(0,7)+b(0,-1), \\ ER(1,5)+b(-1,1) & , & ER(1,6)+b(-1,0) \quad , \quad ER(1,7)+b(-1,-1) \end{array}\right\} \\
\vdots & \\
OP(6,0) = \max & \left\{\begin{array}{lll} ER(5,-1)+b(1,1) & , & ER(5,0)+b(1,0) \quad , \quad ER(5,1)+b(1,-1), \\ ER(6,-1)+b(0,1) & , & ER(6,0)+\underline{\mathbf{b(0,0)}} \quad , \quad ER(6,1)+b(0,-1), \\ ER(7,-1)+b(-1,1) & , & ER(7,0)+b(-1,0) \quad , \quad ER(7,1)+b(-1,-1) \end{array}\right\} \\
\cdots & \\
OP(6,6) = \max & \left\{\begin{array}{lll} ER(5,5)+b(1,1) & , & ER(5,6)+b(1,0) \quad , \quad ER(5,7)+b(1,-1), \\ ER(6,5)+b(0,1) & , & ER(6,6)+\underline{\mathbf{b(0,0)}} \quad , \quad ER(6,7)+b(0,-1), \\ ER(7,5)+b(-1,1) & , & ER(7,6)+b(-1,0) \quad , \quad ER(7,7)+b(-1,-1) \end{array}\right\}
\end{array}\right] \tag{4.3.13}$$

The closing will be:

$$\left[\begin{array}{ll}
CL(0,0) = \min & \left\{\begin{array}{lll} DI(-1,-1)-b(-1,-1), & DI(-1,0)-b(-1,0), & DI(-1,1)+b(-1,1), \\ DI(0,-1)-b(0,-1), & DI(0,0)-\underline{\mathbf{b(0,0)}}, & DI(0,1)-b(0,1), \\ DI(1,-1)-b(1,-1), & DI(1,0)-b(1,0), & DI(1,1)-b(1,1) \end{array}\right\} \\
\cdots & \\
CL(0,6) = \min & \left\{\begin{array}{lll} DI(-1,5)-b(-1,-1), & DI(-1,6)-b(-1,0), & DI(-1,7)-b(-1,1), \\ DI(0,5)-b(0,1), & DI(0,6)-\underline{\mathbf{b(0,0)}}, & DI(0,7)-b(0,1), \\ DI(1,5)-b(1,-1), & DI(1,6)-b(1,0), & DI(1,7)-b(1,1) \end{array}\right\} \\
\vdots & \\
CL_{6,0} = \min & \left\{\begin{array}{lll} DI(5,-1)-b(-1,-1), & DI(5,0)-b(1,0), & DI(5,1)-b(-1,1), \\ DI(6,-1)-b(0,1), & DI(6,0)-\underline{\mathbf{b(0,0)}}, & DI(6,1)-b(0,1), \\ DI(7,-1)-b(1,-1), & DI(7,0)-b(1,0), & DI(7,1)-b(1,1) \end{array}\right\} \\
\cdots & \\
CL(6,6) = \min & \left\{\begin{array}{lll} DI(5,5)-b(-1,-1), & DI(5,6)-b(-1,0), & DI(5,7)-b(-1,1), \\ DI(6,5)-b(0,-1), & DI(6,6)-\underline{\mathbf{b(0,0)}}, & DI(6,7)-b(0,1), \\ DI(7,5)-b(1,-1), & DI(7,6)-b(1,0), & DI(7,7)-b(1,1) \end{array}\right\}
\end{array}\right] \tag{4.3.14}$$

We let the following assumption for the terms of G:

$$\begin{array}{lll}
\beta_{U_k L_i} = \mathbf{b(0,0)} - b(-k,-i) \quad , & \beta_{U_k C} = \mathbf{b(0,0)} - b(-k,0) \quad , & \beta_{U_k R_j} = \mathbf{b(0,0)} - b(-k,j) \quad , \\
\beta_{C L_i} = \mathbf{b(0,0)} - b(0,-i) & , & \beta_{C R_j} = \mathbf{b(0,0)} - b(0,j) \quad , \\
\beta_{D_l L_i} = \mathbf{b(0,0)} - b(l,-i) \quad , & \beta_{D_l C} = \mathbf{b(0,0)} - b(l,0) \quad , & \beta_{D_l R_j} = \mathbf{b(0,0)} - b(l,j) \quad ,
\end{array}$$

$0 < k \leq p, 0 < l \leq q, 0 < i \leq m, 0 < j \leq n$ (i.e. there are p points above, q points beneath, m points on the left, and n points on the right of the centre

point $\mathbf{b}(0,0)$). The above results will be simplified to:

$$\left[\begin{array}{ll}
\Xi(0,0) = \min & \left\{\begin{array}{ccc} F(-1,-1)+\beta_{U_1L_1} & F(-1,0)+\beta_{U_1C} & F(-1,1)+\beta_{U_1R_1} \\ F(0,-1)+\beta_{CL_1} & F(0,0) & F(0,1)+\beta_{CR_1} \\ F(1,-1)+\beta_{D_1L_1} & F(1,0)+\beta_{D_1C} & F(1,1)+\beta_{D_1R_1} \end{array}\right\} \\
\cdots & \\
\Xi(0,6) = \min & \left\{\begin{array}{ccc} F(-1,5)+\beta_{U_1L_1} & F(-1,6)+\beta_{U_1C} & F(-1,7)+\beta_{U_1R_1} \\ F(0,5)+\beta_{CL_1} & F(0,6) & F(0,7)+\beta_{CR_1} \\ F(1,5)+\beta_{D_1L_1} & F(1,6)+\beta_{D_1C} & F(1,7)+\beta_{D_1R_1} \end{array}\right\} \\
\vdots & \\
\Xi(6,0) = \min & \left\{\begin{array}{ccc} F(5,-1)+\beta_{U_1L_1} & F(5,0)+\beta_{U_1C} & F(5,1)+\beta_{U_1R_1} \\ F(6,-1)+\beta_{CL_1} & F(6,0) & F(6,1)+\beta_{CR_1} \\ F(7,-1)+\beta_{D_1L_1} & F(7,0)+\beta_{D_1C} & F(7,1)+\beta_{D_1R_1} \end{array}\right\} \\
\cdots & \\
\Xi(6,6) = \min & \left\{\begin{array}{ccc} F(5,5)+\beta_{U_1L_1} & F(5,6)+\beta_{U_1C} & F(5,7)+\beta_{U_1R_1} \\ F(6,5)+\beta_{CL_1} & F(6,6) & F(6,7)+\beta_{CR_1} \\ F(7,5)+\beta_{D_1L_1} & F(7,6)+\beta_{D_1C} & F(7,7)+\beta_{D_1R_1} \end{array}\right\}
\end{array}\right] \tag{4.3.15}$$

and the erosion will be

$$ER(i,j) = \Xi(i,j) - \mathbf{b}(0,0), \quad 0 \le i,j \le 6. \tag{4.3.16}$$

$$\left[\begin{array}{ll}
\Delta(0,0) = \max & \left\{\begin{array}{ccc} F(-1,-1)-\beta_{D_1R_1} & F(-1,0)-\beta_{D_1C} & F(-1,1)-\beta_{D_1L_1} \\ F(0,-1)-\beta_{CR_1} & F(0,0) & F(0,1)-\beta_{CL_1} \\ F(1,-1)-\beta_{U_1R_1} & F(1,0)-\beta_{U_1C} & F(1,1)-\beta_{U_1L_1} \end{array}\right\} \\
\cdots & \\
\Delta(0,6) = \max & \left\{\begin{array}{ccc} F(-1,5)-\beta_{D_1R_1} & F(-1,6)-\beta_{D_1C} & F(-1,7)-\beta_{D_1L_1} \\ F(0,5)-\beta_{CR_1} & F(0,6) & F(0,7)-\beta_{CL_1} \\ F(1,5)-\beta_{U_1R_1} & F(1,6)-\beta_{U_1C} & F(1,7)-\beta_{U_1L_1} \end{array}\right\} \\
\vdots & \\
\Delta(6,0) = \max & \left\{\begin{array}{ccc} F(5,-1)-\beta_{D_1R_1} & F(5,0)-\beta_{D_1C} & F(5,1)-\beta_{D_1L_1} \\ F(6,-1)-\beta_{CR_1} & F(6,0) & F(6,1)-\beta_{CL_1} \\ F(7,-1)-\beta_{U_1R_1} & F(7,0)-\beta_{U_1C} & F(7,1)-\beta_{U_1L_1} \end{array}\right\} \\
\cdots & \\
\Delta(6,6) = \max & \left\{\begin{array}{ccc} F(5,5)-\beta_{D_1R_1} & F(5,6)-\beta_{D_1C} & F(5,7)-\beta_{D_1L_1} \\ F(6,5)-\beta_{CR_1} & F(6,6) & F(6,7)-\beta_{CL_1} \\ F(7,5)-\beta_{U_1R_1} & F(7,6)-\beta_{U_1C} & F(7,7)-\beta_{U_1L_1} \end{array}\right\}
\end{array}\right] \tag{4.3.17}$$

and the dilation will be:

$$DI(i,j) = \Delta(i,j) + \mathbf{b}(0,0), \quad 0 \le i,j \le 6. \tag{4.3.18}$$

The opening is simplified to:

$$\left[\begin{array}{ll}
OP(0,0) = \max & \left\{\begin{array}{ccc} \Xi(-1,-1)-\beta_{D_1R_1} & \Xi(-1,0)-\beta_{D_1C} & \Xi(-1,1)-\beta_{D_1L_1} \\ \Xi(0,-1)-\beta_{CR_1} & \Xi(0,0) & \Xi(0,1)-\beta_{CL_1} \\ \Xi(1,-1)-\beta_{U_1R_1} & \Xi(1,0)-\beta_{U_1C} & \Xi(1,1)-\beta_{U_1L_1} \end{array}\right\} \\
\cdots & \\
OP(0,6) = \max & \left\{\begin{array}{ccc} \Xi(-1,5)-\beta_{D_1R_1} & \Xi(-1,6)-\beta_{D_1C} & \Xi(-1,7)-\beta_{D_1L_1} \\ \Xi(0,5)-\beta_{CR_1} & \Xi(0,6) & \Xi(0,7)-\beta_{CL_1} \\ \Xi(1,5)-\beta_{U_1R_1} & \Xi(1,6)-\beta_{U_1C} & \Xi(1,7)-\beta_{U_1L_1} \end{array}\right\} \\
\vdots & \\
OP(6,0) = \max & \left\{\begin{array}{ccc} \Xi(5,-1)-\beta_{D_1R_1} & \Xi(5,0)-\beta_{D_1C} & \Xi(5,1)-\beta_{D_1L_1} \\ \Xi(6,-1)-\beta_{CR_1} & \Xi(6,0) & \Xi(6,1)-\beta_{CL_1} \\ \Xi(7,-1)-\beta_{U_1R_1} & \Xi(7,0)-\beta_{U_1C} & \Xi(7,1)-\beta_{U_1L_1} \end{array}\right\} \\
\cdots & \\
OP(6,6) = \max & \left\{\begin{array}{ccc} \Xi(5,5)-\beta_{D_1R_1} & \Xi(5,6)-\beta_{D_1C} & \Xi(5,7)-\beta_{D_1L_1} \\ \Xi(6,5)-\beta_{CR_1} & \Xi(6,6) & \Xi(6,7)-\beta_{CL_1} \\ \Xi(7,5)-\beta_{U_1R_1} & \Xi(7,6)-\beta_{U_1C} & \Xi(7,7)-\beta_{U_1L_1} \end{array}\right\}
\end{array}\right] \tag{4.3.19}$$

The closing is simplified to:

$$\left[\begin{array}{ll}
CL(0,0) = \min & \left\{\begin{array}{ccc} \Delta(-1,-1)+\beta_{U_1L_1} & \Delta(-1,0)+\beta_{U_1C} & \Delta(-1,1)+\beta_{U_1R_1} \\ \Delta(0,-1)+\beta_{CL_1} & \Delta(0,0) & \Delta(0,1)+\beta_{CR_1} \\ \Delta(1,-1)+\beta_{D_1L_1} & \Delta(1,0)+\beta_{D_1C} & \Delta(1,1)+\beta_{D_1R_1} \end{array}\right\} \\
\cdots & \\
CL(0,6) = \min & \left\{\begin{array}{ccc} \Delta(-1,5)+\beta_{U_1L_1} & \Delta(-1,6)+\beta_{U_1C} & \Delta(-1,7)+\beta_{U_1R_1} \\ \Delta(0,5)+\beta_{CL_1} & \Delta(0,6) & \Delta(0,7)+\beta_{CR_1} \\ \Delta(1,5)+\beta_{D_1L_1} & \Delta(1,6)+\beta_{D_1C} & \Delta(1,7)+\beta_{D_1R_1} \end{array}\right\} \\
\vdots & \\
CL(6,0) = \min & \left\{\begin{array}{ccc} \Delta(5,-1)+\beta_{U_1L_1} & \Delta(5,0)+\beta_{U_1C} & \Delta(5,1)+\beta_{U_1R_1} \\ \Delta(6,-1)+\beta_{CL_1} & \Delta(6,0) & \Delta(6,1)+\beta_{CR_1} \\ \Delta(7,-1)+\beta_{D_1L_1} & \Delta(7,0)+\beta_{D_1C} & \Delta(7,1)+\beta_{D_1R_1} \end{array}\right\} \\
\cdots & \\
CL(6,6) = \min & \left\{\begin{array}{ccc} \Delta(5,5)+\beta_{U_1L_1} & \Delta(5,6)+\beta_{U_1C} & \Delta(5,7)+\beta_{U_1R_1} \\ \Delta(6,5)+\beta_{CL_1} & \Delta(6,6) & \Delta(6,7)+\beta_{CR_1} \\ \Delta(7,5)+\beta_{D_1L_1} & \Delta(7,6)+\beta_{D_1C} & \Delta(7,7)+\beta_{D_1R_1} \end{array}\right\}
\end{array}\right] \tag{4.3.20}$$

The generalised implementation of 2-D operators are shown in Figures 4.3 and 4.4. Each row of the image is controlled by a parallel-in/parallel-out shift register which also has a sequential output as well. The length of each shift register equals to the width of the image (Y). The parallel-load action happens when a row is scanned completely. The parallel-load parts are shown with double thick lines in Figures 4.3 and 4.4. Open-closing and close-opening can be built by proper cascades of the above blocks.

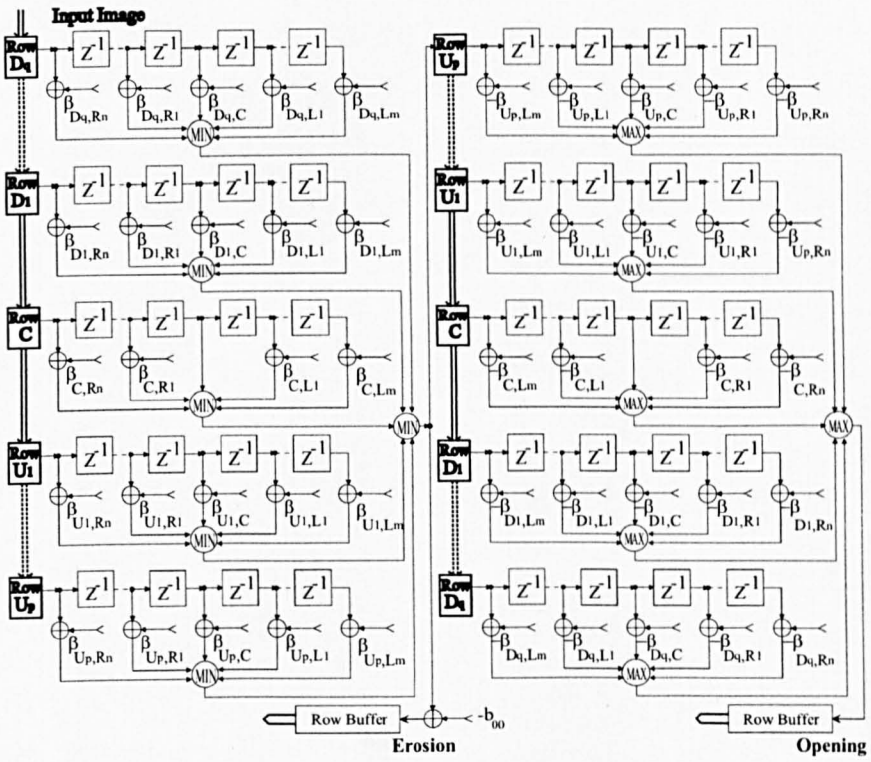


Figure 4.3: The generalised implementation of fast recursive 2-D erosion and opening.

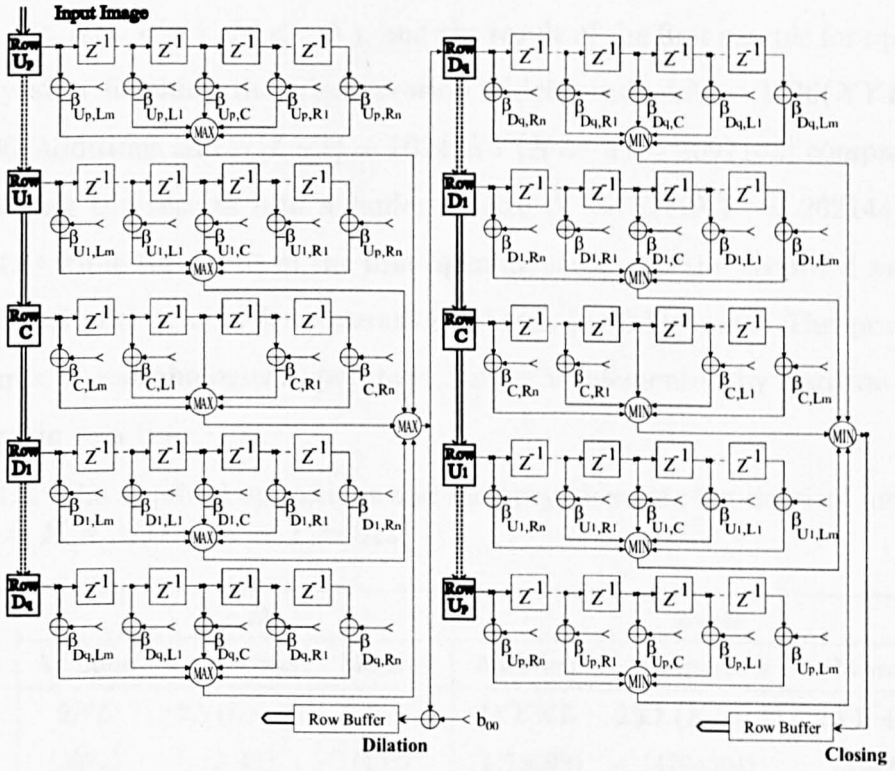


Figure 4.4: The generalised implementation of fast recursive 2-D dilation and closing.

4.4 Efficiency

Our algorithm for real-time MF avoids unnecessary operations and doesn't keep intermediate results. Therefore it is much faster than the classical cascade methods for opening, closing, open-closing and close-opening operations in which the intermediate results are needed. For example if the signal(image) size is $N = 512(X = Y = 512)$ and the GSE's size is $L = 3(K = L = 3)$, we have $L << N(L << Y, K << X)$, and the result of the first sample for opening is ready after finishing the whole erosion which needs $LN = 1536(XYKL = 2359296)$ additions and $N(L-1) = 1024(XY(KL-1) = 2097152)$ comparisons and keeping the results into a buffer of size $N = 512(XY = 262144)$ (see Table 4.2) while the result of the first opening sample in the proposed method is produced only after a few operations shown in Table 4.3. The proposed structures of morphological operators can be implemented by hardware and operated in real time.

Table 4.2: The required operations and memory elements for classical method. (Let $L = K = 3$, $N = X = Y = 512$)

Operations	$1-D$			$2-D$		
	Addition	Comparison	Memory	Addition	Comparison	Memory
op	$2NL$	$2N(L-1)$	$2N+L$	$2XYKL$	$2XY(KL-1)$	$2XY+KL$
	(3072)	(2048)	(1027)	(4718592)	(4194304)	(524297)
opcl	$4NL$	$4N(L-1)$	$2N+L$	$4XYKL$	$4XY(KL-1)$	$2XY+KL$
	(6144)	(4096)	(1027)	(9437184)	(8388608)	(524297)

4.5 Remarks and conclusion

Our major contribution in this chapter is related to the real-time implementation of 1-D and 2-D GS operators. We have tried to highlight the efficiency of the proposed methods. We have shown how our approaches generalise the

Table 4.3: The required operations and memory elements for proposed method. (Let $L = K = 3$, $Y = 512$)

Oper-	$1-D$			$2-D$		
ations	Addition	Comparison	Memory	Addition	Comparison	Memory
op	$2L-2$	$2L-2$	$3L-3$	$2KL-2$	$2KL-2$	$2YK+3KL$
	(4)	(4)	(6)	(16)	(16)	$-2K-1$(3092)
opcl	$4L-4$	$4L-4$	$5L-5$	$4KL-4$	$4KL-4$	$4YK+5KL$
	(8)	(8)	(10)	(32)	(32)	$-4K-1$(6176)

existing methods for any given SE. We have proved that how efficiently they operate, regarding the required hardware and execution time.

Chapter 5

Convolved morphological filters

5.1 Introduction

In previous chapters we have defined GS morphological operators, and presented fast algorithms and also real-time implementation of the operators. This chapter introduces novel convolved morphological operators (CMOs) and convolved morphological filters (CMFs) [74, 77]. As a reminder, a morphological operator is called a morphological filter if it is increasing, idempotent, and extensive or anti-extensive (see Chapter 2). The idea is to combine linear filtering with MF to take the advantage of both methods. The CMF operates using morphological operations and their convolution with selected impulse responses and offers highly accurate characteristics of high-, low-, and band-pass filtering. Designing a suitable structuring element based on the characteristics of the input signal is discussed and hardware implementation of CMF is also investigated.

By contrast to linear filtering which blurs the image, the MFs provide image enhancement without blurring the features. As examples, consider the quality of MF on aperiodic signals compared with linear filtering in Figures 5.1 and 5.2. The SE applied in all cases is a flat SE of size 3 for 1-D and 3×3 for 2-D

applications.

However there is no analytical method available for MF analysis due to their nonlinear features. Reconstruction of the original signal is impossible to be achieved after morphological operations (i.e. MF operators are idempotent: $\psi(\psi(f)) = f$), but linear methods are transferable in the time and frequency domains for analysis and design. Some efforts have been made to combine MF and convolution [84, 12]. This section proposes a novel method for design of MFs with enhancement provided by convolution, which results in a much superior performance compared with that obtained by linear filters.

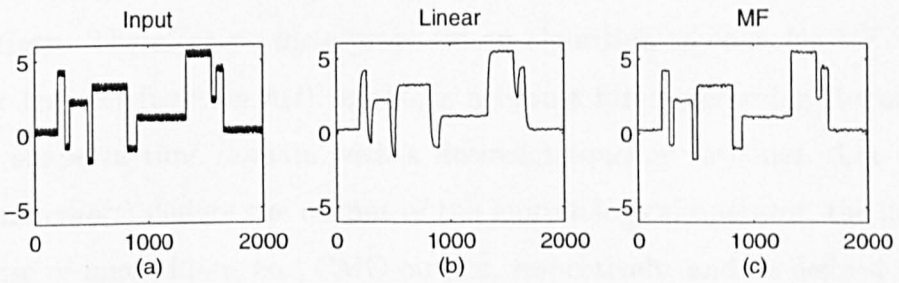


Figure 5.1: Noisy aperiodic square signal.

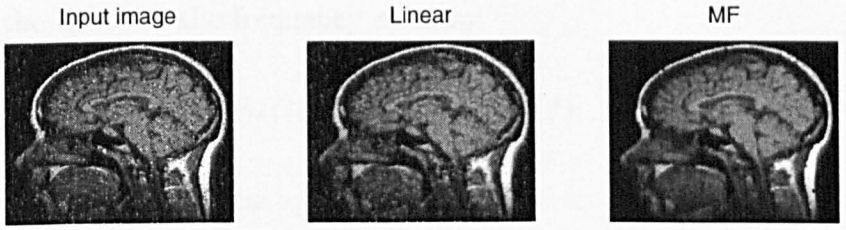


Figure 5.2: Noisy MRI image.

5.2 Definition of CMO

Let $in(t)$ be an input signal of the form

$$in(t) = A_1 \cos(2\pi \frac{f_1}{f_s} t + \phi_1) + \cdots + A_k \cos(2\pi \frac{f_k}{f_s} t + \phi_k) + e(t),$$

where f_s is the sampling frequency and $e(t)$ is white noise. $0 < f_1 < f_2 < \cdots < f_k < \frac{f_s}{2}$. We want to extract the desired frequencies singly/or in a band. Linear filters will cause a time delay which is not desirable. However the frequency response of the MF, despite avoiding the time delays (preserving the original signal shape), is not as valid as for linear filters due to its nonlinear operations. Therefore we have proposed an algorithm to convolve MF with a proper transfer function $h(t)$ to obtain a robust filter preserving the original signal shape in time domain with a desired frequency response. Let $mo(t)$, $h(t)$ and $cmo(t)$ denote the output of the morphological operator, the impulse response of linear filter, and CMO output, respectively, and be defined as

$$cmo(t) = h(t) * mo(t) = \sum_{\tau} h(\tau) \cdot mo(t - \tau), \tag{5.2.1}$$

where $\tau \in D_h, t - \tau \in D_{mo}$, and $*$ denotes convolution. Equivalently we can express the terms in the frequency domain:

$$CMO(f) = H(f) \cdot MO(f). \tag{5.2.2}$$

As convolved morphological operators, convolved erosion $cer(t)$, dilation $cdi(t)$, opening $cop(t)$, closing $ccl(t)$, open-closing $copcl(t)$ and close-opening $cclop(t)$ are defined as $(t, u, v, w \in D_g$ and $f(\alpha), \alpha \in D_f)$

$$cer(t) = \sum_{\tau} h(\tau) \cdot er(t - \tau) = \sum_{\tau} h(\tau) \cdot \min_{u} \{f(t + u - \tau) - g(u)\} \tag{5.2.3}$$

$$cdi(t) = \sum_{\tau} h(\tau) \cdot di(t - \tau) = \sum_{\tau} h(\tau) \cdot \max_{u} \{f(t - u - \tau) + g(u)\} \tag{5.2.4}$$

$$\begin{aligned} cop(t) &= \sum_{\tau} h(\tau) \cdot op(t-\tau) \\ &= \sum_{\tau} h(\tau) \cdot \max_{v}\{\min_{u}\{f(t+u-v-\tau) - g(u) + g(v)\}\} \end{aligned} \tag{5.2.5}$$

$$\begin{aligned} ccl(t) &= \sum_{\tau} h(\tau) \cdot cl(t-\tau) \\ &= \sum_{\tau} h(\tau) \cdot \min_{v}\{\max_{u}\{f(t-u+v-\tau) + g(u) - g(v)\}\} \end{aligned} \tag{5.2.6}$$

$$\begin{aligned} copcl(t) &= \sum_{\tau} h(\tau) \cdot opcl(t-\tau) = \sum_{\tau} h(\tau) \cdot \max_{w}\{\min_{v}\{\min_{u}\{\max_{s}\{ \\ & f(t-s+u+v-w-\tau) + g(t) - g(u) - g(v) + g(w)\}\}\}\} \end{aligned} \tag{5.2.7}$$

$$\begin{aligned} cclop(t) &= \sum_{\tau} h(\tau) \cdot cl(t-\tau) = \sum_{\tau} h(\tau) \cdot \min_{w}\{\max_{v}\{\max_{u}\{\min_{s}\{ \\ & f(t+s-u-v+w-\tau) - g(t) + g(u) + g(v) - g(w)\}\}\}\} \end{aligned} \tag{5.2.8}$$

5.3 Convolved morphological filters

We will consider the above CMOs, as convolved morphological filters (CMFs) except *cer* and *cdi* which are only operators and not filters as they are not idempotent. In this section we will use $(opcl + clop)/2$ as morphological filter to be convolved with $h(t)$.

Figures 5.3- 5.7 show an example when $k = 2$, $f_1 = 5$, $f_2 = 20$, $f_s = 128$ Hz, $A_1 = A_2 = 1$ and

$$h(t) = 2\sqrt{\frac{\alpha}{\pi}} \cdot \cos(2\pi f_0 t) \cdot e^{-\alpha t^2}, \tag{5.3.1}$$

where $f_0 = f_1$ for low-pass and $f_0 = f_2$ for band-pass filtering, and α is a parameter ($=\pi^2$). The linear filter applied for the example is a low- and band-pass butterworth of order 5.

5.3.1 Choosing proper impulse response

In this part we explain the reasons for choosing $h(t)$ as mentioned in Eq. 5.3.1 which seems to be a proper impulse response. If we assume $h_1(t)$ as

defined below:

$$h_1(t) = \sqrt{\frac{\alpha}{\pi}} \cdot e^{-\alpha t^2}. \tag{5.3.2}$$

In this case the frequency response $H_1(f)$ is

$$H_1(f) = e^{-\frac{\pi^2 \cdot f^2}{\alpha}}. \tag{5.3.3}$$

$h_1(t)$ and $H_1(f)$ have both a shape with peak at $n = 0$ and $f = 0$ respectively, and they both decay exponentially when moving away from 0 based on α. Therefore it can be ideal for clearing the unwanted frequencies using a proper α. If we shift $H_1(f)$ with f_0, then

$$H_2(f) = H_1(f - f_0) = e^{-\frac{\pi^2 \cdot (f - f_0)^2}{\alpha}}. \tag{5.3.4}$$

A shift in frequency domain will result in

$$h_2(t) = h_1(t) \cdot e^{j2\pi f_0 t}. \tag{5.3.5}$$

If we consider the mirror frequency as well, we can develop $H(f)$ as

$$H(f) = H_1(f - f_0) + H_1(f + f_0) = (e^{-\frac{\pi^2 \cdot (f - f_0)^2}{\alpha}} + e^{-\frac{\pi^2 \cdot (f + f_0)^2}{\alpha}}). \tag{5.3.6}$$

Then in time domain, we can develop the following relation

$$\begin{aligned} h(t) &= h_1(t) \cdot e^{j2\pi f_0 t} + h_1(t) \cdot e^{-j2\pi f_0 t} = 2h_1(t) \cdot \cos(2\pi f_0 t) \\ &= 2\sqrt{\tfrac{\alpha}{\pi}} \cdot e^{-\alpha t^2} \cdot \cos(2\pi f_0 t). \end{aligned} \tag{5.3.7}$$

5.3.2 Towards CMF

We will use the impulse response $h(t)$ defined in previous section. If we choose the structuring element SE_1 having the shape of half of a sinus with its peak at its centre and a length of $\frac{f_s/2}{f_2}$, the following equation will remove the shapes smaller than the size of SE_1:

$$residue(t) = [opcl(in(t)) + clop(in(t))]/2. \tag{5.3.8}$$

We apply the average of *opcl* and *clop* to have a symmetric result. As a reminder, we know that the higher the frequency of a signal, the smaller its shape in time domain. The signal $residue(t)$ as the output of MF in Eq. 5.3.8 contains the shapes greater than the shape of SE_1. As the shape of SE_1 is half a sinus, it will remove the shapes smaller than SE_1 (i.e. smaller than the shapes with higher frequencies than f_2). The size of half a sinus is enough considering the effect of erosion and dilation on SEs. Therefore we will have the components related to f_1 and f_2 preserved in $residue(t)$. If we increase the size of SE from $SE_1 = \frac{f_s/2}{f_2}$ to $SE_2 = \frac{f_s/2}{f_1}$, the following equation will act as low-pass MF, removing f_2 from $residue(t)$, for the same reason as mentioned above:

$$MFlow(t) = [opcl(residue(t)) + clop(residue(t))]/2. \tag{5.3.9}$$

As MF preserves the original structure, we can subtract $MFlow(t)$ from $residue(t)$ to extract f_2 acting as a band-pass MF:

$$MFband(t) = residue(t) - MFlow(t). \tag{5.3.10}$$

If we apply a convolution to Eqs. 5.3.9 and 5.3.10, we will have the desired CMF as low-pass and band-pass filters:

$$CMFlow(t) = Mflow(t) * h(t), \tag{5.3.11}$$

$$CMFband(t) = Mfban(t) * h(t). \tag{5.3.12}$$

Now we take the following steps as a general rule for periodic signals:

1. Choose a proper structuring element regarding both its shape and size.

2. Apply a proper morphological operator to extract the desired shapes in time domain based on the relations discussed in section B.1. Experimental results have proven that the best operators are $(opcl + clop)/2$.

3. Determine a suitable $h(t)$ for convolution regarding both time and frequency domain.

4. Convolve MF with $h(t)$.

Figures 5.8- 5.12 show another example when $k = 2$, $f_1 = 10$, $f_2 = 40$, $f_s = 1024$ Hz. The rest of the operators are the same as for 5.3- 5.7. Similar discussion exists for 2-D periodic signals (See Figures 5.13- 5.17 as an example of 2-D periodic signal and Figures 5.18- 5.22 as alternative example of 2-D periodic signal with increased sampling frequency).

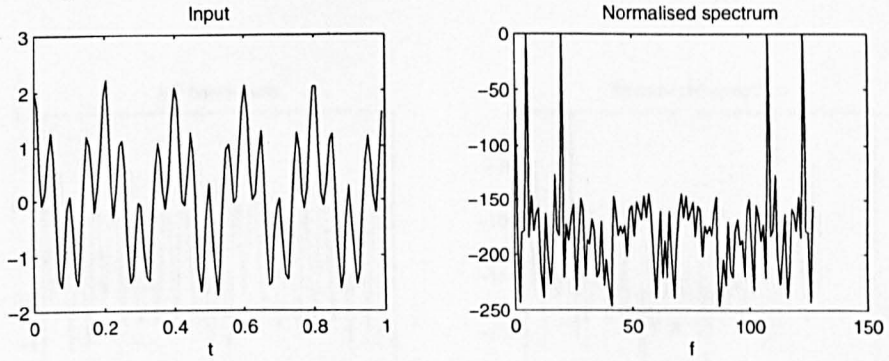


Figure 5.3: The input signal.

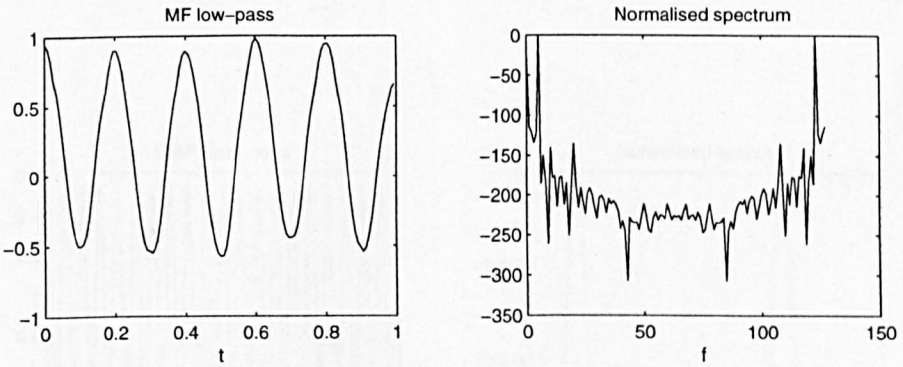


Figure 5.4: MF low-pass signal.

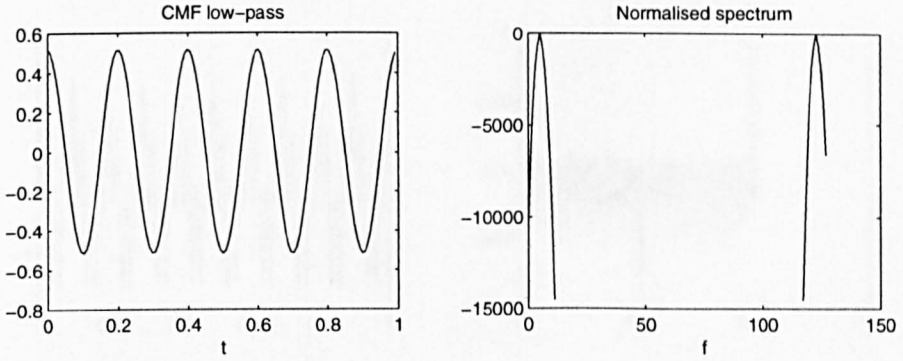


Figure 5.5: CMF low-pass signal.

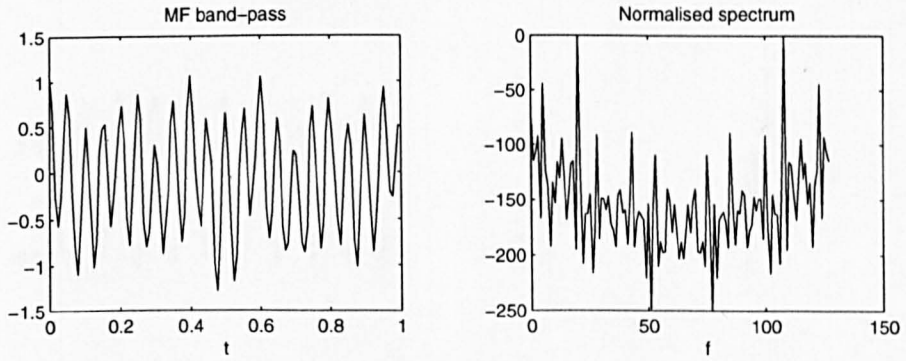


Figure 5.6: MF band-pass signal.

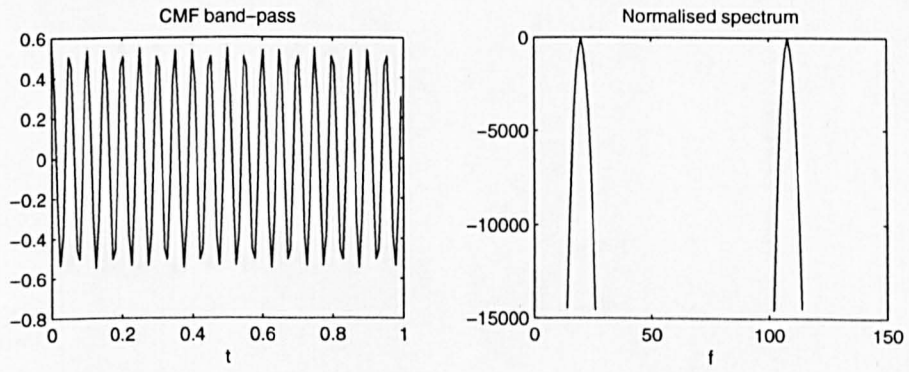


Figure 5.7: CMF band-pass signal.

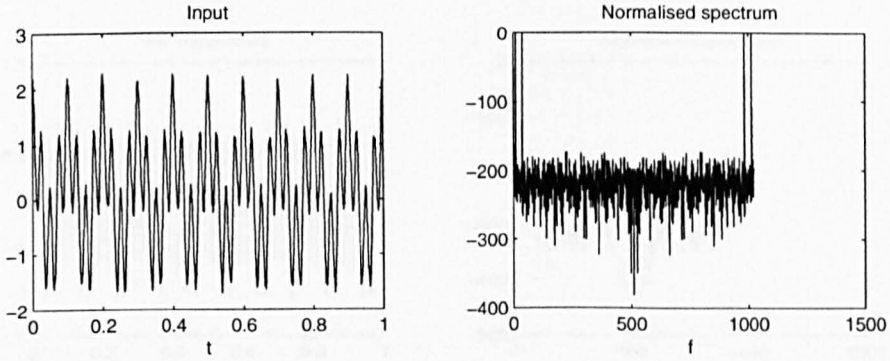


Figure 5.8: The input signals(example 2).

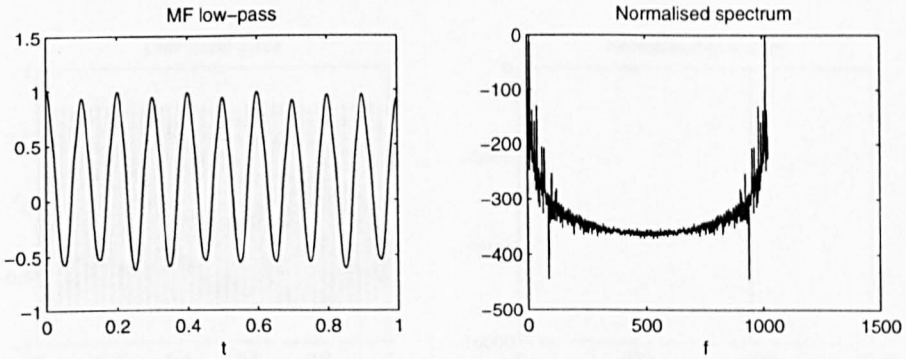


Figure 5.9: MF low-pass signal(example 2).

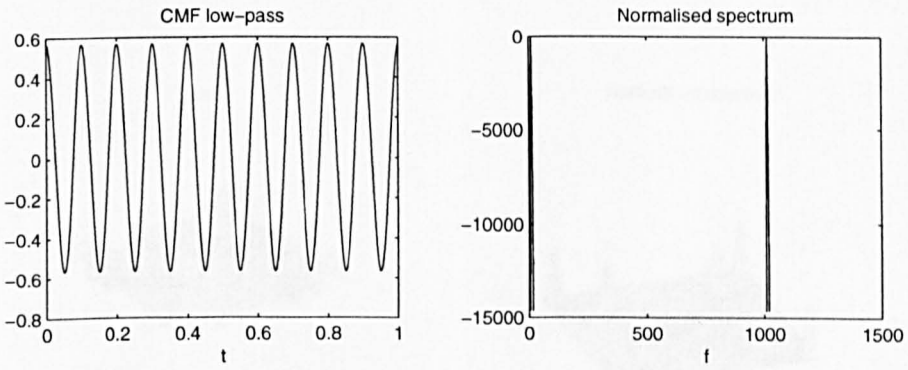


Figure 5.10: CMF low-pass signal(example 2).

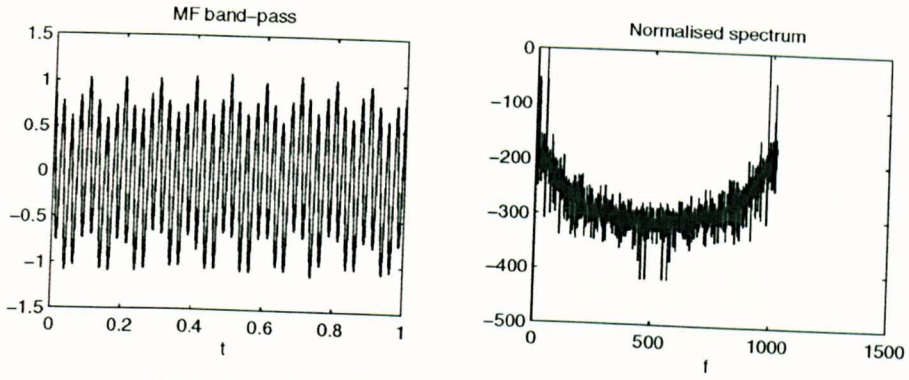


Figure 5.11: MF band-pass signal(example 2).

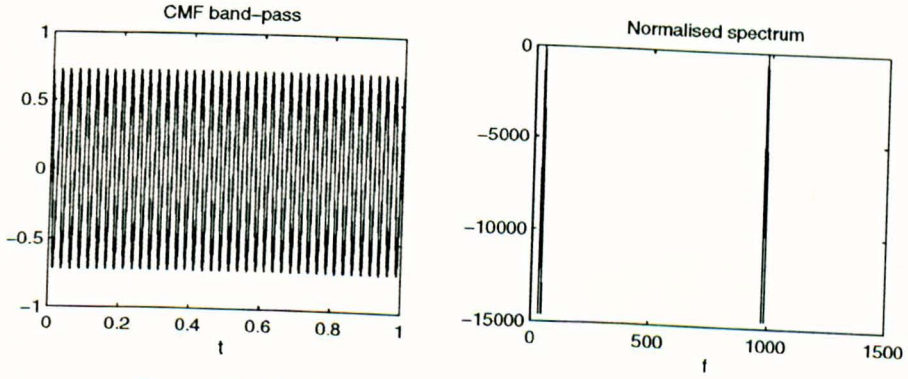


Figure 5.12: CMF band-pass signal(example 2).

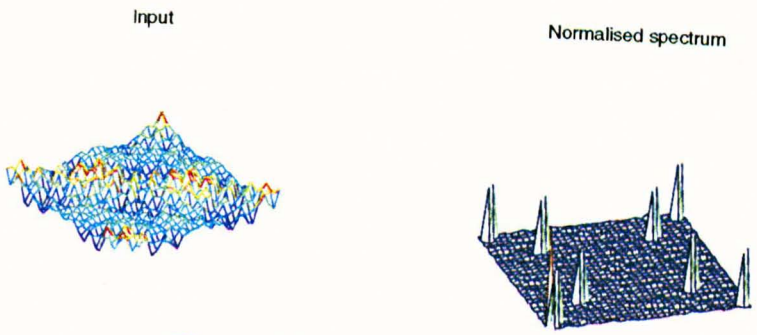


Figure 5.13: The input signals.

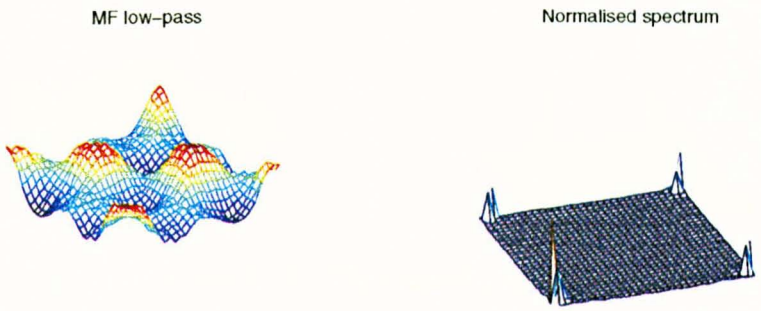


Figure 5.14: MF low-pass signal.

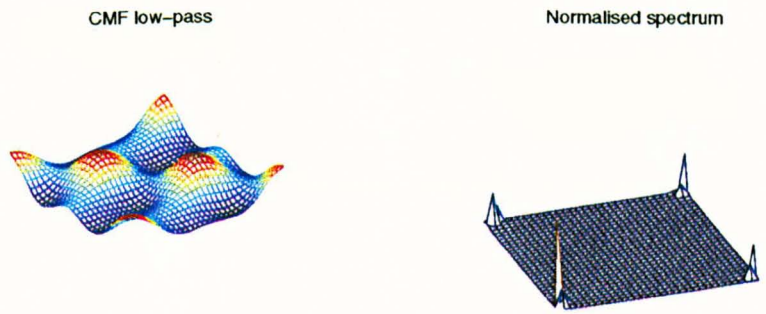


Figure 5.15: CMF low-pass signal.

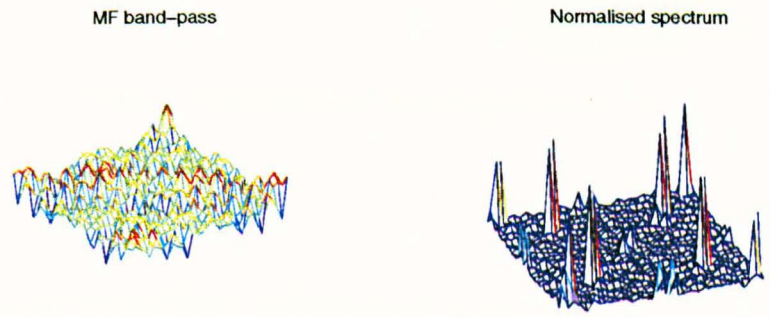


Figure 5.16: MF band-pass signal.

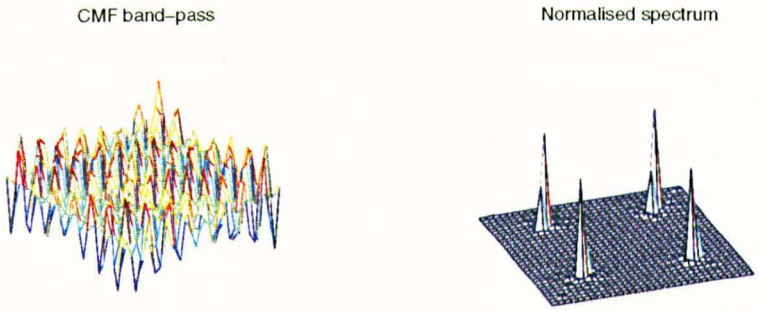


Figure 5.17: CMF band-pass signal.

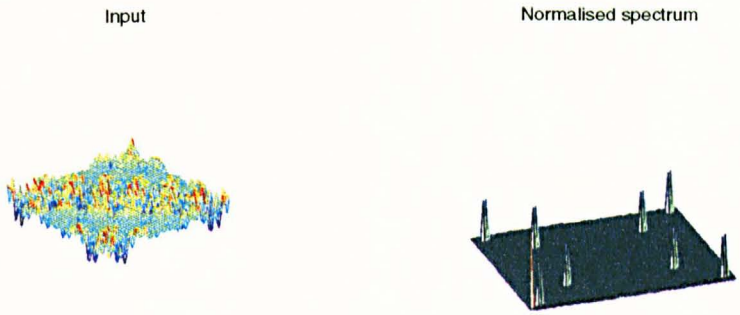


Figure 5.18: The input signals(example 2).

Figure 5.19: MF low-pass signal(example 2).

5.4 Structuring element design

We have empirically shown that the following rule exists for the design of the structuring element (SE). For periodic signals (with a limited spectrum),

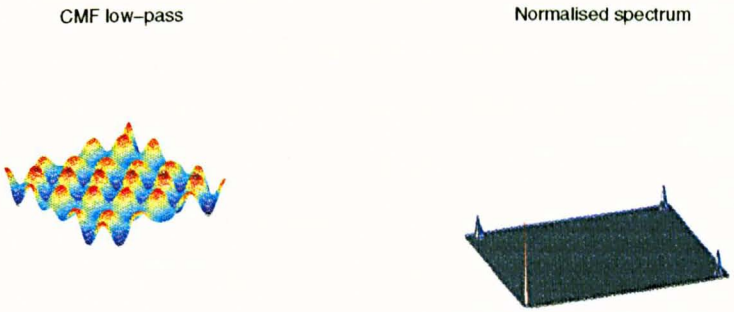


Figure 5.20: CMF low-pass signal(example 2).

Figure 5.21: MF band-pass signal(example 2).

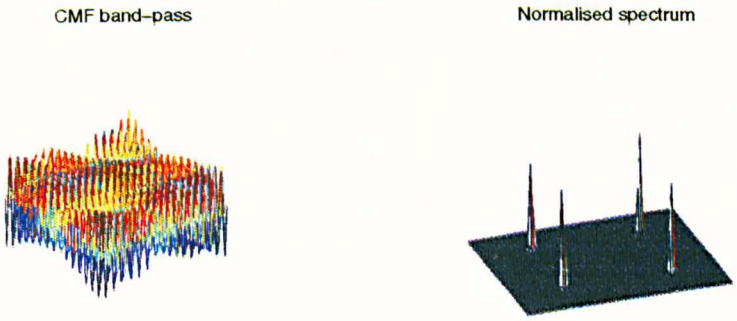


Figure 5.22: CMF band-pass signal(example 2).

the size of SE should approximately equal the number of samples building the corresponding frequency in time domain. For example for $MFlow(t)$, as defined before, the length of SE should be equal to 11 ($\frac{f_s/2}{f_1} = 128/2/5 = 12.8 \rightarrow 11$. It is a symmetric SE with the origin at its centre, and the size of SE should be odd). similarly the proper SE size for $residue(t)$ is $\frac{f_s/2}{f_1} = 128/2/20 = 3.2 \rightarrow 3$. After the size of SE is determined, its shape should be constructed similar to the shape of the desired signal (i.e. sinusoidal in the above example). For aperiodic signals we can design SE similar to the shape and size of the parts of the signal to be removed or preserved. If there is no pre-knowledge about the shape, the best guess is to apply a flat SE. With trial and error, the size of SE can be determined.

5.4.1 Hardware implementation

Based on the techniques of the previous chapter, we can realise CMFs by hardware (see Figure 5.23). Similar hardware for real-time implementation of other CMF operators can be developed.

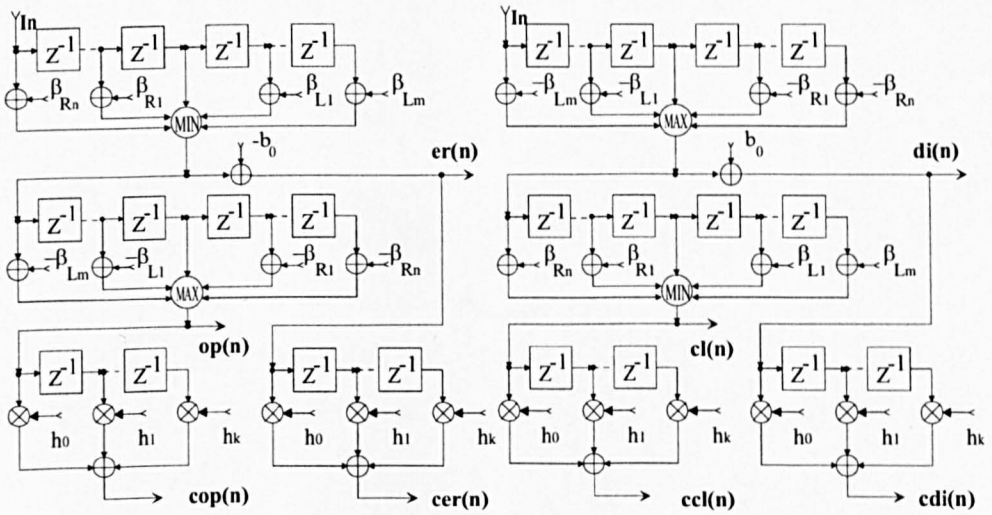


Figure 5.23: Hardware implementation of CMF.

5.5 Remarks and conclusion

We have discussed the design of CMF and shown its super performance in time domain and accurate characteristics in frequency domain when applied for signal/image filtering. CMFs' outstanding performance can be seen in both time and frequency domain. We have tested the proposed method with different periodic waveforms. In all cases, the benefits of applying the MF part in time domain, and sharp and accurate frequency response as a result of convolution part, have accompanied us.

Chapter 6

Weighted morphological filters

6.1 Introduction

Previous chapter introduced a technique to improve the efficiency of MFs in frequency domain by convolving them with linear methods. This chapter presents a novel morphological filter using weighted morphological operators (WMOs) [76]). The newly introduced operators employ a weighted structuring element and apply multiplication and division in place of addition and subtraction in classical morphological operations. Experimental results prove that the new operators' performance dominate over classical ones for signals/images buried in salt&pepper, speckle and Gaussian noises. Some outstanding approaches to promoting the efficiency of classical MF have been proposed [91, 83, 84]. In this chapter we introduce a weighted structuring element (SE) and new operators and present the results obtained using the novel weighted morphological filters (WMFs) for signals/images when buried in different noises.

6.2 Weighted morphological operators

Classical 2-D grey-scale (GS) erosion (ER) and dilation (DI) are defined as (Eqs. 2.6.9, 2.6.10): $ER(k,l) = \min_{u,v}\{f(k+u,l+v) - g(u,v)\}$, $DI(k,l) = \max_{u,v}\{f(k-u,l-v) + g(u,v)\}$, where f and g are the input image and SE respectively. Opening and closing are cascades of erosion and dilation in the proper order [33]. We define weighted erosion (WER) and dilation (WDI) as:

$$WER(k,l) = \min_{u,v}\{f(k+u,l+v)/g(u,v)\}, \tag{6.2.1}$$

$$WDI(k,l) = \max_{u,v}\{f(k-u,l-v)\cdot g(u,v)\}. \tag{6.2.2}$$

The other operators like weighted opening (WOP) and closing (WCL) are simply the cascades of weighted erosion and dilation as $WOP(f) = WDI(WER(f))$ and $WCL(f) = WER(WDI(f))$, respectively. Weighted open-closing ($WOPCL$) and close-opening ($WCLOP$) are defined as

$$WOPCL(f) = WCL(WOP(f)), \tag{6.2.3}$$

$$WCLOP(f) = WOP(WCL(f)), \tag{6.2.4}$$

respectively. The structuring element g has a normalised weight factor and its elements are calculated such that the centre point's weight is 1 and the farthest point's weight is assigned a weight factor $0 < w < 1$, leading to an emphasis over the effect of the central point and a reduction of the effect of the neighbourhood points. The rest of the weights are calculated based on an increment $\Delta w = (1-w)/d$, where d is the distance between the centre point and the farthest point from the centre. In the vertical and horizontal directions, the weights decrease by Δw, each step starting from the origin. In the oblique directions, the weights decrease by $2\Delta w$, each step starting from the centre. The centre can be located at any point of SE. For example, for SE of size 3×3 with its centre at its origin (i.e. $(2,2)$), if $w_1 = 0.1$, the SE will look like g_1, where $\Delta w = 0.45$, and for $w_2 = 0.9$, for the same centre point

(underlined), it will change to g_2, where $\Delta w = 0.05$. For SE of size 7×7 with its centre at $(1,5)$ and $w_3 = 0.5$, the SE will be g_3, where $\Delta w = 0.05$.

$$g_1 = \begin{bmatrix} 0.10 & 0.55 & 0.10 \\ 0.55 & \underline{1.00} & 0.55 \\ 0.10 & 0.55 & 0.10 \end{bmatrix} \tag{6.2.5}$$

$$g_2 = \begin{bmatrix} 0.90 & 0.95 & 0.90 \\ 0.95 & \underline{1.00} & 0.95 \\ 0.90 & 0.95 & 0.90 \end{bmatrix} \tag{6.2.6}$$

$$g_3 = \begin{bmatrix} 0.80 & 0.85 & 0.90 & 0.95 & \underline{1.00} & 0.95 & 0.90 \\ 0.75 & 0.80 & 0.85 & 0.90 & 0.95 & 0.90 & 0.85 \\ 0.70 & 0.75 & 0.80 & 0.85 & 0.90 & 0.85 & 0.80 \\ 0.65 & 0.70 & 0.75 & 0.80 & 0.85 & 0.80 & 0.75 \\ 0.60 & 0.65 & 0.70 & 0.75 & 0.80 & 0.75 & 0.70 \\ 0.55 & 0.60 & 0.65 & 0.70 & 0.75 & 0.70 & 0.65 \\ 0.50 & 0.55 & 0.60 & 0.65 & 0.70 & 0.65 & 0.60 \end{bmatrix}. \tag{6.2.7}$$

6.3 Properties

The weighted morphological operators (WMOs) imply the WER, WDI, WOP and WCL. They have the following properties if $0 < w < 1$:

1. WMOs are increasing ($f_1 \leq f_2 \Rightarrow WMO(f_1) \leq WMO(f_2)$).

2. WDI and WCL are extensive (i.e. $WDI(f) \geq f$ and $WCL(f) \geq f$), and WER and WOP are anti-extensive (i.e. $WER(f) \leq f$ and $WOP(f) \leq f$),

3. WER and WDI are not idempotent (i.e. $WER(WER(f)) \neq WER(f)$ and $WDI(WDI(f)) \neq WDI(f)$). However WOP and WCL are idempotent (i.e. $WOP(WOP(f)) = WOP(f)$ and $WCL(WCL(f)) = WCL(f)$).

These properties exist similarly in classical MFs and are consistent with the established theory of mathematical morphology. A morphological filter should be increasing, idempotent, and extensive or anti-extensive. [78]. Therefore the WMOs can be used to construct a WMF, because they have all the above three required conditions and We will use WCL or WOP as a filter (called WMF). However WER and WDI are considered as WMO.

6.4 Results and discussion

Figure 6.1-d confirms that the performance of WMF acts better than MF while MF removes salt&pepper noise more efficiently than the linear filter. Figures 6.2 and 6.3 show that for both Speckle and Gaussian noises, the linear filtering is better than MF and WMF. However they confirm that WMF is still better than MF.

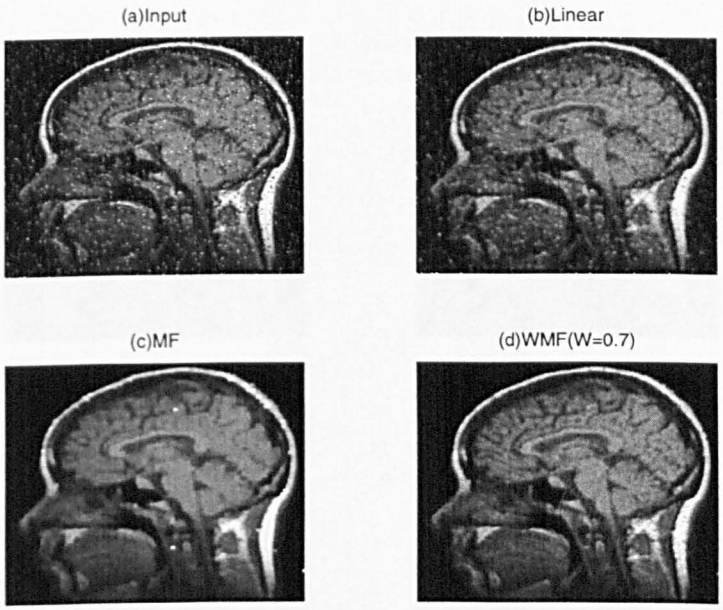


Figure 6.1: Salt&pepper noise removal with linear, MF and WMF.

The following results prove the dominance of WMF over MF in different

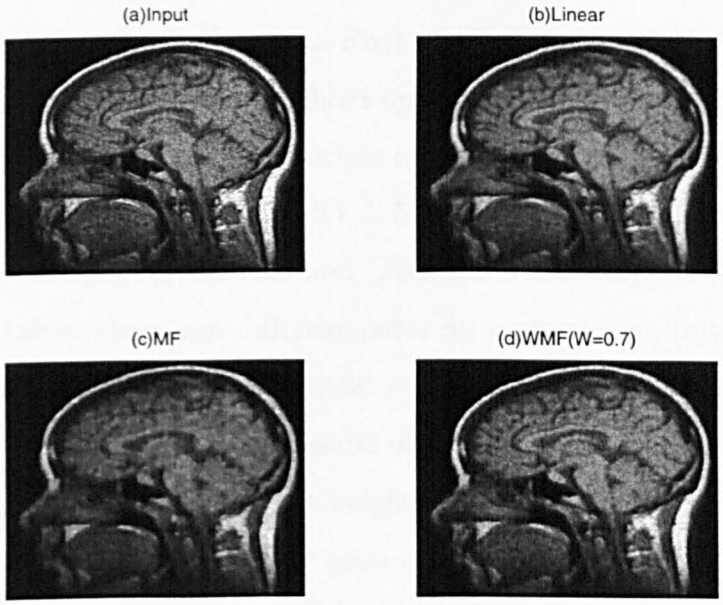


Figure 6.2: Speckle noise removal with linear, MF and WMF.

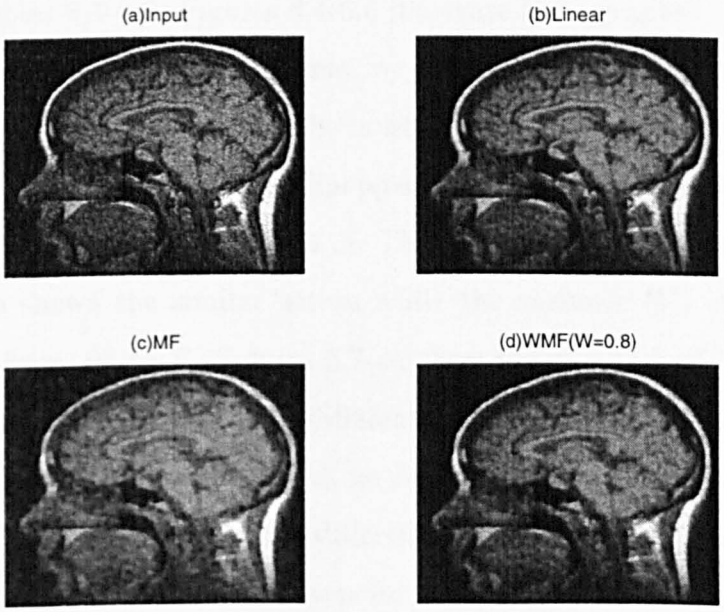


Figure 6.3: Gaussian noise removal with linear, MF and WMF.

noisy environments. We consider the correlation (λ) of the processed image with a reference (noise-free) image. First we test the effect of the size of SE on the results of classical and weighted open-closing and close-opening. In all experiments, the centre of SE is its origin to preserve the symmetry. Tables 6.1-6.3 show the analysis for SE of size 3×3, 5×5, 7×7 and 9×9, when the image is buried in salt&pepper, speckle and Gaussian noise, respectively. The first rows of the tables show two different parts for open-closing (*opcl*) and close-opening (*clop*) operations. The second rows are about the size of SEs. The third rows of the tables show the results of classical MFs. The rest of the rows are related to WMFs with different weights. In Table 6.1, for SE of size 3×3, we get the best results if $w > .4$ for open-closing and $w > .5$ for close-opening, where $\lambda > 0.9$. While the size of SE increases, the efficiency of both filters fall although WMF always seems to be better than MF. Similar discussion can be carried out for speckle and Gaussian noise (see Tables 6.2, 6.3). The relative performance of WMF over MF (λ_w/λ) based on the above results have been shown in Tables 6.4-6.6. Figures 6.4-6.6 illustrate their graphs.

In another sets of the experiments, we have tested the efficiency of WMFs while changing the parameters of the noises. Figure 6.7-a illustrates the result of weighted close-opening for salt&pepper noise while the parameter D (the noise intensity) changes from .01 to .5. The best results come true for $w > .4$. Figure 6.7-b shows the similar action while the variance (V) of the speckle noise varies from .01 to .5. Figures 6.7-c:f show the results for Gaussian noise with different variances (V) upon different mean values (M). We realise that the effect of M for Gaussian noise is negligible and it only produces an offset to the same shapes generated with different Ms. In all cases, SE's size is 3×3 with its centre at $(2,2)$ and the step for increasing noise parameters is .01. Tables 6.7-6.12 show the about results numerically.

Table 6.1: Salt&pepper noise ($D = 0.05$)

λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
-	0.9268	0.7045	0.4668	0.2299	0.9180	0.7043	0.4835	0.2606
0.1	0.7212	0.6595	0.4769	0.2447	0.6530	0.5464	0.4998	0.4475
0.2	0.8318	0.7461	0.5159	0.2493	0.7070	0.6162	0.5538	0.4762
0.3	0.8905	0.7615	0.5123	0.2468	0.7788	0.6962	0.5920	0.4814
0.4	0.9198	0.7610	0.5035	0.2445	0.8585	0.7490	0.6099	0.4640
0.5	0.9354	0.7547	0.4951	0.2426	0.9110	0.7769	0.6104	0.4271
0.6	0.9429	0.7457	0.4876	0.2404	0.9391	0.7833	0.5977	0.3848
0.7	0.9448	0.7349	0.4819	0.2382	0.9466	0.7733	0.5744	0.3465
0.8	0.9420	0.7239	0.4763	0.2351	0.9420	0.7535	0.5434	0.3120
0.9	0.9354	0.7136	0.4713	0.2324	0.9311	0.7287	0.5111	0.2831

Table 6.2: Speckle noise ($V = 0.04$)

λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
-	0.9381	0.7913	0.6374	0.5510	0.9254	0.7733	0.6907	0.6435
0.1	0.9247	0.8920	0.7557	0.6478	0.9224	0.8821	0.8092	0.7354
0.2	0.9293	0.8742	0.7115	0.6069	0.9245	0.8871	0.8116	0.7379
0.3	0.9369	0.8551	0.6884	0.5916	0.9315	0.8871	0.8007	0.7277
0.4	0.9428	0.8415	0.6770	0.5829	0.9388	0.8721	0.7820	0.7158
0.5	0.9469	0.8320	0.6700	0.5764	0.9452	0.8525	0.7632	0.7025
0.6	0.9492	0.8251	0.6643	0.5710	0.9483	0.8334	0.7439	0.6858
0.7	0.9507	0.8189	0.6588	0.5668	0.9469	0.8153	0.7268	0.6716
0.8	0.9496	0.8116	0.6524	0.5624	0.9415	0.7996	0.7127	0.6608
0.9	0.9453	0.8021	0.6449	0.5571	0.9341	0.7860	0.7012	0.6512

Table 6.3: Gaussian noise ($M = 0$, $V = 0.01$)

λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
-	0.8715	0.6862	0.4420	0.3210	0.8330	0.6703	0.5868	0.5334
0.1	0.8504	0.7784	0.5191	0.3564	0.8192	0.7694	0.7101	0.6581
0.2	0.8550	0.7440	0.4841	0.3389	0.8228	0.7583	0.6957	0.6433
0.3	0.8595	0.7209	0.4655	0.3289	0.8294	0.7542	0.6917	0.6405
0.4	0.8644	0.7121	0.4587	0.3270	0.8386	0.7480	0.6760	0.6258
0.5	0.8693	0.7088	0.4558	0.3253	0.8469	0.7324	0.6548	0.6042
0.6	0.8744	0.7062	0.4542	0.3249	0.8525	0.7168	0.6364	0.5837
0.7	0.8782	0.7039	0.4532	0.3244	0.8535	0.7017	0.6208	0.5668
0.8	0.8792	0.6998	0.4509	0.3240	0.8489	0.6899	0.6075	0.5520
0.9	0.8772	0.6941	0.4470	0.3227	0.8423	0.6799	0.5970	0.5417

Table 6.4: Relative comparison for salt&pepper noise ($D = 0.05$)

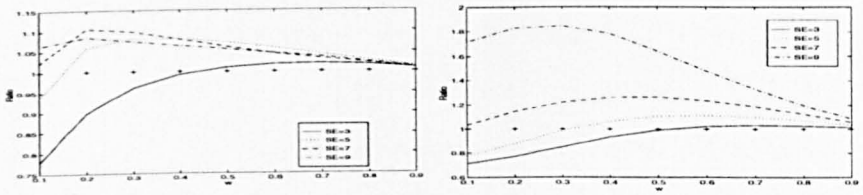
λ_w/λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
.1	0.7782	0.9361	1.0216	1.0644	0.7113	0.7758	1.0337	1.7172
.2	0.8975	1.0590	1.1052	1.0844	0.7702	0.8749	1.1454	1.8273
.3	0.9608	1.0809	1.0975	1.0735	0.8484	0.9885	1.2244	1.8473
.4	0.9924	1.0802	1.0786	1.0635	0.9352	1.0635	1.2614	1.7805
.5	1.0093	1.0713	1.0606	1.0552	0.9924	1.1031	1.2625	1.6389
.6	1.0174	1.0585	1.0446	1.0457	1.0230	1.1122	1.2362	1.4766
.7	1.0194	1.0432	1.0323	1.0361	1.0312	1.0980	1.1880	1.3296
.8	1.0164	1.0275	1.0204	1.0226	1.0261	1.0699	1.1239	1.1972
.9	1.0093	1.0129	1.0096	1.0109	1.0143	1.0346	1.0571	1.0863

Table 6.5: Relative comparison for Speckle noise ($V = 0.04$)

λ_w/λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
.1	0.9857	1.1273	1.1856	1.1757	0.9968	1.1407	1.1716	1.1428
.2	0.9906	1.1048	1.1163	1.1015	0.9990	1.1472	1.1750	1.1467
.3	0.9987	1.0806	1.0800	1.0737	1.0066	1.1472	1.1593	1.1308
.4	1.0050	1.0634	1.0621	1.0579	1.0145	1.1278	1.1322	1.1124
.5	1.0094	1.0514	1.0511	1.0461	1.0214	1.1024	1.1050	1.0917
.6	1.0118	1.0427	1.0422	1.0363	1.0247	1.0777	1.0770	1.0657
.7	1.0134	1.0349	1.0336	1.0287	1.0232	1.0543	1.0523	1.0437
.8	1.0123	1.0257	1.0235	1.0207	1.0174	1.0340	1.0319	1.0269
.9	1.0077	1.0136	1.0118	1.0111	1.0094	1.0164	1.0152	1.0120

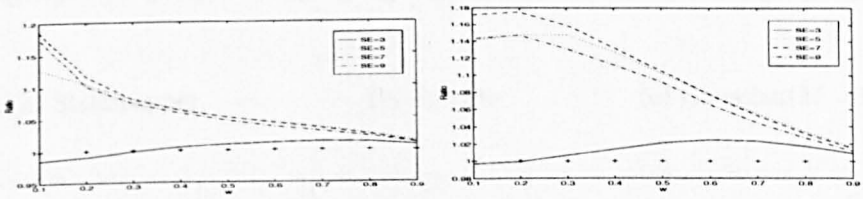
Table 6.6: Relative comparison for Gaussian noise ($M = 0$, $V = 0.01$)

λ_w/λ	SE for *opcl*				SE for *clop*			
w	3	5	7	9	3	5	7	9
.1	0.9758	1.1344	1.1744	1.1103	0.9834	1.1478	1.2101	1.2338
.2	0.9811	1.0842	1.0952	1.0558	0.9878	1.1313	1.1856	1.2060
.3	0.9862	1.0506	1.0532	1.0246	0.9957	1.1252	1.1788	1.2008
.4	0.9919	1.0377	1.0378	1.0187	1.0067	1.1159	1.1520	1.1732
.5	0.9975	1.0329	1.0312	1.0134	1.0167	1.0926	1.1159	1.1327
.6	1.0033	1.0291	1.0276	1.0121	1.0234	1.0694	1.0845	1.0943
.7	1.0077	1.0258	1.0253	1.0106	1.0246	1.0468	1.0579	1.0626
.8	1.0088	1.0198	1.0201	1.0093	1.0191	1.0292	1.0353	1.0349
.9	1.0065	1.0115	1.0113	1.0053	1.0112	1.0143	1.0174	1.0156



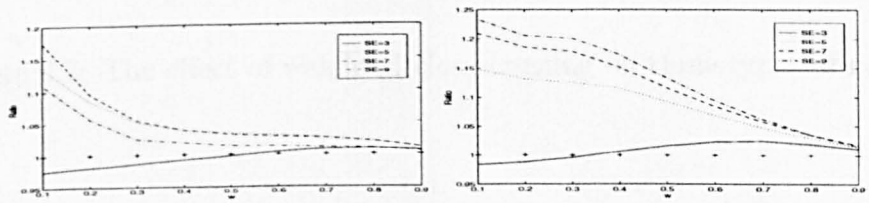
(a) λ_w/λ for open-closing (b) λ_w/λ for close-opening

Figure 6.4: Relative comparison for salt&pepper noise ($D = 0.05$)



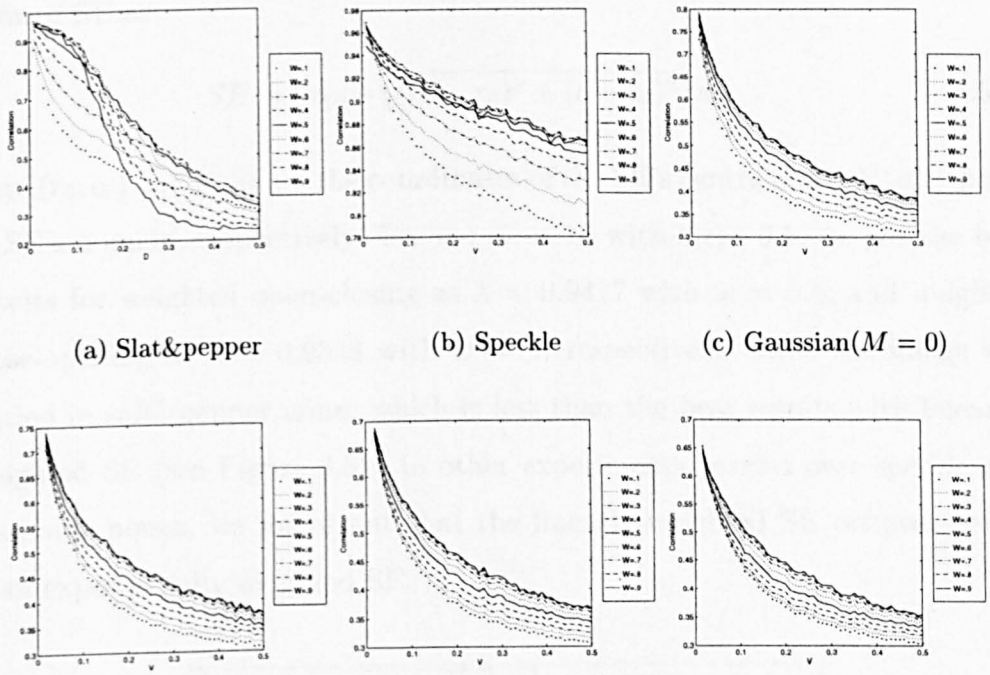
(a) λ_w/λ for open-closing (b) λ_w/λ for close-opening

Figure 6.5: Relative comparison for Speckle noise ($V = 0.04$)



(a) λ_w/λ for open-closing (b) λ_w/λ for close-opening

Figure 6.6: Relative comparison for Gaussian noise ($M = 0$, $V = 0.01$)



(a) Slat&pepper (b) Speckle (c) Gaussian($M = 0$)

(d) Gaussian($M = .05$) (e) Gaussian($M = .1$) (f) Gaussian($M = .15$)

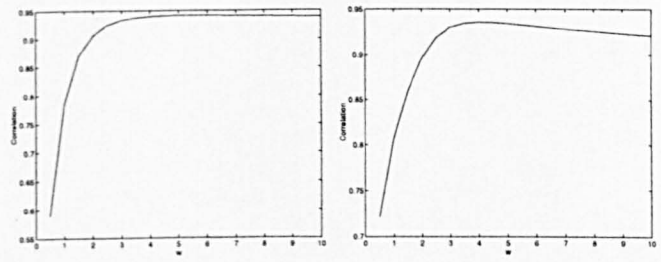
Figure 6.7: The effect of weighted close-opening on three types of noises.

6.5 Remarks and conclusion

In conclusion we realise the superb efficiency of WMF over MF in all noisy environments. If the average of the weighted open-closing and close-opening is applied, it will be the most ideal operator. However for most cases such an accuracy is not required as it needs twice the operation necessary for single operators leading to slow speed of processing. As another experiment, we defined SE as:

$$SE = \exp(-\sqrt{(r-r_0)^2 + (c-c_0)^2}/w), \tag{6.5.1}$$

were (r_0, c_0) and (r, c) are the coordinates of the SE's centre and arbitrary point in SE's domain, respectively. For $w : .5 \rightarrow 10$ with steps 0.5, we got the best results for weighted open-closing as $\lambda = 0.9417$ with $w = 5.5$, and weighted close-opening as $\lambda = 0.9358$ with $w = 4$, respectively, while the image was buried in salt&pepper noise, which is less than the best results with linearly-weighted SE (see Figure 6.8). In other experiments carried over speckle and Gaussian noises, we found out that the linearly weighted SE behaves better than exponentially weighted SE.



(a) *WOPCL* (b) *WCLOP*

Figure 6.8: Exponentially weighted SE for salt&pepper noise.

In this chapter we proposed new operators called WMOs. Their performance over MF was highlighted through this chapter. While MF fails to re-

move speckle and Gaussian noise, compared to linear methods, WMF improves the relevant performance. The only drawback of WMF is its speed compared to MF as they use multiplication/division instead of addition/subtraction, unless fast existing methods for linear filter coefficient design are applied [10, 42].

Table 6.7: The effect of WMF on salt&pepper

	Weight coefficient								
D	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.8723	0.8839	0.9104	0.9442	0.9640	0.9712	0.9694	0.9628	0.9536
0.02	0.7926	0.8253	0.8734	0.9196	0.9493	0.9628	0.9637	0.9573	0.9480
0.03	0.7426	0.7819	0.8362	0.8953	0.9346	0.9548	0.9587	0.9541	0.9449
0.04	0.6842	0.7374	0.7978	0.8681	0.9172	0.9431	0.9508	0.9483	0.9401
0.05	0.6502	0.7052	0.7787	0.8582	0.9099	0.9372	0.9441	0.9400	0.9295
0.06	0.6241	0.6859	0.7549	0.8360	0.8948	0.9285	0.9396	0.9360	0.9243
0.07	0.5961	0.6604	0.7354	0.8201	0.8852	0.9240	0.9383	0.9365	0.9259
0.08	0.5831	0.6455	0.7221	0.8088	0.8713	0.9051	0.9136	0.9056	0.8873
0.09	0.5601	0.6257	0.7049	0.7919	0.8597	0.8995	0.9130	0.9068	0.8876
0.10	0.5434	0.6114	0.6889	0.7770	0.8443	0.8851	0.8959	0.8851	0.8624
0.11	0.5299	0.5984	0.6778	0.7691	0.8396	0.8795	0.8882	0.8735	0.8444
0.12	0.5242	0.5952	0.6712	0.7556	0.8190	0.8543	0.8585	0.8384	0.8057
0.13	0.5039	0.5754	0.6494	0.7290	0.7897	0.8210	0.8168	0.7862	0.7398
0.14	0.5021	0.5744	0.6472	0.7268	0.7849	0.8118	0.7999	0.7613	0.7098
0.15	0.4975	0.5710	0.6442	0.7199	0.7740	0.7963	0.7804	0.7380	0.6833
0.16	0.4846	0.5517	0.6191	0.6879	0.7353	0.7491	0.7235	0.6722	0.6100
0.17	0.4747	0.5424	0.6111	0.6800	0.7234	0.7338	0.7053	0.6493	0.5834
0.18	0.4676	0.5370	0.6053	0.6680	0.6996	0.6943	0.6554	0.5944	0.5285
0.19	0.4592	0.5314	0.6014	0.6628	0.6954	0.6915	0.6505	0.5868	0.5172
0.20	0.4498	0.5175	0.5803	0.6341	0.6585	0.6475	0.6020	0.5372	0.4705
0.21	0.4422	0.5098	0.5744	0.6269	0.6464	0.6313	0.5815	0.5130	0.4455
0.22	0.4319	0.4980	0.5578	0.6065	0.6203	0.5980	0.5435	0.4739	0.4083
0.23	0.4360	0.5004	0.5608	0.6072	0.6223	0.5978	0.5403	0.4697	0.4036
0.24	0.4256	0.4891	0.5465	0.5891	0.5960	0.5674	0.5098	0.4420	0.3805
0.25	0.4189	0.4828	0.5374	0.5755	0.5778	0.5439	0.4840	0.4163	0.3551
0.26	0.4185	0.4809	0.5353	0.5722	0.5717	0.5362	0.4759	0.4095	0.3520
0.27	0.4109	0.4708	0.5224	0.5511	0.5425	0.5007	0.4394	0.3768	0.3236
0.28	0.4015	0.4576	0.5038	0.5276	0.5143	0.4697	0.4093	0.3487	0.2984
0.29	0.3976	0.4543	0.4970	0.5199	0.5064	0.4625	0.4042	0.3456	0.2968
0.30	0.3957	0.4518	0.4960	0.5155	0.5007	0.4556	0.3964	0.3385	0.2909
0.31	0.3875	0.4408	0.4837	0.5003	0.4829	0.4381	0.3826	0.3296	0.2867
0.32	0.3825	0.4335	0.4699	0.4825	0.4619	0.4157	0.3594	0.3081	0.2666
0.33	0.3863	0.4349	0.4738	0.4874	0.4652	0.4179	0.3623	0.3119	0.2722
0.34	0.3816	0.4298	0.4645	0.4756	0.4535	0.4070	0.3537	0.3068	0.2695
0.35	0.3686	0.4108	0.4408	0.4474	0.4229	0.3782	0.3296	0.2882	0.2555
0.36	0.3653	0.4095	0.4396	0.4459	0.4209	0.3753	0.3264	0.2847	0.2523
0.37	0.3638	0.4059	0.4336	0.4379	0.4120	0.3683	0.3212	0.2799	0.2481
0.38	0.3541	0.3957	0.4231	0.4263	0.4014	0.3595	0.3154	0.2792	0.2513
0.39	0.3514	0.3913	0.4164	0.4181	0.3919	0.3498	0.3069	0.2714	0.2439
0.40	0.3489	0.3868	0.4115	0.4118	0.3853	0.3442	0.3032	0.2696	0.2438
0.41	0.3355	0.3721	0.3952	0.3950	0.3685	0.3285	0.2898	0.2591	0.2356
0.42	0.3443	0.3792	0.4021	0.4021	0.3771	0.3396	0.3038	0.2752	0.2527
0.43	0.3380	0.3743	0.3958	0.3944	0.3690	0.3320	0.2961	0.2678	0.2458
0.44	0.3342	0.3679	0.3869	0.3844	0.3596	0.3225	0.2875	0.2600	0.2391
0.45	0.3243	0.3544	0.3713	0.3682	0.3449	0.3120	0.2813	0.2582	0.2406
0.46	0.3225	0.3521	0.3706	0.3685	0.3459	0.3138	0.2841	0.2614	0.2437
0.47	0.3162	0.3434	0.3600	0.3578	0.3357	0.3049	0.2771	0.2564	0.2403
0.48	0.3133	0.3395	0.3553	0.3526	0.3308	0.3013	0.2748	0.2550	0.2401
0.49	0.3107	0.3369	0.3515	0.3473	0.3259	0.2968	0.2710	0.2523	0.2378
0.50	0.3048	0.3306	0.3446	0.3421	0.3236	0.2973	0.2738	0.2559	0.2423

Table 6.8: The effect of WMF on Speckle noise

	Weight coefficient								
V	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.9695	0.9666	0.9664	0.9666	0.9674	0.9668	0.9633	0.9569	0.9485
0.02	0.9521	0.9498	0.9513	0.9543	0.9575	0.9587	0.9567	0.9510	0.9431
0.03	0.9365	0.9364	0.9406	0.9466	0.9518	0.9537	0.9526	0.9479	0.9408
0.04	0.9232	0.9257	0.9319	0.9388	0.9453	0.9480	0.9466	0.9413	0.9333
0.05	0.9110	0.9156	0.9238	0.9332	0.9404	0.9434	0.9420	0.9373	0.9298
0.06	0.9004	0.9072	0.9183	0.9286	0.9362	0.9401	0.9390	0.9340	0.9264
0.07	0.8911	0.9002	0.9127	0.9244	0.9324	0.9364	0.9362	0.9314	0.9234
0.08	0.8840	0.8947	0.9084	0.9200	0.9287	0.9323	0.9318	0.9268	0.9196
0.09	0.8777	0.8902	0.9048	0.9176	0.9267	0.9312	0.9318	0.9280	0.9206
0.10	0.8727	0.8865	0.9017	0.9135	0.9231	0.9273	0.9264	0.9215	0.9144
0.11	0.8667	0.8815	0.8969	0.9096	0.9187	0.9249	0.9253	0.9213	0.9136
0.12	0.8606	0.8768	0.8939	0.9071	0.9176	0.9230	0.9233	0.9194	0.9130
0.13	0.8596	0.8762	0.8930	0.9068	0.9162	0.9206	0.9218	0.9186	0.9123
0.14	0.8529	0.8701	0.8872	0.9002	0.9099	0.9151	0.9159	0.9134	0.9077
0.15	0.8502	0.8687	0.8854	0.8980	0.9083	0.9143	0.9147	0.9114	0.9056
0.16	0.8483	0.8668	0.8848	0.8982	0.9081	0.9134	0.9146	0.9119	0.9062
0.17	0.8442	0.8636	0.8821	0.8962	0.9054	0.9115	0.9133	0.9106	0.9049
0.18	0.8411	0.8606	0.8797	0.8931	0.9025	0.9069	0.9082	0.9058	0.9005
0.19	0.8355	0.8557	0.8752	0.8895	0.8996	0.9060	0.9064	0.9022	0.8957
0.20	0.8331	0.8535	0.8731	0.8868	0.8977	0.9045	0.9055	0.9027	0.8968
0.21	0.8320	0.8532	0.8724	0.8872	0.8971	0.9019	0.9016	0.8991	0.8928
0.22	0.8279	0.8499	0.8697	0.8849	0.8956	0.9009	0.9027	0.9014	0.8961
0.23	0.8282	0.8496	0.8681	0.8820	0.8917	0.8976	0.9000	0.8989	0.8931
0.24	0.8247	0.8464	0.8651	0.8799	0.8907	0.8960	0.8982	0.8966	0.8924
0.25	0.8251	0.8476	0.8671	0.8811	0.8912	0.8979	0.9006	0.8999	0.8956
0.26	0.8162	0.8393	0.8604	0.8760	0.8857	0.8909	0.8915	0.8879	0.8825
0.27	0.8205	0.8429	0.8623	0.8756	0.8841	0.8890	0.8907	0.8895	0.8854
0.28	0.8165	0.8399	0.8606	0.8752	0.8854	0.8916	0.8946	0.8933	0.8893
0.29	0.8151	0.8386	0.8582	0.8726	0.8835	0.8895	0.8916	0.8905	0.8862
0.30	0.8120	0.8354	0.8563	0.8715	0.8830	0.8899	0.8918	0.8900	0.8857
0.31	0.8145	0.8375	0.8573	0.8716	0.8807	0.8870	0.8901	0.8892	0.8851
0.32	0.8121	0.8353	0.8549	0.8705	0.8817	0.8894	0.8914	0.8903	0.8872
0.33	0.8072	0.8309	0.8502	0.8643	0.8743	0.8801	0.8837	0.8841	0.8808
0.34	0.8039	0.8276	0.8470	0.8617	0.8728	0.8796	0.8821	0.8814	0.8777
0.35	0.8050	0.8281	0.8484	0.8624	0.8727	0.8802	0.8837	0.8832	0.8800
0.36	0.8069	0.8302	0.8501	0.8640	0.8731	0.8800	0.8829	0.8831	0.8799
0.37	0.8030	0.8265	0.8460	0.8604	0.8709	0.8789	0.8827	0.8824	0.8798
0.38	0.7996	0.8239	0.8438	0.8576	0.8681	0.8749	0.8773	0.8777	0.8738
0.39	0.7966	0.8210	0.8416	0.8572	0.8679	0.8738	0.8761	0.8754	0.8733
0.40	0.7957	0.8205	0.8411	0.8549	0.8647	0.8717	0.8755	0.8765	0.8734
0.41	0.7944	0.8188	0.8400	0.8543	0.8638	0.8703	0.8727	0.8721	0.8692
0.42	0.7924	0.8175	0.8377	0.8517	0.8613	0.8677	0.8689	0.8662	0.8618
0.43	0.7889	0.8140	0.8346	0.8495	0.8599	0.8662	0.8699	0.8702	0.8665
0.44	0.7913	0.8160	0.8364	0.8508	0.8613	0.8680	0.8713	0.8715	0.8682
0.45	0.7865	0.8111	0.8320	0.8474	0.8566	0.8621	0.8651	0.8644	0.8619
0.46	0.7854	0.8110	0.8317	0.8460	0.8560	0.8625	0.8660	0.8660	0.8629
0.47	0.7837	0.8083	0.8287	0.8443	0.8548	0.8616	0.8657	0.8662	0.8637
0.48	0.7895	0.8137	0.8318	0.8449	0.8546	0.8617	0.8658	0.8672	0.8653
0.49	0.7859	0.8105	0.8306	0.8431	0.8535	0.8602	0.8646	0.8666	0.8653
0.50	0.7824	0.8075	0.8284	0.8431	0.8540	0.8609	0.8634	0.8635	0.8614

Table 6.9: The effect of WMF on Gaussian noise ($M = 0$)

	Weight coefficient								
V	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.8189	0.8222	0.8288	0.8379	0.8459	0.8522	0.8534	0.8505	0.8441
0.02	0.7319	0.7392	0.7520	0.7663	0.7797	0.7893	0.7929	0.7906	0.7846
0.03	0.6726	0.6823	0.6968	0.7134	0.7294	0.7409	0.7472	0.7478	0.7429
0.04	0.6228	0.6346	0.6526	0.6725	0.6901	0.7042	0.7117	0.7121	0.7082
0.05	0.5898	0.6027	0.6203	0.6418	0.6616	0.6770	0.6864	0.6892	0.6864
0.06	0.5568	0.5706	0.5899	0.6120	0.6324	0.6473	0.6553	0.6573	0.6547
0.07	0.5364	0.5509	0.5706	0.5931	0.6133	0.6290	0.6394	0.6426	0.6393
0.08	0.5200	0.5345	0.5555	0.5789	0.5994	0.6143	0.6232	0.6265	0.6251
0.09	0.4903	0.5063	0.5277	0.5534	0.5750	0.5907	0.6002	0.6031	0.6009
0.10	0.4783	0.4933	0.5141	0.5391	0.5614	0.5773	0.5866	0.5901	0.5883
0.11	0.4608	0.4759	0.4967	0.5209	0.5439	0.5619	0.5721	0.5754	0.5750
0.12	0.4542	0.4705	0.4923	0.5169	0.5377	0.5533	0.5619	0.5667	0.5678
0.13	0.4414	0.4562	0.4776	0.5023	0.5252	0.5418	0.5521	0.5560	0.5559
0.14	0.4305	0.4457	0.4661	0.4896	0.5132	0.5308	0.5419	0.5473	0.5466
0.15	0.4250	0.4401	0.4611	0.4856	0.5086	0.5263	0.5393	0.5454	0.5465
0.16	0.4111	0.4269	0.4478	0.4723	0.4944	0.5105	0.5211	0.5264	0.5285
0.17	0.4045	0.4196	0.4396	0.4624	0.4830	0.4984	0.5085	0.5135	0.5134
0.18	0.4002	0.4148	0.4347	0.4585	0.4806	0.4974	0.5081	0.5136	0.5149
0.19	0.3959	0.4110	0.4321	0.4557	0.4762	0.4915	0.5024	0.5085	0.5109
0.20	0.3880	0.4032	0.4235	0.4469	0.4675	0.4830	0.4919	0.4964	0.4988
0.21	0.3835	0.3991	0.4187	0.4416	0.4621	0.4784	0.4883	0.4929	0.4939
0.22	0.3751	0.3902	0.4104	0.4324	0.4527	0.4678	0.4773	0.4822	0.4839
0.23	0.3693	0.3833	0.4022	0.4243	0.4447	0.4605	0.4704	0.4766	0.4785
0.24	0.3696	0.3846	0.4026	0.4229	0.4425	0.4585	0.4690	0.4756	0.4787
0.25	0.3645	0.3786	0.3970	0.4174	0.4374	0.4530	0.4638	0.4688	0.4702
0.26	0.3605	0.3755	0.3945	0.4146	0.4327	0.4475	0.4574	0.4635	0.4665
0.27	0.3542	0.3677	0.3850	0.4048	0.4234	0.4382	0.4483	0.4529	0.4542
0.28	0.3535	0.3660	0.3807	0.3987	0.4169	0.4302	0.4392	0.4446	0.4470
0.29	0.3461	0.3593	0.3759	0.3940	0.4116	0.4252	0.4346	0.4408	0.4433
0.30	0.3460	0.3596	0.3754	0.3931	0.4098	0.4230	0.4324	0.4383	0.4406
0.31	0.3445	0.3572	0.3736	0.3905	0.4076	0.4205	0.4286	0.4334	0.4356
0.32	0.3435	0.3563	0.3713	0.3888	0.4049	0.4190	0.4289	0.4354	0.4389
0.33	0.3415	0.3546	0.3711	0.3892	0.4067	0.4213	0.4314	0.4375	0.4410
0.34	0.3348	0.3479	0.3634	0.3799	0.3949	0.4085	0.4185	0.4241	0.4259
0.35	0.3392	0.3516	0.3663	0.3838	0.3996	0.4114	0.4192	0.4238	0.4252
0.36	0.3398	0.3530	0.3677	0.3840	0.3997	0.4130	0.4232	0.4288	0.4317
0.37	0.3316	0.3444	0.3597	0.3761	0.3916	0.4041	0.4135	0.4204	0.4231
0.38	0.3305	0.3437	0.3589	0.3759	0.3921	0.4057	0.4159	0.4210	0.4233
0.39	0.3266	0.3391	0.3529	0.3687	0.3839	0.3960	0.4038	0.4085	0.4115
0.40	0.3254	0.3377	0.3524	0.3690	0.3847	0.3973	0.4069	0.4114	0.4133
0.41	0.3241	0.3363	0.3498	0.3652	0.3810	0.3944	0.4040	0.4092	0.4120
0.42	0.3157	0.3278	0.3412	0.3567	0.3709	0.3823	0.3904	0.3959	0.3981
0.43	0.3207	0.3337	0.3481	0.3625	0.3754	0.3862	0.3941	0.3980	0.3995
0.44	0.3189	0.3306	0.3438	0.3583	0.3725	0.3839	0.3929	0.3991	0.4031
0.45	0.3150	0.3273	0.3405	0.3550	0.3696	0.3818	0.3906	0.3956	0.3973
0.46	0.3132	0.3261	0.3380	0.3515	0.3650	0.3763	0.3845	0.3897	0.3922
0.47	0.3120	0.3241	0.3368	0.3499	0.3617	0.3714	0.3783	0.3814	0.3830
0.48	0.3116	0.3235	0.3363	0.3490	0.3614	0.3718	0.3801	0.3853	0.3873
0.49	0.3113	0.3236	0.3366	0.3502	0.3623	0.3729	0.3802	0.3846	0.3863
0.50	0.3126	0.3234	0.3351	0.3474	0.3591	0.3700	0.3783	0.3828	0.3849

Table 6.10: The effect of WMF on Gaussian noise ($M = 0.05$)

	Weight coefficient								
V	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.7890	0.7863	0.7850	0.7866	0.7907	0.7948	0.7958	0.7939	0.7881
0.02	0.7055	0.7059	0.7095	0.7176	0.7273	0.7359	0.7404	0.7394	0.7347
0.03	0.6483	0.6511	0.6576	0.6685	0.6815	0.6923	0.6994	0.7013	0.6979
0.04	0.6013	0.6055	0.6154	0.6306	0.6459	0.6583	0.6674	0.6696	0.6670
0.05	0.5702	0.5763	0.5870	0.6033	0.6206	0.6351	0.6448	0.6494	0.6478
0.06	0.5393	0.5462	0.5586	0.5758	0.5939	0.6084	0.6176	0.6212	0.6198
0.07	0.5195	0.5282	0.5416	0.5594	0.5774	0.5924	0.6030	0.6082	0.6067
0.08	0.5045	0.5133	0.5278	0.5468	0.5654	0.5802	0.5897	0.5944	0.5941
0.09	0.4769	0.4866	0.5014	0.5221	0.5428	0.5580	0.5689	0.5732	0.5727
0.10	0.4661	0.4756	0.4897	0.5101	0.5307	0.5466	0.5570	0.5619	0.5617
0.11	0.4490	0.4592	0.4743	0.4939	0.5143	0.5319	0.5440	0.5488	0.5495
0.12	0.4427	0.4527	0.4690	0.4896	0.5094	0.5249	0.5347	0.5404	0.5429
0.13	0.4310	0.4409	0.4564	0.4767	0.4978	0.5147	0.5258	0.5316	0.5324
0.14	0.4209	0.4315	0.4469	0.4662	0.4871	0.5048	0.5167	0.5233	0.5243
0.15	0.4144	0.4250	0.4405	0.4613	0.4821	0.4995	0.5129	0.5205	0.5234
0.16	0.4029	0.4138	0.4292	0.4492	0.4698	0.4860	0.4974	0.5039	0.5075
0.17	0.3951	0.4062	0.4217	0.4420	0.4616	0.4767	0.4867	0.4928	0.4945
0.18	0.3890	0.3989	0.4138	0.4328	0.4511	0.4659	0.4770	0.4841	0.4881
0.19	0.3875	0.3984	0.4130	0.4320	0.4516	0.4681	0.4799	0.4858	0.4881
0.20	0.3812	0.3926	0.4085	0.4272	0.4453	0.4611	0.4725	0.4794	0.4827
0.21	0.3747	0.3845	0.3986	0.4165	0.4347	0.4504	0.4606	0.4670	0.4701
0.22	0.3677	0.3787	0.3937	0.4118	0.4301	0.4449	0.4551	0.4611	0.4639
0.23	0.3643	0.3762	0.3906	0.4077	0.4248	0.4396	0.4508	0.4581	0.4617
0.24	0.3613	0.3720	0.3865	0.4040	0.4211	0.4355	0.4466	0.4537	0.4570
0.25	0.3533	0.3644	0.3781	0.3944	0.4120	0.4265	0.4369	0.4443	0.4481
0.26	0.3525	0.3636	0.3776	0.3952	0.4129	0.4282	0.4402	0.4488	0.4532
0.27	0.3529	0.3630	0.3763	0.3927	0.4104	0.4247	0.4358	0.4432	0.4469
0.28	0.3464	0.3570	0.3697	0.3852	0.4018	0.4155	0.4267	0.4345	0.4389
0.29	0.3428	0.3534	0.3669	0.3818	0.3973	0.4107	0.4209	0.4285	0.4328
0.30	0.3419	0.3522	0.3653	0.3808	0.3971	0.4102	0.4199	0.4263	0.4290
0.31	0.3386	0.3501	0.3641	0.3792	0.3952	0.4085	0.4184	0.4242	0.4263
0.32	0.3345	0.3448	0.3573	0.3721	0.3862	0.3978	0.4056	0.4107	0.4136
0.33	0.3348	0.3448	0.3569	0.3708	0.3851	0.3985	0.4082	0.4148	0.4183
0.34	0.3349	0.3452	0.3576	0.3717	0.3864	0.3991	0.4089	0.4156	0.4200
0.35	0.3304	0.3403	0.3517	0.3652	0.3790	0.3920	0.4024	0.4096	0.4134
0.36	0.3273	0.3384	0.3508	0.3653	0.3801	0.3928	0.4026	0.4095	0.4121
0.37	0.3240	0.3353	0.3483	0.3631	0.3763	0.3878	0.3964	0.4014	0.4037
0.38	0.3230	0.3336	0.3460	0.3590	0.3717	0.3832	0.3920	0.3976	0.4016
0.39	0.3219	0.3324	0.3443	0.3570	0.3697	0.3812	0.3901	0.3954	0.3978
0.40	0.3228	0.3338	0.3461	0.3598	0.3731	0.3846	0.3935	0.3989	0.4025
0.41	0.3192	0.3299	0.3419	0.3552	0.3693	0.3816	0.3908	0.3966	0.3993
0.42	0.3173	0.3271	0.3382	0.3510	0.3640	0.3742	0.3812	0.3871	0.3912
0.43	0.3179	0.3274	0.3386	0.3510	0.3643	0.3763	0.3857	0.3917	0.3944
0.44	0.3112	0.3222	0.3337	0.3450	0.3560	0.3655	0.3728	0.3776	0.3804
0.45	0.3174	0.3280	0.3400	0.3533	0.3659	0.3767	0.3844	0.3892	0.3906
0.46	0.3149	0.3250	0.3362	0.3473	0.3583	0.3675	0.3747	0.3791	0.3810
0.47	0.3162	0.3258	0.3374	0.3498	0.3615	0.3715	0.3789	0.3844	0.3863
0.48	0.3073	0.3176	0.3288	0.3399	0.3506	0.3598	0.3670	0.3711	0.3729
0.49	0.3092	0.3195	0.3311	0.3425	0.3539	0.3635	0.3712	0.3763	0.3786
0.50	0.3062	0.3162	0.3273	0.3389	0.3501	0.3602	0.3677	0.3728	0.3740

Table 6.11: The effect of WMF on Gaussian noise ($M = 0.1$)

	Weight coefficient								
V	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.7575	0.7526	0.7470	0.7427	0.7416	0.7417	0.7417	0.7391	0.7334
0.02	0.6791	0.6765	0.6740	0.6748	0.6788	0.6842	0.6876	0.6878	0.6843
0.03	0.6258	0.6252	0.6259	0.6309	0.6394	0.6487	0.6549	0.6574	0.6546
0.04	0.5871	0.5878	0.5907	0.5991	0.6105	0.6218	0.6295	0.6329	0.6323
0.05	0.5567	0.5586	0.5634	0.5735	0.5865	0.5993	0.6076	0.6108	0.6089
0.06	0.5294	0.5323	0.5391	0.5508	0.5662	0.5801	0.5897	0.5943	0.5933
0.07	0.5036	0.5073	0.5155	0.5300	0.5464	0.5621	0.5718	0.5759	0.5754
0.08	0.4831	0.4879	0.4964	0.5107	0.5281	0.5436	0.5541	0.5596	0.5601
0.09	0.4682	0.4738	0.4840	0.4995	0.5173	0.5332	0.5446	0.5511	0.5540
0.10	0.4546	0.4603	0.4704	0.4868	0.5039	0.5197	0.5309	0.5372	0.5387
0.11	0.4430	0.4489	0.4596	0.4756	0.4944	0.5106	0.5214	0.5275	0.5292
0.12	0.4301	0.4361	0.4458	0.4618	0.4809	0.4968	0.5074	0.5131	0.5150
0.13	0.4238	0.4301	0.4393	0.4531	0.4700	0.4857	0.4958	0.5023	0.5049
0.14	0.4107	0.4168	0.4277	0.4435	0.4615	0.4772	0.4884	0.4956	0.4985
0.15	0.4050	0.4121	0.4223	0.4368	0.4543	0.4700	0.4815	0.4888	0.4919
0.16	0.3979	0.4049	0.4153	0.4302	0.4472	0.4624	0.4740	0.4803	0.4837
0.17	0.3918	0.3990	0.4092	0.4249	0.4422	0.4575	0.4691	0.4769	0.4804
0.18	0.3786	0.3861	0.3965	0.4109	0.4280	0.4431	0.4543	0.4620	0.4655
0.19	0.3780	0.3855	0.3963	0.4113	0.4273	0.4416	0.4525	0.4606	0.4644
0.20	0.3755	0.3826	0.3935	0.4087	0.4254	0.4400	0.4516	0.4586	0.4616
0.21	0.3677	0.3754	0.3861	0.4008	0.4170	0.4321	0.4439	0.4519	0.4559
0.22	0.3625	0.3705	0.3807	0.3946	0.4102	0.4251	0.4368	0.4446	0.4485
0.23	0.3566	0.3650	0.3761	0.3907	0.4064	0.4205	0.4312	0.4392	0.4428
0.24	0.3544	0.3621	0.3721	0.3849	0.3994	0.4138	0.4260	0.4342	0.4394
0.25	0.3506	0.3583	0.3681	0.3815	0.3957	0.4083	0.4182	0.4248	0.4283
0.26	0.3512	0.3599	0.3709	0.3851	0.4008	0.4156	0.4273	0.4345	0.4382
0.27	0.3438	0.3528	0.3628	0.3753	0.3887	0.4016	0.4112	0.4182	0.4227
0.28	0.3437	0.3510	0.3599	0.3715	0.3847	0.3978	0.4094	0.4185	0.4232
0.29	0.3434	0.3515	0.3620	0.3751	0.3894	0.4029	0.4128	0.4200	0.4241
0.30	0.3397	0.3490	0.3607	0.3738	0.3878	0.3998	0.4094	0.4168	0.4215
0.31	0.3346	0.3426	0.3528	0.3655	0.3793	0.3925	0.4038	0.4115	0.4157
0.32	0.3308	0.3397	0.3500	0.3633	0.3770	0.3895	0.3990	0.4049	0.4078
0.33	0.3321	0.3403	0.3491	0.3608	0.3739	0.3862	0.3967	0.4046	0.4095
0.34	0.3284	0.3365	0.3457	0.3567	0.3689	0.3805	0.3893	0.3945	0.3971
0.35	0.3281	0.3360	0.3455	0.3568	0.3694	0.3810	0.3904	0.3957	0.3986
0.36	0.3233	0.3322	0.3422	0.3530	0.3647	0.3754	0.3842	0.3899	0.3917
0.37	0.3245	0.3329	0.3423	0.3537	0.3656	0.3761	0.3849	0.3908	0.3947
0.38	0.3215	0.3297	0.3391	0.3497	0.3611	0.3709	0.3788	0.3847	0.3877
0.39	0.3212	0.3301	0.3406	0.3520	0.3636	0.3750	0.3838	0.3897	0.3929
0.40	0.3166	0.3247	0.3336	0.3445	0.3561	0.3666	0.3753	0.3813	0.3842
0.41	0.3175	0.3271	0.3369	0.3480	0.3592	0.3690	0.3768	0.3814	0.3838
0.42	0.3139	0.3223	0.3318	0.3423	0.3533	0.3630	0.3712	0.3762	0.3784
0.43	0.3139	0.3229	0.3317	0.3414	0.3515	0.3611	0.3683	0.3728	0.3744
0.44	0.3095	0.3182	0.3274	0.3377	0.3479	0.3580	0.3658	0.3713	0.3749
0.45	0.3110	0.3193	0.3285	0.3385	0.3484	0.3575	0.3644	0.3689	0.3716
0.46	0.3107	0.3192	0.3281	0.3379	0.3478	0.3561	0.3624	0.3666	0.3689
0.47	0.3093	0.3181	0.3277	0.3377	0.3479	0.3568	0.3637	0.3678	0.3698
0.48	0.3081	0.3168	0.3261	0.3353	0.3447	0.3534	0.3604	0.3651	0.3669
0.49	0.3074	0.3159	0.3249	0.3347	0.3446	0.3537	0.3614	0.3667	0.3699
0.50	0.3045	0.3135	0.3228	0.3328	0.3424	0.3514	0.3589	0.3635	0.3655

Table 6.12: The effect of WMF on Gaussian noise ($M = 0.15$)

	Weight coefficient								
V	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
0.01	0.7213	0.7159	0.7091	0.7018	0.6963	0.6926	0.6914	0.6901	0.6865
0.02	0.6529	0.6493	0.6447	0.6418	0.6424	0.6453	0.6484	0.6495	0.6479
0.03	0.6016	0.5993	0.5968	0.5973	0.6011	0.6078	0.6145	0.6181	0.6164
0.04	0.5702	0.5689	0.5684	0.5711	0.5778	0.5867	0.5938	0.5969	0.5963
0.05	0.5406	0.5405	0.5413	0.5474	0.5573	0.5678	0.5760	0.5816	0.5817
0.06	0.5138	0.5144	0.5163	0.5232	0.5344	0.5457	0.5554	0.5613	0.5621
0.07	0.4943	0.4957	0.4985	0.5073	0.5200	0.5332	0.5433	0.5498	0.5517
0.08	0.4750	0.4768	0.4811	0.4900	0.5029	0.5162	0.5265	0.5329	0.5354
0.09	0.4580	0.4605	0.4652	0.4753	0.4905	0.5055	0.5173	0.5242	0.5264
0.10	0.4476	0.4500	0.4543	0.4643	0.4786	0.4935	0.5051	0.5124	0.5160
0.11	0.4300	0.4330	0.4384	0.4486	0.4632	0.4779	0.4902	0.4991	0.5031
0.12	0.4232	0.4273	0.4341	0.4463	0.4616	0.4761	0.4862	0.4924	0.4948
0.13	0.4163	0.4199	0.4253	0.4363	0.4513	0.4659	0.4781	0.4870	0.4923
0.14	0.4027	0.4072	0.4142	0.4267	0.4422	0.4577	0.4712	0.4785	0.4829
0.15	0.3952	0.4002	0.4074	0.4189	0.4342	0.4489	0.4611	0.4694	0.4747
0.16	0.3887	0.3932	0.4000	0.4121	0.4263	0.4403	0.4510	0.4571	0.4611
0.17	0.3804	0.3850	0.3919	0.4033	0.4168	0.4302	0.4416	0.4495	0.4551
0.18	0.3778	0.3826	0.3895	0.4004	0.4145	0.4291	0.4415	0.4499	0.4539
0.19	0.3743	0.3793	0.3865	0.3975	0.4116	0.4255	0.4365	0.4432	0.4463
0.20	0.3677	0.3735	0.3813	0.3932	0.4079	0.4223	0.4341	0.4424	0.4467
0.21	0.3616	0.3666	0.3731	0.3836	0.3969	0.4107	0.4230	0.4318	0.4367
0.22	0.3556	0.3611	0.3688	0.3797	0.3932	0.4062	0.4167	0.4247	0.4296
0.23	0.3561	0.3617	0.3682	0.3784	0.3908	0.4036	0.4147	0.4227	0.4274
0.24	0.3553	0.3609	0.3678	0.3776	0.3909	0.4045	0.4156	0.4244	0.4296
0.25	0.3481	0.3545	0.3620	0.3725	0.3855	0.3979	0.4081	0.4156	0.4210
0.26	0.3488	0.3546	0.3620	0.3732	0.3869	0.3996	0.4100	0.4177	0.4228
0.27	0.3416	0.3480	0.3555	0.3652	0.3773	0.3892	0.3994	0.4066	0.4111
0.28	0.3393	0.3453	0.3530	0.3627	0.3745	0.3854	0.3955	0.4022	0.4065
0.29	0.3380	0.3439	0.3511	0.3608	0.3727	0.3841	0.3939	0.4015	0.4059
0.30	0.3361	0.3420	0.3493	0.3590	0.3709	0.3827	0.3935	0.4012	0.4056
0.31	0.3332	0.3395	0.3469	0.3561	0.3670	0.3775	0.3863	0.3924	0.3963
0.32	0.3269	0.3336	0.3414	0.3507	0.3608	0.3707	0.3796	0.3854	0.3885
0.33	0.3279	0.3343	0.3409	0.3494	0.3597	0.3701	0.3788	0.3846	0.3881
0.34	0.3243	0.3304	0.3372	0.3456	0.3557	0.3659	0.3747	0.3818	0.3862
0.35	0.3267	0.3333	0.3404	0.3496	0.3601	0.3711	0.3805	0.3871	0.3913
0.36	0.3215	0.3284	0.3358	0.3447	0.3541	0.3641	0.3734	0.3806	0.3850
0.37	0.3223	0.3301	0.3383	0.3476	0.3576	0.3680	0.3766	0.3824	0.3860
0.38	0.3188	0.3255	0.3328	0.3422	0.3516	0.3611	0.3692	0.3751	0.3785
0.39	0.3160	0.3223	0.3295	0.3382	0.3471	0.3554	0.3628	0.3685	0.3712
0.40	0.3150	0.3221	0.3297	0.3382	0.3474	0.3567	0.3641	0.3691	0.3721
0.41	0.3123	0.3192	0.3266	0.3351	0.3444	0.3533	0.3606	0.3653	0.3683
0.42	0.3104	0.3180	0.3254	0.3338	0.3425	0.3509	0.3578	0.3625	0.3644
0.43	0.3113	0.3188	0.3263	0.3347	0.3445	0.3534	0.3610	0.3666	0.3700
0.44	0.3108	0.3176	0.3244	0.3317	0.3403	0.3491	0.3566	0.3608	0.3624
0.45	0.3105	0.3176	0.3247	0.3320	0.3402	0.3481	0.3541	0.3583	0.3601
0.46	0.3108	0.3180	0.3250	0.3328	0.3413	0.3488	0.3555	0.3593	0.3613
0.47	0.3061	0.3136	0.3205	0.3277	0.3356	0.3438	0.3512	0.3564	0.3588
0.48	0.3064	0.3140	0.3210	0.3287	0.3370	0.3449	0.3510	0.3553	0.3578
0.49	0.3041	0.3113	0.3181	0.3251	0.3328	0.3401	0.3462	0.3506	0.3525
0.50	0.3045	0.3122	0.3203	0.3287	0.3371	0.3445	0.3501	0.3532	0.3544

Chapter 7

Fingerprint classification

7.1 Introduction

Previous chapters have all demonstrated the efficiency of MF. This chapter introduces the application of MF in fingerprint processing as a pre-processor for the first time.

The quest for reliable personal identification in computerised access control has resulted in an increasing interest in biometrics [48]. Biometrics indicates physical or behavioural characteristics which uniquely identifies people. Fingerprint is one of the important biometrics. Secure personal identification is required in many other areas, except forensic use, such as secure systems, banking systems, replacing PIN codes and etc. [86]. All these areas need secure and fast algorithms for identification. Fingerprint processing includes operations such as enhancing contrast of the ridges, segmenting the image to separate ridges from the background and extracting structural features of the image [55]. Identification of fingerprint patterns has always been of scientists' favourites and also important for law enforcement authorities [57]. The fast-growing number of fingerprints gathered and the requirement for a fast recognition system have tremendously increased the demand for an automatic

fingerprint system [57]. An automatic system for fingerprint recognition should satisfy the following criteria [57]: either the system must match two fingerprint impressions and conclude that the two patterns are exactly the same, or it should provide a technique which can significantly improve the present cumbersome manual matching process.

A fingerprint presents a directional image consisting of many ridges at different directions. Its structural information lies in the position and direction of its constituent ridges [41]. The characteristics or features, which identifies each fingerprint uniquely, are called minutiae [57]. Minutiae are interruptions of normal flow of the ridges such as abrupt ridge ending, dots, short ridge, logs, branches, and mergers. The minutiae and their relative locations are so crucial that although each fingerprint pattern has about 100 minutiae, as few as a dozen is considered sufficient to identify a pattern [15, 86].

The basic idea in fingerprint processing is thus to detect local ridges orientations. This can be done by directional filtering. Experimental results show that a better performance can be achieved if directional operators are employed instead of classical methods used for image enhancement and segmentation which involve homogeneous operators [34]. Maio et. al. [51] suggest a direct gray-scale minutiae detection in fingerprints without thinning and binarization. Most of the methods presented for fingerprint processing are efficient only when the quality of images are fairly good which are not always the case, especially when images are acquired from on-paper fingerprints [41]. Rao et. al. [63] focuses on finding the core points in fingerprint. A syntactic approach for fingerprint impression has been proposed in [32] by concentrating on topological representation of the patterns. Ratha et. al. [65] presents a real-time matching system for large fingerprint databases. Kouta et. al. [45] present a graph-based structure to capture the topological relations within the fingerprint. Another syntactic method based on the ridge flow in the pattern area and the presence of deltas in the fingerprint can be found in [57]. The

main problem with this method happens when deltas are not detected at first level. A hierarchial fuzzy approach for fingerprint processing is presented in [41]. It looks very satisfactory when applying a mask with different sizes. However it can not be applied robustly when the ratio of signal/noise is low. The best solution seems to be obtained from a kind of filtering that removes noise without affecting the details of the ridges. By experience we recommend morphological filtering as a pre-processor [71]. With fingerprint pre-processing and directional filtering, fingerprint classification is undertaken based on pattern conversion to strings of symbols. Our algorithm extends the conversion approach proposed in [64] to improve the quality of fingerprint classification.

This chapter presents a syntactic approach to fingerprint classification based on the ridge flow of fingerprint impression, rather similar to the method proposed in [64]. It improves different parts of the algorithm which will be mentioned later. The approach is composed of the raw image pre-processing using the morphological erosion, background normalisation, directional filtering adopted to find the dominant directions of the ridges for each subpicture, pattern trace and conversion to the strings of symbols [71]. The procedure of implementation is presented in detail. Figure 7.1 shows the block diagram applied for fingerprint classification.

7.2 Fingerprint pre-processing

Sherlock et. al. [82] apply position-dependant Fourier domain filters to produce a directionally smoothed image which is then thresholded, yielding the enhanced image leading to a frequency-domain analysis rather than spatially, over the entire image rather than within small blocks. Their algorithm is based on linear filtering and takes the advantage of linear filters. We have previously discussed about the differences between linear filters and MFs.

Morphological operators are faster than convolution methods and avoid

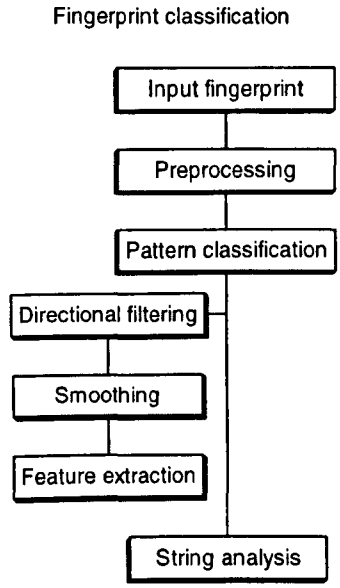


Figure 7.1: Block diagram of fingerprint classification.

blurring the edges of an image when processed. These two key factors have made MF as a popular nonlinear processing tool. Applying MFs for removing noise from fingerprint shows promising results [71]. Experiments show that erosion is one of the best operators among other morphological operators for such a task.

The input data (Figure 7.2), denoted $IN(m,n)$ is eroded with a flat square structuring element $B(m,n)$ of size 3×3 and then subtracted from the input data generating a clear image of the important edges related to the ridges of fingerprint denoted $OUT_1(m,n)$:

$$OUT_1(m,n) = IN(m,n) - (IN(m,n) \ominus B(m,n)). \tag{7.2.1}$$

A thresholding is also introduced to normalise the background producing a clearer image (Figure 7.3). Thresholding is necessary to improve accuracy [82]. There is no specific pre-processing in [64]. Modifying the size of the filter mask is considered for noise reduction in [41]. However we apply the above algorithm as a very effective pre-processor.

Figure 7.2: Input fingerprint.

Figure 7.3: Pre-processing of input fingerprint with MF.

7.3 Pattern classification

In this section we explain the terms, used frequently, in fingerprint classification.

1. *Delta:* The delta point is the point on the ridge closest to the divergence centre of the ridge flow.

2. *Loop:* A loop is that type of fingerprint in which one or more of the ridges enter on either side of the impression, recurve, and terminate on the same side of the impression. Loops whose ridges flow towards the thumb are called radial loops or simply loops, and those whose ridges flow in the direction towards the little finger are called ulnar loops.

3. *Whorl:* A whorl is any pattern with two deltas and at least one recurving ridge which may be a spiral or any variation of a circle.

4. *Twin loop:* The double (twin) loop pattern consists of two deltas and two separate loops, with separate distinct shoulders.

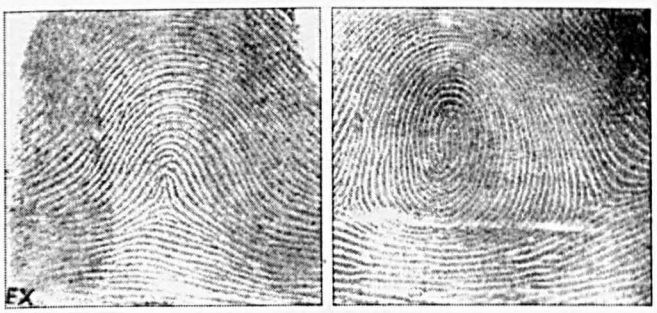
5. *Arch:* An arch is any pattern in which ridges enter on one side, rise in the middle, and flow out on the other side. The tented arch has the same tendency to enter from one side and flow out on the other side, with the exception that the ridges form an upward thrust at the centre.

6. *Transients:* Transients are the prints which do not belong to any of the types mentioned above.

Figure 7.4 shows some fingerprints of different type.

One of the difficulties of the classical classifiers, named "Henry classifiers", is to determine the deltas. That is why Rao et. al. [64] has used the classification based on ridge pattern curves. Based on the direction of the recurving ridge, loops, ulnar loops and twin loops are divided into left and right subclasses. Moreover we divide arch and tented arch in four subclasses based

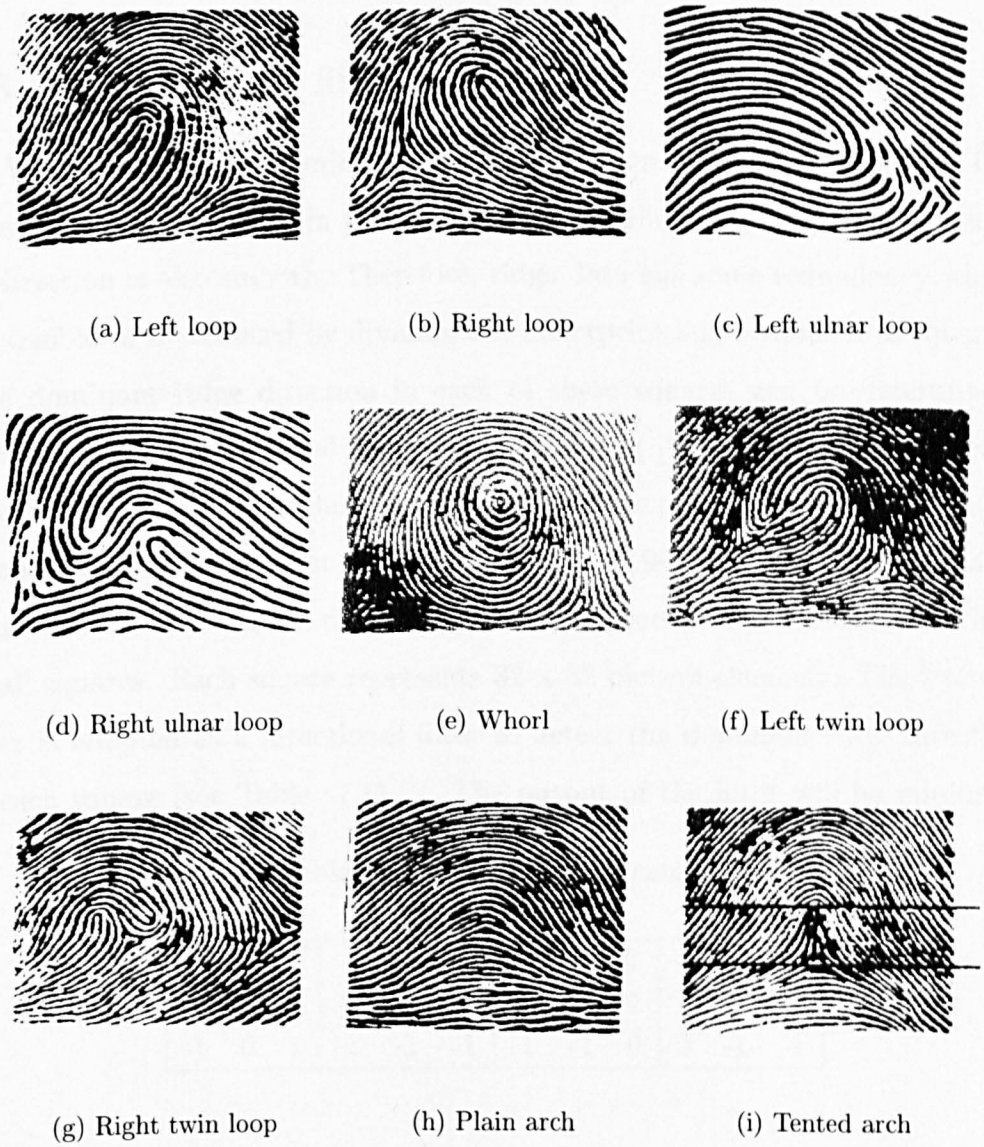
(a) Left loop (b) Right loop (c) Plain arch



(d) Tented arch (e) Whorl

Figure 7.4: Differenet fingerprints.

on the angle as $0°, 90°, 180°, 270°$, thus making sixteen types for classification scheme as follows (there are 10 schemes in [64]): left loop, right loop, left ulnar loop, right ulnar loop, whorl, left twin loop, right twin loop, plain arch (0°), plain arch (90°), plain arch (180°), plain arch (270°), tented arch (0°), tented arch (90°), tented arch (180°), tented arch (270°), transient. Figure 7.5 shows another set of fingerprint of different type as defined above. Pattern



(a) Left loop (b) Right loop (c) Left ulnar loop

(d) Right ulnar loop (e) Whorl (f) Left twin loop

(g) Right twin loop (h) Plain arch (i) Tented arch

Figure 7.5: Another set of different fingerprints [64].

classification involves three parts.

1. Directional filtering.
2. Smoothing.
3. Feature extraction.

Each item is described as follows.

7.3.1 Directional filtering

We use directional filtering as an approximation for the ridge pattern. The reason is that the ridges in a fingerprint are parallel in nature and variations of direction is also smooth. Therefore, ridge data has some redundancy which is possible to be reduced by dividing the fingerprint into a number of squares. The dominant ridge direction in each of these squares can be determined. The matrix, which is called *the sampling matrix* ([26]), can approximate the original print, preserving the global structure of the print. The directional filter applied, detects four major directions: $-45, 0, 45, 90$. It is accurate enough to detect the direction of the ridges [40]. The pre-processed image is divided into small squares. Each square represents 32×32 picture elements. The Prewitt filter is adopted as a directional filter to detect the dominant ridge direction in each square (see Table 7.1). The output of the filter will be minimum

Table 7.1: Directional operators

↔			↕			↗			↘		
-1	0	1	1	1	1	0	1	1	1	1	0
-1	0	1	0	0	0	-1	0	1	1	0	-1
-1	0	1	-1	-1	-1	-1	-1	0	0	-1	-1

for dominant direction, and maximum for the direction perpendicular to the dominant one. If there is no dominant direction for a block, the corresponding

block is left empty. The reason for not having a dominant direction in a window is that either the quality of the image is not good and it has been blurred, or the region, in fact, has not a dominant direction. Leaving a block empty, is preferred to assigning a random direction. Table 7.2 shows an example of a sampling matrix after applying directional filtering.

Table 7.2: Sampling matrix

/	/	/	/	/	-	\	\
/	/	/	/	-	\	\	\
/	/	/	/	\	\	\	\
/	/	|	|	/	|	|	\
/	|	|	|	/	|	|	|
|	|	\	\	/	/	|	
\	\	\	\	-	/	/	/
\	\	\	\	-	/	/	/

We could use Sobel filter instead of Prewitt filter. However our experimental results show better performance with the latter one.

7.3.2 Smoothing

To ease tracing the ridge flow in every part of sampling matrix as well as removing ambiguities, a smoothing process is required. Some ambiguity may occur such as Figure 7.6. It is rather difficult to decide which path (1 or

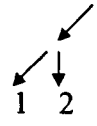


Figure 7.6: Ambiguity in direction.

2) should be followed [64]. A special smoothing technique can remove such

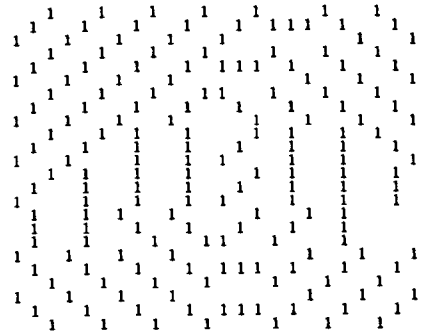
ambiguity. Each of four direction codes in sampling matrix is first converted into a 3×3 matrix (as shown in Table 7.3) leading to an expansion of the size of the sampling matrix (new width and height is three times as big as the previous size).

Table 7.3: Matrix capture

↔			↕			↗			↖		
0	0	0	0	1	0	0	0	1	1	0	0
1	1	1	0	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	1

The result is shown in Table 7.4 as the sampling matrix after expansion and before smoothing.

Table 7.4: Sampling matrix before smoothing.



Then, sliding from top to bottom and from left to right of the sampling matrix, the central point is modified from 0 to 1 if any of the neighbourhood cases 1-4 shown in Table 7.5 exists. Similarly if any of the neighbourhood cases a1-d4 shown in Tables 7.6- 7.9 exists, the central point will be changed from 1 to 0.

The effect of smoothing on the expanded sampling matrix is shown in Table 7.10. Considering the above smoothing method as shown in Tables 7.5 and 7.6, we realise how powerful, our algorithm smoothes the ambiguities compared

Table 7.5: Modifying the central point from $\underline{0}$ to $\underline{1}$

case 1			case 2			case 3			case 4		
0	1	0	x	x	x	x	1	0	0	1	x
1	$\underline{0}$	1	1	$\underline{0}$	1	x	$\underline{0}$	1	1	$\underline{0}$	x
x	x	x	0	1	0	x	1	0	0	1	x

Table 7.6: Modifying the central point from $\underline{1}$ to $\underline{0}$: case a

a1			a2			a3			a4		
x	x	x	1	1	1	1	x	x	x	x	1
x	$\underline{1}$	x	x	$\underline{1}$	x	1	$\underline{1}$	x	x	$\underline{1}$	1
1	1	1	x	x	x	1	x	x	x	x	1

Table 7.7: Modifying the central point from $\underline{1}$ to $\underline{0}$: case b

b1			b2			b3			b4		
x	1	x	x	1	x	x	1	1	x	0	x
0	$\underline{1}$	1	1	$\underline{1}$	0	1	$\underline{1}$	0	1	$\underline{1}$	0
x	0	1	1	0	x	x	0	x	x	1	1

b5			b6			b7			b8		
x	0	1	1	0	x	1	1	x	x	0	x
0	$\underline{1}$	1	1	$\underline{1}$	0	0	$\underline{1}$	1	0	$\underline{1}$	1
x	1	x	x	1	x	x	0	x	1	1	x

Table 7.8: Modifying the central point from $\underline{0}$ to $\underline{1}$: case c

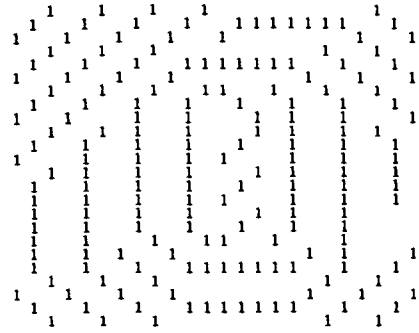
c1					c2					c3					c4				
x	x	x	1	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	1	x	x	x	x	1	x	x	x	x	0	x	1	1	x	0	x	x
x	1	$\underline{1}$	0	x	x	0	$\underline{1}$	1	x	x	0	$\underline{1}$	1	x	x	1	$\underline{1}$	0	x
1	x	0	x	x	x	x	0	x	1	x	x	1	x	x	x	x	1	x	x
x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x	1	x

Table 7.9: Modifying the central point from $\underline{0}$ to $\underline{1}$: case d

d1					d2					d3					d4				
x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	1	x	x
x	1	x	x	x	x	x	x	1	x	x	x	1	x	x	x	x	1	x	x
x	x	$\underline{1}$	1	1	1	1	$\underline{1}$	x	x	x	x	$\underline{1}$	1	1	1	1	$\underline{1}$	x	x
x	x	1	x	x	x	x	1	x	x	x	1	x	x	x	x	x	x	1	x
x	x	1	x	x	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x

with the proposed method in [64] which only covers a few cases of the above-mentioned ambiguities.

Table 7.10: Sampling matrix after smoothing.



7.3.3 Feature extraction

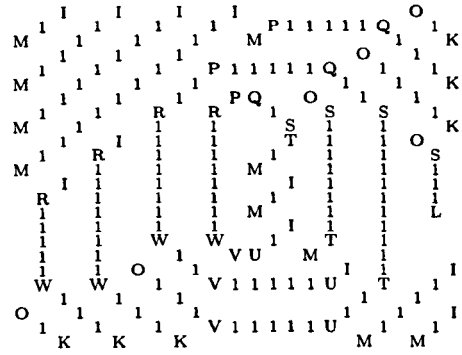
The algorithm applied for feature extraction determines the end points and the points where two lines make an angles [64]. Four different directions make eight possible endpoints, or equivalently starting points, as ($\rightarrow$:N), ($\nearrow$:M), ($\uparrow$:L), ($\nwarrow$:K), ($\leftarrow$:J), ($\swarrow$:I), ($\downarrow$:H), and ($\searrow$:O). A line in one direction can make an angle with another line in six possible ways (e.g. $\dashv$ $\urcorner$ $\lrcorner$ $\urcorner$ $\dashv$ $\urcorner$). The last two possibilities (angle $< 90°$) are eliminated by smoothing. Table 7.11 shows the symbols used for various endpoints and the angle points. We use the similar notation as used in [64] to make the comparison of two methods easier. Table 7.12 displays how the smoothed sampling matrix will

be labelled using these features.

Table 7.11: Features used in classification

→N			←J			↑L			↓H			↗M		
0	0	x	x	0	0	x	1	x	0	0	0	0	x	1
0	1	1	1	1	0	0	1	0	0	1	0	0	1	x
0	0	x	x	0	0	0	0	0	x	1	x	0	0	0
↙I			↖K			↘O			⌟U			└V		
0	0	0	1	x	0	0	0	0	x	0	1	1	0	x
x	1	0	x	1	0	0	1	x	1	1	0	0	1	1
1	x	0	0	0	0	0	x	1	0	0	0	0	0	0
⌐ P			¬ Q			⌋ T			⌊ W			⌋ R		
0	0	0	0	0	0	x	1	0	0	1	x	0	0	1
0	1	1	1	1	0	0	1	0	0	1	0	0	1	0
1	0	x	x	0	1	1	0	0	0	0	1	0	1	x
⌊ S			⌐ A			¬ X			⌟ Y			└ Z		
1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0	1	0	0	1	1	1	1	0	1	1	0	0	1	1
x	1	0	0	1	0	0	1	0	0	0	0	0	0	0

Table 7.12: Sampling matrix with pattern labels



7.4 String analysis

String analysis plays a major role in fingerprint classification. The number of strings belonging to plain arch and tented arch are very limited. In

these cases matching the string with prototype seems to be better than using grammars, but in the case of loops, whorl, and twin loops, the total number of strings is enormous. Therefore a grammar is used to analyse them. As the 1's between the endpoints and angles (in Table 7.12) carry no extra information about the patterns, and they only provide connection between the symbols, they are ignored. As an example, two first strings (left top) in Table 7.12 are as *"I1M"*, and *"I111M"*. They can both be simplified to identical symbol as *"IM"* showing two 45° lines. The strings left represent the characteristic ridge flow pattern and they can be used for classification. The algorithm applied to find the remaining strings is to search the matrix from left to right and top to bottom for endpoints. After an endpoint is found, it should be tracked by picking the connected symbols sequentially until another endpoint is found. We only need to keep one of each type of the strings. Therefore the number of patterns, to be analysed and classified, will be reduced. The following strings are attributed to different ridge patterns:

1. **Plain Arch:**
   - Plain Arch (0°): $MPQK$, $KQPM$.
   - Plain Arch (90°): $IRWK$, $KWRI$.
   - Plain Arch (180°): $OVUI$, $IUVO$.
   - Plain Arch (270°): $OSTM$, $MTSO$.
2. **Tented Arch:**
   - Tented Arch(0°): NYH, HYN, JZH, HZJ.
   - Tented Arch(180°): NXL, LXN, JAL, LAJ.
3. **Right Ulnar Loop:**
   - $KQAZUPQK$, $KQPUZAQK$,

- $KQAZUPQVJ$, $JVQPUZAQK$,
- $JAZUPJ$, $JPUZAJ$,
- $JVQAZUPQVJ$, $JVQPUZAQVJ$.

4. **Left Ulnar Loop:**
   - $NUPQVYXPM$, $MPXYVQPUN$,
   - $NXYVQN$, $NQVYXN$,
   - $MPXYVQPM$, $MPQVYXPM$,
   - $MPQVYXPUN$, $NUPXYVQPM$.

5. **Right Loop:**
   - $KQAWK$, $KWAQK$,
   - $KQAZJ$, $JZAQK$,
   - $JVQAWK$, $KWAQVJ$,
   - $JVQAZJ$, $JZAQVJ$.

6. **Left Loop:**
   - $MPXTM$, $MTXPM$,
   - $NUPXYN$, $NYXPUN$,
   - $MPXYN$, $NYXPM$,
   - $NUPXTM$, $MTXPUN$.

7. **Whorl:** (Φ : is an empty set, and β is any element of the symbolic set)
   $XYZ\alpha\beta$, where $\alpha = (AXYZ)^n$, $n \geq 0$ and $\beta \in \{AXY, AX, A, \Phi\}$.
   $YZA\alpha\beta$, where $\alpha = (XYZA)^n$, $n \geq 0$ and $\beta \in \{XYZ, XY, X, \Phi\}$.
   $ZAX\alpha\beta$, where $\alpha = (YZAX)^n$, $n \geq 0$ and $\beta \in \{YZA, YZ, Y, \Phi\}$.
   $AXY\alpha\beta$, where $\alpha = (ZAXY)^n$, $n \geq 0$ and $\beta \in \{ZAX, ZA, Z, \Phi\}$.

$ZYX\alpha\beta$, where $\alpha = (AZYX)^n$, $n \geq 0$ and $\beta \in \{AZY, AZ, A, \Phi\}$.

$AZY\alpha\beta$, where $\alpha = (XAZY)^n$, $n \geq 0$ and $\beta \in \{XAZ, XA, X, \Phi\}$.

$YXA\alpha\beta$, where $\alpha = (ZYXA)^n$, $n \geq 0$ and $\beta \in \{ZYX, ZY, Z, \Phi\}$.

$XAZ\alpha\beta$, where $\alpha = (YXAZ)^n$, $n \geq 0$ and $\beta \in \{YXA, YX, Y, \Phi\}$.

8. **Left Twin Loop:** (Φ : is an empty set, and α is any element of the symbolic set)

   $\alpha\beta AXZ$, where $\beta = (AXYZ)^n$, $n \geq 0$ and $\alpha \in \{XYZ, YZ, Z, \Phi\}$.

   $\alpha\beta ZXA$, where $\beta = (ZAXY)^n$, $n \geq 0$ and $\alpha \in \{AXY, XY, Y, \Phi\}$.

   $\alpha\beta XZY$, where $\beta = (XYZA)^n$, $n \geq 0$ and $\alpha \in \{YZA, ZA, A, \Phi\}$.

   $\alpha\beta YZX$, where $\beta = (YZAX)^n$, $n \geq 0$ and $\alpha \in \{ZAX, AX, X, \Phi\}$.

9. **Left Loop+Loop3:**

   Loop3:

   - $IRZUI$, $IUZRI$,
   - $IRZYH$, $HYZRI$,
   - $IRZUPJ$, $JPUZRI$,
   - $HTRZUI$, $IUZRTH$,
   - $HTRZYH$, $HYZRTH$,
   - $HTRZUPJ$, $JPUZRTH$,
   - $JAZUI$, $IUZAJ$.

10. **Right Twin Loop:** (Φ : is an empty set, and α is any element of the symbolic set)

    $\alpha\beta XAY$, where $\alpha \in \{AZY, ZY, Y, \Phi\}$ and $\beta = (XAZY)^n$, $n \geq 0$.

    $\alpha\beta YAX$, where $\alpha \in \{XAZ, AZ, Z, \Phi\}$ and $\beta = (YXAZ)^n$, $n \geq 0$.

    $\alpha\beta AYX$, where $\alpha \in \{YZX, ZX, X, \Phi\}$ and $\beta = (AYZX)^n$, $n \geq 0$.

    $\alpha\beta XYA$, where $\alpha \in \{AYZ, YZ, Z, \Phi\}$ and $\beta = (XAYZ)^n$, $n \geq 0$.

11. **Right Loop+Loop4:**

    Loop4:

    - $OVYXN, NXYVO,$
    - $HWSYVQN, NQVYSWH,$
    - $OSYVO, OVYSO,$
    - $OSYZH, HZYSO,$
    - $OSYVQN, NQVYSO,$
    - $HWSYVO, OVYSWH,$
    - $HWSYZH, HZYSWH.$

The total number of the patterns may be reduced by using the following string transformations:

1. $PR \rightarrow A$, $RP \rightarrow A$,
2. $QS \rightarrow X$, $SQ \rightarrow X$,
3. $TU \rightarrow Y$, $UT \rightarrow Y$,
4. $VW \rightarrow Z$, $WV \rightarrow Z$.

Figure 7.7 shows the resulted symbols after conversion.

In the case of three right angles (or more) for a string, the following modifications are also used (it happens for whorl, twin loops):

i. Leading/lagging lines before/after A, X, Y, Z (i.e. L, H, J, N) are ignored (see Figure 7.8).

ii. IR, RI, LR, RL and JP, PJ, MP, PM are modified to A (Figure 7.9-line 1).

iii. LS, SL, OS, SO and KQ, QK, NQ, QN are changed to X (Figure 7.9-line 2).

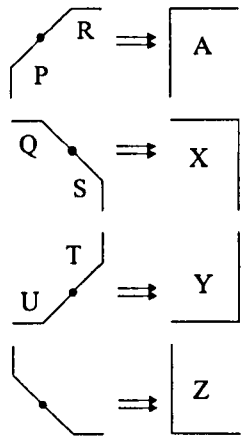


Figure 7.7: Some transformations required for simplification.

iv. HT, TH, MT, TM and IU, UI, NU, UN are modified to Y (Figure 7.9-line 3).

v. KW, WK, HW, WH and JV, VJ, OV, VO are changed to Z (Figure 7.9-line 4).

vi. UP, PU, QV, VQ, WS, SW, TR, RT are ignored (Figure 7.9-line 5).

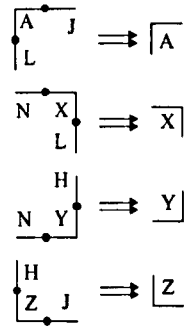


Figure 7.8: Redundant features in whorl and twin loops.

The grammar applied is shown in Tables 7.13 and 7.14. If the patterns in the second column of Table 7.14 match with the pattern of the string,

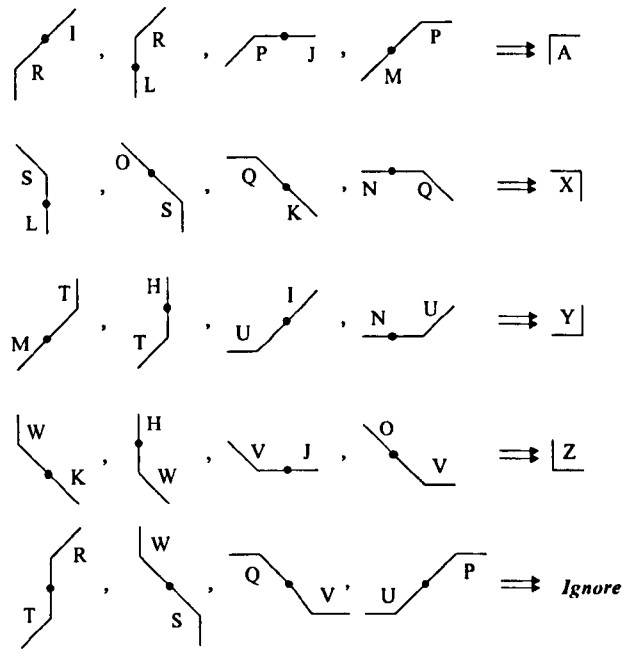


Figure 7.9: Some simplifications for whorl and twin loops.

the third column will be the next location of the table to be compared with next pattern of the string as a confirmation of matching, otherwise the current pattern of the string will be compared with the pattern of the location defined by the fourth column jump. If the third column jump equals to -1, it means the search should be started from the first for the rest of the string. The fifth column indicates the type of the string if end of string. Similar discussion holds for the fourth column if it equals to -1. Depending on the parsing path, the strings can be classified into one of the classes as defined before. For sampling matrix shown in Table 7.12, the strings and the paths they are classified, are as follows:

- *IM*, *OK*: Line.
- *MPQK*: Plain Arch.
- *OSTM*: Plain Arch(270°).

- $IRWK$: Plain Arch(90°).
- OSL: Angular Lines.
- $MTSQPRWVUI$ ($\rightarrow YXAZY$): the following lines of the grammar in Table 7.14 will be examined to detect the pattern:
  1, 11, 24, 34, 46, 59, 78, 81, 82, 83, 91, 111, 112, 110, 109, 111.
  The result is a Whorl.
  $OVUTSQPRWVUI$ ($\rightarrow ZYXAZY$): similarly the following lines will be checked: 1, 11, 24, 34, 46, 59, 78, 81, 82, 83, 84, 95, 103, 104, 102, 101, 103, 104. The result is again a Whorl.

We notice that the results of classification agree with the sampling matrix.

Table 7.13: Grammar syntax for line, arch, and tented arch detection

No.	pt	go	fail	Type	No.	pt	go	fail	Type	No.	pt	go	fail	Type
1	H	2	3		28	O	-1	-1	AL	55	W	56	-1	
2	L	-1	17	Line	29	X	30	31		56	H	-1	71	AL
3	L	4	5		30	N	-1	-1	TA180	57	V	58	59	
4	H	-1	25	Line	31	A	32	-1		58	J	-1	73	AL
5	N	6	7		32	J	-1	-1	TA180	59	S	60	-1	
6	J	-1	33	Line	33	Y	34	35		60	L	-1	77	AL
7	J	8	9		34	H	-1	-1	TA	61	P	62	63	
8	N	-1	41	Line	35	X	36	37		62	J	-1	65	AL
9	I	10	11		36	L	-1	-1	TA	63	T	64	-1	
10	M	-1	49	Line	37	U	38	39		64	H	-1	79	AL
11	M	12	13		38	I	-1	-1	AL	65	Q	66	-1	
12	I	-1	61	Line	39	Q	40	-1		66	K	-1	-1	PA
13	O	14	15		40	K	-1	-1	AL	67	P	68	-1	
14	K	-1	57	Line	41	Z	42	43		68	M	-1	-1	PA
15	K	16	-1		42	H	-1	-1	TA	69	W	70	-1	
16	O	-1	53	Line	43	A	44	45		70	K	-1	-1	PA90
17	W	18	19		44	L	-1	-1	TA180	71	R	72	-1	
18	K	-1	-1	AL	45	V	46	47		72	I	-1	-1	PA90
19	T	20	21		46	O	-1	-1	AL	73	U	74	-1	
20	M	-1	-1	AL	47	P	48	-1		74	I	-1	-1	PA180
21	Y	22	23		48	M	-1	-1	AL	75	V	76	-1	
22	N	-1	-1	TA	49	U	50	51		76	O	-1	-1	PA180
23	Z	24	-1		50	N	-1	75	AL	77	T	78	-1	
24	J	-1	-1	TA	51	R	52	-1		78	M	-1	-1	PA270
25	R	26	27		52	L	-1	69	AL	79	S	80	-1	
26	I	-1	-1	AL	53	Q	54	55		80	O	-1	-1	PA270
27	S	28	29		54	N	-1	67	AL					

AL:Angular Line, TA:Tented Arch, TA180:Tented Arch (180°)
PA:Plain Arch, PA90:Plain Arch (90°), PA180:Plain Arch (180°), PA270: Plain Arch (270°)

Table 7.14: Grammar syntax for the rest of patterns

No.	pt	go	fail	Type	No.	pt	go	fail	Type	No.	pt	go	fail	Type
1	K	2	11		44	X	39	76		87	Y	88	115	Whorl
2	Q	4	3		45	M	-1	-1	LUL	88	Z	86	127	Whorl
3	W	4	-1		46	H	47	59		89	Y	91	117	Whorl
4	A	5	20		47	T	48	56		90	X	89	-1	Whorl
5	Q	6	9		48	R	49	60		91	Z	92	111	Whorl
6	K	-1	7	RL	49	Z	50	-1		92	A	90	118	Whorl
7	V	-1	-1		50	Y	51	52		93	Z	95	-1	Whorl
8	J	-1	14	RL	51	H	-1	-1	Loop3	94	Y	93	117	Whorl
9	W	6	10		52	U	53	57		95	A	96	103	Whorl
10	Z	8	-1		53	I	-1	54	Loop3	96	X	94	-1	Whorl
11	J	12	24		54	P	55	58		97	A	99	118	Whorl
12	Z	4	13		55	J	-1	-1	Loop3	98	Z	97	-1	Whorl
13	V	2	4		56	Y	49	65		99	X	100	107	Whorl
14	U	15	-1		57	R	53	61		100	Y	98	117	Whorl
15	P	16	53		58	T	51	-1		101	Z	103	121	Whorl
16	Q	17	18		59	I	48	78		102	A	101	-1	Whorl
17	V	18	19		60	U	49	-1		103	Y	104	118	Whorl
18	J	-1	-1	RUL	61	A	55	-1		104	X	102	122	Whorl
19	K	-1	-1	RUL	62		-1	-1		105	A	107	-1	Whorl
20	P	21	-1		63	P	49	-1		106	X	105	122	Whorl
21	U	22	-1		64	A	49	-1		107	Z	108	121	Whorl
22	Z	23	-1		65	Z	75	66		108	Y	106	-1	Whorl
23	A	16	-1		66	W	67	-1		109	Y	111	-1	Whorl
24	M	25	34		67	S	68	-1		110	Z	109	121	Whorl
25	P	26	31		68	Y	69	-1		111	X	112	122	Whorl
26	X	27	42		69	V	70	72		112	A	110	-1	Whorl
27	T	28	29		70	Q	71	74		113	X	115	122	Whorl
28	M	-1	-1	LL	71	N	-1	-1	Loop4	114	Y	113	-1	Whorl
29	Y	30	32		72	Z	73	-1		115	A	116	117	Whorl
30	N	-1	37	LL	73	H	-1	-1	Loop4	116	Z	114	121	Whorl
31	T	26	-1		74	O	-1	-1	Loop4	117	Z	120	-1	LTL
32	P	33	-1		75	Y	76	80		118	X	119	-1	LTL
33	U	30	28		76	S	77	-1		119	A	121	126	LTL
34	N	35	46		77	W	73	74		120	Y	-1	-1	LTL
35	Y	26	36		78	O	79	81		121	Y	123	-1	RTL
36	U	25	26		79	V	75	67		122	A	121	126	RTL
37	V	38	-1		80	X	71	-1		123	Z	124	125	RTL
38	Q	39	43		81	A	99	82	Whorl	124	X	122	-1	RTL
39	P	40	41		82	X	87	83	Whorl	125	X	-1	-1	RTL
40	U	41	45		83	Y	91	84	Whorl	126	Y	127	-1	RTL
41	N	-1	-1	LUL	84	Z	95	-1	Whorl	127	A	-1	-1	RTL
42	Q	37	-1		85	X	87	117	Whorl					
43	Y	44	70		86	A	85	118	Whorl					

RL:Right Loop, LL:Left Loop, RUL:Right Ulnar Loop
LUL:Left Ulnar Loop, LTL:Left Twin Loop, RTL: Right Twin Loop

7.5 Decision

The decision for matching a fingerprint to a similar impression among a large database is a very difficult task, because ([65]):

i. there may be no exact match (yes/no type),

ii. the input image can be different from the stored image in the database even though they represent the same finger,

iii. the input may be noisy and distorted,

iv. the input image may contain only a partial image with severe distortions as in the case of a scene-of-crime fingerprint,

Our algorithm can not however be applied for police investigation, as the size of the database is huge and we need a complete fingerprint. Based on these assumptions the proposed algorithm may be applied for personnel identification in not huge companies.

7.6 Discussion and conclusion

We have introduced the idea of applying MF as a pre-processor for fingerprint processing. The application of MF speeds up pre-processing with a reliable output compared to the existing methods. The proposed algorithm for fingerprint classification extends and improves the approach in [64]. The proposed method in [64] is one of the best syntactic analysis for fingerprint classification. However our approach dominates over [64] as follows. It presents more powerful smoothing procedure, whereas there is no definite pre-processor in latter method. Our algorithm for smoothing the pre-processed data removes all possible ambiguities compared to the latter method. The integrated programmed grammar covers a complete set of patterns, it may detect some patterns such as $NXTM$, $NYXPM$, $NYXPUN$, $KQPRK$ which [64] suggests

but can not detect. Our integrated programmed grammar may be developed for newer features very conveniently, and it is expandable for possible future developments (see line 62 in Table 7.14), while there is no chance to develop the grammar table in [64] without modifying it generally. The algorithm presented in this chapter shows an improvement of fingerprint processing and classification, compared to the existing methods, owing to a powerful morphological pre-processor and robust technique for smoothing and an efficient grammar. It can be implemented to process and match fingerprints with reliability.

Chapter 8

ECG waves detection

8.1 Introduction

This chapter concentrates on one of the applications of morphological filtering (MF) for ECG filtering. One of the major problems in recording ECGs is that the measurement is degraded by additive 50 or 60 Hz power line interference [62]. Existing computerised algorithms of rhythm analysis are not satisfactory for detecting complex atrial arrhythmias in ECG. They can detect normal sinus rhythms and sinus arrhythmias, rather perfectly, where the P waves and QRS complexes exhibit a 1:1 ratio and the PR interval is constant, but their performance is not good in detecting complex arrhythmias [67]. Several methods have been reported in the literature to detect the atrial waves. Most of the methods limit their search to a pre-defined window in front of each QRS. However P waves are not always located before QRS. When the ECG signal is noisy, it is difficult to detect the P waves buried in noise, while their shapes and amplitudes change. In most cases they will not be detected and, instead, some spurious waves will be detected as P waves. Reddy et. al. [67] apply a pre-processor for atrial wave detection. They subtract a median complex of leads II and V1 from the rhythm data of the corresponding leads.

These signals are then low-pass filtered and decimated to 125 samples/sec, differentiated (1st and 2nd), rectified and added, to form a composite detection function. Median values of these measurements are computed and measurements of the candidate P waves are compared for closeness to the median. If they are within pre-defined limits, they are given positive scores representing a likelihood of a true P wave. This method seems to be very efficient for detecting complex arrhythmias. We, in general, have empirically experienced that the proposed method in [67] is the best to detect P waves, compared to other existing methods, and we will apply it on our own way which will be discussed later in this chapter.

An approach to QRS complex detection using MM is presented in [87]. The proposed algorithm leads to an accurate QRS complex detection. A preprocessor using linear filtering is applied to enhance QRS complexes and suppress the other parts of the signal as well as noise. The decision rule operates on this output and classifies the dominant events as QRS complexes or not. We confirm applying MM operators for ECG wave detection. However we do not agree with the idea of suppressing the other parts of the signal such as P, S, T waves. ECG signals are frequently plagued by impulsive noise, e.g., due to muscle activities and power line interference [11]. Impulsive noise has very large positive or negative values of short duration. Moreover, background normalisation is required to correct the baseline drift of the signal caused by the respiration and motion of the subject [2].

We apply MM for noise suppression and background normalisation in ECG signals with similar method as what proposed in [11].

8.2 Definition of the terms in ECG

The electrocardiogram (ECG) is the graphical representation of the potential difference between two points on a body surface, versus time. One of

universally accepted form of ECG is 12-lead ECG. The detailed explanation of other ECG methods and the physics of ECG can be found in literature [31, 24, 18].

The heartbeat usually starts in the sinus node (1 in Fig. 8.1) located in the right atrium (http://www.atlcard.com/hartbeat.html). The sinus node sends an electrical signal throughout the atria (2 in Fig. 8.1) and to the atrioventricular (AV) node (3 in Fig. 8.1). The signal then travels down the special pathways that conduct it to the ventricles (4,5 in Fig. 8.1). As the signal travels through the heart, the heart contracts or beats. Figure 8.2 (http://www.mei.com/resource/arrhythm/welcome.html) shows different parts of the heart.

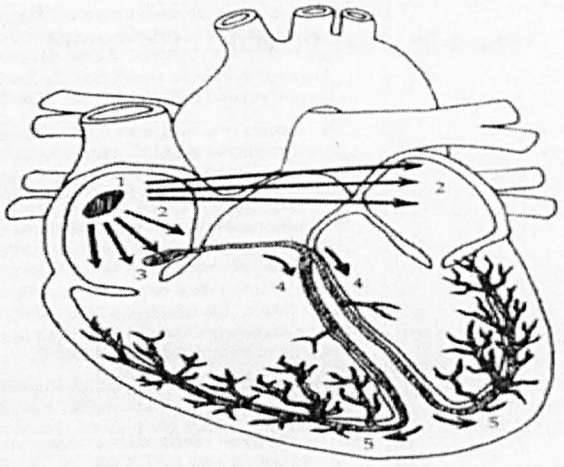


Figure 8.1: Heart beat.

Figure 8.3 (http://www.mei.com/resource/arrhythm/welcome.html) shows a sample ECG wave on ECG sheet.

In this part we will briefly explain the different waves in ECG:

i. *P wave:* The P wave of the ECG represents atrial depolarisation. P wave is best viewd on leads II and V1 and may be upward, downward and or diphasic. Its duration indicates the time required for an impulse

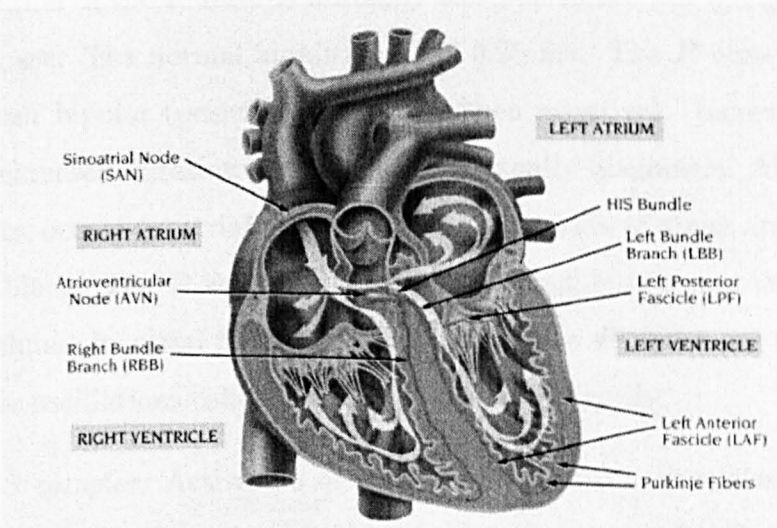


Figure 8.2: Different parts of heart.

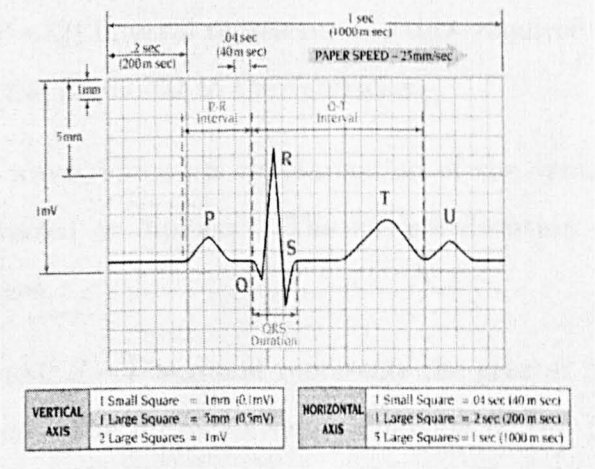


Figure 8.3: A sample ECG wave.

to pass from the SA node to the AV node (atrial conduction time). The duration of the P wave is normally $0.07 - 0.10$ sec but not greater than 0.11 sec. The normal amplitude is ≤ 0.25 mv. The P wave in lead V1 is seen bipolar (positive at first and then negative). Increased voltage or increased duration of the P wave is usually diagnostic. Absence of P waves occurs in atrial standstill, during periods of sinus arrest, and in SA block. The P wave may not be recognisable in some AV junctional rhythms. In atrial flutter and fibrillation, the P waves are replaced by other oscillations called F and f waves, respectively.

ii. *QRS complex:* Activation or depolarisation of the ventricles is reflected by QRS complex. An initial upward and downward deflexion after P wave are called Q and R waves respectively. S wave usually represents the terminal part of ventricular activation. There may be a second R (called R') or S (called S') waves, or only a single negative deflection (called QS) representing the entire QRS complex. The maximum normal duration of the QRS complex is $0.08 - 0.10$ sec.

iii. $P - R$ (or $P - Q$) interval represents the time required for an impulse to travel from the SA nodal to the ventricles.

iv. *T wave:* T wave represents repolarisation of the ventricles. It may be upright, inverted or diphasic. The normal duration of the T wave is $0.10 - 0.25$ sec.

v. *S – T segment:* $S - T$ segment represents the greater part of ventricular repolarisation. The normal average duration of the $S - T$ segment is $0.05 - 0.15$ sec.

vi. *U wave:* U wave is a small rounded deflexion which occurs immediately after the T wave and it is normally in the same direction as the T wave. It is usually best seen in leads V2 to V4. The deflexion may be so small

as to make accurate recognition difficult. The genesis of the U wave is uncertain and remains controversial.

vii. $Q-T$: $Q-T$ interval represents the total duration of the ventricular activity. In some cases it is difficult to measure it. There are special tables to calculate another factor named $Q-T_c$ instead of $Q-T$ [18].

There are a lot of irregularities which make the waves look different from the normal shape. Appendix C introduces some of the arrhythmia including tables and figures as a reference.

8.3 Pre-processing

In this section we apply MM for impulsive noise rejection and baseline drift removal. Noise suppression is typically the first step in ECG signal processing [60]. Linear low-pass and high-pass filtering are commonly used for noise suppression and baseline drift removal respectively [3]. Linear filtering is ineffective in impulsive noise suppression, and in general, regarding our previous discussions of comparison of linear filters versus nonlinear ones, about preserving the shape of the original signal, we reject the idea of applying linear filters for ECG wave pre-processing. There is a report about applying median filters [21] and ranked-ordering methods [90].

The block diagram of the method applied is shown in Figure 8.4. The algorithm applied is similar to the proposed method in [11]. However Trahanias [87] uses open-closing instead of the average of open-closing and close-opening for baseline drift removal. The advantage of the method in [87] is that it is simpler, and therefore, faster than the method in [11], but less efficient. We have empirically proven that the best morphological pre-processor for ECG signal is the one shown in Figure 8.4.

The algorithm applied is as follows:

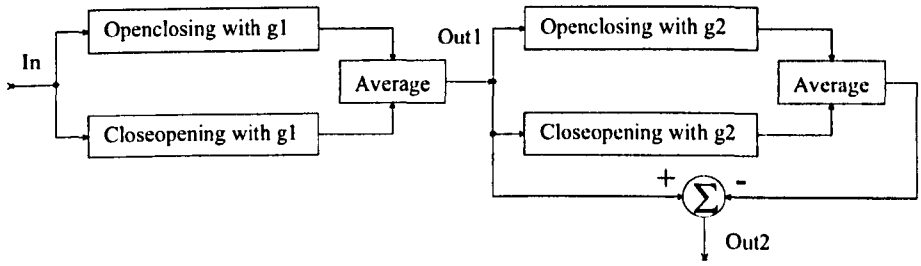


Figure 8.4: Block diagram of ECG pre-processing.

i. A flat SE of size 3 with its origin at its centre (i.e. $g_1 = \{0, \underline{0}, 0\}$) is taken. The length of SE should be smaller than the smallest valuable component in ECG and bigger than the undesired parts such as noise. If the duration of the smallest valuable wave is T sec, and the sampling frequency is f_s Hz, then the corresponding wave will be presented with $T \times f_s$ samples. Therefore the length of the SE should be $< (T \times f_s)/2$. If the sampling frequency is, for example, 200 sps (sample per second), then 3 samples will be equal to $3/200 = 15$ msec. This size is smaller than any sharp wave in ECG and can be applied as the size of SE to remove the impulsive noise from ECG. With g_1 as SE, we apply the following equation to remove the noise:

$$out_1(n) = (opcl(in(n)) + clop(in(n)))/2, \tag{8.3.1}$$

ii. To remove the baseline drift, we need to increase the SE size to the size which is bigger than the size of the biggest ECG desirable component. For instance we use a flat SE of size 51 (g_2) with its origin at its centre. If we do the similar operation on $out_1(n)$ as done on $in(n)$ in Eq. 8.3.1, with g_2 as SE, we will then remove all desirable components from $out_1(n)$, leaving the components related to baseline. If such a result is subtracted from $out_1(n)$, we will, instead, get rid of baseline drift. Therefore the

operation applied, is

$$out_2(n) = out_1(n) - (opcl(out_1(n)) + clop(out_1(n)))/2. \qquad (8.3.2)$$

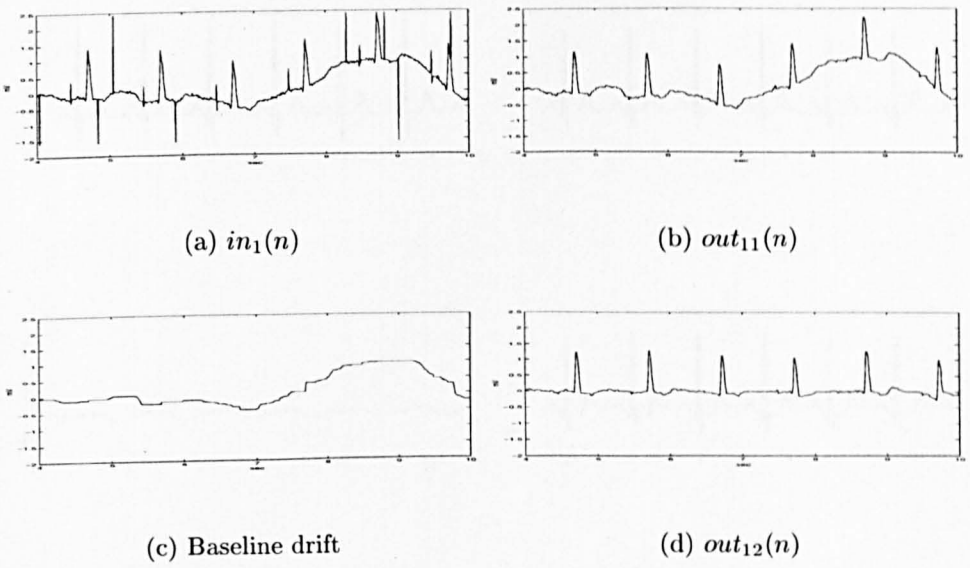
Figure 8.5-a shows an input ECG that has baseline drift and is plagued by noise. Figure 8.5-b shows the result of noise removing. This is considered as $out_1(n)$ in Figure 8.4. The baseline drift detected is shown in Figure 8.5-c. Figure 8.5-d shows the desired output $out_2(n)$. We notice how well the morphological filtering has removed noise and baseline leaving an ideal signal for the next stages of ECG wave detection. Figures 8.6-a:d show the same input when the SE, applied for noise removal, is $g_1 = \{0, \underline{1}, 0\}$. We have shown that when impulsive noise and baseline drift exist in ECG wave, our algorithm removes them successfully without losing the important details. However we need to be confident that if there is no noise and baseline drift, still applying the above-mentioned filters would cause no adverse effect. Figures 8.7-a:d illustrate that idea, illustrating that in the case of having a pure ECG wave, our pre-processor does not affect adversely. Figures 8.8-a:d show the same input when the SE, applied for noise removal, is $g_1 = \{0, \underline{1}, 0\}$. One of the most important benefits of applying MM, besides what mentioned above, is that the baseline of the processed data will move to 0 volt and it will cease the next processing as the reference line will be 0 volt.

8.4 QRS detection

This section reviews briefly the existing methods for *QRS* detection and then proposes a better method.

8.4.1 The existing *QRS* detectors

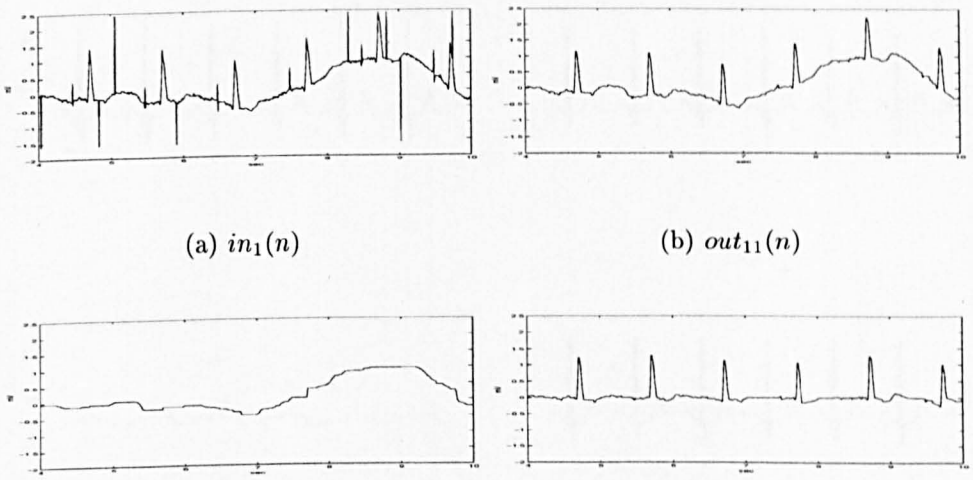
The following algorithms exist for *QRS* detection in literature [19]:



(a) $in_1(n)$ (b) $out_{11}(n)$

(c) Baseline drift (d) $out_{12}(n)$

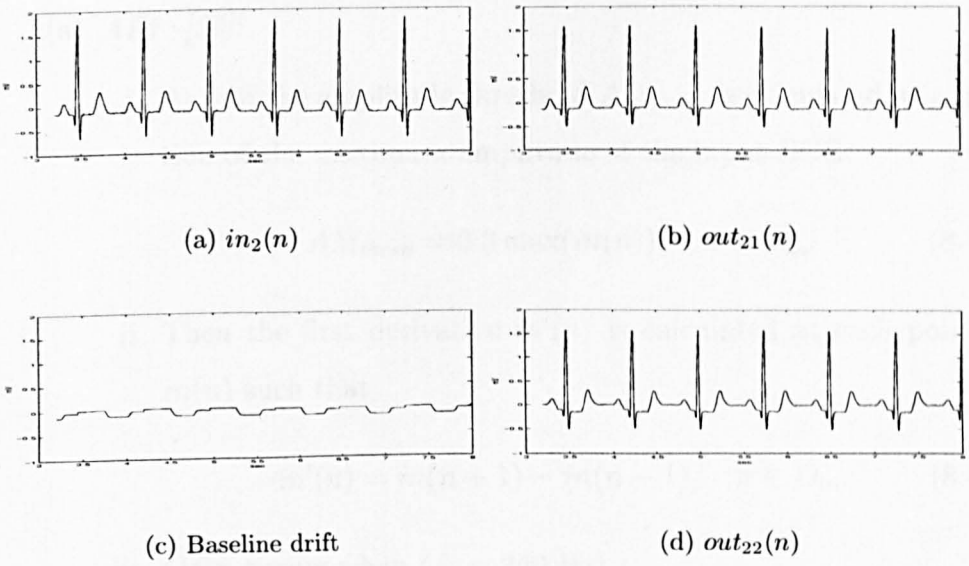
Figure 8.5: ECG noise suppression and baseline drift removal(1).



(a) $in_1(n)$ (b) $out_{11}(n)$

(c) Baseline drift (d) $out_{12}(n)$

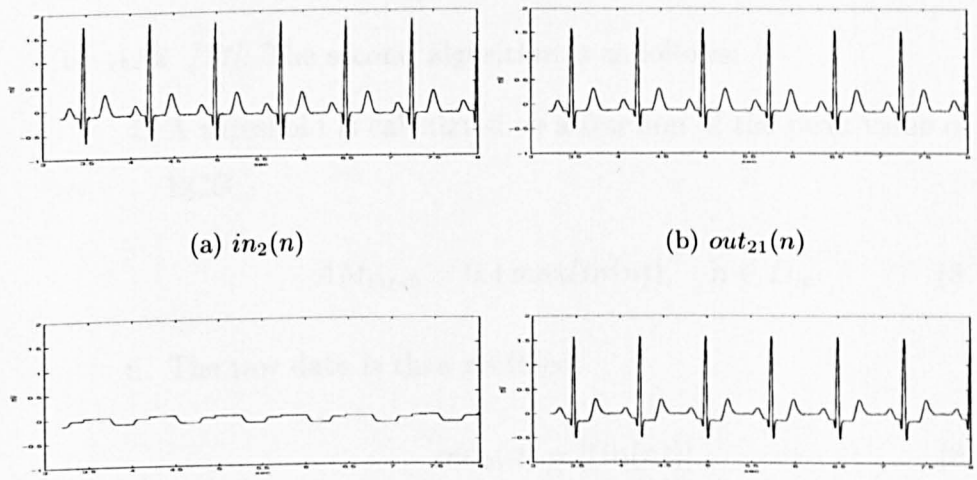
Figure 8.6: ECG noise suppression and baseline drift removal(2).



(a) $in_2(n)$ (b) $out_{21}(n)$

(c) Baseline drift (d) $out_{22}(n)$

Figure 8.7: ECG noise suppression and baseline drift removal(3).



(a) $in_2(n)$ (b) $out_{21}(n)$

(c) Baseline drift (d) $out_{22}(n)$

Figure 8.8: ECG noise suppression and baseline drift removal(4).

i. **Algorithms based on amplitude and first derivative**

(a) *AF1 [50]:*

i. At first the amplitude threshold AM_{thrsh} is calculated as a fraction of the maximum amplitude of the input ECG:

$$AM_{thrsh} = 0.3 \max(in(n)), \quad n \in D_{in}. \tag{8.4.1}$$

ii. Then the first derivative $in'(n)$ is calculated at each point of $in(n)$ such that

$$in'(n) = in(n+1) - in(n-1), \quad n \in D_{in}. \tag{8.4.2}$$

iii. *QRS* occurs when ($f_s = 250$ Hz)

A. $in'(n), in'(n+1), in'(n+2) > 0.5$, and

B. $in'(m), in'(m+1) < -0.3$, where $(n+2) < m < (n+25)$, and

C. $in(n), in(n+1), \cdots, in(m+1) \geq AM_{thrsh}$.

(b) *AF2 [17]:* The second algorithm is as follows:

i. A threshold is calculated as a fraction of the peak value of the ECG:

$$AM_{thrsh} = 0.4 \max(in(n)), \quad n \in D_{in}. \tag{8.4.3}$$

ii. The raw data is then rectified:

$$out_0(n) = |(in(n))|. \tag{8.4.4}$$

iii. The rectified ECG is passed through a low-level clipper:

$$out_1(n) = \begin{cases} out_0(n) & \text{if } out_0(n) \geq AM_{thrsh} \\ AM_{thrsh} & otherwise \end{cases} \tag{8.4.5}$$

iv. The first derivative is calculated at each point of the clipped rectified data:

$$out_1'(n) = out_1(n+1) - out_1(n-1), \quad n \in D_{out_1}. \tag{8.4.6}$$

v. A *QRS* candidate occurs when a point in $out_1'(n)$ exceeds the fixed constant threshold: $out_1'(n) > 0.7$.

(c) *AF3 [29]:* The third algorithm is as follows:

i. The first derivative $in'(n)$ is calculated at each point of $in(n)$ such that

$$in'(n) = in(n+1) - in(n-1), \quad n \in D_{in}. \tag{8.4.7}$$

ii. The result is searched for points which exceed a constant threshold: $in'(n) \geq 0.15$.

iii. Then the next three derivative values $in'(n+1)$, $in'(n+2)$ and $in'(n+3)$ must also exceed 0.15.

iv. The next two sample points must have positive slope-amplitude products:

$$in'(n+1) \cdot in(n+1) > 0 \quad \text{and} \quad in'(n+2) \cdot in(n+2) > 0.$$

If all the above conditions exist, then the current point is a *QRS* point.

ii. **Algorithms based on first derivative only**

(a) *FDI [56]:* The proposed algorithm is as follows:

i. The first derivative is calculated as:

$$\begin{aligned} &in'(n) = \\ &-2in(n-2) - in(n-1) + in(n+1) + 2in(n+2). \end{aligned} \tag{8.4.8}$$

ii. The slope threshold SL_{thrsh} is calculated as a fraction of the maximum slope for $in'(n)$:

$$SL_{thrsh} = 0.70 \max(in'(n)). \tag{8.4.9}$$

iii. $in'(n)$ is searched for points which exceed the slope threshold. The first point that exceeds the slope threshold is taken as the onset of a QRS candidate.

(b) *FD2 [38]:* The modified algorithm is as follows:

i. The first derivative $in'(n)$ is calculated at each point of $in(n)$ such that

$$in'(n) = in(n+1) - in(n-1), \quad n \in D_{in}. \tag{8.4.10}$$

ii. $in'(n)$ is searched until a point exceeds the slope threshold: $in'(n) > 0.45$.

iii. A QRS candidate occurs if another point in the next three sample points also exceeds the threshold: $in'(n+1) > 0.45$, or $in'(n+2) > 0.45$, or $in'(n+3) > 0.45$.

iii. **Algorithms based on first and second derivative**

(a) *FS1 [5]:*

i. The absolute values of the first and second derivative are calculated:

$$\begin{aligned} in'(n) &= |in(n+1) - in(n-1)| \\ in''(n) &= |in(n+2) - 2in(n) + in(n-2)| \end{aligned} \tag{8.4.11}$$

ii. These two data are scaled and then summed:

$$out(n) = 1.3in'(n) + 1.1in''(n). \tag{8.4.12}$$

iii. $out(n)$ is scanned until a threshold is met: $out(n) \geq 1.0$.

iv. Once this occurs, the next eight points are compared to the threshold. If at least six of these points meet the threshold, the corresponding point is a candidate for QRS.

(b) *FS2 [1]:*

i. The rectified first derivative is calculated:

$$out_0(n) = |in(n+1) - in(n-1)|. \tag{8.4.13}$$

ii. The above result is then smoothed:

$$out_1(n) = (in(n-1) + 2in(n) + in(n+1))/4. \tag{8.4.14}$$

iii. The rectified second derivative is calculated:

$$out_2(n) = |in(n+2) - 2in(n) + in(n-2)|. \tag{8.4.15}$$

iv. The above two results are added to each other:

$$out_3(n) = out_1(n) + out_2(n). \tag{8.4.16}$$

v. The maximum value of this array is determined and scaled to serve as primary and secondary thresholds:

$$Prime_{thrsh} = 0.8 \max(out_3(n)) \tag{8.4.17}$$

$$Secnd_{thrsh} = 0.1 \max(out_3(n)) \tag{8.4.18}$$

vi. $out_3(n)$ is scanned from the first up to the last point until a point exceeds $Prime_{thrsh}$. In order to be classified as a QRS candidate, the next six consecutive points must all be equal to, or greater than $Secnd_{thrsh}$ (i.e. $out_3(i) \geq Prime_{thrsh}$, and $out_3(i+1), out_3(i+2), \cdots, out_3(I+6) > Secnd_{thrsh}$.

iv. **Algorithms based on digital filters**

(a) *DF1 [16]:*

i. The input signal is passed through a differentiator with a 62.5 Hz notch filter.

$$out_0(n) = in(n) - in(n-4). \tag{8.4.19}$$

ii. $out_0(n)$ is then passed through a digital low-pass filter:

$$out_1(n) = out_0(n) + 4out_0(n-1) + 6out_0(n-2) + 4out_0(n-3) + 4out_0(n-4). \tag{8.4.20}$$

iii. Two thresholds are used, equal in magnitude but opposite in polarity. $out_1(n)$ is scanned until a point with amplitude greater than the positive threshold is reached. This point is the onset of the search. The number of alternate threshold crossings is used to classify the initial crossing as either a baseline shift, a *QRS* candidate, or as noise: If $out_1(i) > 21.0$, then search region onset=i. If no other threshold crossings occur within the entire search, the occurrence is classified as a baseline shift. Otherwise, the following three conditions are tested:

A. if $out_1(i+j) < -21.0, \quad 0 < j < 40,$

B. if $out_1(i+j) < -21.0, \quad 0 < j < 40,$ and $out_1(i+k) > 21.0, \quad j < k < 40,$

C. $out_1(i+j) < -21.0, \quad 0 < j < 40,$ and $out_1(i+k) > 21.0, \quad j < k < 40,$ and $out_1(i+l) < -21.0, \quad k < l < 40.$

If any of the above conditions apply, the occurrence is classified as a *QRS*. If additional threshold crossings occur, the occurrence is classified as noise.

(b) *DF2 [59]:*

i. The first stage smoothes the input ECG using a three-point moving average filter:

$$out_0(n) = (in(n-1) + 2in(n) + in(n+1))/4. \tag{8.4.21}$$

ii. $out_0(n)$ is then passed through a low-pass filter:

$$out_1(n) = \frac{1}{2m+1} \sum_{k=n-m}^{n+m} out_0(k), \quad m < n < N - m, \tag{8.4.22}$$

where N is the total length of the input samples.

iii. The difference between $out_0(n)$ and $out_1(n)$ is squared:

$$out_2(n) = (out_0(n) - out_1(n))^2, \quad m < n < N - m. \tag{8.4.23}$$

iv. The squared difference is then filtered:

$$out_3(n) = out_2(n) \left(\sum_{k=n-m}^{n+m} out_2(k) \right)^2 \quad m < n < N - m. \tag{8.4.24}$$

v. $out_4(n)$ is made as: $out_4(n) = out_3(n)$ if $(out_0(n) - out_0(n - m))(out_0(n) - out_0(n + m)) > 0$, otherwise $out_4(n) = 0$.

vi. The maximum value of $out_4(n)$ is determined and scaled to form the threshold:

$$Thrsh = 0.125 \max(out_4(n)), \quad m < n < N - m. \tag{8.4.25}$$

vii. A QRS occurs when a point in $out_4(n)$ exceeds $Thrsh$.

Experiments show that as m increases, the performance increases along with computational demands. The best value for m seems to be 6.

For a detailed discussion about the comparison of the above methods, the paper by Frisen et. al. [19] is a good reference. What we can briefly conclude, is that none of the forementioned techniques does not act perfectly due to lack of a robust pre-processor. Therefore we apply our technique for QRS detection as explained in the next part.

8.4.2 Our approach to QRS detection

In some cases the amplitude of the T wave may be as big as of QRS on some leads. Therefore it is not wise to use the algorithms using the amplitude. Fulton [20] applies the absolute sum of all twelve leads. However regarding that only the leads I, II, V1, V2, V3, V4, V5 and V6 are independent for twelve-lead ECG, we apply the absolute sum of the independent leads to detect the approximate place of QRS complex. We choose leads II and V1 for single lead investigation, as P waves are usually (and not always) seen better in these leads. The algorithm applied is as follows:

i. Apply a smoothing filter to each independent lead to remove small fluctuations caused by the pre-processors.

$$y_i(n) = (x_i(n-2) + x_i(n-1) + 2x_i(n) + x_i(n+1) + x_i(n+2))/6 \quad (8.4.26)$$

where $i = 1 \cdots 8$ for each independent lead.

ii. Make the absolute sum of all $y_i(n)$, denoted $sum(n)$:

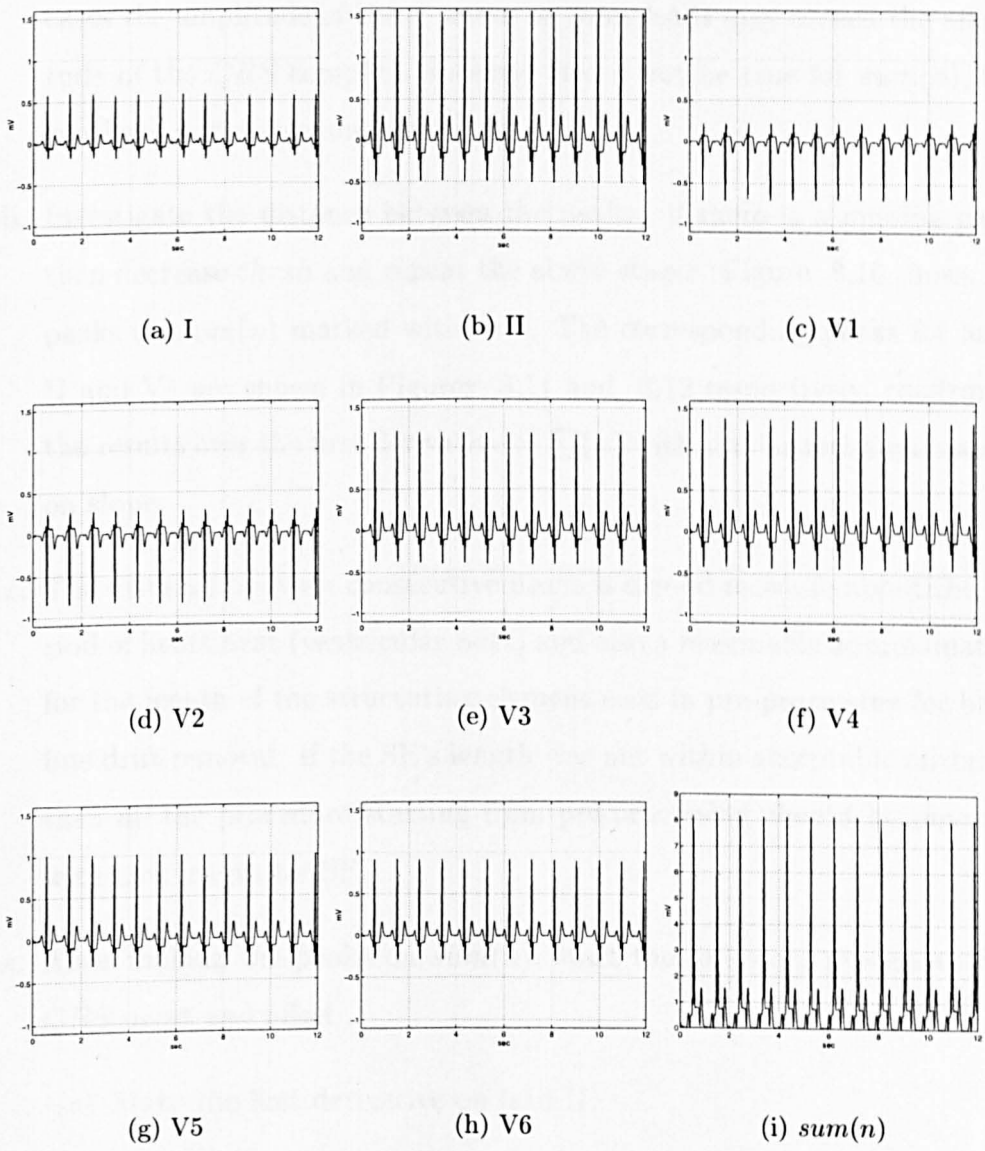
$$sum(n) = \sum_{i=1}^{8} y_i(n). \quad (8.4.27)$$

Figure 8.9 shows eight independent pre-processed leads and $sum(n)$.

iii. Find the global maximum amplitude of $sum(n)$ (denoted $MAXSUM$).

iv. Assign a threshold (denoted $thrsh$) and make an amplitude threshold: $AMP_{thrsh} = thrsh * MAXSUM$. If $thrsh$ is too low, a lot of false peaks will be detected, and if it is too high, some peaks will be missing. Therefore a trade-off should be considered. We apply $thrsh = 0.8$. If it is too high, it will be reduced as explained below.

v. Scan $sum(n)$ to find the points that exceed AMP_{thrsh}.



(a) I (b) II (c) V1

(d) V2 (e) V3 (f) V4

(g) V5 (h) V6 (i) $sum(n)$

Figure 8.9: Eight pre-processed leads and their absolute sum.

vi. Investigate any continuous interval exceeding AMP_{thrsh}. Then find the local maximum amplitude inside every interval that exceeds AMP_{thrsh}.

vii. Remove false peaks due to large pacemaker spikes and T waves. In some cases the amplitude of the T waves on some leads may exceed the amplitude of the QRS complex. However it can not be true for $sum(n)$, but can have noticeable amplitude.

viii. Investigate the distance between the peaks. If there is a missing peak, then decrease *thrsh* and repeat the above stages. Figure 8.10 shows the peaks on $sum(n)$ marked with "<". The corresponding peaks for leads II and V1 are shown in Figures 8.11 and 8.12 respectively, confirming the results over the first derivative of $\sum(n)$ with similar techniques acted on slope.

ix. The distance between consecutive peaks is a good measure about the period of heart beat (ventricular beat) and also a reasonable approximation for the length of the structuring element used in pre-processing for baseline drift removal. If the SE's length was not within acceptable criterion, then all the procedure starting from pre-processing should be repeated with new length for SE.

x. After making the peaks on $sum(n)$, start the following process to find QRS onset and offset:

   (a) Make the first derivative on lead II.

   (b) Start from the location of the first corresponding peak on lead II. Evaluate the location and amplitude of the relevant peak by searching a neighbourhood around the location of the peak. It may slightly differ, regarding the effect of the smoothing filter $y_2(n)$ and based on the fact that different peaks (positive or negative) may occur

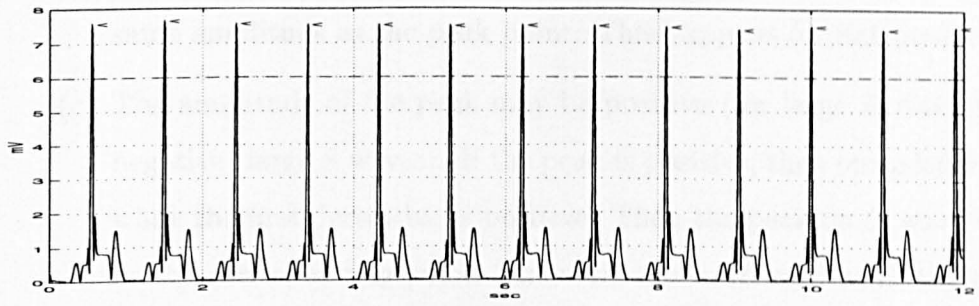


Figure 8.10: The peaks on $sum(n)$.

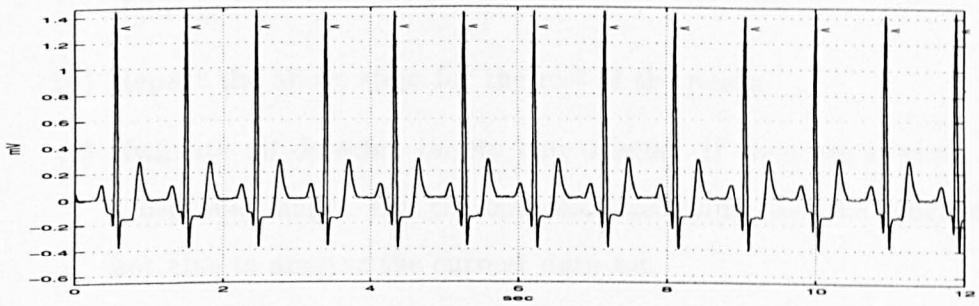


Figure 8.11: The peaks on II.

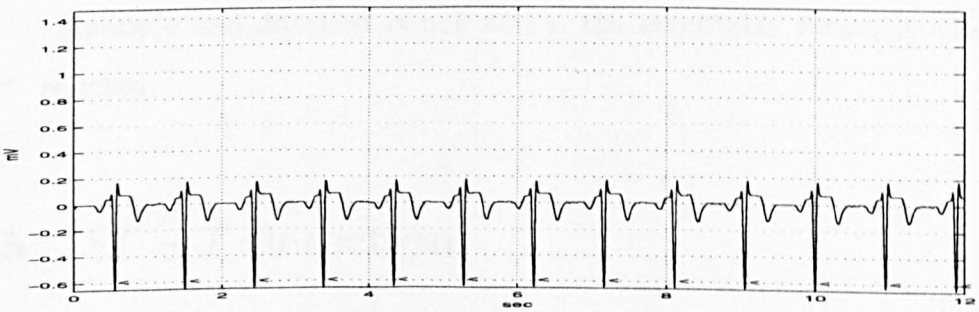


Figure 8.12: The peaks on V1.

on different leads. Scan leftward all adjacent points which have the same amplitude as the peak point. This happens for flat peaks.

(c) The amplitude of the peak may be positive (for large R waves) or negative (large S waves). If the peak is positive, then move leftward while the first derivative is positive. Then the peak on Q wave will be the corresponding point. Then still move leftward while the first derivative is negative. Then at this point the onset of QRS has been detected. If the peak is negative, a dual tracking technique should be considered. The same technique but dual track for slopes should be considered for detecting QRS offset while the movement will be rightward.

(d) Repeat the above steps for the rest of the peaks.

(e) Evaluate all detected onsets and offsets. If they are beyond the acceptable ranges, exit the operation declaring that the program is not able to analyse the current data set.

Similar procedure is applied for lead V1. The result of the proposed method is shown in Figures 8.13- 8.16. The QRS complexes are marked with red colour. After finding QRS onsets and offsets, the validity of them (duration and amplitude) should be evaluated. If any complex's amplitude and duration is not within the acceptable range, it will be rejected.

8.5 $ST-T$ detection

The applied algorithm for $ST-T$ detection, is alert enough to detect the buried P waves on T waves. In some cases, as mentioned, P wave can be on top of a T wave, changing the normal shape of the T wave. Therefore it will be

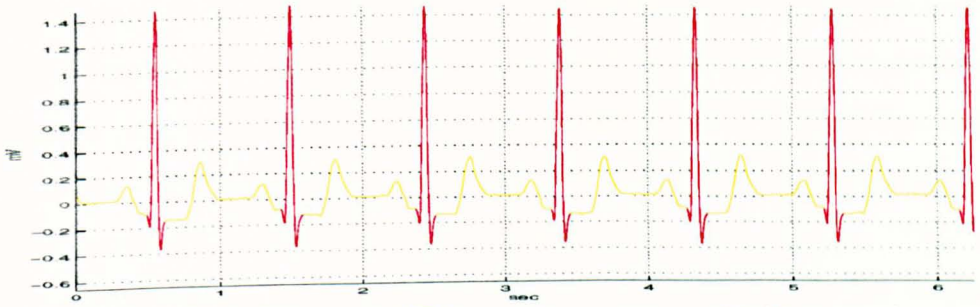


Figure 8.13: Detected QRS complexes on lead II (first part).

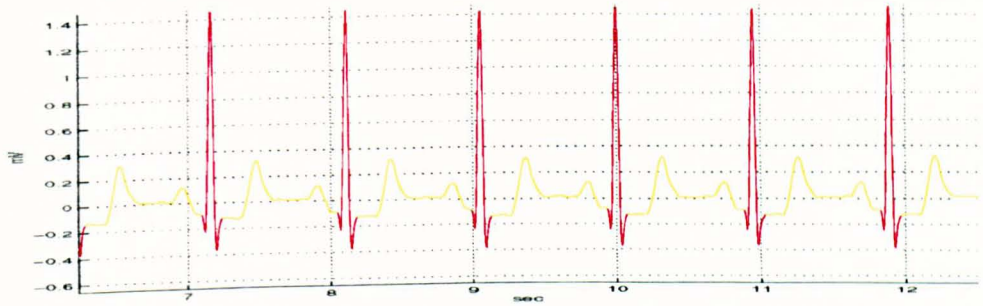


Figure 8.14: Detected QRS complexes on lead II (second part).

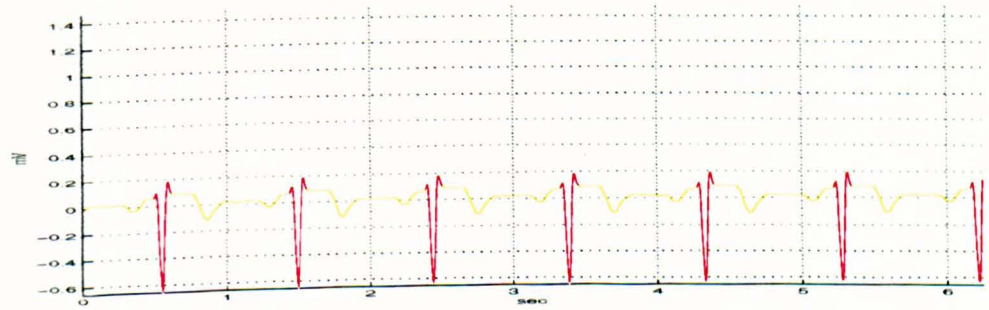


Figure 8.15: Detected QRS complexes on lead V1 (first part).

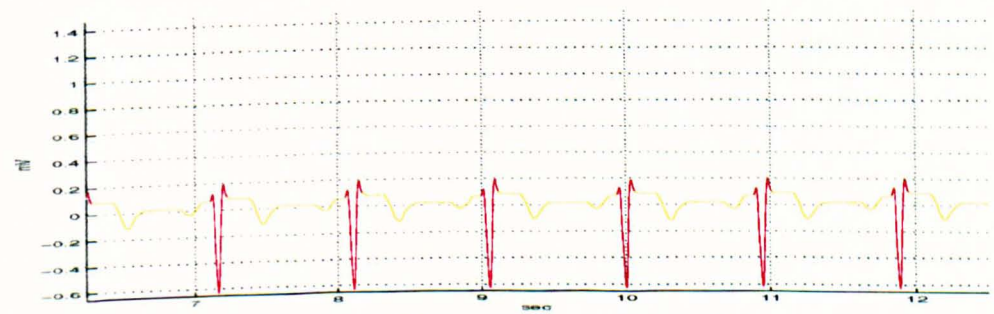


Figure 8.16: Detected QRS complexes on lead V1 (second part).

a big mistake to try to identify the T waves without considering the location of the P waves, as it exists among the current T wave detection algorithms in literature (e.g. [27, 14]). In fact we concentrate on $ST-T$ detection instead of T waves. When later the P waves are detected, T waves can be separated. The T waves are positive or sometimes negative or even bipolar. They may occur immediately after QRS offset. The T waves are usually flat compared to the duration of QRS and they have higher amplitude than the P waves. The following part illustrates the algorithm for ST detection. This procedure should be repeated for each $ST-T$ segment.

i. Make a search over the interval between the current QRS offset and the next QRS onset for every QRS to investigate the following instructions.

ii. Apply open-closing (denoted by *opcl*) for the search area. Experimental results show that a flat structuring element of size 9 satisfies the demands.

iii. Construct the first derivative of the filtered data for the given interval (denoted $deriv1$).

iv. Find the maximum of absolute value among 1/8, of the full interval range of *opcl* and remember its location. Find out the real sign of the relevant maximum. Denote it as $max18$.

v. The T waves can never occur too much closer to QRS complexes. Therefore bypass possible fluctuations and/or ST segment elevations by moving rightward from the location of $max18$ and regarding $deriv1$ and the amplitude of *opcl* to reach to a flat area.

vi. Find the maximum of absolute value of *opcl* from the above point up to half the full range as the most probable location for the T wave. Denote it as $max12$.

vii. Move leftward and rightward from the location of $max12$ to find the onset and offset of the T wave with the similar technique as mentioned for finding QRS onset and offset, bearing in mind that the T waves are always flatter than the QRS complexes.

viii. Evaluate the detected onset and offset. They should be in appropriate ranges and proper distances from the offset of the corresponding QRS complexes. If they are beyond the acceptable ranges, then reject them.

Figures 8.17- 8.20 show the detected $ST-T$ segments marked by green colour on leads II and V1.

8.6 Residue

For difficult rhythms (where atrial waves are on top of ST segments, or they are spread all over the ECG with no fixed temporal relation to the QRS), it is better to form one single template of the ST for all the complexes which look alike. Moreover if the $ST-T$ segment is zeroed, then we will not be able to see any atrial waves buried inside the T waves or in the ST segments. Reddy et. al. [67] apply ST median and QRS interpolation to leave a residue of carrying the information of the P waves as described below. We have carefully examined the recommended algorithm and discovered that it superbly helps for next stage to detect the P waves. Therefore our algorithm for making the residue is adapted from [67]. The proposed method is as follows.

i. The correlation among all detected $ST-T$ segments is evaluated. If the correlation exceeds a threshold, the corresponding segments are categorised as the same template, otherwise a new template is built. The value for thresholding should be decided very carefully. The more the level of the threshold, the more, the number of templates, and vice versa.

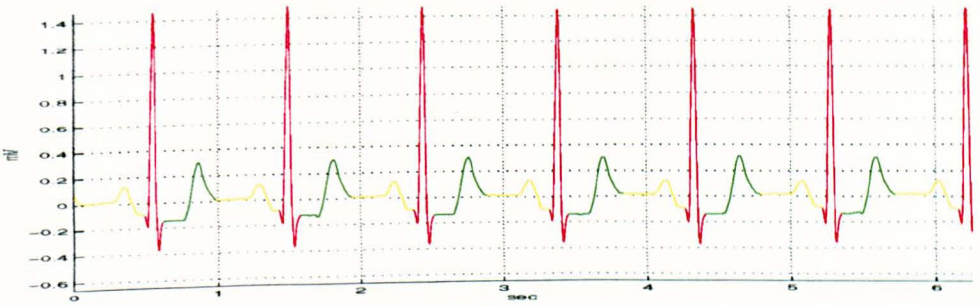


Figure 8.17: Detected $QRST$s on lead II (first part).

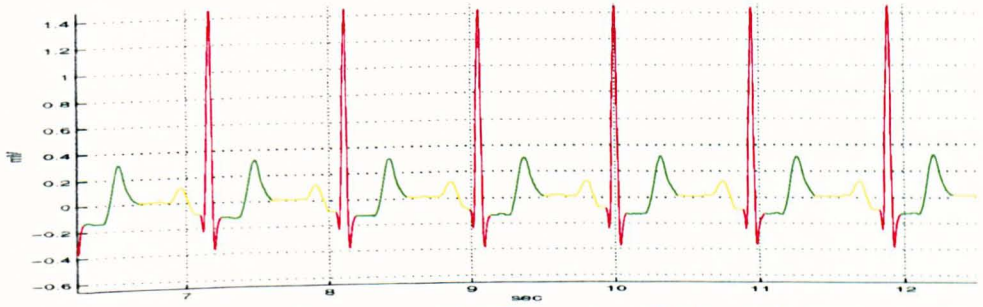


Figure 8.18: Detected $QRST$s on lead II (second part).

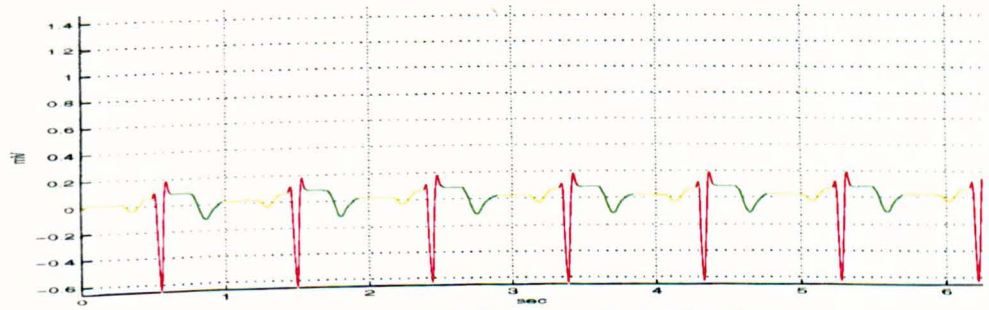


Figure 8.19: Detected $QRST$s on lead V1 (first part).

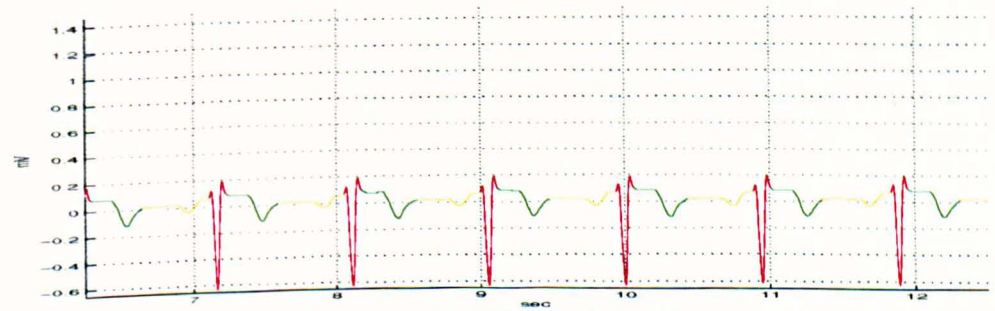


Figure 8.20: Detected $QRST$s on lead V1 (second part).

ii. After categorising the ST segments into different groups, the median of each group is found. There are advantages of applying median instead of averaging [21].

iii. Each ST segment is subtracted from the median of its group.

iv. All QRS complexes are substituted by interpolation of the values of their onsets and offsets. They are not replaced by zero values in order to avoid the abrupt discontinuities in the subtracted waveform.

Applying the above algorithm will leave a residue mostly carrying information about the P waves. Figures 8.21- 8.24 show the residual signals on leads II and V1 respectively.

8.7 P wave detection

Reddy et. al. [66] apply a nine-point central Differentiator upon the residue as below.

$$\begin{aligned} y(n) = \; & -x(n-4)/256 - x(n-3)*3/32 - x(n-2)/2 - x(n-1) \\ & +x(n+4)/256 + x(n+3)*3/32 + x(n+2)/2 + x(n+1). \end{aligned} \tag{8.7.1}$$

Then they investigate the second difference computed as:

$$z(n) = y(n) - y(n-1). \tag{8.7.2}$$

A composite function f is then considered by rectifying and adding the first and second differences as:

$$f(n) = |y(n)| + |z(n)|. \tag{8.7.3}$$

We, instead, use morphological filtering and get a better result than theirs. Our approach is as follows:

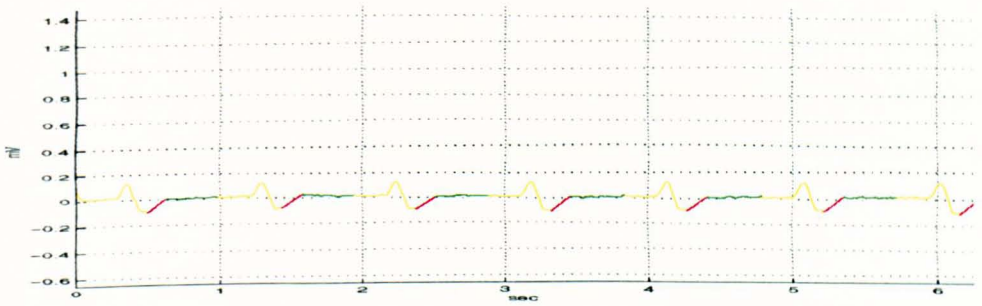


Figure 8.21: Residue on lead II (first part).

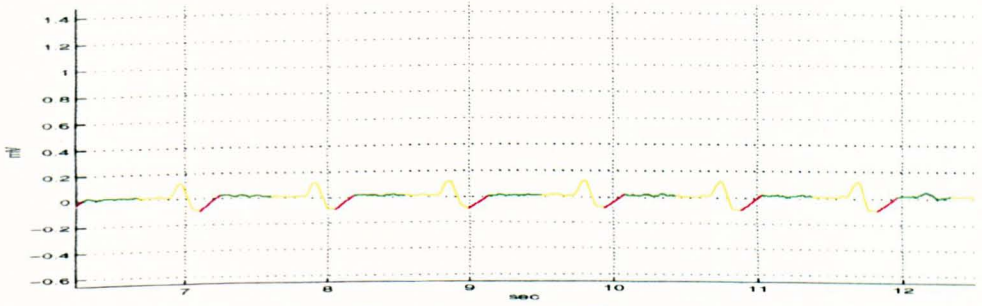


Figure 8.22: Residue on lead II (second part).

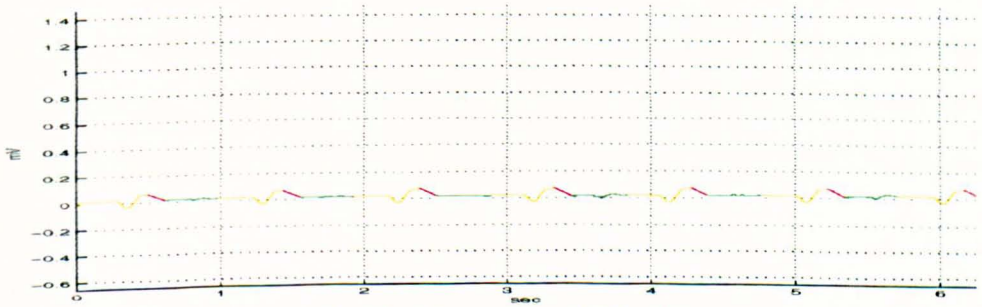


Figure 8.23: Residue on lead V1 (first part).

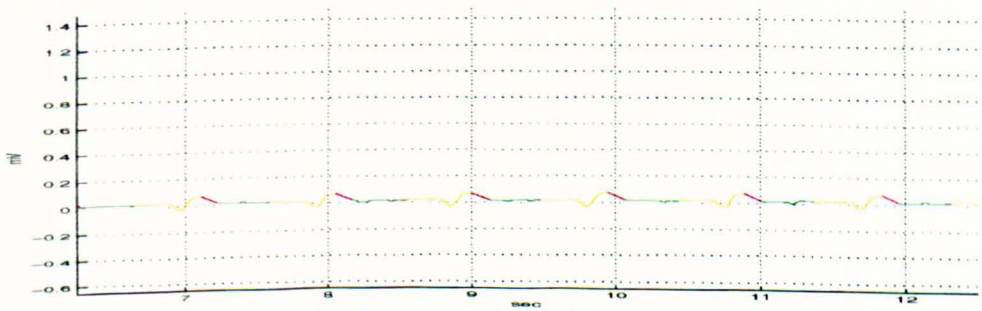


Figure 8.24: Residue on lead V1 (second part).

1. Compute the absolute value of the open-close of the residual signal on leads II and V1. A flat structuring element of size 9 seems to be adequate. Take the average of the above signals for next stage. We denote it as $opcl_2(n)$.

2. Make the first derivative of the above signal.

3. Investigate all the intervals between the offset of the current QRS and the onset of the next one.

4. Find the local maxima inside the intervals.

5. Find the onset and offset around each maximum with the similar techniques as mentioned before.

6. Evaluate the validity of the points candidated as the onsets and offsets of different P waves by investigating their amplitudes and widths.

Figures 8.25 and 8.26 show the detected P waves on $opcl_2(n)$ marked by blue colour. Figures 8.27- 8.30 illustrate the detected P waves on the corresponding residues. Figures 8.31- 8.34 show the corresponding results on the original signal.

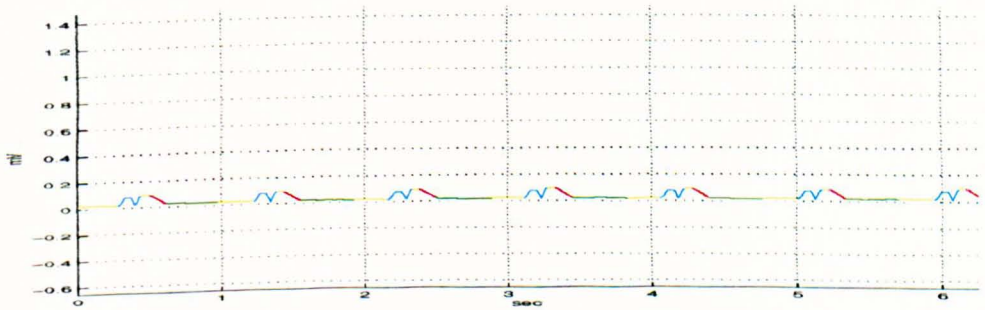


Figure 8.25: $opcl_2(n)$ (first part).

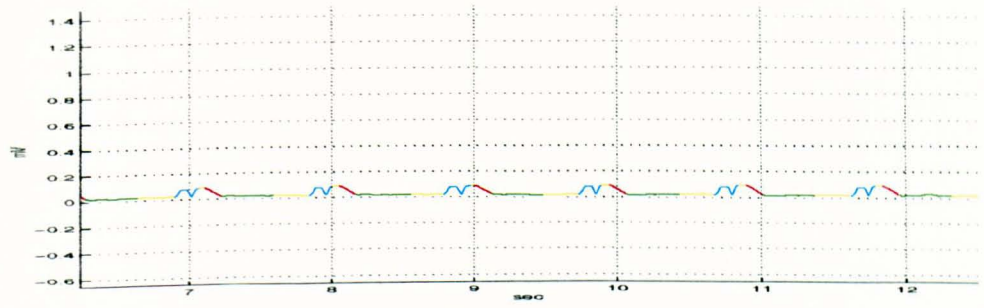


Figure 8.26: $opcl_2(n)$ (second part).

Figures 8.35- 8.48 illustrate the whole procedure for a case which has buried P on top of T waves.

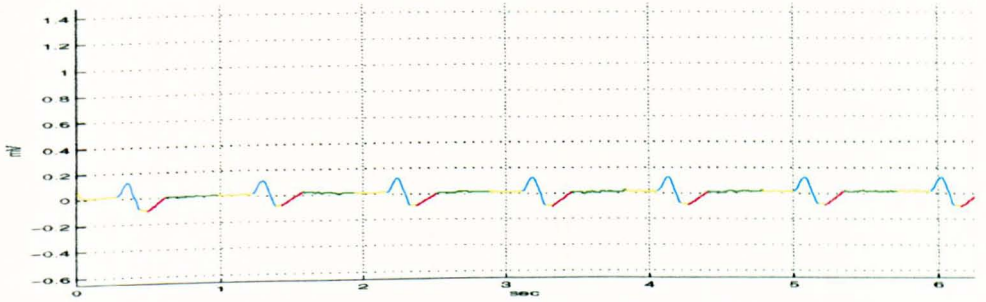


Figure 8.27: P waves on the residue of lead II (first part).

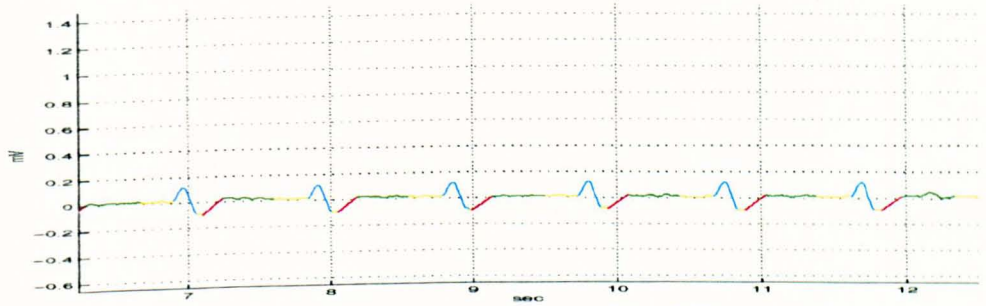


Figure 8.28: P waves on the residue of lead II (second part).

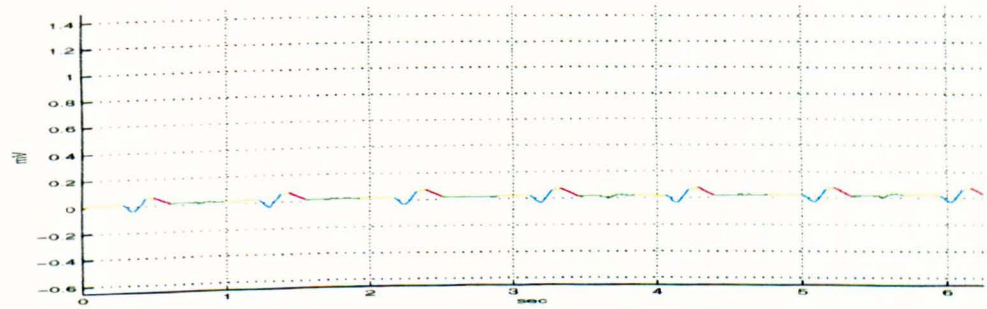


Figure 8.29: P waves on the residue of lead V1 (first part).

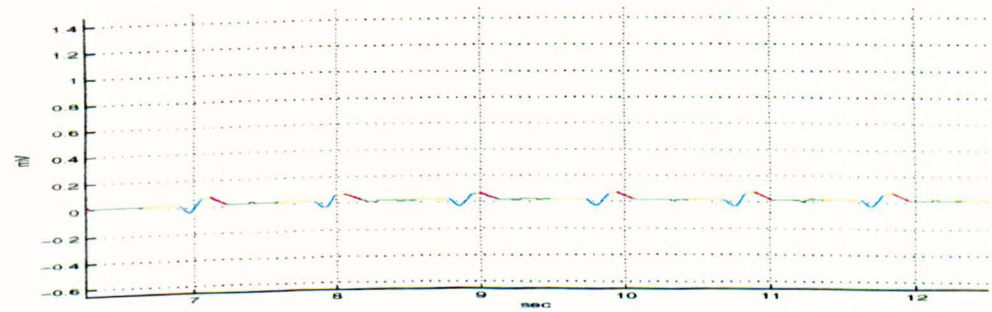


Figure 8.30: P waves on the residue of lead V1 (second part).

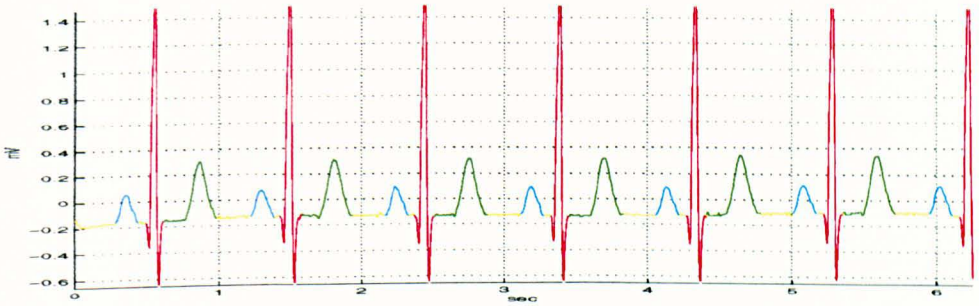


Figure 8.31: P waves on the original signal of lead II (first part).

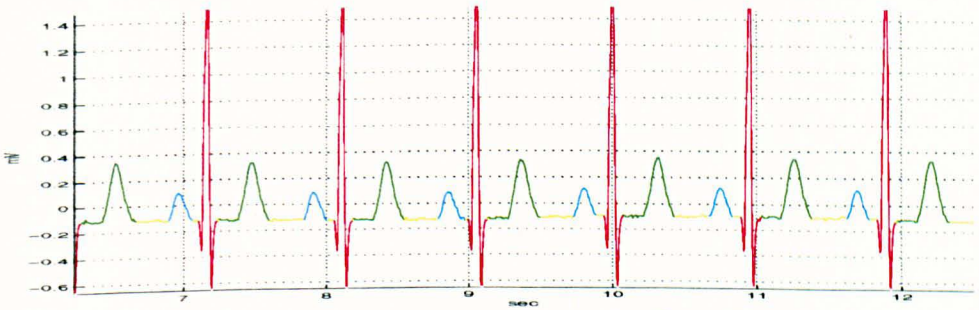


Figure 8.32: P waves on the original signal of lead II (second part).

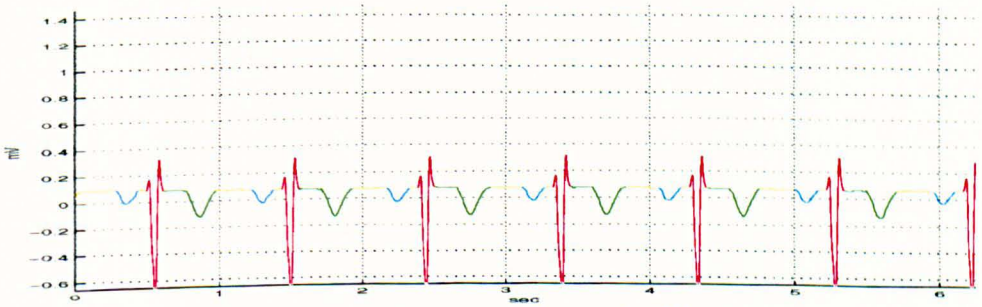


Figure 8.33: P waves on the original signal of lead V1 (first part).

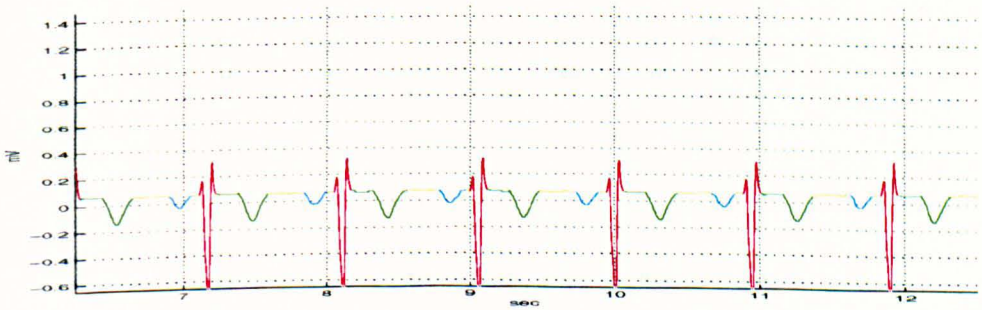


Figure 8.34: P waves on the original signal of lead V1 (second part).

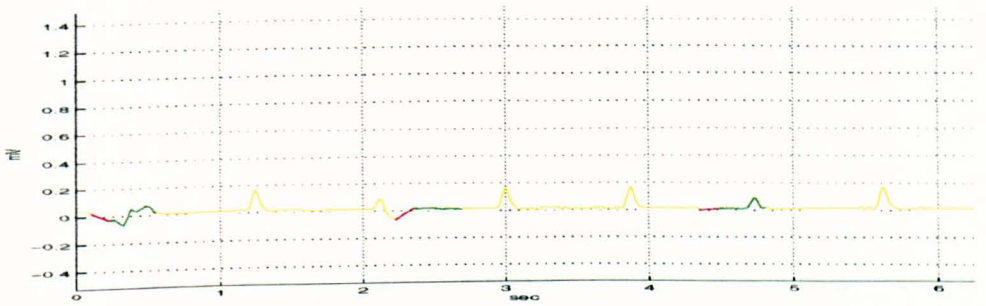


Figure 8.35: Residue on lead II (1st.) in a complete heart block.

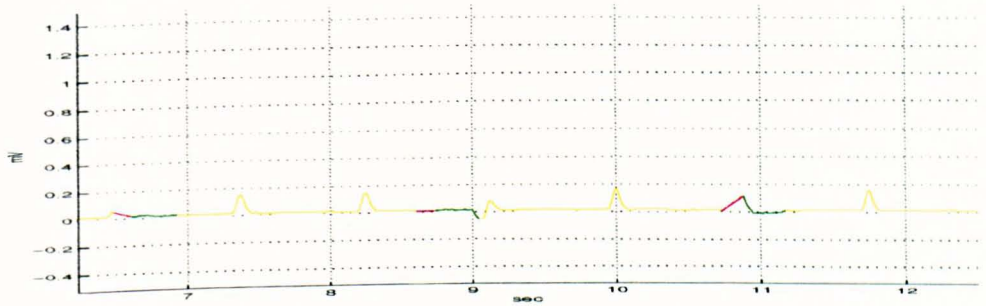


Figure 8.36: Residue on lead II (2nd.) in a complete heart block.

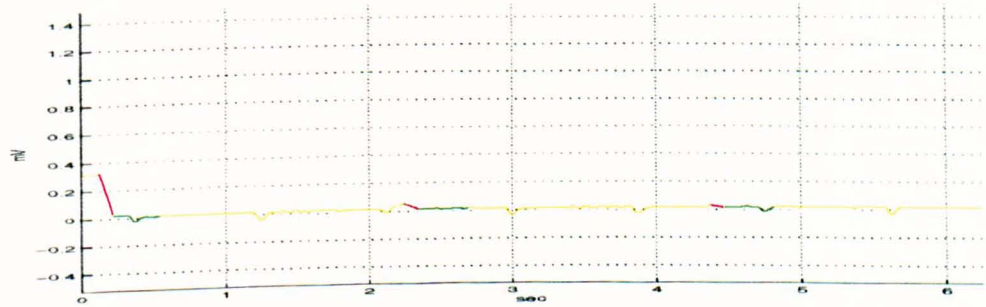


Figure 8.37: Residue on lead V1 (1st) in a complete heart block.

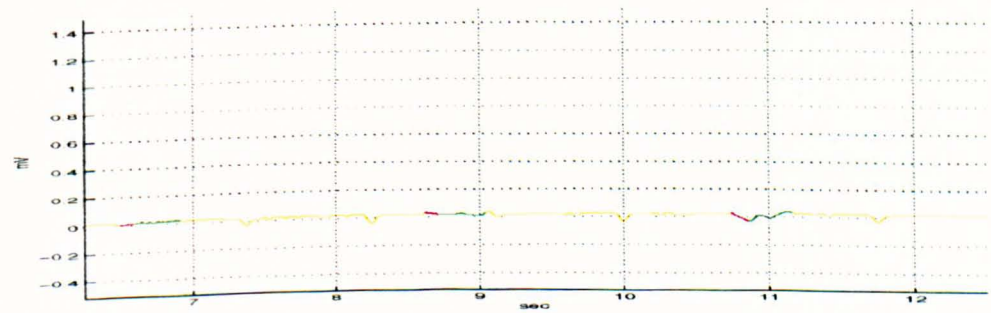


Figure 8.38: Residue on lead V1 (2nd.) in a complete heart block.

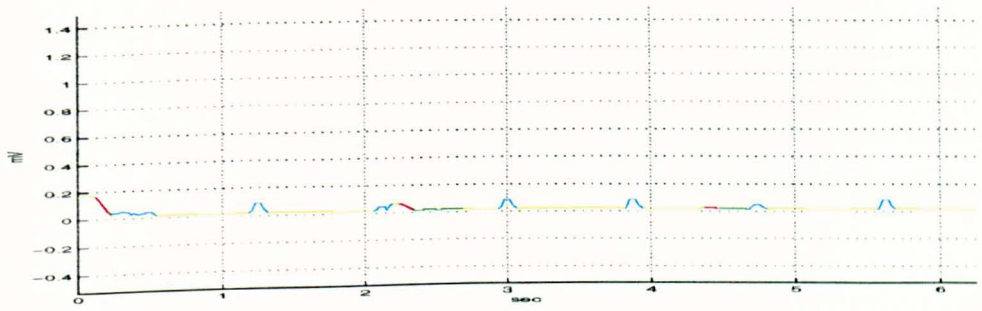


Figure 8.39: $opcl_2(n)$ (first part) in a complete heart block.

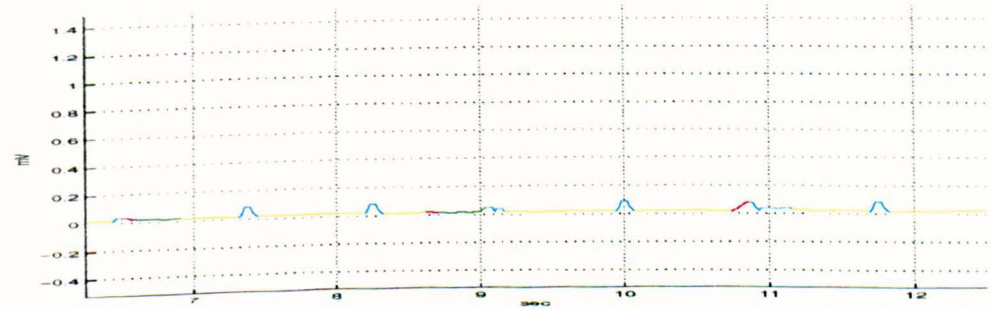


Figure 8.40: $opcl_2(n)$ (second part) in a complete heart block.

8.8 Discussion and conclusion

We have tried our algorithm on the database obtained from Royal Liverpool University Hospital and Marquette Electronics Co. Ltd.. One of the disadvantages of our algorithm, despite its accuracy, is that due to a heavy computation, it is not suitable for ambulatory cases, or generally speaking, for real-time implementations, unless the algorithm is applied via the real-time morphological operators as explained in chapter 4 and suitable hardware for the rest of the operations.

8.9 Future work

The algorithm, applied for ECG analysis, has been evaluated by Liverpool Women's Hospital experts and upon their demands, a similar robust technique has been required for fetal ECG monitoring. The algorithms for fetal ECG is supposed to differ from what applied for adult ECG processing. We will develop new algorithms based on their demands.

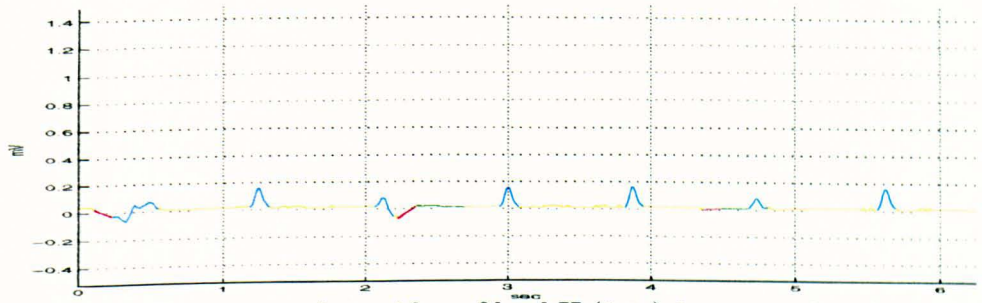


Figure 8.41: P waves on the residue of lead II (1st.) in a complete heart block.

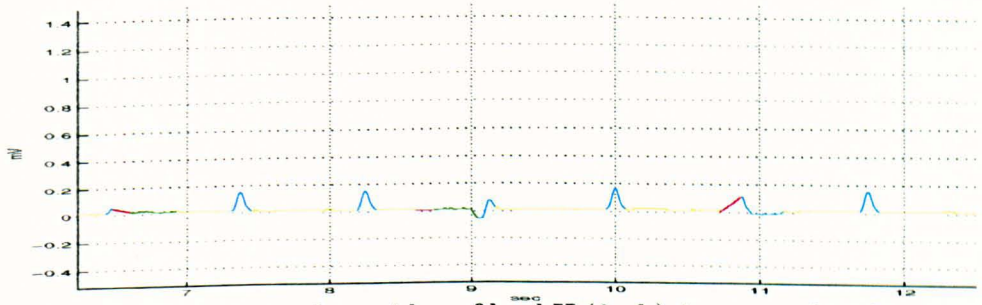


Figure 8.42: P waves on the residue of lead II (2nd.) in a complete heart block.

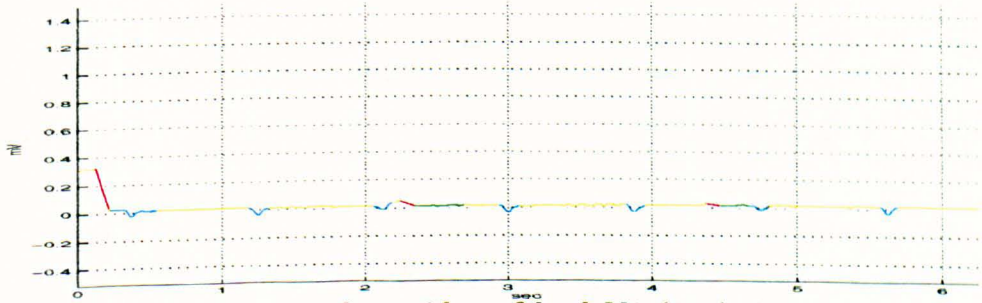


Figure 8.43: P waves on the residue of lead V1 (1st.) in a complete heart block.

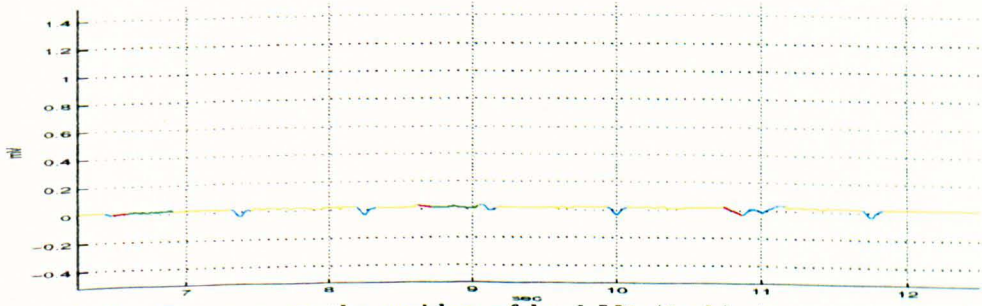


Figure 8.44: P waves on the residue of lead V1 (2nd.) in a complete heart block.

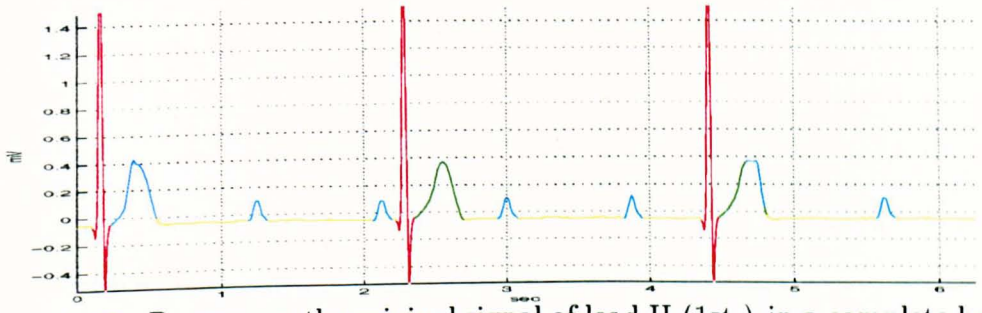


Figure 8.45: P waves on the original signal of lead II (1st.) in a complete heart block.

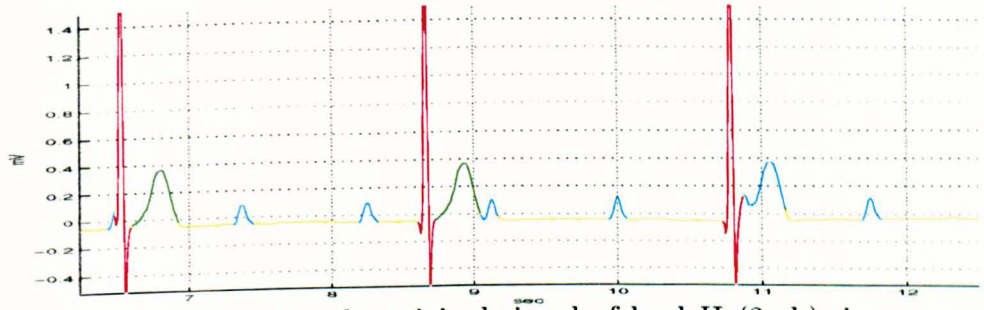


Figure 8.46: P waves on the original signal of lead II (2nd.) in a complete heart block.

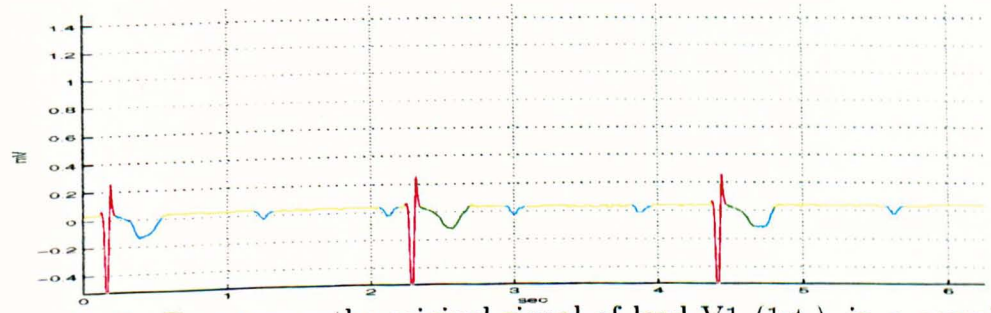


Figure 8.47: P waves on the original signal of lead V1 (1st.) in a complete heart block.

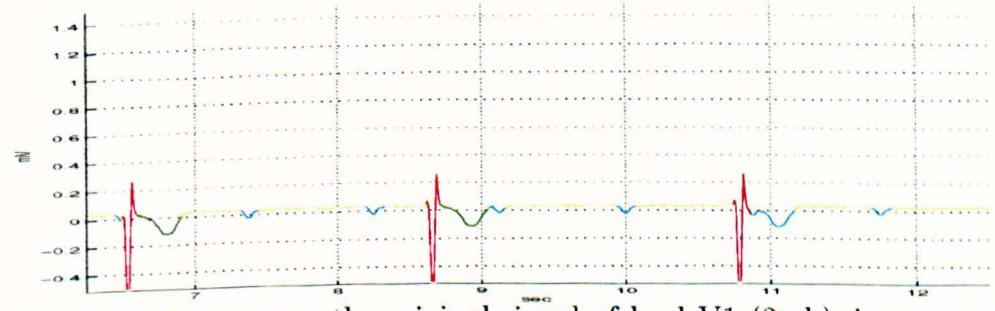


Figure 8.48: P waves on the original signal of lead V1 (2nd.) in a complete heart block.

Chapter 9

Summary and conclusion

9.1 General Remarks

This chapter concludes the dissertation. It summarizes the major results of the presented research work and indicates directions for further investigations based on this work.

9.2 Summary of the thesis

This thesis has traced the development of MM from its beginning to the latest propositions. The point of departure is the consideration of the problem of obtaining efficient and real-time morphological operators in 1-D and 2-D, and applying them in different areas where traditional methods fail to be applied successfully.

In the preceding chapters we have presented the following work and obtained promising results.

Literature review was done in *Chapter 1.* The importance of considering MM as an advanced image processing tool was explained. Historical notes about MM were reviewed. A comparison between MM versus non-MF was

demonstrated. Also major contributions and overview of the thesis were indicated.

Chapter 2 introduced a theoretical background on mathematical morphology. It also explained the existing confusion in literature about the definition of the operators. Definition of the terms and operators in binary and grey-scale modes were discussed. Properties of morphological operators were also highlighted.

Chapter 3 presented new fast algorithms in 1-D and 2-D for morphological erosion, dilation, opening, closing, open-closing and close-opening, relying on avoiding redundant comparisons.

Chapter 4 improved and generalised real-time implementation of 1-D and 2-D GS morphological operators.

Chapter 5 proposed a technique to combine MF with convolution, both in software and hardware, emphasising the power of our algorithm in using any kind of SE.

Chapter 6 introduced new novel operators called weighted morphological filters and emphasized their superb performance compared with classical MFs for removing salt&pepper, speckle and Gaussian noises with different noise parameters.

Chapter 7 showed how MF could be applied in fingerprint processing as a pre-processor. It also introduced a syntactic approach to fingerprint classification including the details of pattern classification and string analysis.

Chapter 8 was about ECG waves detection using MF in different stages. After an introduction and definition of the terms in ECG, a pre-processor based on MF was introduced. It emphasised the efficiency of pre-processor in removing noise and baseline drift without loosing necessary details of the signal. The existing QRS detectors were reviewed and our contribution was included. The proposed method for $ST - T$ detection was then followed. A residual signal was produced, resulting from interpolation of QRS and subtracting the

median(s) of $ST - T$ segments from the corresponding parts. Later step was about analysing the residue and detecting P waves, even when they were buried on T waves.

Our major contribution was concentrated on the following topics:

i. **Fast algorithms for 1-D and 2-D GS morphological operators ([69, 70]).**

ii. **Real-time implementation of GS morphological filtering ([72, 75, 73]).**

iii. **Convolved morphological filters ([74, 77]).**

iv. **Linearly-weighted morphological operators ([76]).**

v. **Morphological filters in fingerprint processing ([71]).**

vi. **ECG waves detection using morphological filtering.**

9.3 Limitations of the approach

One of the major limitations using MM was the lack of analytical methods, due to the nonlinearity of the MF operators. The next drawback was about the evaluation of fingerprint recognition. We could not test our approach in real environment. As we reckoned, it could only be used for small companies where the size of database was limited. Besides time limitation, we did not have access to a good experimental database. In general, it has not been applied for a real-time fingerprint recognition.

Another limitation of the research was about the adaptive structuring element design. We tried genetic algorithms to design SE for ECG wave analysis and fingerprint processing. However the procedure was very slow, specially in 2-D, and we did not get the satisfactory results. Therefore we only used SEs based on a pre-knowledge about the nature of the signals and images.

9.4 Recommendations for future work

Designing the structuring element has been left as an open research topic in MM. It would be worth of having a research for SE design using another techniques instead of genetic algorithms. Our research about convolved and weighted morphological operators can be deeply extended to design a hybrid embedded operator.

Fingerprint recognition using MF can be another topic for a research. The most important pre-requisite is obtaining a large data base to test the developed algorithms. Our research can be continued towards fingerprint recognition. We stopped at classification step. Stretching MF in m-D, could be another interesting research aspect, for example finding applications in colour image processing. Finally our research about ECG wave detection can be followed to get better results leading to commercial products.

Appendix A

Morphological citations in English

Table A.1: Abbreviation of Journals, etc.

Abrev.	Description
CSICC	Annual Conf. of Computer Society of Iran
AIDIM	Advances in digital image proc., edit=P. Stucki, Plentum, New York
AAECC	Applicable Algebra in Engineering, Communication and Computing
BOOK-DOUGH	MM in Image Processing, E. R. Dougherty, Marcel Dekker, New York
BOOK-HARAL	MM. Theory and Hardware, ed.=R.M. Haralick, Oxford Univ. Press, New York
BOOK-O	Shape in Picture: Mathematical Description of Shape in Grey-Level Images, ed.=Y-L. O et. al., NATO ASI Series, Driebergen, The Netherlands, Springer, Berlin, vol.=126
BOOK-SER-94	MM and its applications to image proc., Kluwer Academic, The Netherlands
CSSP	Circuits, Systems, Sig. Proc.
CVGIP	Comp. Vision, Graph. and Image Proc.
CVGIPIU	Comp. Vision, Graph. and Image Proc.:Image understanding
CWI	Centrum voor Wiskunde en Informatica
ECOLE	Ecole Nationale Supérieure des Mines de Paris, Fontainebleau
ELEC-LET	Electronics Letters
ELSSIG	Signal Processing
ELSSIGIC	Signal Processing: Image Communication
GMIP	Graphical Models and Image Processing
ICEE	Iranian Conf. on Electrical Eng.
IECIPAIA	IEE Conf. on Imag. Proc. and its applic.
IEECSP	IEEE Computer Society Press
IEEPRC	Proceedings of the IEEE
IEESL1	IEEE Signal Proc. Letters
IEETA1	IEEE Trans. Acoust. Speech Signal Process.
IEETB1	IEEE Trans. Biomed. Eng.
IEETC1	IEEE Trans. Circuits and Systems
IEETC4	IEEE Trans. Comput.

continued on next page

continued from previous page (Citation:Theory and tutorial)

Abrev.	Description
IEETI1	IEEE Trans. Image Proc."
IEETP2	IEEE Trans. Pattern Anal., Machine Intell.
IEETS1	IEEE Trans. Signal Proc."
IEEWNSIP	IEEE Workshop on Non-Linear Sig. and Image Proc.
ISCSDSP	International Symp.on Communic. Systems and DSP
ISMM	International Symposium On Mathematical Morphology and its Applications to Image and Signal Processing IV
JEI	Journal of Electronic Imaging
JMIV	Journal of Mathematical Imaging and Vision
JVCIR	Journal of Visual Comm. and Image Rep.
PATR	Pattern Recognition
PATRL	Pattern Recognition Letters
PhD	PhD Thesis
PROCMM	Proc. of the Intern. Symp. on Mathematical Morphology
PROCIEVISP	Proc. IEE Vis., Image, Sig. Proc.
PROCIEE18	Proc. IEEE 18th Convention of Elec. Eng.
PROCIWMM	Proc. Intern. Workshop on Mathematical Morphology
PROCISSSE	Proc. Internat. Symp. on Signal Systems, and Electronics
SSPR	Shape, Structure and Pattern Recognition, Edit.= D. Dori et. al., World Scientific
PROCWATRS	Proc. Workshop on Applic.s and Theory of Random Sets, Minneapolis, edit=J. Goutsias et. al., Springer, New York
PROCSSMISP	Proc. Summer School on Morpholo. Image and Signal Proc.
RSE	Remote Sensing of the Environment
RTI	Real-Time Imaging
RTMIP	Real-time medical image proc., edit.=M. Onoe et. al., Plentum, London
SIAMJ	SIAM J. Appl. Math.
SIRL	Signal and Image Research Laboratory
SPIE-OEP	SPIE Optical Engineering Press
SPIEPRC-IAMIP	SPIE Proc. Image Algebra and Morphological Image Proc.
SPIEPRC-CVCIP	Proc. SPIE-Conf. on Visual Comm. and Image Proc.
SPIE-NFIP	SPIE, Nonlinear Filters for Image Proc.

Table A.2: Citation:Theory and tutorial.

Author	Yr	Title	Description	Pages
L. Doyen et. al.	94	Mutational equations of morphological dilation tubes	BOOK-SER-94	13-20
H. J. A. M. Heijmans	87	MM: an algebraic approach	CWI Newsletter, vol.=4	7-27
H. J. A. M. Heijmans	89	Iteration of morphological transformations	CWI Quarterly, vol.=2	19-36
H. J. A. M. Heijmans	90	Morphological filtering and iteration	SPIEPRC-CVCIP, Lausanne,vol.=1360	166-175
H. J. A. M. Heijmans et. al.	90	The algebraic basis of MM: I. dilations and erosions	CVGIP, vol.=50	245-295
H. J. A. M. Heijmans	91	Theoretical aspects of gray-level morphology	IEETP2, vol.=13, no.=6	568-582

continued on next page

continued from previous page (Citation:Theory and tutorial)

Author	Yr	Title	Description	Pages
Y. Hsueh	94	Thresholdings, umbrae, residuals, and surpluses of l-images	BOOK-SER-94	21-28
P. Maragos	87	Tutorial on Advances in Morphological Image Processing and Analysis	Optical Engineering, vol.=26	623-632
J. Serra	86	Introduction to MM	CVGIP, vol.=35, no.=3	283-305
J. Serra	93	Anamorphoses and function lattices (multivalued morphology	MM in Image Processing: E.R. Dougherty	483-523

- Table A.3: Citation:Statistical analysis.

Author	Yr	Title	Description	Pages
M. Charif-Chefchaouni et. al.	95	Morphological representation of order-statistics filters	IEETI1, vol.=4, no.=6	838-845
C.-S. Chen et. al.	96	Statistical analysis of space-varying morphological openings with flat SEs	IEETS1, vol.=44, no.=4	1010-1014
H. A. David	92	Some properties of order-statistics filters	CSSP, vol.=11, no.=1	109-114
M. A. Mohamed et. al.	95	Statistical evaluation of sequential morphological operations	IEETS1, vol.=43, no.=7	1703-1709
A. Morales et. al.	93	Statistical Analysis of Morphological Openings	IEETS1, vol.=41 , no.=10	3052-3056
A. Morales et. al.	94	A study of statistical properties of GS compound morphological operators using the basis matrix	SPIE, vol.=2180, no.=V	124-135
S. Na et. al.	95	Second-order statistics of morphological dilation and erosion of a memoryless source	IEETS1, vol.=43, no.=10	2418-2422
J. Neejärvi et. al.	92	Statistical analysis of median type and morphological filters	SPIEPRC-CVCIP, vol.=1818	366-375
C. Regazzoni et. al.	94	Statistical pattern spectrum for binary pattern recognition	BOOK-SER-94	185-192
M. Ropert et. al.	94	Synthesis of adaptive weighted order statistic filters with gradient algorithms	BOOK-SER-94	37-44
R. L. Stevenson et. al.	87	Morphological filters: statistics and further syntactic properties	IEETC1, vol.=CAS-34, no.=11	1292-1305
C. Wang et. al.	95	Some statistical properties of MM	IEETS1, vol.=43, no.=8	1955-1965

- Table A.4: Citation:Representations and overviews.

Author	Yr	Title	Description	Pages
G. J. F. Banon et. al.	91	Minimal representation for translation-invariant set mappings by mathematical morphology	SIAM J. Appl. Math., vol.=51	1782-1798

continued on next page

continued from previous page (Citation:Representations and overviews)

Author	Yr	Title	Description	Pages
P. Bhattacharya et. al.	93	An algebraic approach for morphological operations on 2D and 3D images	PATR, vol.=26	1785-1796
I. Bloch et. al.	95	Fuzzy mathematical morphologies: a comparative study	PATR, vol.=28, no.=9	1341-1387
M. Charif-Chefchaouni et. al.	92	Morphological bounds on nonlinear filters	SPIEPRC-CVCIP, vol.=1818	414-425
M. Charif-Chefchaouni et. al.	94	On the invertibility of the morphological representation of binary images	IEETI1, vol.=3	847-849
M. Charif-Chefchaouni et. al.	94	Morphological representation of nonlinear filters	JMIV, vol.=4	215-232
M. Charif-Chefchaouni et. al.	95	Generalized morphological center: convergence	IEEWNSIP, vol.=I, ed.=I. Pitas, Greece	325-328
M. Charif-Chefchaouni et. al.	95	Generalized morphological center: self-duality, idempotence, and convergence	SIRL, Univ. of Illinois, Report, Chicago	
D. Coltuc et. al.	96	Morphological residual representations of signals	IEETRI1, vol.=5, no.=11	1569-1572
J. Crespo et. al.	97	Locality and adjacency stability constraints for morphological connected operators	JMIV, vol.=7, no.=1	85-102
J. Crespo et. al.	95	Theoretical aspects of morphological filters by reconstruction	ELSSIG vol.=47, no.=2	201-225
J. Crespo et. al.	98	New results on the theory of morphological filters by reconstruction	PATR, vol.=31, no.=4	419-429
L. Dorst et. al.	94	Two dual representations of morphology based on the parallel normal transport property	BOOK-SER-94	161-170
L. Dorst et. al.	95	Orientation-based representations for MM	SSPR	13-22
E. R. Dougherty et. al.	93	Precision of Morphological estimation	SPIE, vol.=1902, no.=IV	65-76
E.R. Dougherty	94	Minimal representation of τ-openings via pattern bases	PATRL, vol.=15	1029-1033
E.R. Dougherty et. al.	94	Precision of morphological-representation estimators for translation-invariant binary filters: increasing and nonincreasing	ELSSIG, vol.=40	129-154
E. R. Dougherty et. al.	95	Computational gray-scale mathematical morphology on lattices (a comparator-based image algebra) Part 1: architecture	RTI, vol.=1, no.=1	69-85

continued on next page

continued from previous page (Citation:Representations and overviews)

Author	Yr	Title	Description	Pages
E. R. Dougherty et. al.	95	Computational gray-scale mathematical morphology on lattices (a comparator-based image algebra) Part 2: Image operators	RTI, vol.=1, no.=1	283-295
L. Doyen et. al.	95	Mutational equations of the morphological dilation tubes	JMIV, vol. =5, no.=3	219-230
J. Goutsias	??	Morphological Transformations of Image Sequences: A Lattice Theory Approach	SPIEPRC-IAMIP, Belingham, USA, vol.=III	20-22
J. Goutsias	92	Morphological Analysis of Discrete Random Shapes	JMIV, vol. 2	193-215
J. Goutsias	??	Modeling Random Shapes: An Introduction to Random Set Theory	BOOK-HARAL	
J. Goutsias et. al.	91	Morphological Representation of Discrete and Binary Images	IEETS1, vol.=39	1369-1379
R. M. Haralick et. al.	87	Image analysis using MM	IEETP2, Vol.:PAMI-9, No:4	532-550
R. M. Haralick et. al.	87	Multiresolution morphology	IEEPRC, First Int. Conf. Comp. Vision	516-520
R. M Haralick et. al.	95	Model-based morphology: the opening spectrum	GMIP, vol.=57, no.=1	1-12
H. J. A. M. Heijmans et. al.	92	Graph morphology	JVCIR, vol.=3	24-38
H. J. A. M. Heijmans et. al.	92	Convergence, continuity and iteration in MM	JVCIR, vol.=3	84-102
H. J. A. M. Heijmans et. al.	93	Graph morphology in image analysis	BOOK-DOUGH	171-203
H. J. A. M. Heijmans	93	A note on the umbra transform in gray-scale morphology	PATRL, vol.=14	877-881
H. J. A. M. Heijmans	94	Dominance and incidence structures with applications to stochastic geometry and MM	BOOK-SER-94	171-178
H.J.A.M. Heijmans	95	MM: basic principles	PROCSSMISP	
H. J. A. M. Heijmans	95	MM: a modern approach in image processing based on algebra and geometry	SIAM Review, vol.=37, no.=1	1-36
H.J.A.M. Heijmans et. al.	96	Similarity and Symmetry Measures for Convex Sets Based on Minkowski Addition	Report BS-R9610, CWI, Amsterdam	
R. P. Loce et. al.	95	Mean-absolute-error representation and optimization of computational-morphological filters	GMIP, vol.=57, no.=1	27-37
P. Maragos	89	A representation theory for morphological image and signal processing	IEETP2, vol.=11, no.=6	586-599
P. Maragos	89	Pattern spectrum and multiscale shape representation	IEETP2, vol.=11, no.=7	701-716
P. Maragos et. al.	90	Threshold superposition in morphological image analysis systems	IEETP2, vol.=12	498-504

continued on next page

continued from previous page (Citation:Representations and overviews)

Author	Yr	Title	Description	Pages
P. Maragos	90	Affine morphology and affine signal models	SPIEPRC-IAMIP, San Diego, Vol.=1350	31-43
P. Maragos et. al.	90	Morphological systems for multidimensional signal processing	IEEPRC, vol.=78, no.=4	690-710
J Mattioli	95	Minkowski operations and vector spaces	Set-Valued Analysis, vol.=3, no.=1	33-50
J. Mattioli et. al.	94	On information contained in the erosion curve	BOOK-O	177-195
P. F. M. Nacken	93	Chamfer metrics in MM	CWI, Amsterdam, Report, no.=BS-R9309	
P.F.M. Nacken	94	Chamfer metrics in mathematical morphology	JMIV, vol.=4	233-253
P. F. M. Nacken	96	Chamfer metrics, the medial axis and MM	JMIV, vol.=6, no.=2/3	235-248
Y. L. O et. al.	91	MM in hierarchical image representation	Medical Images: Formation, Handling and Evaluation, Ed.= A. E. Todd-Pokropek et. al., NATO ASI Series F, Pub.=Springer, Heidelberg	447-462
A.T. Popov	97	Convexity Indicators Based On Fuzzy Morphology	PATRL, vol.=18, no.=3	259-267
J. A. Rea et. al.	96	Fuzzy logic and MM: implementation by stack filters	IEETS1, vol.=44, no.=1	142-147
J. M. Reinhardt et. al.	96	Efficient morphological shape representation	IEETC1, vol.=5, no.=1	89-101
J. B. T. M. Roerdink et. al.	88	MM for structures without translation-symmetry	ELSSIG, vol.=15	271-277
J. B. T. M. Roerdink	89	MM on homogeneous spaces - Part I: The simply transitive case	CWI, Amsterdam , report, no.=AM-R8924	
J. B. T. M. Roerdink	90	MM on homogeneous spaces - Part II: The transitive case	CWI, Amsterdam , report, no.=AM-R9006	
J. B. T. M. Roerdink	90	MM on the sphere	SPIEPRC-CVCIP, Lausanne, vol. 1360	263-271
J. B. T. M. Roerdink	93	MM with non-commutatitive symmetry groups	MM in Image Processing, BOOK-DOUGH	205-254
J. B. T. M. Roerdink	??	On the construction of translation and rotation invariant morphological operators	BOOK-HARAL	
J. B. T. M. Roerdink	94	Manifold shape: from differential geometry to MM	BOOK-O	209-223
C. Ronse	89	Fourier analysis, MM, and vision	Working Document, no.=WD54, Philips Research Laboratory, Brussels, Belgium	
C. Ronse	90	Regular open or closed sets	Working Document, no.=WD, Philips Research Laboratory, Brussels, Belgium	
C. Ronse	90	Why mathematical morphology needs complete lattices	ELSSIG, vol.=21	129-154
C. Ronse et. al.	91	Morphological shape and region description	ELSSIG, vol.=25	91-105

continued on next page

continued from previous page (Citation:Representations and overviews)

Author	Yr	Title	Description	Pages
C. Ronse et. al.	91	The algebraic basis of mathematical morphology – Part II: Openings and closings	CVGIPIU, vol.=54	74-97
C. Ronse	92	Toggles of openings, and a new family of idempotent operators on partially ordered sets	AAECC, vol.=3	99-128
C. Ronse	94	Lattice-theoretical fixpoint theorems in morphological image filtering	JMIV, vol.=4	19-41
C. Ronse	96	A lattice-theoretical morphological view on template extraction in images	JVCIR, vol.=7, no.=3	273-295
D. Schonfeld et. al.	90	Robust Morphological Representation of Binary Images	Proceedings of the "International Conference on Acoustics, Speech, and Signal Processing", Albuquerque, New Mexico, pub.=IEEE	3-6
D. Schonfeld et. al.	91	On the Morphological Representation of Binary Images in a Noisy Environment	JVCIR, vol.=2	17-30
J. Serra	89	Toggle mappings	From Pixels to Features, ed.=J. C. Simon, Pub.=North Holland, Amsterdam	61-72
J. Serra et. al.	89	Elements of functional analysis in MM	Report, no.=N-39/90/MM, ECOLE	
J. Serra et. al.	92	An overview of morphological filtering	CSSP, Vol.=11, no.=1	47-108
J. Serra	94	Morphological filtering: an overview	ELSSIG, vol.=38	3-11
F. Y. Shih et. al.	92	A mathematical morphology approach to Euclidean distance transformation	IEETI1, vol.=1	197-204
N. D. Sidiropoulos et. al.	96	Further results on MAP optimality and strong consistency of certain classes of morphological filters	IEETI1, vol.=5, no.=5	762-764
N. D. Sidiropoulos et. al.	96	MAP signal estimation in noisy sequences morphologically smooth images	IEETI1, vol.=5, no.=5	1088-1093
A. Skowron et. al.	B. 96	Analytical morphology: mathematical morphology of decision tables	Fundamenta Informaticae, vol.=27	255-271
J. Song et. al.	90	The analysis of morphological filters with multiple SEs	CVGIP, vol.=50	308-328
A. V. Tuzikov et. al.	97	Convex set symmetry measurement via Minkowski addition	JMIV, vol.=7, no.=1	53-68
L. Vincent	93	Morphological algorithms	BOOK-DOUGH	255-288
X. Wang et. al.	92	Some Sequential Algorithms for a Generalized Distance Transformation Based on Minkowski Operations	IEETP2, vol.=14, no.=11	1114-1121
S. S. Wilson	92	Theory of matrix morphology	IEETP2, vol.=14	636-652
S. S. Wilson	93	Training structuring elements in morphological networks	BOOK-DOUGH	1-41

continued on next page

continued from previous page (Citation:Representations and overviews)

Author	Yr	Title	Description	Pages
Z. Xiaoqi	95	Shape description and recognition using the high order morphological pattern spectrum	PATR, vol.=28, no.=9	1333-1340

- Table A.5: Citation:General filtering.

Author	Yr	Title	Description	Pages
J. A. Bangham et. al.	96	Scale-Space from Nonlinear Filters	IEETP2, vol.=18, no.=5	520-528
R.W. Brockett et. al.	94	Evolution equations for continuous-scale morphological filtering	IEETSP1, vol.=42, no.=12	3377-3386
M. H. Chen et. al.	89	A Multiscale Approach Based on Morphological Filtering	IEETP2, vol.=11, no.=7	694-700
Y. Chen et. al.	94	Adaptive parameterized openings	BOOK-SER-94	29-36
E. R. Dougherty et. al.	93	Representation of finite-range increasing filters in the context of computational morphology	SPIE, vol.=1902, no.=IV	53-63
E. R. Dougherty et. al.	87	Error bounds for morphologically derived measurements	SIAMJ, vol.47	425-440
E. D. Dougherty	92	Morphological pseudoconvolutions: one-parameter families of derived filters with increased invariant classes	CSSP, vol.=11, no.=1	195-228
M. Fathy et. al.	95	An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis	PATRL, vol.=16, no.=12	1321-1330
P. K. Ghosh	96	MM operations of boundary-represented geometric objects	JMIV, vol.=6, no.=2/3	199-222
A. Gross et. al.	B. 95	Digitizations preserving topological and differential geometric properties	CVIU, vol.=62, no.=3	370-381
C.-C. Han et. al.	94	A greedy and branch and bound searching algorithm for finding the optimal morphological erosion filter on binary images	IEESL1, vol.=1	41-44
N.R. Harvey et. al.	94	Using genetic algorithms in the design of morphological filters	BOOK-SER-94	53-60
N.R. Harvey et. al.	96	The use of genetic algorithms in morphological filter design	ELSSIG, vol.=8, no.=1	55-71
H. J. A. M. Heijmans	91	Morphological discretization	Geometrical Problems of Image Proc., Ed.=U. Eckhardt et. al., pub.= Akademie Verlag, Berlin	99-106
H. J. A. M. Heijmans	92	Discretization of morphological operators	JVCIR, vol.=3	182-193

continued on next page

continued from previous page (Citation:General filtering)

Author	Yr	Title	Description	Pages
H.J.A.M. Heijmans	93	Apects of the theory of morphological operators and filters	Workshop on Design Methodologies for Microelectronics and Signal Processing , Gliwice-Cracow, Poland	377-387
H. J. A. M. Heijmans	94	Construction of self-dual morphological operators and modifications of the median	IEECSP, Proceedings ICIP-94, Los Alamitos, vol.=II	492-496
H. J. A. M. Heijmans	94	On the construction of morphological operators which are self-dual and activity-extensive	ELSSIG, vol.=38	13-19
H. J. A. M. Heijmans	95	Composing morphological filters	CWI, Research report, Amsterdam, no.=BS-R9504	
H. J. A. M. Heijmans	95	Morphological filters	PROCSSMISP, Zakopane, Poland	
H. J. A. M. Heijmans	96	Self-dual morphological operators and filters	JMIV, vol.=6, no.=1	15-36
H.J.A.M. Heijmans	97	Composing Morphological Filters	IEETI1, vol.=6, no.=5	713-723
H.J.A.M. Heijmans	97	Connected Morphological Operators for Binary Images	Report PNA-R9708, CWI, Amsterdam	
H. J. A. M. Heijmans	97	Easy recipes for morphological filters	SPIE-NFIP, Ed.=E. R. Dougherty et. al.	
P. T. Jackway	94	Properties of multiscale morphological smoothing by poweroids	PATRL, vol.=15	135-140
X. C. Jin et. al.	95	A domain operator for binary morphological processing	IEETI1, vol.=4, no.=7	1042-1046
R. Jones et. al.	94	Morphological filtering as template matching	IEETP2, vol.=16	438-443
K.D. Lee et. al.	94	Threshold Boolean filters	IEETS1, vol.=42	2022-2036
J. Mattioli	94	Minimal generator basis of a finite structure opening	BOOK-SER-94	61-68
P. Maragos et. al.	87	Morphological filters-Part I: Their set-theoretic analysis and relations to linear shift-invariant filters	IEETA1, vol.=ASSP-35, no.=8	1153-1169
P. Maragos et. al.	87	Morphological filters-Part II: Their relations to median, order-statistic, and stack filters	IEETA1, vol.=ASSP-35, no.=8	1170-1184
P. Kraft et. al.	95	Parallel genetic algorithms for optimizing morphological filtera	IECIPAIA, no.=410	762-767
P. F. M. Nacken	94	Openings Can Introduce Zero Crossings in Boundary Curvature	IEETP2, vol.=16, no.=6	656-658
J. Neejärvi et. al.	90	Sinusoidal and pulse responses of morphological filters	IEEPRC vol.=78	2136-2139
S.-C. Pei et. al.	97	An efficient class of alternating sequential filters in morphology	GMIP, vol.=59, no.=2	109-116
A. T. Popov	95	Morphological operations on fuzzy sets	IECIPAIA, no.=410	837-840
C. Ronse	??	Openings: main properties, and how to construct them	BOOK-HARAL	

continued on next page

continued from previous page (Citation:General filtering)

Author	Yr	Title	Description	Pages
C. Ronse	98	Set-theoretical algebraic approaches to connectivity in continuous or digital spaces	JMIV, vol.=8	41-58
M. A. Schulze et. al.	93	linear combinations of morphological operators: the midrange, pseudomedian, and LOCO filters	IEEPRC vol.=	V-57:V-60
M. H. Sedaaghi et. al.	98	The power of morphological filters alone and when combined with linear filtering	ISMM'98, Amsterdam, 6-8 Apr.	
N. D. Sidiropoulos et. al.	92	Optimal morphological filters for discrete random sets under a union or intersection noise model	SPIEPRC-CVCIP, vol.=1818	402-413
K. Sivakumar et. al.	97	On the discretization of morphological operators	JVCIR, vol.=8, no.=1	39-49
J. Song et. al.	92	A study of the generalized morphological filter	CSSP, vol.=11, no.=1	229-252
S. R. Sternberg	86	Grayscale morphology	CVGIP, vol.=35, no.=3	333-355
I. D. Svalbe	91	The geometry of Basis Sets for Morphologic Closing	IEETP2, vol.=13, no.=12	1214-1224
L. Vincent	91	Morphological transformations of binary images with arbitrary SEs	ELSSIG, vol.=22	3-23
L. Vincent	94	Morphological area openings and closings for grey-scale images	BOOK-O	197-208
R. Vogt	94	A spatially variant, locally adaptive, background normalization operator	BOOK-SER-94	45-52
D. Wang et. al.	92	Compared performances of morphological, median type and running mean filters	SPIEPRC-CVCIP, vol.=1818	384-391
Q. Wang et. al.	93	Root properties of morphological filters	ELSSIG, vol.=34	131-148
Q. Wang et. al.	93	Adaptation of grayscale morphological filters	SPIE, vol.=1902, no.=IV	2-7
G. R. Wilson	96	Morphological operations on crack coded binary images	PROCIEVISP, vol.=143, no.=3	171-176
P. Deng-Wong et. al.	92	Adaptive morphological filters for color image enhancement	SPIEPRC-CVCIP, vol.=1818	358-365
Y. Yao et. al.	94	Image enhancement using MM with adaptive SEs	SPIE-NFIP, vol.=2180, no.=V	198-208
D. Zhao	93	Adaptive thresholding through morphological filtering	SPIE, vol.=1902, no.=IV	148-158

- Table A.6: Citation:Hit-miss filtering.

Author	Yr	Title	Description	Pages
D. Bloomberg et. al.	90	Generalized hit-miss operations	SPIEIAMIP, vol.=1350	116-128
E. R. Dougherty	93	Optimal morphological hit-or-miss filtering of gray-scale images	SPIE, vol.=1902, no.=IV	30-40

continued on next page

continued from previous page (Citation:Hit-miss filtering)

Author	Yr	Title	Description	Pages
E.R. Dougherty	94	Optimal mean-absolute-error filtering of gray-scale signals by the morphological hit-or-miss transform	JMIV, vol.=4	255-271
M. Khosravi and R.W. Schafer	96	Template matching based on GS hit-or-miss transform	IEETI1, vol.=5, no.=6	1060-1066

- Table A.7: Citation:Soft and recursive filtering.

Author	Yr	Title	Description	Pages
S. Chen et. al.	95	Recursive erosion, dilation, opening, and closing transforms	IEETI1, vol.=4, no.=3	335-345
P. Kuosmanen et. al.	93	Analysis and extensions of soft morphological filters	SPIE, vol.=1902, no.=IV	41-52
P. Kuosmanen et. al.	95	Soft morphological filtering	JMIV, vol.=5, no.=3	231-262
P. Kuosmanen et. al.	95	Shape preservation criteria and optimal soft morphological filtering	IEETI1, vol.=5, no.=4	319-335
C. Laÿ	87	Recursive algorithms in MM	Acta Stereologica, vol.=6	691-696
D. C. Pu et. al.	95	Threshold decomposition of gray-scale soft morphology into binary soft morphology	GMIP, vol.=57, no.=6	522-526
D. C. Pu et. al.	95	Threshold decomposition algorithm for gray-scale soft morphological operations part II:erosion	IECIPAIA, no.=410	757-761
F.Y. Shih et. al.	95	Pipeline architecture for recursive morphological operations	IEETI1, vol.=4, no.=1	11-18
F. Y. Shih et. al.	95	Recursive soft morphological filters	IEETI1, vol.=4, no.=7	1027-1032
F. Y. Shih et. al.	95	Analysis of the properties of soft morphological filtering using threshold decomposition	IEETS1, vol.=43, no.=2	539-544
P. Soille et. al.	96	Recursive implementation of erosions and dilations along discrete lines at arbitrary angles	IEETP2, vol.=18, no.=5	562-567
D. Wang et. al.	95	Shape decomposition and representation using a recursive morphological operation	PATR, vol.=28, no.=11	1783-1792
R. J. D. Wang et. al.	96	Binary Image representation and coding by a double-recursive morphological algorithm	ELSSIGIC, vol.=8	241-266
M. A. Zmuda et. al.	96	Efficient algorithms for the soft morphological operators	IEETP2, vol.=18, no.=11	1142-1147

- Table A.8: Citation:Annular filtering.

Author	Yr	Title	Description	Pages
H. J. A. M. Heijmans et. al	96	Annular filters for binary images	CWI, Research report, Amsterdam, no.=BS-R9604	

continued on next page

continued from previous page (Citation:Annular filtering)

Author	Yr	Title	Description	Pages
C. Ronse et. al.	96	A lattice-theoretical framework for annular filters in morphological image processing	LSIIT, Strasbourg, report, no.=ERII-RR96/19	

- Table A.9: Citation:Slope transforms.

Author	Yr	Title	Description	Pages
L. Dorst et. al.	94	Morphological signal processing and slope transform	ELSSIG, vol.=38	79-98
H. J. A. M. Heijmans et. al.	95	Lattice Calculus of the Morphological Slope Transform	Report BS-R9531, CWI, Amsterdam	
H. J. A. M. Heijmans et. al.	96	Morphology on Convolution Lattices with Application to the Slope Transform and Random Set Theory	Report BS-R9603, CWI, Amsterdam	
H. J. A. M. Heijmans et. al.	97	Lattice Calculus of the Morphological Slope Transform	ELSSIG, vol.=59, no.=1	17-42
H. J. A. M. Heijmans et. al.	98	Morphology on convolution lattices with applications to the slope transform and random set theory	JMIV, vol.=8, no.=3	199-214
P. Maragos	94	Morphological systems: slope transforms and max-min difference and differential equations, and sampling	BOOK-SER-94	149-160
P. Maragos	94	Morphological systems: slope transforms and max-min difference and differential equations	ELSSIG, vol.=38	57-77
P. Maragos	95	Slope transforms: theory and application to nonlinear signal processing	IEETS1, vol.=43, no.=4	864-877
P. Maragos	96	Differential morphology and image processing	IEETI1, vol.=5, no.=6	922-937

- Table A.10: Citation:Sequential filtering.

Author	Yr	Title	Description	Pages
J. Goutsias et. al.	95	Morphological Operators for Image Sequences	CVIU, vol. 62	326-346
H. J. A. M. Heijmans	95	A new class of alternating sequential filters	IEEPRC: Workshop on Nonlinear Signal and Image Proc., ed.= I. Pitas, Greece, vol.=I	30-33
A. Morales et. al.	B. 95	Morphological pyramids with alternating sequential filters	IEETI1, vol.=4, no.=7	965-977

- Table A.11: Citation:Scale-space.

Author	Yr	Title	Description	Pages
R. van den Boomgaard et. al.	94	Towards a morphological scale-space theory	BOOK-O	631-640

continued on next page

continued from previous page (Citation:Scale-space)

Author	Yr	Title	Description	Pages
R. van den Boomgaard	94	The morphological structure of images: the differential equations of morphological scale space	IEETP2, vol.=16	1101-1113
Paul T. Jackway	95	On Dimensionality in Multiscale Morphological Scale-Space with Elliptic Poweroid Structuring Functions	JVCIR, vol.=6, no.=2	189-195
Paul T. Jackway et. al.	96	Scale-Space Properties of the Multiscale Morphological Dilation-Erosion	IEETP2, vol.=18, no.=1	38-51
K. R. Park et. al.	96	Scale-space using MM	IEETP2, vol.=18, no.=11	1121-1126
G. Sapiro et. al.	95	Area and length preserving geometric invariant scale-spaces	IEETP2, vol.=17, no.=1	67-72

- Table A.12: Citation:Sampling.

Author	Yr	Title	Description	Pages
D. Florêncio et. al.	94	Critical morphological sampling and applications to image coding	BOOK-SER-94	109-116
R. M. Haralick et. al.	89	The digital morphological sampling theorem	IEETA1, vol.=37	2067-2090
H. J. A. M. Heijmans et. al.	91	Morphological sampling	CVGIPIU, vol.=54	384-400
S.-C. Pei et. al.	94	Image sampling structure conversion by morphological filters	ELSSIGIC, vol.=6	13-24
J. Serra	94	A sampling approach based on equicontinuity	BOOK-SER-94	117-124
K. Sivakumar et. al.	96	Binary random fields, random closed sets, and morphological sampling	IEETI1, vol.=5, no.=6	899-912

- Table A.13: Citation:Geodesic methods.

Author	Yr	Title	Description	Pages
L. Najman et. al.	96	Geodesic saliency of watershed contours and hierarchical segmentation	IEETP2, vol.=18, no.=12	1163-1173
P. Soille	94	Generalized geodesic distances applied to interpolation and shape description	BOOK-SER-94	193-200

- Table A.14: Citation:Coding and compression.

Author	Yr	Title	Description	Pages
P. Brigger et. al.	95	Morphological shape representation for very low bit-rate video coding	ELSSIG, vol.=7, no.=4-6	297-311
R. J. Chen et. al.	95	Three-dimensional morphological pyramid and its application to color image sequence coding	ELSSIG, vol.=44, no.=2	163-180

continued on next page

continued from previous page (Citation:Coding and compression)

Author	Yr	Title	Description	Pages
P.J. Czerepinski et. al.	96	Morphological video coder with conditional smoothing	ELEC-LET, vol.=32, no.=7	645-647
J. Goutsias et. al.	89	Image Coding Via Morphological Transformations: A General Theory	IEEPRC, Conf. on Computer Vision and Pat. Rec. ,San Diego, California	4-8
C. Gu et. al.	94	Application of morphological filters for contour image sequence coding	BOOK-SER-94	125-132
R. Jeannot et. al.	96	Binary image representation and coding by a double-recursive morphological algorithm	ELSSIGIC, vol.=8, no.=3	241-266
X. Kong et. al.	94	A study of pyramidal techniques for image representation and compression	JVCIR, vol.=5	190-203
B. Macq et. al.	91	Multiscale morphological region coding	SPIEPRC-CVCIP, vol.=1606	165-173
L. A. Overturf et. al.	95	Color Image Coding Using Morphological Pyramid Decomposition	IEETI1, vol.=4, no.=2	177-185
P. Salembier	94	Morphological multiscale segmentation for image coding	ELSSIG, vol.=38	359-386
P. Salembier	96	morphological operators for image and video compression	IEETI1, vol.=5, no.=6	881-898
G. Sapiro et. al.	94	Morphological image coding based on a geometric sampling theorem and a modified skeleton representation	JVCIR, vol.=5	29-40

Table A.15: Citation:Segmentation.

Author	Yr	Title	Description	Pages
J. Crespo et. al.	94	The flat zone approach and color images	BOOK-SER-94	85-92
W. Li et. al.	94	morphological segmentation applied to displaced frame difference coding	ELSSIG, vol.=38	45-56
M. Lybanon et. al.	94	Segmentation of diverse image types using opening and closing	Proc. of the 12th IAPR International Conf. on Pat. Rec., Jerusalem, Vol.=I	347-351
B. Marcotegui et. al.	94	Morphological segmentation of image sequences	BOOK-SER-94	101-108
F. Meyer	94	Minimum spanning forests for morphological segmentation	BOOK-SER-94	77-84
M. Pardás et. al.	94	Joint region and motion estimation with morphological tools	BOOK-SER-94	93-100
M. Pardás et. al.	94	3-D morphological segmentation and motion estimation for image sequences	ELSSIG, vol.=38	31-43
P. Salembier et. al.	92	Morphological multiscale image segmentation	SPIEPRC-CVCIP, vol.=1818	620-631
P. Salembier	94	Morphological multiscale segmentation for image coding	ELSSIG, vol.=38	359-386

continued on next page

continued from previous page (Citation:Segmentation)

Author	Yr	Title	Description	Pages
P. Salembier et. al.	95	Flat zones filtering, connected operators, and filters by reconstruction	IEETI1, vol.=4, no.=8	1153-1160
P. Soille	96	Morphological Partitioning of Multispectral Images	JEI, vol.=5, no.=3	252-265
L. Vincent et. al.	89	The morphological approach to segmentation: an introduction	Report, ECOLE	
L. Vincent et. al.	94	Morphological segmentation for textures and particles	Digital Image Processing Methods, edit.=E. R. Dougherty, pub.=Marcel Dekker, New York	43-102
E. Wang et. al.	93	Texture classification and segmentation based on iterative morphological decomposition	JVCIR, vol.=4	197-214

- Table A.16: Citation:Granulometries and shape description.

Author	Yr	Title	Description	Pages
S. Banerjee et. al.	94	C-factor: a morphological shape descriptor	JMIV, vol.=4	43-55
J. A. Bangham et. al.	94	The multiscale morphology decomposition theorem	BOOK-SER-94	179-184
E. J. Breen et. al.	96	Attribute openings, thinnings, and granulometries	CVGIPIU, vol.=64, no.=3	377-389
Y. Chen et. al.	92	Texture classif. by GS morphological granulometries	SPIEPRC-CVCIP, vol.=1818	931-942
Y. Chen et. al.	97	Optimal and adaptive reconstructive granulometric bandpass filters	ELSSIG, vol.=61, no.=1	65-81
E. R. Dougherty	90	Characterization of gray-scale morphological granulometries	SPIE, vol. 1350	129-137
E. R. Dougherty et. al.	92	Morphological texturebased maximum-likelihood pixel classification based on local granulometric moments	PATR, vol.=25, no.=10	1181-1198
E. R. Dougherty	92	Euclidean gray-scale granulometries: representation and umbra inducement	jmiv, vol.=1 7-21	
E. R. Dougherty et. al.	92	Detection of osteoporosis by morphological granulometries	SPIEPRC-CVCIP, San Jose, vol.=1660	
E.R. Dougherty et. al.	95	Morphological pattern-spectrum classification of noisy shapes exterior granulometries	PATR, vol.=28, no.=1	81-98
E. R. Dougherty et. al.	95	Representation of linear granulometric moments for deterministic and random binary Euclidean images	JVCIR, vol.=6, no.=1	69-79
E. R. Dougherty	??	Morphological τ-openings and granulometries: binary to Euclidean gray-scale	BOOK-HARAL	
E. R. Dougherty et. al.	??	Morphological texture-based maximum-likelihood pixel classification based on local granulometric moments	PATR	

continued on next page

continued from previous page (Citation:Granulometries and shape description)

Author	Yr	Title	Description	Pages
C. Gratin et. al.	94	Texture classification using neural networks and local granulometries	BOOK-SER-94	309-316
H. J. A. M. Heijmans	94	MM as a tool for shape description	Shape in Picture, Ed.="Y-L. O et. al., pub.= pringer, Berlin	147-176
R. Jones et. al.	96	Periodic lines: cascades, and application to granulometries	PATRL, vol.=17	1057-1063
E. J. Kraus et. al.	93	Gray-scale granulometries compatible with spatial scalings	ELSSIG, vol.=34	1-17
C. K. Lee et. al.	96	A mathematical morphological approach for segmenting heavily noise-corrupted images	PATR, vol.=29, no.=8	1347-1358
J. Mattioli et. al.	92	Inverse problems for granulometries by erosion	JMIV, vol.=2	217-232
J. T. Newell et. al.	92	Maximum-likelihood morphological granulometric classifiers	SPIEPRC-IAMIP, vol.=1657	386-395
R. Sabourin et. al.	97	Off-line signature verification by local granulometric size distributions	IEETP2, vol.=19, no.=9	976-988
F. Sand et. al.	98	Asymptotic granulometric mixing theorem: morphological estimation of sizing parameters and mixture proportions	PATR, vol.=31, no.=1	53-61
K. Sivakumar et. al.	96	On Estimating Granulometric Discrete Size Distributions of Random Sets	PROCWATRS	

Table A.17: Citation:Watersheds.

Author	Yr	Title	Description	Pages
M. Baccar et. al.	96	Segmentation of range images via data fusion and morphological watersheds	PATR, vol.=29, no.=10	1671-1687
S. Beucher et. al.	93	The morphological approach to segmentation: the watershed transformation	BOOK-DOUGH	433-481
S. Beucher	94	Watershed, hierarchial segmentation and waterfall algorithm	BOOK-SER-94	69-76
P. T. Jackway	96	Gradient watersheds in morphological scale-space	IEETI1, vol.=5, no.=6	913-921
F. Meyer	94	Topographic distance and watershed lines	ELSSIG, vol.=38	113-125
A. Moga et. al.	94	Implementation of a distributed watershed algorithm	BOOK-SER-94	281-288
A. Moga et. al.	97	Parallel image component labeling with watershed transformations	IEETP2, vol.=19, no.=5	441-450
L. Najman et. al.	94	Watershed of a continuous function	ELSSIG, vol.=38	99-112
L. Shafarenko et. al.	97	Automatic watershed segmentation of randomly textured color images	IEETI1, vol.=6, no.=11	1530-1544

continued on next page

continued from previous page (Citation:Watersheds)

Author	Yr	Title	Description	Pages
L. Vincent et. al.	91	Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations	IEETP2, vol.=13, no.=6	583-598

Table A.18: Citation:Skeletonisation.

Author	Yr	Title	Description	Pages
D. Attali et. al.	97	Computing and simplifying 2D and 3D continuous skeletons	CVGIPIU, vol.=67, no.=3	161-273
S. Beucher	94	Digital skeletons in euclidean and geodesic spaces	ELSSIG, vol.=38	127-141
B. K. Jang et. al.	90	Analysis of Thinning Algorithms Using MM	IEETP2, vol.=12, no.=6	541-551
L. Ji et. al.	92	Fast Homotopy-Preserving Skeletons Using MM	IEETP2, vol.=14, no.=6	653-664
R. Kimmel et. al.	95	Skeletonization via distance maps and level sets	CVIU, vol.=62, no.=3	382-391
R. Kresch et. al.	92	Morphological Multi-Structuring-Element Skeleton and Its Applications	PROCISSSE, Paris	166-169
R. Kresch et. al.	93	Two-Sided Skeleton - A Representation Composed of Both Positive and Negative Morphological Elements	PROCIWMM, Barcelona	145-150
R. Kresch et. al.	94	Morphological reduction of skeleton redundancy	ELSSIG, vol.=38	143-151
R. Kresch et. al.	94	Morphological reduction of skeleton redundancy	PROCIWMM, Barcelona	145-150
R. Kresch et. al.	94	Multi-Parameter Skeleton Decomposition	BOOK-SER-94	141-148
R. Kresch et. al.	94	Multi-Parameter Skeleton Decomposition	PROCMM, ISMM'94	141-148
R. Kresch et. al.	94	Skeleton Redundancy Reduction Based on a Generalization of Convexity	EUSIPCO, Edinburgh	
R. Kresch et. al.	95	An Efficient Coding Scheme for Binary Images Based on the Morphological Skeleton Representation	PROCIEE18, Israel	
R. Kresch et. al.	95	New Morphological Skeleton Properties Leading to Its Efficient Coding	IEEWNSIP, Greece	
R. Kresch et. al.	95	Quadtree and Bit-Plane Decomposition as Particular Cases of the Generalized Morphological Skeleton	IEEWNSIP, Greece	
J. Madrid et. al.	93	Topological considerations on gray level skeletonization	SPIEPRC-CVCIP, vol.=1818	392-401
P. Maragos	86	Morphological skeleton representation and coding of binary images	IEETA1, vol.=34	1228-1244
P. Maragos et. al.	88	Threshold parallelism in morphological feature extraction, skeletonization and pattern spectrum	SPIEPRC-CVCIP	106-115

continued on next page

continued from previous page (Citation:Skeletonisation)

Author	Yr	Title	Description	Pages
G. Matheron	88	Examples of Topological Properties of Skeletons	MM and Image Analysis. II: Theoretical Advances, Ed.=J. Serra, pub.=Academic Press, London	ch.=11
G. Matheron	88	Filters and lattices	MM and Image Analysis. II: Theoretical Advances, Ed.=J. Serra, pub.=Academic Press, London	ch.=6
T.-W. Pai et. al.	94	Boundary-constrained morphological skeleton minimization and skeleton reconstruction	IEETP2, vol.=16, no.=2	201-208
J. M. Reinhardt et. al.	96	Comparison between the morphological skeleton and morphological shape decomposition	IEETP2, vol.=18, no.=9	951-957
J. Serra	91	Skeleton decompositions	Preprint	
M. Schmitt	94	One pixel thick skeletons	BOOK-SER-94	257-264
R. C. Staunton	96	An analysis of hexagonal thinning algorithms and skeletal shape representation	PATR, vol.=29, no.=7	1131-1146
Z. Zhou et. al.	92	Analysis and implementation of morphological skeleton transforms	CSSP, vol.=11, no.=1	253-280

- Table A.19: Citation:Random models and tesselations.

Author	Yr	Title	Description	Pages
A. J. Baddely et. al.	95	Incidence and lattice calculus with applications to stochastic geometry and image analysis	AAECC, vol.=6, no.3	129-146
E. Bertin et. al.	94	Optimization in Voronoï diagrams	BOOK-SER-94	209-216
J. C. Handley et. al.	97	Maximum-likelihood estimation for the two-dimensional discrete Boolean random set and function models using multidimensional linear samples	GMIP, vol.=59, no.=4	221-231
J. Goutsias et. al.	91	Discrete Random Set Models for Shape Synthesis and Analysis	SPIEPRC-CVCIP, Boston, USA	11-13
J. Goutsias	96	Morphological Analysis of Random Sets: An Introduction	PROCWATRS	
J. C. Handley et. al.	96	Optimal nonlinear filter for signal-union-noise and runlength analysis in the directional one-dimensional discrete Boolean random set model	ELSSIG, vol.=51, no.=3	147-166
E. Kalaitzis et. al.	94	Performance analysis of a morphological Voronoï tessellation algorithm	BOOK-SER-94	201-208
T. Lee et. al.	94	A stochastic tessellation of digital space	BOOK-SER-94	217-224

continued on next page

continued from previous page (Citation:Random models and tesselations)

Author	Yr	Title	Description	Pages
J. L. Quenec'h et. al.	94	Liquid phase sintered materials modelling by random closed sets	BOOK-SER-94	225-232
N.D. Sidiropoulos et. al.	94	Algebraic analysis of the generating functional for discrete random sets and statistical inference for intensity in the discrete Boolean random-set model	JMIV, vol.=4	273-290
K. Sivakumar et. al.	94	Monte-Carlo estimation of morphological granulometric discrete size distributions	BOOK-SER-94	233-240

- Table A.20: Citation:Algorithmic techniques.

Author	Yr	Title	Description	Pages
E. Breen et. al.	94	An evaluation of priority queues for MM	BOOK-SER-94	249-256
R. Brémond et. al.	94	Morphogenesis simulations with lattice gas	BOOK-SER-94	297-304
J. Brown et. al.	96	A morphological point thinning algorithm	PATRL, vol.=17, no.=2	197-207
E. R. Dougherty et. al.	94	Computational MM	ELSSIG, vol. =38	21-29
M. v Droogenbroeck	94	On the implementation of morphological operations	BOOK-SER-94	241-248
S. Fejes et. al.	94	An efficient implementation technique of adaptive morphological operations	BOOK-SER-94	273-280
M. Khosravi et. al.	94	Implementation of linear digital filters based on morphological representation theory	IEETS1, vol.=42, no.=9	2264-2275
S. J. Ko et. al.	95	Block basis matrix implementation of morphological open-closing and clos-opening	IEESL1, vol.=2	7-9
L. Lucke et. al.	95	A digit-serial architecture for gray-scale morphological filtering	IEETI1, vol.=4, no.=3	387-391
J. B. T. M. Roerdink et. al.	94	Visualization of Minkowski operations by computer graphics techniques	BOOK-SER-94	289-296

- Table A.21: Citation:Decomposition techniques.

Author	Yr	Title	Description	Pages
J. A. Bangham et. al.	96	Multiscale Nonlinear Decomposition: The Sieve Decomposition Theorem	IEETP2, vol.=18, no.=5	529-539
G. J. F. Banon et. al.	93	Decomposition of mappings between complete lattices by mathematical morphology: Part I. General Lattices	ELSSIG, vol.=38	299-327

continued on next page

continued from previous page (Citation:Decomposition techniques)

Author	Yr	Title	Description	Pages
G. J. F. Banon et. al.	94	Set operator decomposition and conditionally translation invariant elementary operators	BOOK-SER-94 5-12	
R. van den Boomgaard	??	Decomposition of structuring elements	IEETP2	
O. I. Camps et. al.	96	Gray-scale SE decomposition	IEETI1, vol.=5, no.=1	111-120
P. K. Ghosh	96	The indecomposability problem in binary morphology: an algorithmic approach	JMIV, vol.=6, no.=2/3	169-198
C.T. Huang et. al.	94	A Euclidean distance transform using grayscale morphology decomposition	IEETP2, vol.=16	443-448
R. Jones et. al.	94	Algorithms for the Decomposition of Gray-Scale Morphological Operations	IEETP2, vol.=16, no.=6	581-588
X. Kong et. al.	93	A Comparison of Pyramidal Image Decomposition Techniques for Image Compression	SPIEPRC-IAMIP, San Diego, California	11-16
W. Li et. al.	95	Composite morphological filters in multiresolution morphological decomposition	IECIPAIA, no.=410	752-756
F. Mount et. al.	G. 91	Combinatorial and computational aspects of Minkowski decomposition	Con cs, vol.=119	107-124
C. H. Richardson et. al.	91	A lower Bound for SE Decompositions	IEETP2, vol.=13, no.=4	365-369
L.A. Overturf et. al.	95	Color image coding using morphological pyramid decomposition	IEETI1, vol.=4, no.=2	177-185
H. Park et. al.	94	Doptimal ecomposition of Convex morphological SEs for 4-Connected Parallel Array Processors	IEETP2, vol.=16, no.=3	304-313
H. Park et. al.	95	Decomposition of arbitrarily shaped morphological SEs	IEETP2, vol.=17, no.=1	2-15
S.-C. Pei et. al.	94	3-D spatiotemporal subband decompositions for hierarchical compatible video coding by MM	ELSSIGIC, vol.=6	83-99
S.-C. Pei et. al.	95	Hierarchical image representation by mathematical morphology subband decomposition	PATRL, vol.=16	183-192
I. Pitas et. al.	90	Morphological shape decomposition	IEETP2, vol.=12	38-45
P. Salembier et. al.	92	Size-sensitive multiresolution decomposition of images with rank order based filters	ELSSIG, vol.=27, no.=2	205-241
B. Singh et. al.	96	A simplified algorithm for approximate separable decomposition of morphological templates	PATR, vol.=29, no.=9	1519-1522

continued on next page

continued from previous page (Citation:Decomposition techniques)

Author	Yr	Title	Description	Pages
P. Sussner et. al.	97	Decomposition of gray-scale morphological templates using the rank method	IEETP2, vol.=19, no.=6	649-658
A. Toet	89	A morphological pyramidal image decomposition	PATRL, vol.=9	255-261
J. Xu	91	Decomposition of convex polygonal morphological SEs into neighbourhood subsets	IEETP2, vol.=13, no.=2	153-162
J. Xu	96	Morphological decomposition of 2-D binary shapes into simpler shape parts	PATRL, vol.=17, no.=7	759-769
J. Xu	96	Morphological decomposition of 2-D binary shapes into conditionally maximal convex polygons	PATR, vol.=29, no.=7	1075-1104
X. Zhuang et. al.	86	Morphological SE decomposition	CVGIP, vol.=35	370-382
X. Zhuang	94	Decomposition of morphological structuring elements	JMIV, vol.=4	5-18

- Table A.22: Citation:Fast algorithms.

Author	Yr	Title	Description	Pages
A. Bleau et. al.	92	A new set of fast algorithms for mathematical morphology: I. Idempotent geodesic transforms	CVGIPIU, vol.=56	178-209
R. v. d. Boomgaard et. al.	92	Methods for Fast morphological image transforms using bitmapped binary images	CVGIP, vol.=54, no.=3	252-258
P. Brigger et. al.	94	The geodesic morphological skeleton and fast transformation algorithms	BOOK-SER-94	133-140
M. van Droogenbroeck et. al.	96	Fast computation of morphological operations with arbitrary SEs	PATRL, vol.=17, no.=14	1451-1460
J. Kisacanin et. al.	94	A fast thresholded linear convolution representation of morphological operations	IEETI1, vol.=3	455-457
S. J. Ko et. al.	96	Fast recursive algorithms for morphological operators based on the basis matrix representation	IEETI1, vol.=5, no.=6	1073-1077
J. Pecht	85	Speeding up Successive Minkowski Operations	PATRL, vol.=3	113-117
D. Schonfeld and J. Goutsias	88	A Fast Algorithm for the Morphological Coding of Binary Images	SPIEPRC-CVCIP, Cambridge, Massachusetts	9-11
M. H. Sedaaghi	97	Direct Implementation of open-closing in morphological filtering	ELEC-LET, vol.=33, no.=3	198-199
M. H. Sedaaghi et. al.	97	A Direct technique for morphological filters	ICEE, 5th. Conf.	7.80-7.85
M. H. Sedaaghi et. al.	97	Real-time implementation of grey-scale morphological operators	ELEC-LET, vol.=33, no.=21	1761-1763

continued on next page

continued from previous page (Citation:Fast algorithms)

Author	Yr	Title	Description	Pages
M. H. Sedaaghi et. al.	98	A reliable hardware for grey-scale morphological filtering	ISCSDSP 1st. 6-8 April, Sheffield	147-150
M. H. Sedaaghi et. al.	98	Hardware for grey-scale morphological filtering	PDPTA 13-16 July, Las Vegas	
L. Vincent	94	Fast grayscale granulometry algorithms	BOOK-SER-94	265-272
D. Wang et. al.	94	A fast implementation of 1-D GS morphological filters	IEETC1, vol.=41, no.=9	634-636

- Table A.23: Citation:Applications.

Author	Yr	Title	Description	Pages
J. A. Bangham et. al.	94	Multiscale median and morphological filters for 2D pattern recognition	ELSSIG, vol.=38	387-415
J. Cardillo et. al.	96	Target recognition in a cluttered scene using MM	PATR, vol.=29, no.=1	27-49
J. Casas et. al.	94	Morphological scheme for morphological analysis of epidermal biopsy images	BOOK-SER-94	325-332
A. R. Chaudhuri et. al.	95	Detection of occluded circular objects by morphological operators	ELSSIG, vol.=46, no.=2	233-242
M.-Y. Chen et. al.	95	Variable duration hidden Markov model and morphological segmentation for handwritten word recognition	IEETI1, vol.=4, no.=12	1675-1688
C. H. H. Chu et. al.	89	Impulsive noise suppresion and background normalization of ECG signals using morphological operators	IEETB1, vol.=36, no.=2	262-273
E. R. Dougherty et. al.	96	Bayesian morphological peak estimation and its application to chromosome counting via fluorescence in situ hybridization	PATR, vol.=29, no.=6	987-996
M. Duff	79	Parallel processors for digital image processing	AIDIM	265-279
F. Gerritsen et. al.	81	Design and use of DIP-1: A fast flexible and dynamically microprogrammable image processor	PATTERN RECOGNITION, vol.=14	319-330
M. J. E. Golay	69	Hexagonal parallel pattern transformations	IEETC4, vol.=C-18	733-740
D. Graham et. al.	80	The Diff3 analyzer: A parallel/serial Golay image processor	RTMIP	163-182
B. V. Howard	94	Single object geometry - the stereology of registered serial sections	BOOK-SER-94	305-308
J. C. Klein et. al.	77	The texture analyzer	J. Microscopy, vol.=95	349-356
J.-S. Kwon et. al.	96	Obtaining a 3-D orientation of projective textures using a morphological method	PATR, vol.=29, no.=5	725-732

continued on next page

continued from previous page (Citation:Applications)

Author	Yr	Title	Description	Pages
B. Laÿ	94	Image processing: a key to success in industrial applications	BOOK-SER-94	341-352
S. M. Lea et. al.	92	Finding mesoscale ocean structures with MM	RSE, vol.=44	25-33
L. Leboucher et. al.	94	Grey-tone skeletons of elongated objects using the concept of morphological automaton. Application to images of DNA molecules	PATRL, vol.=15	309-315
W. Li et. al.	97	Residues of morphological filtering by reconstruction for texture classification	PATR, vol.=30, no.=7	1081-1093
S. Liang et. al.	94	A morphological approach to text string extraction from regular periodic overlapping text/background images	CVGIP, Graphical Models and Image Proc., vol.=56, no=5	402-413
S. Liang et. al.	97	Segmentation of handwritten interference masks using multiple directional stroke planes and reformalized morphological approach	IEETI1, vol.=6, no=8	1195-1202
T.-S. Liu et. al.	95	Fast color image quantization with error diffusion and morphological operations	ELSSIG, vol.=43, no.=3	293-303
M.W. Mak et. al.	94	A lip-tracking system based on morphological processing and block matching techniques	ELSSIGIC, vol.=6	335-348
P. Maragos	88	Optimal Morphological approaches to Image Matching and Object Detection	Proc. 2nd International Conference on Computer Vision, Florida	
P. Maragos	88	Morphology-based symbolic image modeling, multi-scale nonlinear smoothing, and pattern spectrum	IEEProc. Computer Society Conference on Computer Vision and Pattern Recognition, Ann Arbor	
F. Marqués et. al.	94	Automatic quantification of spine parameters from X-ray images by means of morphological tools	BOOK-SER-94	333-340
S. Marshall et. al.	94	Fusion of MR and CT images of the human brain using multiresolution morphology	BOOK-SER-94	317-324
G. K. Matsopoulos et. al.	95	Application of morphological pyramids: fusion of MR and CT phantoms	JVCIR, vol.=6, no.=2	196-207
C. Mering et. al.	94	Application of morphological operators to supervised multidimensional data classification	BOOK-SER-94	361-368
J. A. Noble	96	The Effect of Morphological Filters on Texture Boundary Localization	IEETP2, vol.=18, no.=5	554-561
R. A. Peters II	95	A new algorithm for image noise reduction using MM	IEETI1, vol.=4, no.=5	554-568

continued on next page

continued from previous page (Citation:Applications)

Author	Yr	Title	Description	Pages
K. Preston et. al.	92	Signal and image processing using 3D binary ranking transforms	CSSP, vol.=11, no.=1	137-151
S. J. Rees et. al.	95	Conditional morphological operators for direct feature extraction and enhancement	IECIPAIA, no.=410	747-751
A. I. T. Rowstron et. al.	95	Implementing mathematical morphological in ISETL-LINDA	IECIPAIA, no.=410	847-851
K. R. Rystrom et. al.	95	Optimal single-stage restoration of subtractive noise corrupted images by a morphological closing	JEI, vol.=4, no.=3	
D. Schonfeld et. al.	89	Parametric Morphological Filters for Pattern Restoration	Proc. of the Workshop "Multidimensional Signal Processing", R, Asilomar Conference Center, Pacific Grove, California	6-8
D. Schonfeld et. al.	89	Optimal Morphological Filters for Pattern Restoration	SPIEPRC-CVCIP, Philadelphia, Pennsylvania	8-10
D. Schonfeld	94	Optimal SEs For the Morphological Pattern Recognition of Binary Images	IEETP2, vol.=16, no.=6	589-601
D. Schonfeld and J. Goutsias	91	Optimal Morphological Pattern Restoration from Noisy Binary Images	IEETP2, vol.=13	14-29
M. H. Sedaaghi et. al.	97	A syntactic approach to fingerprint classification	CSICC'97, Tehran, 22-24 Dec.	31-36
J. Serra	87	Morphological optics	Journal of Microscopy, vol.=145	1-22
N.D. Sidiropoulos et. al.	94	Optimal filtering of digital binary images corrupted by union/intersection noise	IEETI1, vol.=3	382-403
D. Sinha et. al.	90	Discrete Black and White Object Recognition Via Morphological Functions	IEETP2, vol.=12, no.=3	275-293
M. M. Skolnick	86	Application of Morphological Transformations to Athe analysis of 2D Electrophoretic Gels of Biological Materials	CVGIP, vol.=35, no.=3	283-305
X. Song et. al.	93	Robust edge detectors based on morphological filters	PATRL, vol.=14	889-894
J. P. Thiran et. al.	96	Morphological Feature Extraction for the Classification of Digital Images of Cancerous Tissues	IEETB1, vol.=43, no.=10	1011-1020
P. E. Trahanias	93	An approach to QRS complex detection using mathematical morphology	IEETB1, vol.=40, no.=2	201-205
J. G. Verly et. al.	93	Adaptive MM for Range Imagery	IEETI1, vol.=2, no.=2	272-275
T. Viero et. al.	95	Morphological extraction of line networks from noisy low-contrast images	JVCIR, vol.=6, no.=4	335-347
L. Vincent	88	MM on graphs	SPIEPRC-CVCIP, Cambridge	95-105

continued on next page

continued from previous page (Citation:Applications)

Author	Yr	Title	Description	Pages
L. Vincent	89	Graphs and MM	ELSSIG, vol.=16	365-388
L. Vincent	89	MM for graphs applied to image description and segmentation	Proc. Electronic Imaging West, Pasadena, vol.=1	313-318
L. Vincent	93	Morphological GS reconstrcution in image analysis: applications and efficient algorithms	IEETI1, vol.=2, no.=2	176-201
L. Vincent	93	Grayscale area openings and closings, their efficient implementation and applications	MM and its Applications to Signal Processing, Barcelona, Spain	22-27
L. Vincent	93	Morphological grayscale reconstruction in image analysis: efficient algorithms and applications	IEETI1, vol.=2	176-201
D. Wang et. al.	96	Bounded gray-level morphology and its applications to image representation	IEETC1, vol.=5, no.=6	1067-1072
G. R. Wilson	94	Morphological operations on crack coded binary images	IEETI1, vol.=143, no.=3	171-176
D. L. Wilson et. al.	95	Morphological restoration of AFM images	Langmuir, vol.=11	265-272
S. S. Wilson	93	Gray Level Image Enhancement Using a Projected Thickening	SPIE-NFIP, vol.=1902	20-29
H. Yamada et. al.	93	Directional MM and Reformalized Hough Transformation for the Analysis of Topographic Maps	IEETP2, vol.=15, no.=4	380-387
J. Yang et. al.	95	Boundary detection using MM	PATRL, vol.=16, no.=12	1277-1286
P.-F. Yang et. al.	95	Min-max classifiers: learnability, design and application	PATR, vol.=28, no.=6	879-899
J. Yang et. al.	95	Directional morphology and its application in boundary detection	IECIPAIA, no.=410	742-746
R. D. Zhang et. al.	94	Convexity dependent morphological transformations for mode detection in cluster analysis	PATR, vol.=27	135-148

- Table A.24: Citation:Books.

Author	Yr	Title	Description
E. R. Dougherty et. al.	87	Image Processing - Continuous to Discrete	Prentice-Hall, Englewood Cliffs, NJ
E. R. Dougherty	92	An Introduction to Morphological Image Processing	SPIE, Tutorial Texts in Optical Eng., vol.=TT 9, Washington
E. R. Dougherty	93	MM in Image Processing	Marcel Dekker, New York
E. R. Dougherty et. al.	97	Nonlinear Filters for Image Proc.	SPIE-OEP
C. R. Giardina et. al.	88	Morphological Methods in Image and Signal Processing	Prentice-Hall, Englewood Cliffs, NJ.

continued on next page

continued from previous page (Citation:Books)

Author	Yr	Title	Description
R. M. Haralick	??	MM: Theory and Hardware	Oxford University Press
H.J.A.M. Heijmans	94	Morphological Image Operators	Academic Press, Boston
T. S. Huang	81	2D Digital Signal Processing II: Transforms and Median Filters	Springer, Berlin
G. Matheron	75	Random sets and integrated geometry	Wiley, New York
P. Maragos et. al.	96	Mathematical morphology and its applications to image and signal processing	Kluwer Academic Publishers, Computational Imaging and Vision, Boston
I.S. Molchanov	93	Limit theorems for unions of random closed sets	Springer, Lecture Notes in Mathematics, vol.=1561
Y-L. O et. al.	92	Shape in Picture: Mathematical Description of Shape in Grey-Level Images	NATO ASI Series, Driebergen, The Netherlands, Springer, Berlin, vol.=126
J. Serra	82	Image analysis and MM	Academic Press, New York
J. Serra	88	Image analysis and MM	Academic Press, New York, Vol. =2: Theoretical advances
J. Serra et. al.	94	MM and its applications to image processing	Kluwer Academic Publishers, The Netherlands
M. Schmitt et. al.	??	Morphology: Algorithms and Applications	Cambridge University Press
R. Schneider	93	Convex Bodies: the Brunn-Minkowski Theory	Cambridge University Press, Cambridge
J. C. Simon	89	From Pixels to Features	North Holland, Amsterdam

Table A.25: Citation:Thesis.

Author	Yr	Title	Description	Tp
R. van den Boomgaard	92	MM: Extensions towards comp. vision	University of Amsterdam	TH
C. Gu	95	Multivalued morphology and segmentation-based coding	École Polytechnique Fédérale De Lausanne	TH
P. T. Jackway	94	Morphological Scale-Space With Application to 3D Object Recognition	Queensland Univ. of Tech.	TH
R. Kresch	95	Morphological image representation for coding applications	Israel Institute of Tech.	TH
B. Kruse	77	Design and implementation of a picture processor	University of Linkoeping, Linkoeping, Sweden	TH
P. Maragos	85	Unified Theory of Translation-Invariant Systems with Applications to Morphological Analysis and Coding of Images	School of Electrical Engineering, Georgia Ins. of Tech., Atlanta	TH
P. F. M. Nacken	94	Image analysis methods based on hierarchies of graphs and multi-scale MM	University of Amsterdam	TH
B. J. H. Verwer	91	Distance Transform: Metrics, Algorithms and Applications	Delft University of Technology	TH

Appendix B

Existing relations & proof of some equations

B.1 Existing relations

The following relations are valid for the operators defined so far. They have been collected from literature and we have completed and corrected them based on our definitions. Some of the relations are proved in appendix B . The symbol (SA) before some of the equations means that there exists the same relations for dilation/erosion as the relation mentioned for Minkowski addition/subtraction

- $$A \overset{M}{\oplus} B = A \oplus \check{B}. \tag{B.1.1}$$

$$A \overset{M}{\ominus} B = A \ominus \check{B}. \tag{B.1.2}$$

- *Duality:*

$$A \overset{M}{\ominus} B = (A^c \overset{M}{\oplus} B)^c. \tag{B.1.3}$$

$$A \overset{M}{\oplus} B = (A^c \overset{M}{\ominus} B)^c. \tag{B.1.4}$$

$$A \ominus B = (A^c \oplus B)^c. \tag{B.1.5}$$

$$A \oplus B = (A^c \ominus B)^c. \tag{B.1.6}$$

$$f \ominus g = (f^c \oplus \check{g})^c. \tag{B.1.7}$$

$$f \oplus g = (f^c \ominus \check{g})^c. \tag{B.1.8}$$

$$A \circ B = (A^c \bullet \check{B})^c. \tag{B.1.9}$$

$$A \bullet B = (A^c \circ \check{B})^c. \tag{B.1.10}$$

$$f \circ g = (f^c \bullet \check{g})^c. \tag{B.1.11}$$

$$f \bullet g = (f^c \circ \check{g})^c. \tag{B.1.12}$$

- *Commutative:*

$$A \overset{M}{\oplus} B = B \overset{M}{\oplus} A. \tag{B.1.13}$$

$$A \oplus B = \check{B} \oplus \check{A}. \tag{B.1.14}$$

$$f \oplus g = \check{g} \oplus \check{f}. \tag{B.1.15}$$

- *Distributive:* (SA)

$$A \overset{M}{\oplus} (\bigcup_{i \in I} B_i) = \bigcup_{i \in I} (A \overset{M}{\oplus} B_i). \tag{B.1.16}$$

$$A \overset{M}{\ominus} (\bigcap_{i \in I} B_i) = \bigcap_{i \in I} (A \overset{M}{\ominus} B_i). \tag{B.1.17}$$

$$(f \vee g) \oplus h = (f \oplus h) \vee (g \oplus h). \tag{B.1.18}$$

$$(f \wedge g) \ominus h = (f \ominus h) \wedge (g \ominus h). \tag{B.1.19}$$

- (SA)

$$(A \overset{M}{\oplus} B) \overset{M}{\oplus} C = (A \overset{M}{\oplus} C) \overset{M}{\oplus} B. \tag{B.1.20}$$

$$(f \oplus g) \oplus h = (f \oplus h) \oplus g. \tag{B.1.21}$$

- *Associative (chain rule):*

$$A \overset{M}{\oplus} D = A \overset{M}{\oplus} (B \overset{M}{\oplus} C) = (A \overset{M}{\oplus} B) \overset{M}{\oplus} C. \tag{B.1.22}$$

$$(A \overset{M}{\ominus} B) \overset{M}{\ominus} C = (A \overset{M}{\ominus} C) \overset{M}{\ominus} B = A \overset{M}{\ominus} (B \overset{M}{\oplus} C). \tag{B.1.23}$$

$$(A \oplus B) \oplus C = A \oplus (B \overset{M}{\oplus} C). \tag{B.1.24}$$

$$(A \ominus B) \ominus C = A \ominus (B \overset{M}{\oplus} C). \tag{B.1.25}$$

$$(f \oplus g) \oplus h = f \oplus (g \oplus h). \tag{B.1.26}$$

$$(f \ominus g) \ominus h = f \ominus (g \oplus h). \tag{B.1.27}$$

The length (*len*) of the input signal after morphological operations equals to its original size (e.g. $len(f) = len(f \oplus g)$). However Equations B.1.26 and B.1.27 are true only if $len(g \oplus h)$ is extended to $len(g) + len(h) - 1$, otherwise they can not be true. As an example let

$$\begin{aligned} f &= \{8, 2, 2, 7, 1, 1, 3, 0, 4, 7\}, \\ g &= \{1, 2, \underline{3}, 2, 1\}, \\ h &= \{0, \underline{-2}, 0\}, \end{aligned} \tag{B.1.28}$$

(underlined values in g and h show the position of their origins). $len(f) = 10$, $len(f \oplus g) = 10$, $len((f \oplus g) \oplus h) = 10$, $len((f \ominus g) \ominus h) = 10$, $len(g) = 5$,

$len(h) = 3$, $len(g \oplus h) = 7 (> len(g))$. Then we will have ($*$ means it has not been defined)

$$g \oplus h = \{*, -1, 0, \underline{3}, 1, 2, *\},$$
$$f \oplus g = \{*, *, 9, 7, 8, 7, 3, 6, 6, *, *\},$$
$$(f \oplus g) \oplus h = \{*, *, *, 10, 8, 9, 8, 4, *, *, *\} \equiv f \oplus (g \oplus h),$$
$$f \ominus g = \{*, *, 0, 0, 0, -1, -1, 0, -2, *, *\},$$
$$(f \ominus g) \ominus h = \{*, *, *, -1, -2, -2, -1, -3, *, *, *\} \equiv f \ominus (g \oplus h).$$

The extended relation of the above equations can be written as:

$$A \overset{M}{\ominus} (B_1 \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_n) = (\cdots A \overset{M}{\ominus} B_1) \overset{M}{\ominus} \cdots \overset{M}{\ominus} B_n). \tag{B.1.29}$$

$$f \ominus (g_1 \oplus \cdots \oplus g_n) = (\cdots (f \ominus g_1) \ominus \cdots \ominus g_n). \tag{B.1.30}$$

- *Scale invariant:* (SA)

$$\alpha A \overset{M}{\oplus} \alpha B = \alpha (A \overset{M}{\oplus} B). \tag{B.1.31}$$

$$\alpha A \overset{M}{\ominus} \alpha B = \alpha (A \overset{M}{\ominus} B). \tag{B.1.32}$$

$$\alpha f \oplus \alpha g = \alpha (f \oplus g). \tag{B.1.33}$$

$$\alpha f \ominus \alpha g = \alpha (f \ominus g). \tag{B.1.34}$$

$$\alpha A \circ \alpha B = \alpha (A \circ B). \tag{B.1.35}$$

$$\alpha A \bullet \alpha B = \alpha (A \bullet B). \tag{B.1.36}$$

$$\alpha f \circ \alpha g = \alpha (f \circ g). \tag{B.1.37}$$

$$\alpha f \bullet \alpha g = \alpha (f \bullet g). \tag{B.1.38}$$

where α is a real number.

- *Compatibility under change of scale:* (SA)

$$A \overset{M}{\oplus} \alpha B = \alpha(\frac{A}{\alpha} \overset{M}{\oplus} B). \tag{B.1.39}$$

$$A \overset{M}{\ominus} \alpha B = \alpha(\frac{A}{\alpha} \overset{M}{\ominus} B). \tag{B.1.40}$$

$$f \oplus \alpha g = \alpha(\frac{f}{\alpha} \oplus g). \tag{B.1.41}$$

$$f \ominus \alpha g = \alpha(\frac{f}{\alpha} \ominus g). \tag{B.1.42}$$

Similar compatibility exists for opening and closing

- *Scaling with convex sets:* (SA) The set A is convex if $\forall x, y \in A$, $\alpha \in [0,1]$: $\alpha x + (1-\alpha)y \in A$. If A is convex and α and β are positive real scalars, then:

$$\alpha A \overset{M}{\oplus} \beta A = (\alpha + \beta)A. \tag{B.1.43}$$

$$\alpha f \oplus \beta f = (\alpha + \beta)f. \tag{B.1.44}$$

Therefore we can say that a convex set S is divisible for all integers $\gamma > 0$:

$$A = \underbrace{\frac{1}{\gamma}A \overset{M}{\oplus} \frac{1}{\gamma}A \overset{M}{\oplus} \cdots \overset{M}{\oplus} \frac{1}{\gamma}A}_{\gamma \quad terms}. \tag{B.1.45}$$

or equivalently:

$$\gamma A = \underbrace{A \overset{M}{\oplus} A \overset{M}{\oplus} \cdots \overset{M}{\oplus} A}_{\gamma \quad terms}. \tag{B.1.46}$$

$$\gamma f = \underbrace{f \oplus f \oplus \cdots \oplus f}_{\gamma \quad terms}. \tag{B.1.47}$$

- *Translation Invariance:*

$$(A)_x \overset{M}{\oplus} B = (A \overset{M}{\oplus} B)_x. \tag{B.1.48}$$

$$A \overset{M}{\oplus} (B)_x = (A \overset{M}{\oplus} B)_x. \tag{B.1.49}$$

$$(A)_x \oplus B = (A \oplus B)_x. \tag{B.1.50}$$

$$A \oplus (B)_x = (A \oplus B)_{-x}. \tag{B.1.51}$$

$$(f)_x \oplus g = (f \oplus g)_x. \tag{B.1.52}$$

$$f \oplus (g)_x = (f \oplus g)_{-x}. \tag{B.1.53}$$

$$(A)_x \overset{M}{\ominus} B = (A \overset{M}{\ominus} B)_x. \tag{B.1.54}$$

$$A \overset{M}{\ominus} (B)_x = (A \overset{M}{\ominus} B)_{-x}. \tag{B.1.55}$$

$$(A)_x \ominus B = (A \ominus B)_x. \tag{B.1.56}$$

$$A \ominus (B)_x = (A \ominus B)_{-x}. \tag{B.1.57}$$

$$(f)_x \ominus g = (f \ominus g)_x. \tag{B.1.58}$$

$$f \ominus (g)_x = (f \ominus g)_x. \tag{B.1.59}$$

The following equations will then be true:

$$\begin{aligned} &A \overset{M}{\oplus} B_1 \overset{M}{\oplus} \cdots \overset{M}{\oplus} (B_n)_x \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_N = \\ &(A \overset{M}{\oplus} B_1 \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_n \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_N)_x. \end{aligned} \tag{B.1.60}$$

$$\begin{aligned} &f \oplus g_1 \oplus \cdots \oplus (g_n)_x \oplus \cdots \oplus g_N = \\ &(f \oplus g_1 \oplus \cdots \oplus g_n \oplus \cdots \oplus g_N)_{-x}. \end{aligned} \tag{B.1.61}$$

- *Insensitivity of opening and closing to translation of SE:*

$$A \circ (B)_x = A \circ B. \tag{B.1.62}$$

$$A \bullet (B)_x = A \bullet B. \tag{B.1.63}$$

$$f \circ (g)_x = f \circ g. \tag{B.1.64}$$

$$f \bullet (g)_x = f \bullet g. \tag{B.1.65}$$

- *Shift compensation in image by properly shifting of SE:*

$$(A)_x \overset{M}{\oplus} (B)_{-x} = A \overset{M}{\oplus} B. \tag{B.1.66}$$

$$(A)_x \oplus (B)_x = A \oplus B. \tag{B.1.67}$$

$$(f)_x \oplus (g)_x = f \oplus g. \tag{B.1.68}$$

Similarly we can compensate the shift in image with a proper shift in one of the decomposed SEs:

$$\begin{aligned} &(A)_x \overset{M}{\oplus} B_1 \overset{M}{\oplus} \cdots \overset{M}{\oplus} (B_n)_{-x} \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_N = \\ &A \overset{M}{\oplus} B_1 \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_n \overset{M}{\oplus} \cdots \overset{M}{\oplus} B_N. \end{aligned} \tag{B.1.69}$$

$$\begin{aligned} &(A)_x \oplus B_1 \oplus \cdots \oplus (B_n)_x \oplus \cdots \oplus B_N = \\ &A \oplus B_1 \oplus \cdots \oplus B_n \oplus \cdots \oplus B_N. \end{aligned} \tag{B.1.70}$$

$$\begin{aligned} &(f)_x \oplus g_1 \oplus \cdots \oplus (g_n)_x \oplus \cdots \oplus g_N = \\ &f \oplus g_1 \oplus \cdots \oplus g_n \oplus \cdots \oplus g_N. \end{aligned} \tag{B.1.71}$$

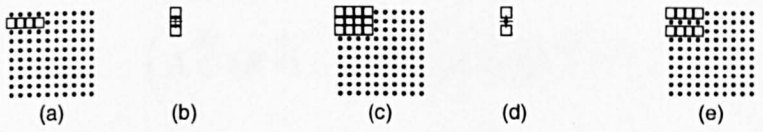


Figure B.1: Extensivity: (a):A, (b):B_1, (c):$A \overset{M}{\oplus} B_1$, (d):$B_2$, (e):$A \overset{M}{\oplus} B_2$.

- *Extensivity:* (SA) Minkowski addition is extensive if the origin belongs to SE. Figures B.1-a, b, d show an image and two SEs denoted by A, B_1 and B_2 respectively. B_2 does not include the origin. Figures B.1-c, e, denoted by $A \overset{M}{\oplus} B_1$ and $A \overset{M}{\oplus} B_2$ show the result of the Minkowski addition of A by B_1 and B_2 respectively. We see that $A \subset A \overset{M}{\oplus} B_1$, but $A \not\subset A \overset{M}{\oplus} B_2$. For extensivity discussion of morphological operators refer to Table 2.3.

- *Anti Extensivity:* (SA) Minkowski subtraction is anti extensive if the origin belongs to SE. Figures B.2-a, b, d show an image and two SEs denoted by A, B_1 and B_2 respectively. B_2 does not include the origin. Figures B.2-c, e, denoted by $A \overset{M}{\ominus} B_1$ and $A \overset{M}{\ominus} B_2$ show the result of the Minkowski subtraction of A by B_1 and B_2 respectively. We see that $A \overset{M}{\ominus} B_1 \subset A$, but $A \overset{M}{\ominus} B_2 \not\subset A$. Similar anti-extensivity exists for erosion. For anti-extensivity discussion of morphological operators refer to Table 2.3.

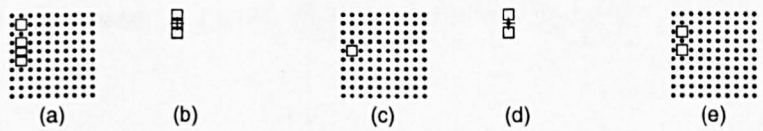


Figure B.2: Anti extensivity: (a):A, (b):B_1, (c):$A \overset{M}{\ominus} B_1$, (d):$B_2$, (e):$A \overset{M}{\ominus} B_2$.

- (SA)

$$\left(A \overset{M}{\oplus} (B \overset{M}{\ominus} C)\right) \subseteq \left((A \overset{M}{\oplus} B) \overset{M}{\ominus} C\right). \tag{B.1.72}$$

- *Adjunction relation:* (SA)

$$B \subseteq (A \overset{M}{\ominus} C) \Longleftrightarrow (B \overset{M}{\oplus} C) \subseteq A. \tag{B.1.73}$$

-

$$\begin{aligned} B \subseteq \quad & (\cdots (A \overset{M}{\ominus} C_1) \overset{M}{\ominus} \cdots) \overset{M}{\ominus} C_N \\ & (\cdots (B \overset{M}{\oplus} C_1) \overset{M}{\oplus} \cdots) \overset{M}{\oplus} C_N \quad \subseteq A. \end{aligned} \quad \Longleftrightarrow \tag{B.1.74}$$

- *Increasing:* (SA)

$$B \subseteq C \Rightarrow (B \overset{M}{\oplus} D) \subseteq (C \overset{M}{\oplus} D). \tag{B.1.75}$$

$$B \subseteq C \Rightarrow (B \overset{M}{\ominus} D) \subseteq (C \overset{M}{\ominus} D). \tag{B.1.76}$$

Figure B.3 illustrates the property of increasing in Minkowski addition and subtraction. Parts a, b and g show B, C and D respectively ($B \subset C$). Parts c, d, e and f show $B \overset{M}{\oplus} D$, $C \overset{M}{\oplus} D$, $B \overset{M}{\ominus} D$ and $C \overset{M}{\ominus} D$ respectively. We notice that $B \overset{M}{\oplus} D \subset C \overset{M}{\oplus} D$ and $B \overset{M}{\ominus} D \subset C \overset{M}{\ominus} D$.

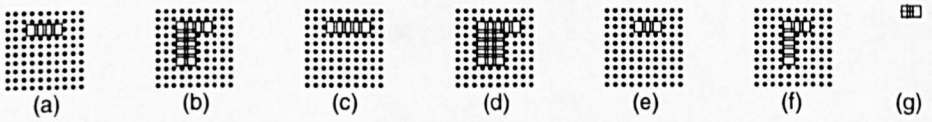


Figure B.3: Increase: (a):B, (b):C, (c):$B \overset{M}{\oplus} D$, (d):$C \overset{M}{\oplus} D$, (e):$B \overset{M}{\ominus} D$, (f):$C \overset{M}{\ominus} D$, (g) D.

$$B \subseteq C \Rightarrow (B \circ D) \subseteq (C \circ D). \tag{B.1.77}$$

$$B \subseteq C \Rightarrow (B \bullet D) \subseteq (C \bullet D). \tag{B.1.78}$$

$$f_1 \leq f_2 \Rightarrow f_1 \oplus g \leq f_2 \oplus g. \tag{B.1.79}$$

$$f_1 \leq f_2 \Rightarrow f_1 \ominus g \leq f_2 \ominus g. \tag{B.1.80}$$

$$f_1 \leq f_2 \Rightarrow f_1 \circ g \leq f_2 \circ g. \tag{B.1.81}$$

$$f_1 \leq f_2 \Rightarrow f_1 \bullet g \leq f_2 \bullet g. \tag{B.1.82}$$

- *Anti-increasing:* (SA)

$$B \subset C \Rightarrow (A \overset{M}{\ominus} B) \supset (A \overset{M}{\ominus} C). \tag{B.1.83}$$

Figure B.4 illustrates Equation B.1.83: Parts a, b and c show A, B and C respectively ($B \subset C$). Parts d and e show $A \overset{M}{\ominus} B$ and $A \overset{M}{\ominus} C$ respectively ($A \overset{M}{\ominus} B \supset A \overset{M}{\ominus} C$).

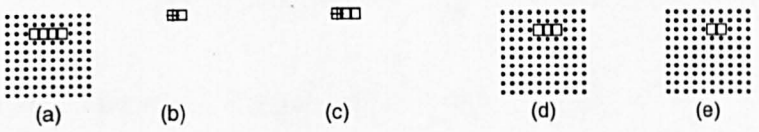


Figure B.4: Anti increasing: (a):A, (b):B, (c):C, (d):$A \overset{M}{\ominus} B$, (e):$A \overset{M}{\ominus} C$.

$$g_1 \leq g_2 \Rightarrow f \oplus g_1 \leq f \oplus g_2. \tag{B.1.84}$$

$$g_1 \leq g_2 \Rightarrow f \ominus g_1 \geq f \ominus g_2. \tag{B.1.85}$$

- (SA)

$$A \overset{M}{\ominus} \hat{A} = \{o\}. \tag{B.1.86}$$

- (SA)

$$A \overset{M}{\oplus} (B \cup C) = (A \overset{M}{\oplus} B) \cup (A \overset{M}{\oplus} C). \tag{B.1.87}$$

- (SA)

$$A \overset{M}{\ominus} (B \cup C) = (A \overset{M}{\ominus} B) \cap (A \overset{M}{\ominus} C). \tag{B.1.88}$$

- (SA)

$$(B \cap C) \overset{M}{\ominus} A = (B \overset{M}{\ominus} A) \cap (C \overset{M}{\ominus} A). \tag{B.1.89}$$

- (SA)

$$A \overset{M}{\oplus} (B \cap C) \subseteq (A \overset{M}{\oplus} B) \cap (A \overset{M}{\oplus} C). \tag{B.1.90}$$

- (SA)

$$A \overset{M}{\ominus} (B \cap C) \supseteq (A \overset{M}{\ominus} B) \cup (A \overset{M}{\ominus} C). \tag{B.1.91}$$

- (SA)

$$(B \cup C) \overset{M}{\ominus} A \supseteq (B \overset{M}{\ominus} A) \cup (C \overset{M}{\ominus} A). \tag{B.1.92}$$

- *Open set:* The set A is open with respect to B if

$$A \circ B = A. \tag{B.1.93}$$

- *Closed set:* The set A is close with respect to B if

$$A \bullet B = A. \tag{B.1.94}$$

- *Sieving:* If B is open with respect to C, then:

$$(A \circ B) \subset (A \circ C) \subset A \subset (A \bullet C) \subset (A \bullet B). \tag{B.1.95}$$

- (SA) If B and g are symmetric, then

$$A \overset{M}{\oplus} B = (A \overset{M}{\oplus} B) \circ B = (A \bullet B) \overset{M}{\oplus} B. \tag{B.1.96}$$

$$f \oplus g = (f \oplus g) \circ g = (f \bullet g) \oplus g. \tag{B.1.97}$$

- $$A \circ B = \bigcup \{Bx : B_x \subseteq A\}. \tag{B.1.98}$$

- $$A \bullet B = \{x : x \in \check{B}_y \Rightarrow \check{B}_y \cap A \neq \emptyset\}. \tag{B.1.99}$$

- *Morphological gradient:* Dilation and erosion often are used to compute the morphological gradient of an image denoted by $GRAD$:

  $$GRAD = (f \oplus g) - (f \ominus g). \tag{B.1.100}$$

  As opposed to gradients obtained using methods such as a Sobel operation, morphological gradients obtained using symmetrical structuring elements tend to depend less on edge directionality. This price paid for this advantage is significant increase in computational requirements.

- *Top-hat transformation:* The so called morphological top-hat transformation of an image, denoted by $TOPHAT$, is defined as

  $$TOPHAT = f - (f \circ g). \tag{B.1.101}$$

  It is useful for enhancing detail in the presence of shading.

- *Granulometry:* Granulometry is a field that, among other things, deals with determining the size distribution of particles in an image. The following morphological approach can be used to determine size distribution. Opening operations with structuring elements of increasing size are performed on the original image. The difference between the original image and its opening is computed after each pass with a different structuring element is completed. At the end of the process, these differences are normalised and then used to construct a histogram of particle-size distribution.

- *Hit-or-miss operator* $\circledast$*:* Hit-or-miss operator ([78]) uses two binary SEs B and C which have empty intersection (otherwise it will not be a meaningful operation) and is defined as:

$$\begin{aligned} A \circledast (B,C) &= \{x : B_x \subseteq A, C_x \subseteq A^c\} \\ &= (A \ominus B) \cap (A^c \ominus C). \end{aligned} \tag{B.1.102}$$

If B and C have even one pixel in common, the sesult will be empty set. In hit-or-miss, some properties similar to B in foreground and C in background of the image is required. Hit-or-miss is not increasing but translation invariant:

$$(A)_x \circledast (B,C) = [A \circledast (B,C)]_x. \tag{B.1.103}$$

Figure B.5 illustrates an example. Parts a, b, c and d respectively show A, $A \circledast (B,C)$, B and C. Hit-or-miss transformations are often used

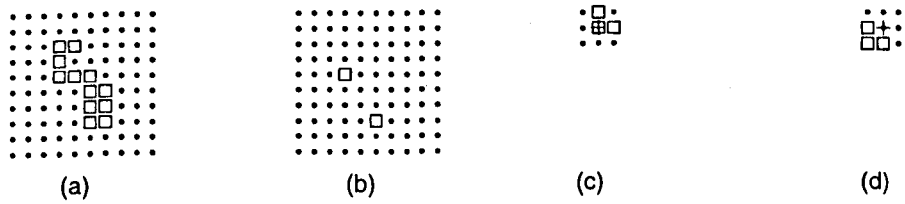


Figure B.5: Hit-or-miss operation. (a):A, (b):$A \circledast (B,C)$, (c):B, (d):C.

in digital topology where these transformations can be used to detect specific topological configuration in an image. Some applications of hit-or-miss operator can be found in [78, 36].

- *Boundary extraction:* The boundary of a set A, denoted by $\mathbb{B}(A)$, can be obtained by

$$\mathbb{B}(A) = A - (A \ominus B). \tag{B.1.104}$$

where B is a suitable SE.

- *Convex hull:*

  Let $B^i, i = 1,2,3,4$ represent four structuring elements. The procedure to make convex hull is as follows.

$$X_k^i = (X \circledast B^i) \cup A, \quad i = 1,2,3,4 \ \textit{and} \ k = 1,2,3,\ldots \tag{B.1.105}$$

  with $X_0^i = A$. Now let $D^i = X_{conv}^i$, where the subscript "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. Then the convex hull of A is

$$C(A) = \bigcup_{i=1}^{4} D^i. \tag{B.1.106}$$

  In other words, the procedure consists of iteratively applying the hit-or-miss transform to A with B^1; when no further changes occur, we apply the union with A and call the result D^1. The procedure is replaced with B^2 until no further changes occur, and so on. The union of the four resulting D's constitutes the convex hull of A.

- *Thinning:* The thinning of a set A by a structuring element B, denoted by $A \otimes B$ can be defined in terms of the hit-or-miss transform:

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c. \tag{B.1.107}$$

  A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:

$$\{B\} = \{B^1, B^2, B^3, \ldots B^n\} \tag{B.1.108}$$

  where B^i is a rotated version of B^{i-1}. Using this concept, we now define thinning by a sequence of SEs as

$$A \otimes B = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n). \tag{B.1.109}$$

  In other words, the process is to thin A by one pass with B^1, then thin the result with one pass of B^2, and so on, until A is thinned with one pass of B^n. The entire process repeats until no further changes occur.

- *Thickening:* Thickening is the morphological dual of thinning and is defined by the expression

$$A \odot B = A \cup (A \circledast B). \tag{B.1.110}$$

where B is a structuring element suitable for thickening. Thickening can be defined as a sequential operaton:

$$A \odot B = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n). \tag{B.1.111}$$

The usual procedure practical for thickening is to thin the background of the set in question and then complement the result. Depending on the nature of A, this complementary procedure may result in some disconnected points. Hence thickening by this method usually is followed by a simple post-processing step to remove disconnected points.

- *Skeletons:* Serra [78] showed that the skeleton of a set (region) A can be expressed in terms of erosions and openings. That is, with $S(A)$ denoting the skeleton of A, it can be shown that

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \tag{B.1.112}$$

with

$$S_k(A) = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\} \tag{B.1.113}$$

where B is a structuring element, $(A \ominus kB)$ indicates k successive erosions of A; that is

$$(A \ominus kB) = ((\ldots(A \ominus B) \ominus B) \ominus \ldots) \ominus B$$

k times, and K is the last iterative step before A erodes to an empty set. In other words,

$$K = \max\{k \mid (A \ominus kB) \neq \emptyset\}.$$

- *Annular opening and closing:* With annular opening, parts of an image are removed on the basis of their isolation, while with the usual structural opening, parts of an image are removed on the basis of their size [35]. It is defined as:

$$OP_{Ann} = A \cap (A \oplus B). \tag{B.1.114}$$

Similarly annular closing is defined as

$$CL_{Ann} = A \cup (A \ominus B). \tag{B.1.115}$$

and it removes isolated hole points.

- *Geodesic operators and operators by reconstruction:* Geodesic GS dilation $(\delta^1(f,r))$ and erosion $(\epsilon^1(f,r))$ of size 1 are defined respectively as: ([68])

$$\delta^1(f,r) = \min\{\delta_1(f), r\}, \tag{B.1.116}$$

$$\epsilon^1(f,r) = -\delta^1(-f,-r), \tag{B.1.117}$$

where r is a reference, and δ_1 is dilation. Reconstruction by dilation is defined as:

$$\gamma^{rec}(f,r) = \delta^{(\infty)}(f,r) = \cdots \delta^{(1)}(\cdots \delta^{(1)}(f,r)\cdots, r). \tag{B.1.118}$$

Reconstruction by erosion is defined as:

$$\phi^{rec}(f,r) = \epsilon^{(\infty)}(f,r) = \cdots \epsilon^{(1)}(\cdots \epsilon^{(1)}(f,r)\cdots, r). \tag{B.1.119}$$

Opening by reconstruction of opening is defined as: $\gamma^{(rec)}(op(f), f)$. Closing by reconstruction of closing is similarly defined as: $\phi^{(rec)}(cl(f), f)$. Opening by partial reconstruction is defined as: $\gamma^{(rec)}(\epsilon_n(f), \gamma_k(f))$. Similarly closing by partial reconstruction is defined as: $\phi^{(rec)}(\delta_n(f), \phi_k(f))$. In above equations if $k = n$, then the above relations will be classical opening and closing. For $k = 0$, they will be opening and closing by reconstruction.

B.2 Proof of some morphological operations

i. Proof of Eq. 2.5.6:

$$
\begin{aligned}
A \overset{M}{\ominus} B &\overset{?}{=} \bigcap_{b\in B} (A)_b && (?)\\
(\check{B})_x &= \{x-b : b\in B\} &&,\\
A \overset{M}{\ominus} B &\overset{Eq.\,2.5.5}{=} \{y : (\check{B})_y \subseteq A\} = \{y-b\in A, b\in B\} &&,\\
z\in\left(\bigcap_{b\in B}(A)_b\right) &\Rightarrow z\in(A)_b, b\in B \Rightarrow z-b\in A, b\in B &&\Rightarrow\\
A \overset{M}{\ominus} B &= \bigcap_{b\in B}(A)_b && \blacksquare
\end{aligned}
$$

ii. Proof of Eq. B.1.13:

$$
\begin{aligned}
A \overset{M}{\oplus} B &\overset{?}{=} B \overset{M}{\oplus} A && (?)\\
A \overset{M}{\oplus} B &= \{a+b : a\in A, b\in B\} = \{b+a : b\in B, a\in A\} &&=\\
&= B \overset{M}{\oplus} A && \blacksquare
\end{aligned}
$$

iii. Proof of Eq. B.1.22:

$$
\begin{aligned}
A \overset{M}{\oplus} (B \overset{M}{\oplus} C) &\overset{?}{=} (A \overset{M}{\oplus} B) \overset{M}{\oplus} C && (?)\\
A \overset{M}{\oplus} (B \overset{M}{\oplus} C) &= \{a+(b+c) : a\in A, b+c\in B \overset{M}{\oplus} C\} &&=\\
&= \{a+b+c : a\in A, b\in B, c\in C\} &&=\\
&= \{(a+b)+c : a+b\in A \overset{M}{\oplus} B, c\in C\} &&=\\
&= (A \overset{M}{\oplus} B) \overset{M}{\oplus} C && \blacksquare
\end{aligned}
$$

iv. Proof of Eq. B.1.26

The length of $g \oplus k$ will be extended to Length(g)+Length(h)-1 (see Eq. B.1.28). Let $\gamma = \beta - \alpha$:

$$
\begin{aligned}
(f\oplus g)\oplus h &\overset{?}{=} f\oplus(g\oplus h) && (?)\\
((f\oplus g)\oplus h)(n) &= \max_{v\in D_h}\{\max_{u\in D_g}\{f(n-u-v)+g(u)+h(v)\}\} && (1),\\
(g\oplus h)(m) &= k(m) = \max_{\alpha\in D_h}\{g(m-\alpha)+h(\alpha)\} && (2),\\
(f\oplus k)(l) &= \max_{\beta\in D_k}\{f(l-\beta)+k(\beta)\} && (2) =\\
&= \max_{\beta\in D_k}\{f(l-\beta)+\max_{\alpha\in D_h}\{g(\beta-\alpha)+h(\alpha)\}\} && (2),\\
(f\oplus k)(l) &= \max_{\alpha+\gamma\in D_g}\{\max_{\alpha\in D_h}\{f(l-\alpha-\gamma)+g(\gamma)+h(\alpha)\}\} && (2)\Leftrightarrow\\
((f\oplus g)\oplus h)(n) &= (f\oplus(g\oplus h))(n) && \blacksquare
\end{aligned}
$$

v. Proof of Eq. B.1.48:

$$
\begin{aligned}
(A)_x \overset{M}{\oplus} B &\overset{?}{=} (A \overset{M}{\oplus} B)_x && (?)\\
(A)_x \overset{M}{\oplus} B &= \{d : d = c+b, c\in(A)_x, b\in B\} &&=\\
&= \{d : d = (a+x)+b, a\in A, b\in B\} &&=\\
&= \{d : d = (a+b)+x, a\in A, b\in B\} &&=\\
&= \{d : d = e+x, e\in A \overset{M}{\oplus} B\} &&=\\
&= (A \overset{M}{\oplus} B)_x && \blacksquare
\end{aligned}
$$

vi. Proof of Eq. B.1.53:

$$
\begin{aligned}
f \oplus (g)_x &\overset{?}{=} (f \oplus g)_{-x} && (?)\\
(f \oplus (g)_x)(n) &= \max_{u \in D_{(g)_x}} \{f(n-u) + (g)_x(u)\} && =\\
&= \max_{v \in D_g} \{f(n-v-x) + g(v)\} && =\\
&= (f \oplus g)_{-x}(n) && \blacksquare
\end{aligned}
$$

vii. Proof of Eq. B.1.54:

$$
\begin{aligned}
(A)_x \overset{M}{\ominus} B &\overset{?}{=} (A \overset{M}{\ominus} B)_x && (?)\\
(A \overset{M}{\ominus} B)_x &= \{y : (\check{B})_y \subset A\}_x \overset{Eq.\ 2.5.1}{=} \{y : (\check{B})_y - x \subset A\} && =\\
&= \{y : (\check{B})_y \subset (A)_x\} && =\\
&\overset{Eq.\ 2.5.5}{=} (A)_x \overset{M}{\ominus} B && \blacksquare
\end{aligned}
$$

viii. Proof of Eq. B.1.64:

$$
\begin{aligned}
f \circ (g)_x &\overset{?}{=} f \circ g && (?)\\
(f \circ (g)_x)(n) &= \max_{v \in D_{(g)_x}} \{ \min_{u \in D_{(g)_x}} \{f(n+u-v) - (g)_x(u) + (g)_x(v)\}\} && =\\
&= \max_{\beta \in D_g} \{ \min_{\alpha \in D_g} \{f(n+(\alpha+x)-(\beta+x)) - g(\alpha) + g(\beta)\}\} && =\\
&= (f \circ g)(n) && \blacksquare
\end{aligned}
$$

ix. Proof of Eq. B.1.55:

$$
\begin{aligned}
A \overset{M}{\ominus} (B)_x &\overset{?}{=} (A \overset{M}{\ominus} B)_{-x} && (?)\\
(A \overset{M}{\ominus} B)_{-x} &= \{y : (\check{B})_y \subseteq A\}_{-x} \overset{Eq.\ 2.5.1}{=} \{y : (\check{B})_y + x \subseteq A\} && =\\
&= \{y : ((\check{B})_x)_y \subset A\} && =\\
&\overset{Eq.\ 2.5.5}{=} A \overset{M}{\ominus} (B)_x && \blacksquare
\end{aligned}
$$

x. Proof of Eq. B.1.66:

$$
\begin{aligned}
(A)_x \overset{M}{\oplus} (B)_{-x} &\overset{?}{=} A \overset{M}{\oplus} B && (?)\\
(A)_x \overset{M}{\oplus} (B)_{-x} &\overset{Eq.\ B.1.48}{=} (A \overset{M}{\oplus} (B)_{-x})_x = (A \overset{M}{\oplus} B)_{x-x} && =\\
&= A \overset{M}{\oplus} B && \blacksquare
\end{aligned}
$$

xi. Proof of Eq. B.1.72:

$$
\begin{aligned}
\left(A \overset{M}{\oplus} (B \overset{M}{\ominus} C)\right) &\overset{?}{\subseteq} \left((A \overset{M}{\oplus} B) \overset{M}{\ominus} C\right) && (?)\\
B \overset{M}{\ominus} C &= \{x : x + c \in B, c \in C\} && ,\\
A \overset{M}{\oplus} (B \overset{M}{\ominus} C) &= \{a + x + c : a \in A, x + c \in B, c \in C\} && \Rightarrow\\
\left(A \overset{M}{\oplus} (B \overset{M}{\ominus} C)\right) &\subseteq \{a + x + c : a \in A, x \in B, c \in C\} && \Rightarrow\\
\left(A \overset{M}{\oplus} (B \overset{M}{\ominus} C)\right) &\subseteq \{(a+x) + c : a + x \in A \overset{M}{\oplus} B, c \in C\} && \Longrightarrow\\
\left(A \overset{M}{\oplus} (B \overset{M}{\ominus} C)\right) &\subseteq \left((A \overset{M}{\oplus} B) \overset{M}{\ominus} C\right) && \blacksquare
\end{aligned}
$$

xii. Proof of Eq. B.1.73:

$$
\begin{aligned}
B \subseteq (A \overset{M}{\ominus} C) &\overset{?}{\Longleftrightarrow} (B \overset{M}{\oplus} C) \subseteq A && (?)\\
B \subseteq (A \overset{M}{\ominus} C) &\overset{?1}{\Longrightarrow} (B \overset{M}{\oplus} C) \subseteq A && (1?)\\
(B \overset{M}{\oplus} C) \subseteq A &\overset{?2}{\Longrightarrow} B \subseteq (A \overset{M}{\ominus} C) && (2?)\\
B \overset{M}{\oplus} C &= \{x : x = b + c, b \in B, c \in C\} && (1) \Rightarrow\\
B \overset{M}{\oplus} C &\subseteq \{x : x = b + c, b \in (A \overset{M}{\ominus} C), c \in C\} && (1)\\
B \overset{M}{\oplus} C &\subseteq (\{x : x = b + c, b + c \in A, c \in C\} = \{x : x \in A\}) && (1)\blacksquare\\
B \overset{M}{\oplus} C &= \{x : x = b + c, b \in B, c \in C\} \subseteq A && (2) \Rightarrow\\
&\Rightarrow \{b + c : b + c \in A, c \in C\} && (2) \Rightarrow\\
&\Rightarrow b \in (A \overset{M}{\ominus} C) && (2) \Rightarrow\\
&\Rightarrow B \subseteq (A \overset{M}{\ominus} C) && (2)\blacksquare
\end{aligned}
$$

xiii. Proof of Eq. B.1.75:

$$
\begin{aligned}
B \subset C &\overset{?}{\Rightarrow} (B \overset{M}{\oplus} D) \subset (C \overset{M}{\oplus} D) && (?)\\
B \overset{M}{\oplus} D &= \{b + d : b \in B, d \in D\}, C \overset{M}{\oplus} D = \{c + d : c \in C, d \in D\} && ,\\
&, \quad B \subset C \rightarrow b \in (B \subset C) && \Rightarrow\\
&\Rightarrow (B \overset{M}{\oplus} D) \subset (C \overset{M}{\oplus} D) && \blacksquare
\end{aligned}
$$

xiv. Proof of Eq. B.1.80:

$$
\begin{aligned}
f_1 \le f_2 &\overset{?}{\Rightarrow} f_1 \ominus g \le f_2 \ominus g && (?)\\
f_1(n) \le f_2(n) &\Rightarrow f_1(n+u) \le f_2(n+u) \Rightarrow f_1(n+u) - g(u) \le f_2(n+u) - g(u) && \Rightarrow\\
&\Rightarrow \min_{u \in D_g} \{f_1(n+u) - g(u)\} \le \min_{u \in D_g} \{f_2(n+u) - g(u)\} && \Rightarrow\\
f_1(n) \le f_2(n) &\Rightarrow (f_1 \ominus g)(n) \le (f_2 \ominus g)(n) && \blacksquare
\end{aligned}
$$

xv. Proof of Eq. B.1.83:

$$
\begin{aligned}
B \subset C &\overset{?}{\Rightarrow} (A \overset{M}{\ominus} B) \supset (A \overset{M}{\ominus} C) && (?)\\
B \subset C &\overset{Eq.\ B.1.75}{\Rightarrow} A^c \overset{M}{\oplus} B \subset A^c \overset{M}{\oplus} C \Rightarrow (A^c \overset{M}{\oplus} B)^c \supset (A^c \overset{M}{\oplus} C)^c && \Longrightarrow\\
&\overset{Eq.\ 2.5.5}{\Longrightarrow} A \overset{M}{\ominus} B \supset A \overset{M}{\ominus} C && \blacksquare
\end{aligned}
$$

xvi. Proof of Eq. B.1.87:

$$
\begin{aligned}
A \overset{M}{\oplus} (B \cup C) &\overset{?}{=} (A \overset{M}{\oplus} B) \cup (A \overset{M}{\oplus} C) && (?)\\
A \overset{M}{\oplus} (B \cup C) &\overset{Eq.\ 2.5.4}{=} \bigcup_{x \in A} (B_x \cup C_x) = \left(\bigcup_{x \in A} B_x\right) \cup \left(\bigcup_{x \in A} C_x\right) && =\\
&= (A \overset{M}{\oplus} B) \cup (A \overset{M}{\oplus} C) && \blacksquare
\end{aligned}
$$

xvii. Proof of Eq. B.1.88:

$$
\begin{aligned}
A \overset{M}{\ominus} (B \cup C) &\overset{?}{=} (A \overset{M}{\ominus} B) \cap (A \overset{M}{\ominus} C) && (?)\\
A \overset{M}{\ominus} (B \cup C) &\overset{Eq.\ 2.5.5}{=} \left(A^c \overset{M}{\oplus} (B \cup C)\right)^c \overset{Eq.\ B.1.87}{=} \left((A^c \overset{M}{\oplus} B) \cup (A^c \overset{M}{\oplus} C)\right)^c && =\\
&= (A^c \overset{M}{\oplus} B)^c \cap (A^c \overset{M}{\oplus} C)^c && =\\
&\overset{Eq.\ B.1.87}{=} (A \overset{M}{\ominus} B) \cap (A \overset{M}{\ominus} C) && \blacksquare
\end{aligned}
$$

xviii. Proof of Eq. B.1.89

$$
\begin{array}{lll}
(B\cap C)\overset{M}{\ominus}A & \overset{?}{=} & (B\overset{M}{\ominus}A)\cap(C\overset{M}{\ominus}A) \qquad (?)\\
(B\cap C)\overset{M}{\ominus}A & \overset{Eq.\ 2.5.5}{=} & \left((B\cap C)^c\overset{M}{\oplus}A\right)^c=\left((B^c\cup C^c)\overset{M}{\oplus}A\right)^c \qquad =\\
 & \overset{Eq.\ B.1.87}{=} & \left(B^c\overset{M}{\oplus}A)\cup(C^c\overset{M}{\oplus}A)\right)^c=(B^c\overset{M}{\oplus}A)^c\cap(C^c\overset{M}{\oplus}A)^c \qquad =\\
 & \overset{Eq.\ 2.5.5}{=} & (B\overset{M}{\ominus}A)\cap(C\overset{M}{\ominus}A) \qquad \blacksquare
\end{array}
$$

xix. Proof of Eq. B.1.90

$$
\begin{array}{rcl}
\left(A\overset{M}{\oplus}(B\cap C)\right) & \overset{?}{\subseteq} & \left((A\overset{M}{\oplus}B)\cap(A\overset{M}{\oplus}C)\right) \qquad (?)\\
(B\cap C)\subseteq B \quad , & & (B\cap C)\subseteq C \qquad \overset{Eq.\ B.1.75}{\Longrightarrow}\\
\left(A\overset{M}{\oplus}(B\cap C)\right)\subseteq(A\overset{M}{\oplus}B) \quad , & & \left(A\overset{M}{\oplus}(B\cap C)\right)\subseteq(A\overset{M}{\oplus}C) \qquad \Longrightarrow\\
\left(A\overset{M}{\oplus}(B\cap C)\right) & \subseteq & \left((A\overset{M}{\oplus}B)\cap(A\overset{M}{\oplus}C)\right) \qquad \blacksquare
\end{array}
$$

xx. Proof of Eq. B.1.91

$$
\begin{array}{lll}
\left(A\overset{M}{\ominus}(B\cap C)\right) & \overset{?}{\supseteq} & \left((A\overset{M}{\ominus}B)\cup(A\overset{M}{\ominus}C)\right) \qquad (?)\\
A\overset{M}{\ominus}(B\cap C) & \overset{Eq.\ 2.5.5}{=} & \left(A^c\overset{M}{\oplus}(B\cap C)\right)^c \qquad ,\\
\left(A^c\overset{M}{\oplus}(B\cap C)\right) & \overset{Eq.\ B.1.90}{\subseteq} & \left((A^c\overset{M}{\oplus}B)\cap(A^c\overset{M}{\oplus}C)\right) \qquad \Longrightarrow\\
\left(A^c\overset{M}{\oplus}(B\cap C)\right)^c & \supseteq & \left((A^c\overset{M}{\oplus}B)\cap(A^c\overset{M}{\oplus}C)\right)^c \qquad \Longrightarrow\\
 & \supseteq & \left((A^c\overset{M}{\oplus}B)^c\cup(A^c\overset{M}{\oplus}C)^c\right) \qquad \overset{Eq.\ 2.5.5}{\Longrightarrow}\\
\left(A\overset{M}{\ominus}(B\cap C)\right) & \supseteq & \left((A\overset{M}{\ominus}B)\cup(A\overset{M}{\ominus}C)\right) \qquad \blacksquare
\end{array}
$$

xxi. Proof of Eq. B.1.92

$$
\begin{array}{lll}
\left((B\cup C)\overset{M}{\ominus}A\right) & \overset{?}{\supseteq} & \left((B\overset{M}{\ominus}A)\cup(C\overset{M}{\ominus}A)\right) \qquad (?)\\
(B\cup C)\overset{M}{\ominus}A & = & \left((B\cup C)^c\overset{M}{\oplus}A\right)^c=(B^c\cap C^c)\overset{M}{\oplus}A \qquad ,\\
\left((B^c\cap C^c)\overset{M}{\oplus}A\right) & \overset{Eq.\ B.1.90}{\subseteq} & \left((B^c\overset{M}{\oplus}A)\cap(C^c\overset{M}{\oplus}A)\right) \qquad \Longrightarrow\\
\left((B^c\cap C^c)\overset{M}{\oplus}A\right)^c & \overset{Eq.\ B.1.90}{\supseteq} & \left((B^c\overset{M}{\oplus}A)\cap(C^c\overset{M}{\oplus}A)\right)^c \qquad ,\\
 & \supseteq & \left((B^c\overset{M}{\oplus}A)^c\cup(C^c\overset{M}{\oplus}A)^c\right) \qquad \Longrightarrow\\
\left((B\cup C)\overset{M}{\ominus}A\right) & \supseteq & \left((B\overset{M}{\ominus}A)\cup(C\overset{M}{\ominus}A)\right) \qquad \blacksquare
\end{array}
$$

xxii. Proof of Eq. B.1.43

$$
\begin{array}{rcl}
\alpha A\overset{M}{\oplus}\beta A & \overset{?}{=} & (\alpha+\beta)A \qquad (?)\\
\alpha A\overset{M}{\oplus}\beta A & = & \{a+b:a\in\alpha A,b\in\beta A\}=\{\alpha a+\beta b:a\in A,b\in A\} \qquad =\\
 & = & (\alpha+\beta)\{\tfrac{\alpha}{\alpha+\beta}a+\tfrac{\beta}{\alpha+\beta}b:a\in A,b\in A\} \qquad ,\\
\lambda & = & \tfrac{\alpha}{\alpha+\beta}\Rightarrow 1-\lambda=\tfrac{\beta}{\alpha+\beta},0\le\lambda\le 1 \quad (\text{A is convex}) \quad c=\lambda a+(1-\lambda b)\in A \Rightarrow\\
\alpha A\overset{M}{\oplus}\beta A & = & (\alpha+\beta)\{c:c\in A\}=(\alpha+\beta)A \qquad \blacksquare
\end{array}
$$

xxiii. Proof of Eq. B.1.96

$$
\begin{aligned}
A \overset{M}{\oplus} B &\overset{?}{=} (A \overset{M}{\oplus} B) \circ B \overset{?}{=} (A \bullet B) \overset{M}{\oplus} B && (?)\\
L &= A \overset{M}{\oplus} B, M = L \overset{M}{\ominus} B, N = M \overset{M}{\oplus} B && \Rightarrow\\
L &\overset{?}{=} N && (?)\\
L &= A \overset{M}{\oplus} B \overset{Eq.\ B.1.73}{\Rightarrow} A \subseteq L \overset{M}{\ominus} B \Rightarrow A \subseteq M && \Rightarrow\\
&\Rightarrow A \overset{M}{\oplus} B \subseteq M \overset{M}{\oplus} B && \Rightarrow\\
&\Rightarrow L \subseteq N && (1)\blacksquare\\
M &= L \overset{M}{\ominus} B \overset{Eq.\ B.1.73}{\Rightarrow} M \overset{M}{\oplus} B \subseteq L && \Rightarrow\\
&\Rightarrow N \subseteq L && (2)\blacksquare
\end{aligned}
$$

Appendix C

Arrhythmia types

The current appendix introduces some of arrhythmia types adopted from literature [9, 7]. An arrhythmia is a change in the regular beat of the heart. The heart may seem to skip a beat or beat irregularly or very fast or very slowly (http://www.atlcard.com/arrhth.html). Arrhythmias may occur in people who do not have heart disease. In most cases, there is no recognisable cause of an arrhythmia. One reason can be heart disease. However other causes can be tobacco, alcohol, diet pills, stress, caffeine, and cough and cold medicines. Most of the people with arrhythmias have nothing to fear. They do not need extensive exams or special treatments for their condition unless arrhythmias are associated with heart disease. In these cases, heart disease, and not the arrhythmias, causes the greatest threat to the patient. In a very small number of people with serious symptoms, arrhythmias are really dangerous. These arrhythmias require medical treatment to keep the heartbeat regular. For example, a few people have a very slow heartbeat (bradycardia), causing them to feel lightheaded or faint. If left untreated, the heart may stop beating and these people could die. Arrhythmias occur commonly in middle-age adults. As people get older, they are more likely to experience an arrhythmia. Most people have felt their heart beat very fast, experienced a fluttering in their

chest, or noticed that their heart skipped a beat. Almost everyone has also felt dizzy, faint, or out of breath or had chest pains at one time or another. One of the most common arrhythmias is sinus arrhythmia, the change in heart rate that can occur normally when we take a breath. These experiences may cause anxiety, but for the majority of people, they are completely harmless. Describing how the heart beats, normally helps to explain what happens during an arrhythmia. The heart is a muscular pump divided into four chambers–two atria located on the top and two ventricles located on the bottom. Normally each heartbeat starts in the right atrium. Here, a special group of cells called the sinus node, or natural pacemaker, sends an eletrical signal. The signal spreads throughout the atria to the area called the atrioventricular (AV) node. The AV node connects to a group of special pathways that conduct the signal to the ventricles below. As the signal travels through the heart, the heart contracts. First the atria contract, pumping blood into the ventricles a fraction of a second later, the ventricles contract, sending blood throughout the body. Usually the whole heart contracts between 60 and 100 times per minute. Each contraction equals one heartbeat. An arrhythmia may occur for one of several reasons

- Instead of beginning in the sinus node, the heartbeat begins in another part of the heart.
- The sinus node develops an abnormal rate or rhythm.
- A patient has a heart block.

Heart block is a condition in which the electrical signal cannot travel normally down the special pathways to the ventricles. For example, the signal from the atria to the ventricles may be

i. delayed, but each one conducted;

ii. delayed with only some getting through; or

iii. completely interrupted.

If there is no conduction, the beat generally originates from the ventricles and is very slow. There are many types of arrhythmias. Arrhythmias are identified by where they occur in the heart (atria or ventricles) and by what happens to the heart's rhythm when they occur. Arrhythmias arising in the atria are called atrial or supraventricular (above the ventricles) arrhythmias. Ventricular arrhythmias begin in the ventricles. In general, ventricular arrhythmias caused by heart disease are the most serious.

Sometimes an arrhythmia can be detected by listening to the heart with a stethoscope. However, the electrocardiogram is the most precise method for diagnosing the arrhythmia. An arrhythmia may not occur at the time of the exam even though the symptoms are present at other times. In such cases, tests will be done if necessary to find out whether an arrhythmia is causing the symptoms. First the doctor will take a medical history and do a thorough physical exam. The one or more tests may be used to check for an arrhythmia and to decide whether it is caused by heart disease. Many arrhythmias require no treatement whatsoever. Serious arrhythmias are treated in several ways depending on what is causing the arrhythmia. Sometimes the heart disease is treated to control the arrhythmia. Or, the arrhythmia itself may be treated using one or more of the following treatments.

- *Drugs :* There are several kinds of drugs used to treat arrhythmias. One or more drugs may be used. Drugs are carefully chosen because they can cause arrhythmias or make arrhythmias worse. For this reason, the benefits of the drug are carefully weighed against any risks associated with taking it. It is important not to change the dose or type of your medication unless you check with your doctor first. If you are taking drugs for arrhythmia, one of the following tests will probably be used to see whether treatment is working: a 24-hour electrocardiogram (ECG) while

you are on drug therapy, and exercise ECG, or a special technique to see how easily the arrhythmia can be caused. Blood levels of antiarrhythmic drugs may also be checked.

- *Cardioversion :* To quickly restore a heart to its normal rhythm, the doctor may apply an electrical shock to the chest wall. Called cardioversion, this treatment is most often used in emergency situations. After cardioversion, drugs are usually prescribed to prevent the arrhythmia from recurring.

- *Automatic implantable defibrillators :* These devices are used to correct serious ventricular arrhythmias that can lead to sudden death. The defibrillator is surgically placed inside the patient's chest. There, it monitors the heart's rhythm and quickly identifies serious arrhythmias. With an electrical shock, it immediately disrupts a deadly arrhythmia.

- *Artificial pacemaker :* An artificial pacemaker can take charge of sending electrical signals to make the heart beat if the heart's natural pacemaker is not working properly or its electrical pathway is blocked. During a simple operation, this electrical device is placed under the skin. A lead extends from the device to the right side of the heart, where it is permanently anchored.

- *Surgery :* When an arrhythmia cannot be controlled by other treatmets, doctors may perform surgery. After locating the heart tissue that is causing the arrhythmia, the tissue is altered or removed so that it will not produce the arrhythmia.

If the heart disease is not causing the arrhythmia, the doctor may suggest that you avoid what is causing it. For example, if caffeine or alcohol is the cause, the doctor may ask you not to drink coffee, tea, colas, or alcoholic beverages.

The arrhythmia differ from the originating point of view as:

i. **Originating in the Atria:**

- *Sinus arrhythmia:* Cyclic changes in the heart rate during breathing. Common in children and often found in adults.
- *Sinus tachycardia:* The sinus node sends out electrical signals faster than usual, speeding up the heart rate.
- *Sick sinus syndrome:* The sinus node does not fire its signals properly, so that the heart rate slows down. Sometimes the rate changes back and forth between a slow (bradycardia) and fast (tachycardia) rate.
- *Premature supraventricular contractions or premature atrial contractions (PAC):* A beat occurs early in the atria, causing the heart to beat before the next regular heartbeat.
- *Supraventricular tachycardia (SVT), paroxysmal atrial tachycardia (PAT):* A series of early beats in the atria speed up the heart rate (the number of times a heart beats per minute). In paroxysmal tachycardia, repeated periods of very fast heartbeats begin and end suddenly.
- *Atrial flutter:* Rapidly fired signals cause the muscles in the atria to contract quickly, leading to a very fast, steady heartbeat.
- *Atrial fibrillation:* Electrical signals in the atria are fired in a very fast and uncontrolled manner. Electrical signals arrive in the ventricles in a completely irregular fashion, so the heart beat is completely irregular.
- *Wolff-Parkinson-White syndrome:* Abnormal pathways between the atria and ventricles cause the electrical signal to arrive at the ventricles too soon and to be transmitted back into the atria. Very fast

heart rates may develop as the electrical signal richochets between the atria and ventricles.

ii. **Originating in the Ventricles:**

- *Premature ventricular complexes (PVC).:* An electrical signal from the ventricles causes an early heart beat that generally goes unnoticed. The heart then seems to pause until the next beat of the ventricle occurs in a regular fashion.
- *Ventricular tachycardia:* The heart beats fast due to electrical signals arising from the ventricles (rather than from the atria).
- *Ventricular fibrillation:* Electrical signals in the ventricles are fired in a very fast and uncontrolled manner, causing the heart to quiver rather than beat and pump blood.

In this part we will have a look to the name of some abnormalities:

- *First-degree AV block:* Prolongation of the $P-R$ interval is called first-degree AV block.

The following part shows two tables and some figures about some of the arrhythmia.

Table C.1: Arrhythmia recognition

Disease	Heart rate	Rhythm	P Wave	PR(sec)	QRS(sec)	Figure
Normal Sinus Rhythm	60 – 100 bpm	Regular	Before each QRS, identical	.12 – .20	< .12	C.1
Sinus Arrhythmia	Usually 60 – 100 bpm	Irregular	Before each QRS, identical	.12 – .20	< .12	C.2-a
Sinus Tachycardia	> 100 bpm	Regular	Before each QRS, identical	.12 – .20	< .12	C.2-b
Sinus Bradycardia	< 60 bpm	Regular	Before each QRS, identical	.12 – .20	< .12	C.2-c
Sinus Pause	N/A	Irregular	Before each QRS, identical. Dropped beat. The P to P interval is undisturbed.	.12 – .20	< .12	C.2-d

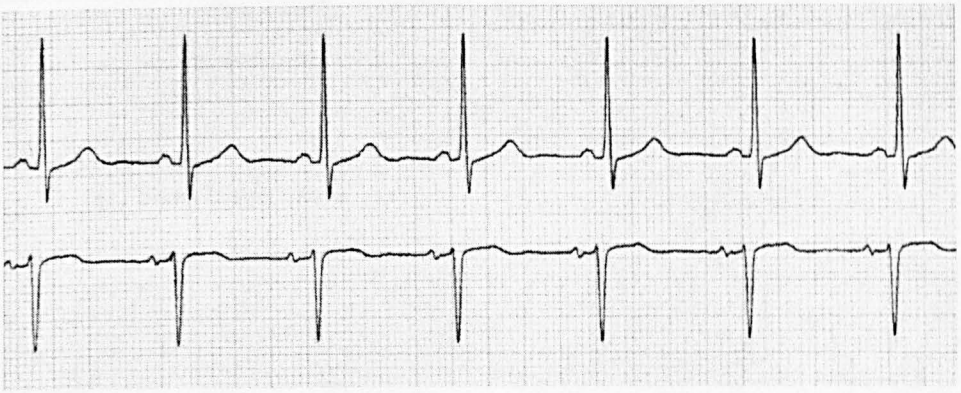
continued on next page

continued from previous page

Disease	Heart rate	Rhythm	P Wave	PR(sec)	QRS(sec)	Figure
Sinus Arrest	N/A	Irregular	Before each QRS, identical. New rhythmbegins after a pause. The P to P interval is disturbed	.12 – .20	< .12	C.2-e
PACs / Isolated PAC's Occur Single	N/A	Irregular	Premature and abnormal or hidden	< .20	< .12	C.2-g
PACs / Paired PAC's Occur in Two's	N/A	Irregular	Premature and abnormal or hidden	< .20	< .12	C.3-a
PACs / Atrial Bigeminy: Every Other Beat Is a PAC	N/A	Irregular	Premature and abnormal or hidden	< .20	< .12	C.4-a
PACs / Abberrantly Conducted PAC: QRS Is Bizarre	N/A	Irregular	Premature and abnormal or hidden	< .20	< .12	C.3-c
Nonconducted PAC	N/A	Irregular	Premature and abnormal or hidden	N/A	Absent	C.3-d
Atrial Tachycardia	140 – 250 bpm	Regular	Abnormal P before each QRS (difficult to see)	< .20	< .12	C.3-e
Atrial Flutter	A: 220 – 430 bpm , V: < 300 bpm	Regular or variable	Sawtoothed appearance	N/A	< .12	C.3-f
Atrial Fibrillation	A: 350 – 650 bpm , V: Slow to rapid	Irregular	Fibrillatory (fine to coarse)	N/A	.12	C.3-g
Junctional Rhythm	40 – 60 bpm	Regular	Inverted, absent or after QRS	< .12	< .12	C.3-h
Accelerated Junctional Rhythm	60 – 100 bpm	Regular	Inverted, absent or after QRS	< .12	< .12	C.4-c
Junctional Tachycardia	> 100bpm	Regular	Inverted, absent or after QRS	< .12	< .12	C.4-d
Wandering Pacemaker	< 60bpm	Irregular	Multiple forms	Variable	< .12	C.4-e

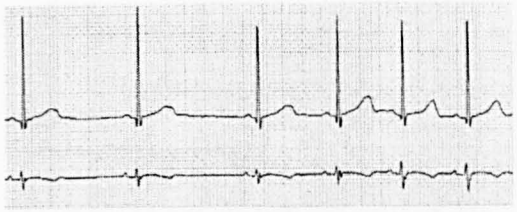
Table C.2: Arrhythmia recognition, continue

Disease	Characteristics	P Wave	PR(sec)	QRS(sec)	Figure
Right Bundle Branch Block	RsR' in V1	Before each QRS, identical	.12–.20	≥ .12	C.4-f
Left Bundle Branch Block	RR' in V5	Before each QRS, identical	.12–.20	≥ .12	C.4-g
Preexcitation Syndrome	Delta wave distorts QRS	Before each QRS, identical	< .12	Usually > .10	C.3-b
First Degree AV Block	Regular rhythm	Before each QRS, identical	> .20	< .12	C.4-h
Second Degree AV Block / Mobitz I (Wenckebach)	QRS dropped in a repeating pattern	Conduction Intermittent	Increasingly prolonged	< .12	C.4-b
Second Degree AV Block / Mobitz II	Fixed ratio of conduction (P:QRS)	Conduction Intermittent	Interval is constant	≥ .11 Usually a BBB pattern	C.2-h
Third Degree (Complete) AV Block	No relationship between P and QRS	Normal but not related to QRS	None	N/A	C.2-f

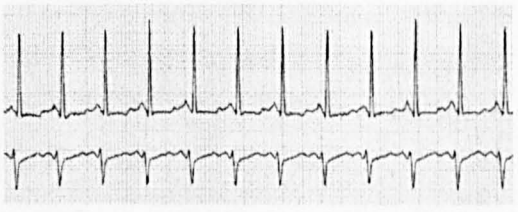


(a)

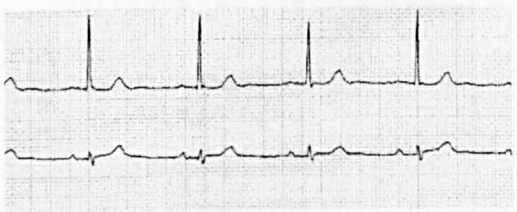
Figure C.1: Normal rhythm.



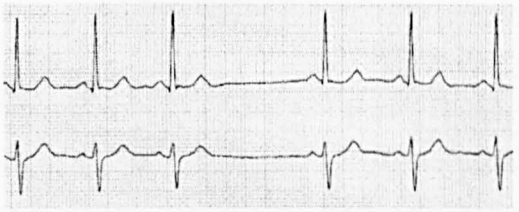
(a) Sinus Arrhythmia



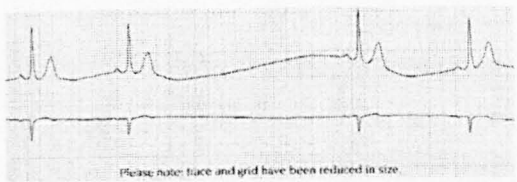
(b) Sinus Tachycardia



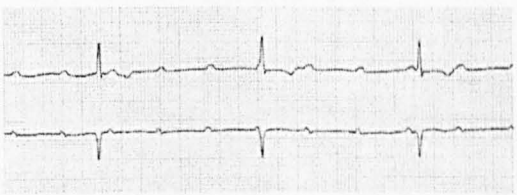
(c) Sinus Bradycardia



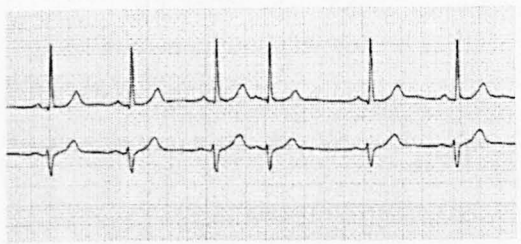
(d) Sinus Pause



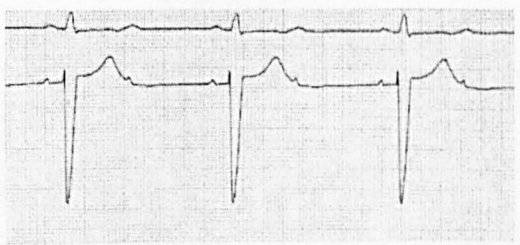
(e) Sinus Arrest



(f) AV Block (3rd)



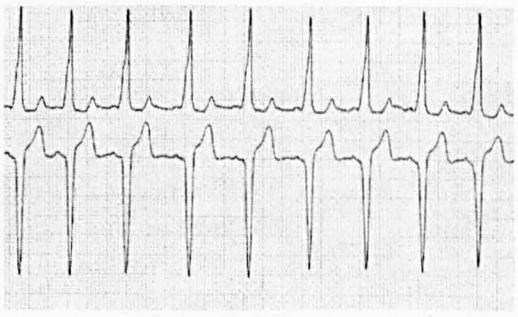
(g) Isolated PAC's Occur Single



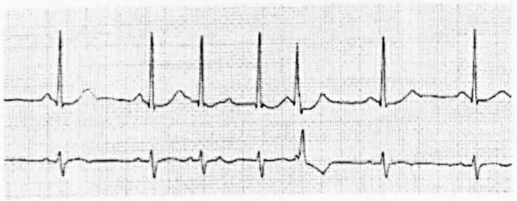
(h) AV Block (2nd)/Mobitz II

Figure C.2: Arrhythmia recognition(1).

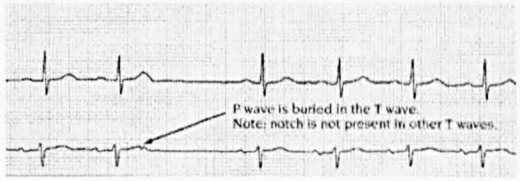
(a) Paired PAC's Occur in Two's



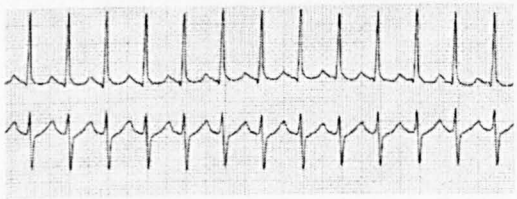
(b) Preexcitation Syndrome



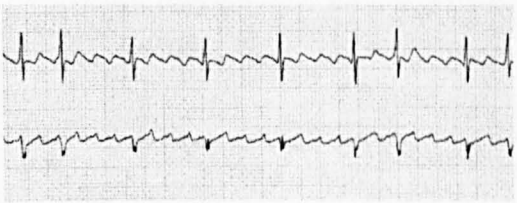
(c) Abberrantly Conducted PAC



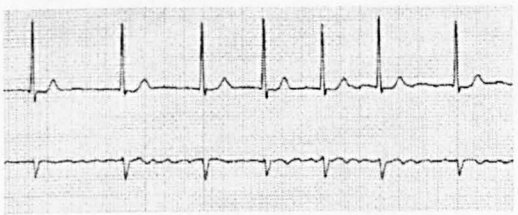
(d) Nonconducted PAC



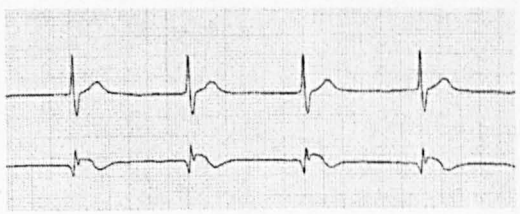
(e) Atrial Tachycardia



(f) Atrial Flutter

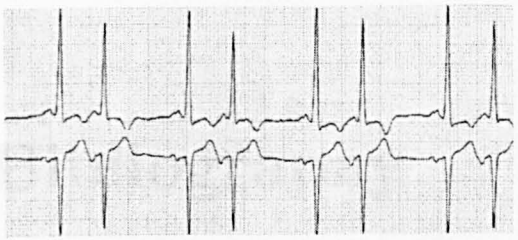


(g) Atrial Fibrillation

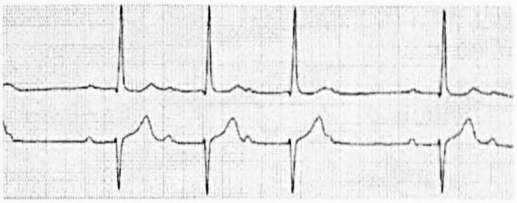


(h) Junctional Rhythm

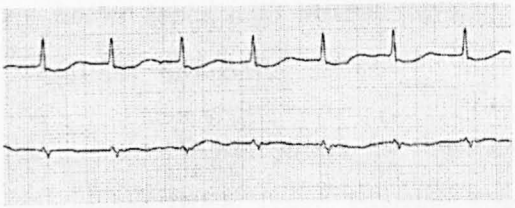
Figure C.3: Arrhythmia recognition(2).



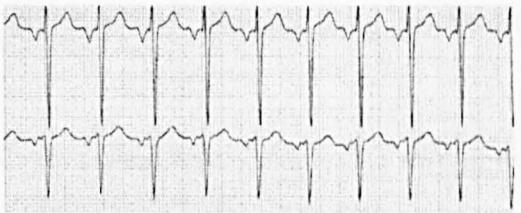
(a) PAC/Atrial Bigeminy



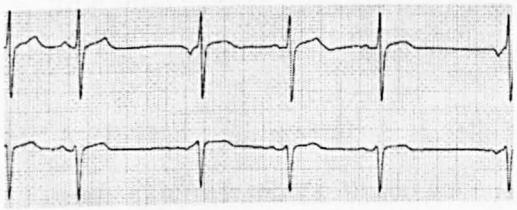
(b) AV Block (2nd) Mobitz I



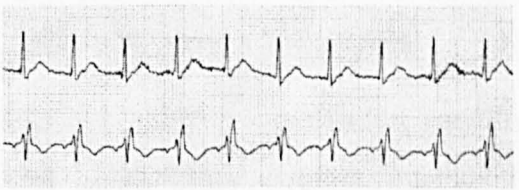
(c) Accelerated Junctional Rhythm



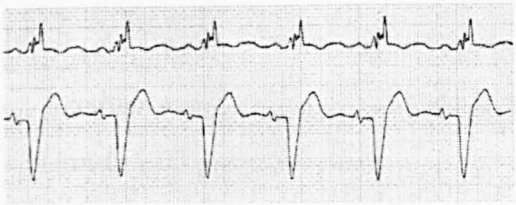
(d) Junctional Tachycardia



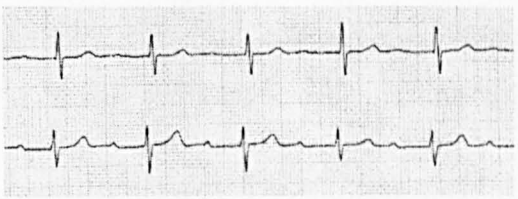
(e) Wandering Pacemaker



(f) Right Bundle Branch Block



(g) Left Bundle Branch Block



(h) AV Block(1st)

Figure C.4: Arrhythmia recognition(3).

Bibliography

[1] M. L. Ahlstrom and W. J. Tompkins, *Automated high-speed analysis of holter tapes with microcomputers*, IEEE Trans. Biomed. Eng. **BME-30** (1983), 651–657.

[2] ———, *Digital filters for real-time ecg signal processing using microprocessors*, IEEE Trans. Biomed. Eng. **BME-32** (1985), 708–713.

[3] J. A. V. Alste and T. S. Schilder, *Removal of baseline wander and power-line interference from the ecg by an efficient fir filter with a reduced number of taps*, IEEE Trans. Biomed. Eng. **BME-32** (1985), 1052-1060.

[4] A. J. Baddeley and H. J. A. M. Heijmans, *Incidence and lattice calculus with applications to stochastic geometry and image analysis*, Applicable Analysis in Engineering, Communication, and Computing **6** (1995), no. 3, 129–146.

[5] R. A. Balda, *The hp ecg analysis program*, Trends in computer processed electrocardiograms (J. H. vanBemnel and J. I. Willems, eds.), North Holland, 1977, pp. 197–205.

[6] R. G. Baraniuk, *Beyond time-frequency analysis: Energy densities in one and many dimensions*, Proceedings ICASSP'94 1994 International Conference on Acoustics, Speech and Signal Processing (Los Alamitos, CA, Adelaide, Australia), IEEE Signal Processing Society, 1994, pp. III:357–III:360.

[7] D. H. Bennett, *Cardiac arrhythmias: practical notes on interpretation and treatment*, Wright, Bristol, 1985.

[8] G. Birkhoff, *Lattice theory*, American Mathematics Society, Providence, Rhode Island, 1979.

[9] E. Braunwald (ed.), *Heart disease, a textbook of cardiovascular medicine*, vol. 1, W. B. Saunders Co., USA, 1997.

[10] D. R. Bull, *Gray-scale structuring element decomposition*, IEEE Trans. Industrial. Electronics **43** (1996), no. 5, 549–558.

[11] C. H. H. Chu and E. J. Delp, *Impulsive noise suppresion and background normalization of electrocardiogram signals using morphological operators*, IEEE Trans. Biomed. Eng. **36** (1989), no. 2, 262–273.

[12] E. R. Dougherty and E. Kraus, *Morphological pseudoconvolutions: one-parameter families of derived filters with increased invariant classes*, Circuits, Systems, Signal Process **11** (1992), no. 1, 195–228.

[13] M. Duff, *Parallel processors for digital image processing*, Advances in digital image processing (P. Stucki, ed.), Plentum, New York, 1979, pp. 265–279.

[14] L. Edenbrandt, B. Devine, and P. W. Macfarlane, *Classification of electrocardiographic st-t segments-human expert vs artificial neural network*, European Heart Journal **14** (1993), 464–468.

[15] M. Eleccion, *Automatic fingerprint identification*, IEEE Spectrum **10** (1973), 36–45.

[16] W. A. H. Engelse and C. Zeelenberg, *A single scan algorithm for qrs detection and feature extraction*, IEEE Comput. Card. (Long Beach), IEEE Computer Society, 1979, pp. 37–42.

[17] J. Fraden and M. R. Neoman, *Qrs wave detection*, Med. Biol. Eng. Comput. **18** (1980), 125–132.

[18] H. H. Friedman, *Diagnostic electrocardiography and vectorcardiography*, McGraw-Hill, New York, 1985.

[19] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, *A comparison of the noise sensitivity of nine qrs detection algorithms*, IEEE Trans. Biomed. Eng. **37** (1990), no. 1, 85–98.

[20] M. C. Fulton, *Ecg p wave analysis using neural networks*, Master's thesis, University of Liverpool, 1994.

[21] M. Gabbouj, E. J. Coyle, and N. C. Gallagher, *An overview of median and stack filtering*, Circuits, Systems, Signal Process **11** (1992), no. 1, 7–45.

[22] F. Gerritsen and L. G. Aardema, *Design and use of DIP-1: A fast flexible and dynamically microprogrammable image processor*, Pattern recognition **14** (1981), 319–330.

[23] M. J. E. Golay, *Hexagonal parallel pattern transformations*, IEEE Trans. Comput. **C-18** (1969), 733–740.

[24] A L. Goldberger and E. Goldberger, *Clinical electrocardiography: a simplified approach*, Mosby, St. Louis, 1994.

[25] D. Graham and P. E. Norgren, *The diff3 analyzer: A parallel/serial golay image processor*, Real-time medical image processing (M. Onoe, K. Preston, and A. Rosenfeld, eds.), Plentum, London, 1980, pp. 163–182.

[26] A. Grasselli, *On the automatic classification of fingerprints- some considerations of the linguistic interpretation of pictures*, Methodologis of pattern recognition (S. Watanabe, ed.), Academic, New York, 1969, pp. 253–273.

[27] F. Gritzali, G. Frangakis, and G. Papakonstantinou, *Detection of the p and t waves in an ecg*, Computers and Biomedical Research **22** (1989), 83–91.

[28] C. Gu, *Multivalued morphology and segmentation-based coding*, PhD thesis, École Polytechnique Fédérale De Lausanne, 1995.

[29] D Gustafson, *Automated vca interpretation studies using signal analysis techniques*, Research report R-1044, Charles Stark Draper Lab., 1977.

[30] H. Hadwiger, *Minkowskische addition und subtraktion beliebiger punktmengen und die theoreme von Erhard Schmidt*, Mathematische Zeitschrift **53** (1950), no. 3, 210–218.

[31] J. R. Hampton, *The ecg in practice*, Chuchill Livingstone, Edinburgh, UK, 1986.

[32] W. J. Hankley and J. T. Tou, *Automatic fingerprint interpretation and classification via contentual analysis and topological coding*, Pictorial Pattern recognition (G. C. Cheng, R. S. Ledley, D. K. Pollack, and A Rosenfeld, eds.), Tompson, Washington DC, 1968, pp. 411–456.

[33] R. M. Haralick, S. R. Sternberg, and X. Zhuang, *Image analysis using mathematical morphology*, IEEE Trans. Pattern Anal., Machine Intell. **PAMI-9** (1987), no. 4, 532–550.

[34] M. Hedlund, G. H. Granlund, and H. Knutsson, *A consistency operation for line and curve enhancement*, Conf. on Pattern Recognition and Image Processing, 1982, pp. 93–96.

[35] H. J. A. M. Heijmans and C. Ronse, *Annular filters for binary images*, Research report BS-R9604, CWI, 1996, Submitted to IEEE Trans. Image Proc.

[36] H.J.A.M. Heijmans, *Morphological Image Operators*, Academic Press, Boston, 1994.

[37] ———, *Mathematical morphology: Basic principles*, Proceedings of Summer School on Morphological Image and Signal Processing (Zakopane, Poland), 1995.

[38] W. P. Holsinger, *A qrs preprocessor based on digital differentiation*, IEEE Trans. Biomed. Eng. **BME-18** (1971), 212–217.

[39] P. T. Jackway, *Morphological scale-space with application to 3d object recognition*, PhD thesis, Queensland University of Technology, 1994.

[40] M. Kass and A. Witkin, *Analysing oriented patterns*, Comput. Vision, Graphics, Image Processing **37** (1987), 362–385.

[41] S. Khorsandi and A. N. Venetsanpoulus, *Hierarchical fuzzy approach for fingerprint processing*, Nonlinear image processing, vol. 1902, SPIE Proceedings, no. IV, 1993, pp. 224–234.

[42] M. Khosravi and R.W. Schafer, *Implementation of linear digital filters based on morphological representation theory*, IEEE Trans. Software Engrg. **42** (1994), no. 9, 2264–2275.

[43] J. C. Klein, L. Cahn, C. Ray, and G. H. Urban, *The texture analyzer*, J. Microscopy **95** (1977), 349–356.

[44] S. J. Ko, A. Morales, and K. H. Lee, *Fast recursive algorithms for morphological operators based on the basis matrix representation*, IEEE Trans. Image Proc. **5** (1996), no. 6, 1073–1077.

[45] M. K. Kouta and R. A. Ragab, *A new technique for fingerprint recognition*, The egyptian computer journal **23** (1995), no. 2, 117–131.

[46] R. Kresch, *Morphological image representation for coding applications*, PhD thesis, Israel Institute of Tech., 1995.

[47] B. Kruse, *Design and implementation of a picture processor*, PhD thesis, University of Linkoeping, Linkoeping, Sweden, 1977.

[48] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, *Face recognition: A convolutional neural network approach*, IEEE Trans. on Neural Networks **8** (1990), no. 1, 98–113.

[49] M. D. Levine, *Vision in Man and Machine*, vol. McGraw-Hill Series in Electrical Engineering, McGraw-Hill, New York, 1985.

[50] P. M. Mahoudeaux, *Simple microprocessor-based system for on-line ecg analysis*, Med. Biol. Eng. Comput. **19** (1981), 497–500.

[51] D. Maio and D. Maltoni, *direct gray-scale minutiae detection in fingerprints*, IEEE Trans. on Pattern Anal. and Machine Int. **19** (1997), no. 1, 27–40.

[52] P. Maragos, *A representation theory for morphological image and signal processing*, IEEE Trans. Pattern Anal., Machine Intell. **11** (1989), no. 6, 586–599.

[53] P. Maragos and R. W. Schafer, *Morphological systems for multidimensional signal processing*, Proceedings of the IEEE **78** (1990), no. 4, 690–710.

[54] G. Matheron, *Random sets and integrated geometry*, Wiley, New York, 1975.

[55] B. M. Mehtre, *Fingerprint image analysis for automatic identification*, Machine Vision and Applications **6** (1993), no. 2–3, 124–139.

[56] A. Menrad, *Dual microprocessor system for cardiovascular data acquisition, processing and recording*, Proceedings 1981 IEEE Int. Conf. Industrial Elect. Contr. Instrument., IEEE, 1981, pp. 64–69.

[57] B. Moayer and K.S. Fu, *A tree system approach for fingerprint pattern recognition*, IEEE Trans. Comput. **C-25** (1976), no. 3, 262–274.

[58] P. F. M. Nacken, *Image analysis methods based on hierarchies of graphs and multi-scale mathematical morphology*, PhD thesis, University of Amsterdam, 1994.

[59] M. Okada, *A digital filter for the qrs complex detection*, IEEE Trans. Biomed. Eng. **BME-26** (1979), 700–703.

[60] O. Pahlm and L. Sornmo, *Data processing of exercise ecg's*, IEEE Trans. Biomed. Eng. **BME-34** (1987), 158–165.

[61] T. Pavlidis, *A critical survey of image analysis methods*, Proc. 8th. Internat. Conf., Pattern Recognition, 1986, pp. 502–511.

[62] S. C. Pei and C. C. Tseng, *Elimination of ac interference in electrocardiogram using iir notch filter with transient suppression*, IEEE Trans. Biomed. Eng. **42** (1995), no. 11, 1128–1132.

[63] C. V. K. Rao and K. Balck, *Finding the core point in a fingerprint*, IEEE Trans. on Computers **C–27** (1978), no. 1, 77–81.

[64] ———, *Type classification of fingerprint: A syntactic approach*, IEEE Trans. on Pattern Anal. and Machine Int. **PAMI–2** (1980), no. 3, 223–231.

[65] N. K. Ratha, K. K. Shaoyun, and A. K. Jain, *A real-time matching system for large fingerprint databases*, IEEE Trans. on Pattern Anal. and Machine Int. **18** (1996), no. 8, 799–813.

[66] B. R. S. Reddy, P. E. Elko, D. W. Christenson, and G. I. Rowlandson, *Detection of p waves in resting ecg:a preliminary study*, Proceedings of the IEEE **80** (1992), 87–90.

[67] B. R. S. Reddy, P. P. Elko, D. W. Christenson, and G. I. Rowlandson, *Detection of complex atrial arrhythmias in resting ecg*, Proceedings of Computers in Cardiology (Bethesda, Maryland), IEEE Computer Society, September 1994, pp. 777–780.

[68] P. Salembier, *Morphological multiscale segmentation for image coding*, Signal Processing **38** (1994), 359–386.

[69] M. H. Sedaaghi, *Direct implementation of open-closing in morphological filtering*, Electronics Letters **33** (1997), no. 3, 198–199.

[70] M. H. Sedaaghi and P. J. G. Lisboa, *A direct technique for morphological filters*, Fifth Iranian Conference on Electrical Eng. (Tehran-Iran) (M. Ehsan, ed.), Sharif Univ. publication, May 1997, pp. 7.80–7.85.

[71] M. H. Sedaaghi and Q. H. Wu, *Application of morphological filtering in fingerprint processing*, Proceeding of the 3rd international Anual Computer Conference of the Computer Society of Iran (CSICC'97) 23-25 December, 1997 (Tehran-Iran), Iran Univ. of Science and Tech., 1997, pp. 31–36.

[72] ———, *Real-time implementation of grey-scale morphological operators*, Electronics Letters **33** (1997), no. 21, 1761–1763.

[73] ———, *Hardware for grey-scale morphological filtering*, CISST'98: The Proceedings of the 1998 International Conference on Imaging Science, Systems, and Technology , July 6-9, 1998 (Monte Carlo Resort & Casino, Las Vegas blvd., Las Vegas, Nevada, USA), CSREA Press, 1998, pp. 389- 395.

[74] ———, *The power of morphological filters alone and when combined with linear filtering*, ISMM'98: International Symposium On Mathematical

Morphology and its Applications to Image and Signal Processing IV, June 3-5, 1998 (Amsterdam, The Netherlands) (H. J. A. M. Heijmans and J. B. T. M. Roerdink, eds.), Computational Imaging and Vision, Kluwer Academic Publishers, 1998, pp. 375–381.

[75] ———, *A reliable hardware for grey-scale morphological filtering*, Proceedings of First International Symposium on Communication Systems and Digital Signal Processing 6-8 April 1998 (Sheffield-UK) (Z. Ghassemlooy and R. Saatchi, eds.), vol. 1, Sheffield Hallam University, 1998, pp. 147–150.

[76] ———, *Weighted morphological filter*, Electronics Letters **34** (1998), no. 16, 1566–1567.

[77] ———, *Morphological filtering and convolution*, UKACC'98 (United Kingdom Automatic Control Council) International Conference on CONTROL, University of Wales, Swansea, UK, 1-4 Sep. 1998, IEE, To appear.

[78] J. Serra (ed.), *Image analysis and mathematical morphology*, Academic Press, New York/London, 1982.

[79] ———, *Morphological filtering: an overview*, Signal Processing **38** (1994), 3–11.

[80] J. Serra and P. Soille, *Mathematical morphology and its applications to image processing*, Kluwer Academic Publishers, The Netherlands, 1994.

[81] J. Serra and L. Vincent, *An overview of morphological filtering*, Circuits Systems and Signal Processing **11** (1992), no. 1, 48–108.

[82] B. G. Sherlock, D. M. Monro, and K. Millard, *Algorithm for enhancing fingerprint images*, Electronics Letters **28** (1992), no. 18, 1720–1721.

[83] F. Y. Shih and P. Puttagunta, *Recursive soft morphological filters*, IEEE Trans. Image Proc. **4** (1995), no. 7, 1027–1031.

[84] J. Song and E. J. Delp, *A study of the generalized morphological filter*, Circuits, Systems, Signal Process **11** (1992), no. 1, 229–252.

[85] S. R. Sternberg, *Grayscale morphology*, Comp. Vision, Graph. and Image Proc. **35** (1986), no. 3, 333–355.

[86] E. N. Székely and V. Székely, *Image recognition problems of fingerprint identification*, Microprocessors and Microsystems **17** (1993), no. 4, 215–218.

[87] P. E. Trahanias, *An approach to qrs complex detection using mathematical morphology*, IEEE Trans. Biomed. Eng. **40** (1993), no. 2, 201–205.

[88] R. van den Boomgaard, *Mathematical morphology: Extensions towards computer vision*, PhD thesis, University of Amsterdam, 1992.

[89] D. Wang and D. C. He, *A fast implementation of 1-D grayscale morphological filters*, IEEE Trans. Circuits and Systems **41** (1994), no. 9, 634–636.

[90] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, *Stack filters*, IEEE Trans. Acoust. Speech Signal Process. **ASSP-34** (1986), 898–911.

[91] Y. Yao, R. Acharya, and S. Srihari, *Image enhancement using mathematical morphology with adaptive structuring elements*, Nonlinear image processing, vol. 2180, SPIE Proceedings, no. V, 1994, pp. 198–208.