

1994

## Digital Morphometry : A Taxonomy Of Morphological Filters And Feature Parameters With Application To Alzheimer's Disease Research

Andrew Mehnert  
*Edith Cowan University*

Follow this and additional works at: <https://ro.ecu.edu.au/theses>



Part of the [Analytical, Diagnostic and Therapeutic Techniques and Equipment Commons](#)

---

### Recommended Citation

Mehnert, A. (1994). *Digital Morphometry : A Taxonomy Of Morphological Filters And Feature Parameters With Application To Alzheimer's Disease Research*. Edith Cowan University. Retrieved from <https://ro.ecu.edu.au/theses/1468>

This Thesis is posted at Research Online.  
<https://ro.ecu.edu.au/theses/1468>

---

# ***DIGITAL MORPHOMETRY:***

## **A Taxonomy of Morphological Filters and Feature Parameters With Application to Alzheimer's Disease Research**

---

*A Thesis*

*Submitted to the*

*Faculty of Science & Technology*

*Edith Cowan University*

*Perth Western Australia*

*by*

*Andrew Mehnert*

*In Fulfilment of the*

*Requirements for the Degree*

*of*

*Master of Applied Science (Mathematics and Planning)*

*April 1994*

---

## Abstract

In this thesis the expression *digital morphometry* collectively describes all those procedures used to obtain quantitative measurements of objects within a two-dimensional digital image. Quantitative measurement is a two-step process: the application of geometrical transformations to extract the features of interest, and then the actual measurement of these features. With regard to the first step the *morphological filters* of *mathematical morphology* provide a wealth of suitable geometric transformations. Traditional radiometric and spatial enhancement techniques provide an additional source of transformations. The second step is more classical (e.g. Underwood, 1970; Bookstein, 1978; and Weibull, 1980); yet here again mathematical morphology is applicable – morphologically derived feature parameters. This thesis focuses on mathematical morphology for digital morphometry. In particular it proffers a taxonomy of morphological filters and investigates the morphologically derived feature parameters (Minkowski functionals) for digital images sampled on a square grid. "As originally conceived by Georges Matheron, mathematical morphology concerns the analysis of binary images by means of probing with structuring elements [typically convex geometric shapes]" (Dougherty, 1993, preface). Since its inception the theory has been extended to grey-level images and most recently to *complete lattices*. It is within the very general framework of the complete lattice that the taxonomy of morphological filters is presented. Examples are provided to help illustrate the behaviour of each type of filter.

This thesis also introduces DIMPAL (Mehnert, 1994) – a PC-based image processing and analysis language suitable for researching and developing algorithms for a wide range of image processing applications. Though DIMPAL was used to produce the majority of the images in this thesis it was principally written to provide an environment in which to investigate the application of mathematical morphology to Alzheimer's disease research. Alzheimer's disease is a form of progressive dementia associated with the degeneration of the brain. It "is the commonest type of dementia and probably accounts for half the dementia of old age" (Forsythe, 1990, p. 21). Post mortem examination of the brain reveals the presence of characteristic neuropathologic lesions; namely neuritic plaques and neurofibrillary tangles. They occur predominantly in the cerebral cortex and hippocampus. Quantitative studies of the distribution of plaques and tangles in normally aged and Alzheimer brains are hampered by the enormous amount of time and effort required to count and measure these lesions. Herein a morphological algorithm is proposed for the automatic segmentation and measurement of neuritic plaques from light micrographs of post mortem brain tissue.

---

## ***Declaration***

I certify that this thesis does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any institution of higher education and that, to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

\_\_\_\_\_  
Date 10/5/94

---

## ***Acknowledgements***

Foremost I wish to thank my principal supervisor, Associate Professor James Cross, for his inspiring enthusiasm, infinite patience, and unceasing support and encouragement. I would also like to express my appreciation to my co-supervisor, Dr Inta Adams, for her considerable efforts in providing me with tissue samples, equipment, and advice in relation to the Alzheimer's disease application (as well as to life in general). I thank John Green and Peter Northcott, Media Photographics, for providing me with photographic prints of the 35 mm slides of the photomicrographs provided by Dr Adams. I thank Suzanne Furby, CSIRO Division of Mathematics and Statistics, Perth WA, for scanning and converting these photographs into a raw digital image format. I thank my fellow students for their criticisms and words of encouragement. In particular I thank Stephan Bettermann for letting me use him as a sounding-board, and for convincing me to port DIMPAL from OS/2 version 1.3 to version 2.1. Last, but by no means least, I thank my parents for their love and support.

---

# *Table of Contents*

Abstract .....	ii
Declaration .....	iii
Acknowledgements .....	iv
 CHAPTER 1. INTRODUCTION.....	 1
1.1. Thesis Rationale .....	1
1.2. What is Digital Morphometry?.....	2
1.3. The Significance of This Research .....	3
1.4. Research Objectives .....	3
1.5 Structure of This Thesis .....	4
 CHAPTER 2. THEORETICAL FRAMEWORK.....	 6
2.1. About This Chapter.....	6
2.2. The Morphological Paradigm.....	6
2.3. What Constitutes an Image Filter? .....	7
2.4. Mathematical Morphology for Complete Lattices .....	8
2.4.1. Complete lattices .....	8
2.4.2. Dilations and erosions for complete lattices .....	10
2.4.3. A representation theorem for increasing mappings .....	13
2.4.4. Morphological openings and closings for complete lattices .....	13
2.4.5. Size distributions .....	14
2.4.6. Monotone continuity.....	15
2.4.7. Complete boolean lattices .....	15
2.4.8. Morphological filters .....	18
2.5. Euclidean Morphology .....	19
2.5.1. Image representation theory.....	19
2.5.1.1. A general representation theory .....	20
2.5.2. Dilation, erosion, opening, and closing.....	21
2.5.3. Duality between dilations/erosions and openings/closings.....	25
2.5.4. Anamorphoses.....	26
2.5.5. Representing functions by sets .....	27
2.5.5.1. Threshold decomposition.....	27
2.5.5.2. The umbra transform.....	28
2.5.5.3. Relating the umbra to the cross-sections of a function.....	28
2.5.6. The umbra homomorphism theorem .....	29

2.5.7. Implementing dilation and erosion for digital grey-level images .....	30
2.5.8. The genesis of mathematical morphology is more than just dilations and erosions .....	32
2.5.9. Topology for digital spaces .....	33
2.6. Linear Versus Morphological Filtering .....	36
 CHAPTER 3. A TAXONOMY OF MORPHOLOGICAL FILTERS .....	39
3.1. About This Chapter.....	39
3.2. A Brief History of Morphological Filters .....	39
3.3. Algebraic Openings and Closings .....	40
3.3.1. The Matheron representation theorem for translation-invariant openings and closings.....	42
3.3.2. How to construct openings and closings .....	43
3.3.2.1. Annular openings and closings.....	44
3.3.2.2. Generating openings from inf-overfilters and closings from sup-underfilters.....	46
3.3.2.3. The Matheron representation theorem for increasing and translation-invariant mappings .....	49
3.3.2.4. Generating openings and closings from increasing digital mappings .....	53
3.3.2.5. Rank-openings .....	54
3.4. Composite Morphological Filters .....	56
3.4.1. Open-closings and close-openings .....	57
3.4.2. General composition.....	58
3.5. The Middle Filter.....	59
3.6. Self Dual Filtering - The Morphological Centre .....	59
3.7. Alternating Sequential Filters .....	65
3.7.1. Derivates of the ASF.....	66
3.7.2. Digital ASFs.....	67
3.8. Multiple Structuring Element Filters.....	70
3.9. Generalised Morphological Filters .....	73
3.10. Soft Morphological Filters.....	75
3.11. Dolby Morphological Filters.....	78
 CHAPTER 4. DIGITAL MORPHOMETRY .....	82
4.1. About This Chapter.....	82
4.2. The Convex Set Model.....	83
4.2.1. Minkowski functionals.....	83

4.3. The Hit-or-Miss Transform.....	84
4.4. Connectivity Number.....	85
4.5. Area.....	88
4.6. Perimeter.....	88
4.6.1. Line segments and intercepts in $Z^2$ .....	89
4.6.2. Principal directions.....	90
4.6.3. A digital interpretation of Crofton's formula.....	90
4.7. Aspect Ratio Correction.....	91
 CHAPTER 5. D.I.M.P.A.L.....	 94
5.1. About This Chapter.....	94
5.2. Introduction.....	94
5.3. Lexical Conventions.....	96
5.3.1. Identifiers.....	96
5.3.2. Constants.....	97
5.3.3. Operators.....	97
5.3.4. Separators.....	98
5.4. Language Grammar.....	98
5.5. Variables.....	98
5.6. Expression Evaluation.....	100
 CHAPTER 6. MATHEMATICAL MORPHOLOGY AS A TOOL FOR THE MORPHOMETRIC INVESTIGATION OF NEURITIC PLAQUES ASSOCIATED WITH ALZHEIMER'S DISEASE.....	   103
6.1. About This Chapter.....	103
6.2. Alzheimer's Disease.....	103
6.3. Previous Research.....	104
6.4. The Investigation.....	106
6.4.1. Classifying neuritic plaques.....	107
6.4.2. Staining and image acquisition.....	107
6.4.3. Noise removal and background normalisation.....	109
6.4.4. Segmentation.....	113
6.4.4.1. Beucher's gradient.....	114
6.4.4.2. Selection of the threshold value.....	115
6.4.4.3. Connected component labelling.....	115
6.4.4.4. Border correction.....	115
6.4.4.5. Eliminating primitive plaques, and blood vessels.....	117
6.4.4.6. Boundary smoothing, and elimination of small plaques.....	117



6.4.5. Counts, area, and perimeter .....	118
6.4.6. Cluster analysis .....	123
6.5. Discussion .....	124
 CHAPTER 7. CONCLUSION.....	 127
7.1. Summary .....	127
7.2. Discussion .....	128
7.3. Future Directions .....	130
 BIBLIOGRAPHY.....	 132
 Appendix A. The Bound Matrix .....	 139
Appendix B. Properties of the Basic Morphological Operators.....	141
Appendix C. Minkowski Functionals in $\mathbf{R}^n$ .....	142
Appendix D. DIMPAL Commands and Functions .....	143
Appendix E. The Syntax of a DIMPAL Statement .....	160
Appendix F. Colour Print of a Light Micrograph.....	162
Appendix G. DIMPAL Script Files Used to Segment and Measure Neuritic Plaques.....	163
Appendix H. Segmented Images.....	169

---

---

# CHAPTER 1.

## INTRODUCTION

---

---

### 1.1. Thesis Rationale

I first developed an interest in the mathematics of digital image processing early in 1983 after reading an article in the now defunct computing journal, *Creative Computing*. The article (Cook, 1983), entitled *Pixel Averaging Smooths Graphics Effects*, detailed an elementary technique used by NASA to enhance satellite and probe imagery. As part of my undergraduate studies in applied mathematics I completed a small research project that dealt with radiometric (contrast) enhancement, spatial enhancement, and Fourier transformation of two-dimensional digital images. For a period of ten weeks, beginning in December 1989, I worked at the National Measurement Laboratory in Lindfield, NSW, on a CSIRO vacation scholarship. My work in the Division of Mathematics and Statistics entailed adding software routines to the division's developmental image processing package called Z. Some of these routines implemented elementary spatial domain filters (such as the moving average filter) with which I was already familiar. However I was also required to implement elementary binary morphology operators. This necessitated some preliminary background reading in mathematical morphology and sparked my interest in the subject. Subsequently, in 1990, I completed my honours dissertation entitled *Digital Image Processing Using Mathematical Morphology*. My honours paper provides a good introduction to the operations of digital binary and grey-scale morphology.

During the course of my honours research I realised that with the practical and theoretical advances in mathematical morphology since Jean Serra's milestone treatise, *Image Analysis and Mathematical Morphology* (1982), there was a need for a comprehensive summary of the morphological filters propounded in the literature. To this end this thesis proffers a taxonomy of morphological filters for digital image processing; the development of this taxonomy is presented in the general setting of the *complete lattice* (digital binary and grey-tone images are particular cases of this more general object space). This then provides the practitioner of morphology with a veritable plethora of filters for solving image analysis problems. This is much needed because although mathematical morphology provides an algebraic framework for describing morphological filters and algorithms, the practitioner still needs to rely heavily on experience and heuristic methods. With the provision of a taxonomy of filter types and examples of their behaviour the practitioner's job is made that little bit easier. In addition the Alzheimer's disease case study presented at the end of the thesis provides a practical example of the development of a morphological algorithm (and in particular the choice of appropriate morphological filters) to solve a *real world* image analysis problem.

---

The mathematical representation of a two-dimensional grey-tone or binary digital image (manifested on a square grid) by a *bound matrix* (see Appendix A) was introduced by Giardina and Dougherty (1988). Image operations and transformations can be readily expressed in terms of these matrices. This thesis introduces DIMPAL<sup>1</sup> (Mehnert, 1994); an acronym for *digital image processing and analysis language*. It is an equation-like language in which variables are collections of one or more bound matrices (images), and functions represent image operations. For example the morphological dilation of an image by a spherical structuring element of radius three is expressed as

```
result=dilate(image,sphere(3)).
```

DIMPAL is written in C for IBM's OS/2<sup>2</sup> operating system. It is, I believe, the only package of its kind available for this platform. Most of the images herein were produced using DIMPAL.

From my consultations with Dr Inta Adams (a neuroanatomist in the Department of Science, Edith Cowan University) during my honours research, I realised the possibility of applying mathematical morphology to the field of Alzheimer's disease research. Current research is hampered by the enormous amount of human effort and time required to perform counts and measure the area and other *morphometric* quantities of various characteristic brain lesions. This thesis presents an investigation (using DIMPAL) of the application of mathematical morphology to digitised light micrographs of neuritic plaques in post mortem brain tissue from Alzheimer's patients.

## 1.2. What is Digital Morphometry?

The word morphometry derives from the Greek words *morphē*, meaning structure or form, and *metrikós*, meaning measure. In this paper *digital morphometry* describes all those procedures used to obtain quantitative structural measurements of objects within a two-dimensional digital image. Typically such measurements include area, perimeter, and quantitative shape measurements of image features. Formally such measures are termed *feature parameters*. This use of the word morphometry contrasts sharply with its meaning in the biological context (Aherne & Dunhill, 1982, preface). Generally biologists use the word morphometry to describe those direct measurements, such as counts, made using procedures such as *serial sectioning*. The word *stereology* is then used to describe those geometrico-statistical procedures used to obtain quantitative measures.

---

<sup>1</sup> © Copyright 1991-1994 by Andrew Mehnert. All rights reserved.

<sup>2</sup> ® IBM, Operating System/2 and OS/2 are registered trademarks of International Business Machines Corporation.

In the context of digital imagery, morphometry necessarily encompasses the traditional radiometric and spatial enhancement techniques as well as those techniques offered by *mathematical morphology*. The application of these techniques is precursory to any mensuration of image features.

For the uninitiated, mathematical morphology is a mathematical formalism pioneered by Georges Matheron at the Paris School of Mines, Fontainebleau, France, in the late 1960s. In contrast to conventional *linear* filters, the sensible application of morphological filters preserves geometric structure within an image. The morphological approach involves the application of one or more geometric shapes (*structuring elements*) to an image in what amounts to a *fitting* operation. The internal geometry of the image under investigation is in essence probed by the chosen structuring elements. Hence mathematical morphology is particularly suited to object recognition, the generation of size distributions, and the generation of feature parameters (morphometrics) such as area and perimeter.

### 1.3. The Significance of This Research

This research is significant in the following respects:

- It provides a taxonomy of morphological filters.
- It provides a general-purpose digital image processing and analysis language (DIMPAL) for IBM's OS/2 operating system. At the time of writing no equivalent software package is available for this platform.
- It details the investigation, implementation, and evaluation of morphological techniques applied to the computer recognition and measurement of brain lesions (neuritic plaques) from digitised photographs of conventional light microscope images of post mortem brain tissue from Alzheimer's patients.

### 1.4. Research Objectives

This research addresses the following major questions:

- What are the fundamental classes of morphological filters, (e.g. opening, closing, open-closing, close-opening, alternating sequential, multiple structuring element) and what are their characteristics?
  - What are the elementary 2-D feature parameters (morphometrics) that can be generated using morphological operators for digital images manifested on the square grid?
  - Can mathematical morphology be used as a tool for the morphometric investigation of neuritic plaques, which are one of the microscopic hallmarks of Alzheimer's disease?
-

## 1.5 Structure of This Thesis

This thesis comprises four parts:

### *PART I*

#### *MORPHOLOGICAL FILTERS*

Chapter 2. Theoretical Framework

Chapter 3. A Taxonomy of Morphological Filters

### *PART II*

#### *PLANAR DIGITAL MEASUREMENT*

Chapter 4. Digital Morphometry

### *PART III*

#### *A DIGITAL IMAGE PROCESSING AND ANALYSIS LANGUAGE*

Chapter 5. D.I.M.P.A.L.

### *PART IV*

#### *ALZHEIMER'S DISEASE CASE STUDY*

Chapter 6. Mathematical Morphology as a Tool for the Morphometric Investigation of Neuritic Plaques Associated with Alzheimer's Disease.

---

---

# PART I.

## MORPHOLOGICAL FILTERS

---

---

---

## CHAPTER 2.

# THEORETICAL FRAMEWORK

---

---

### 2.1. About This Chapter

This chapter begins with a brief account of the history of digital image processing and in particular that of mathematical morphology. A formal definition of a signal (image) filter follows. The definition is sufficiently general so that it includes linear, order statistic (also called rank order), and morphological filters. Euclidean space and its digital derivatives are instances of a more general object space known as the *complete lattice*. This structure gives rise to a single theory that unifies classical Euclidean and digital morphology (for binary and grey-scale images). A review of mathematical morphology for complete lattices appears in section 2.4. The review concludes with the definition of a morphological filter. A treatment of Euclidean morphology (both binary and grey-scale), including its digital analogues, follows. This then establishes the necessary theoretical framework for Chapter 3 – the taxonomy of morphological filters; Chapter 4 – digital image measurement; and Chapter 6 – the Alzheimer's disease case study. The chapter closes with a short discussion of the rationale behind morphological filtering, and a classification of image filters based upon linearity and idempotence properties.

### 2.2. The Morphological Paradigm

Digital image processing and analysis evolved out of the space race begun in October 1957 with the (former) Soviet Union's successful launch of Sputnik 1 into orbit around the earth. The USA's Explorer 1 matched this feat some three months later. In October 1959, the Soviet probe Lunik 3 transmitted pictures of the dark side of the moon back to earth. The Soviets employed analogue methods to enhance these images. However, even after enhancement, the images were noisy, distorted, and out of focus. In comparison, when in July 1964 the American probe Ranger 7 began transmitting images of the lunar surface, NASA employed computers and digital techniques to enhance them. These techniques proved to be far more flexible and effective than the analogue methods used by the Soviets.

Linear filters, such as the mean (moving average) filter, are particularly good at removing noise from an image. However they also have a tendency to degrade the spatial characteristics of the image by blurring contours and edges. Whilst order statistic (also called ranked order) filters, such as the median filter, redress this problem (see Figure 1) they have other drawbacks; e.g. in contrast to the mean filter,

---

the median filter, under self-composition, does not yield another median filter. Whilst the Americans were developing essentially linear methods of digital image processing, such as spatial convolution (weighted moving averages) and frequency domain filtering via the Fourier transform, in Europe a non-linear methodology based on geometric shape was evolving independently and in parallel. "Mathematical morphology was born in 1964 when G. Matheron was asked to investigate the relationships between the geometry of porous media and their permeabilities" and J. Serra was asked to "quantify the petrography of iron ores, in order to predict their milling properties" (Serra, 1982, preface). Around this time the *Centre de Morphologie Mathématique* was established on the campus of the Paris School of Mines at Fontainebleau. Matheron's research culminated in his milestone paper, *Random Sets and Integral Geometry*, published in 1975. Serra (1982) formalised the theory of mathematical morphology borne of the mid to late 60s, in his book entitled *Image Analysis and Mathematical Morphology*. Mathematical morphology was originally formulated for binary images and was later extended to grey-level images by Serra (1982) using *cross-sections* and Sternberg (1986) using *umbræ*. More recently Matheron and Serra (in Serra, 1988) have generalised the theory of mathematical morphology to complete lattices. As a consequence, the theory finds application beyond that of just images; e.g. Vincent (1989) defines mathematical morphology for graphs. In comparison to traditional image processing, mathematical morphology has a strong algebraic basis and is thus more than just an ad hoc collection of procedures and methods.

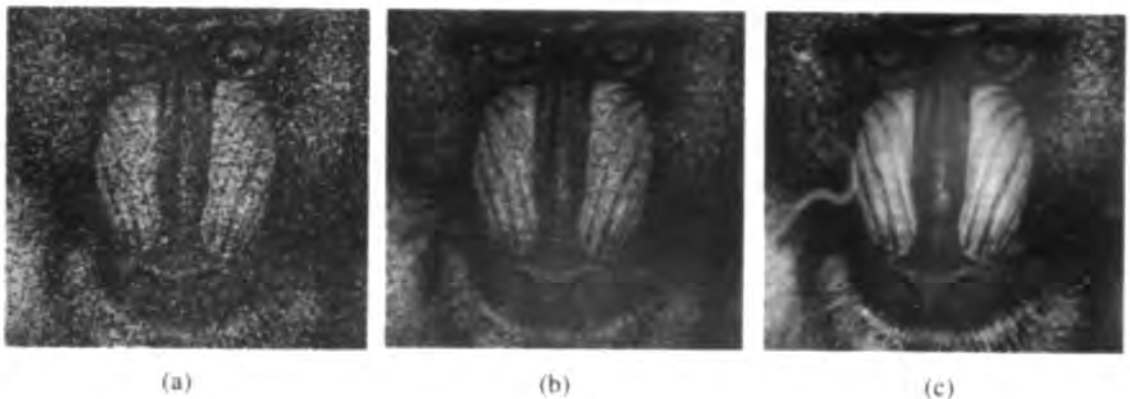


Figure 1. Linear filters tend to blur contours and edges.

- (a) An image corrupted with additive noise.
- (b) Result after applying a 5x5 moving average filter.
- (c) Result after applying a 5x5 median filter.

### 2.3. What Constitutes an Image Filter?

The term *filter* abounds in the image and signal processing literature. Indeed the term has been used thus far without formal definition. So what exactly is an image filter? Obviously it is some kind of



image to image transformation, the nature of the transformation being predetermined by the nature of the image analysis problem; e.g. removal of additive noise – mean filter, correction of motion blur – deconvolution. Serra (1988) states that "when dealing with a signal in one- or multidimensional space, the filter is commonly defined as any operator that is *linear, continuous and invariant under translation*" (p. 102). A transformation  $\psi$  is said to be linear if  $\psi(f + g) = \psi(f) + \psi(g)$  where  $f$  and  $g$  are signals. These transformations "preserve addition, and, beyond addition, the notion of group structure, thereby making reversibility an important feature" (Serra, 1988, p. 10). For example, linear methods are required to restore blurred images (a blurred image is the sum of several non-blurred images; a linear distortion). Morphological filters adopt the opposite tack. They preserve inclusion relationships existing between pairs of objects; i.e. they are *increasing* rather than linear transformations – in fact these two notions are incompatible (Serra, 1986, p. 288). Serra's definition of a filter is a little too restrictive. To accommodate morphological and order statistic filters (i.e. non-linear filters), the linearity requirement must be omitted; this leads to the following definition.

**Definition 1.**

With regard to one- or multidimensional signals (images), a filter is any *continuous and translation-invariant* operator.

## 2.4. Mathematical Morphology for Complete Lattices

The most recent theoretical advance in mathematical morphology has been its reformulation within the very general framework of the complete lattice. Classical Euclidean morphology is just a particular case of the more general theory. "This new framework allows [us] to introduce Mathematical Morphology in algebraic terms" (Vincent, 1989, p. 366). Through generalisation new insight is attained. This section begins with a review of some essential lattice theory. A review of mathematical morphology for complete lattices follows. The section concludes with a formal definition of a morphological filter. For a more complete exposé see Serra (1988), and Heijmans and Ronse (1990).

### 2.4.1. Complete lattices

The following definitions are generalisations of those found in Hungerford (1974), and Heijmans and Ronse (1990).

**Definition 2: The partial order relation.**

A binary relation  $R$  defined on a set  $\mathcal{L}$  is called a partial order relation if it is

- (i) *reflexive*: for all  $X \in \mathcal{L}$ ,  $XR X$  ( $X$  is related to  $X$ );
- (ii) *antisymmetric*: for all  $X, Y \in \mathcal{L}$ ,  $XY$  and  $YX \Rightarrow X=Y$ ;
- (iii) *transitive*: for all  $X, Y, Z \in \mathcal{L}$ ,  $XY$  and  $YZ \Rightarrow XZ$ .

*Example*

Consider the partial order relation *less than or equal to* ( $\leq$ ) defined on the set  $A = \{1, 2, 3\}$ . The relation can be explicitly defined by the following set  $S$  of ordered pairs:

$$S = \{(x, y) \mid x \leq y \text{ and } x, y \in A\} = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}.$$

*Definition 3: The poset.*

A non-empty set  $\mathcal{L}$  together with a partial order relation  $R$  is called a partially ordered set, abbreviated to *poset*, and denoted  $(\mathcal{L}, R)$ .

*Definition 4: The infimum.*

Given the poset  $(\mathcal{L}, R)$ , and  $\mathcal{X} \subseteq \mathcal{L}$ ,  $\mathcal{X} \neq \emptyset$ , then a *lower bound* of  $\mathcal{X}$  is an  $L \in \mathcal{L}$  such that  $L R X$  for every  $X \in \mathcal{X}$ . A *greatest lower bound* (infimum,  $\inf$ , or  $\wedge$ ) of  $\mathcal{X}$  is a lower bound  $L_0$  such that  $L R L_0$  for every other lower bound  $L$  of  $\mathcal{X}$ .

*Definition 5: The supremum.*

Given the poset  $(\mathcal{L}, R)$ , and  $\mathcal{X} \subseteq \mathcal{L}$ ,  $\mathcal{X} \neq \emptyset$ , then an *upper bound* of  $\mathcal{X}$  is a  $U \in \mathcal{L}$  such that  $X R U$  for every  $X \in \mathcal{X}$ . A *least upper bound* (supremum,  $\sup$ , or  $\vee$ ) of  $\mathcal{X}$  is an upper bound  $U_0$  such that  $U_0 R U$  for every other upper bound  $U$  of  $\mathcal{X}$ .

*Example*

For the previous example,  $\inf A = 1$ , and  $\sup A = 3$ .

*Definition 6: The lattice.*

A poset  $(\mathcal{L}, R)$  is a *lattice* if the set  $\{X, Y\}$  has both a supremum and an infimum for all  $X, Y \in \mathcal{L}$ .

*Definition 7: The complete lattice.*

A lattice  $(\mathcal{L}, R)$  is said to be *complete* if every  $\mathcal{X} \subseteq \mathcal{L}$ ,  $\mathcal{X} \neq \emptyset$ , has both a supremum and an infimum.

*Remarks*

- (i) Implicit in this definition is the existence of the *universal element* and the *null element*, respectively  $U = \sup \mathcal{L}$  and  $O = \inf \mathcal{L}$ . It follows that  $\forall X \in \mathcal{L}$ ,  $O R X$  and  $X R U$ .
- (ii) "Any element of  $\mathcal{L}$  is both an upper and a lower bound of the empty subset  $\emptyset$  of  $\mathcal{L}$ " and consequently  $\sup \emptyset = O$  and  $\inf \emptyset = U$  (Heijmans & Ronse, 1990, p. 254). Thus all subsets of  $\mathcal{L}$ , including the empty set  $\emptyset$ , have a supremum and infimum.

### Examples

- (i) The closed set of real numbers  $\overline{\mathbf{R}} = \mathbf{R} \cup \{-\infty, +\infty\}$  equipped with the usual ordering  $\leq$ , the usual supremum and infimum operators, universal element  $+\infty$ , and null element  $-\infty$ , is a complete lattice.
- (ii) The set of subsets of any set  $S$  along with set inclusion  $\subseteq$ , set union  $\cup$ , set intersection  $\cap$ , universal element  $S$ , and null element  $\emptyset$ , is a complete lattice.
- (iii) The set of all upper semi-continuous<sup>3</sup> (u.s.c.) functions  $\mathbf{R}^n \rightarrow \overline{\mathbf{R}}$  equipped with the partial order relation « defined  $(f \ll g \Leftrightarrow \forall x \in \mathbf{R}^n, f(x) \leq g(x))$ , is a complete lattice.

### Proposition 1: The principle of duality.

Let  $<$  be the generic symbol for the partial order relation meaning *is contained in or is less than or equal to*. Let  $>$  be the generic symbol for the partial order relation meaning *contains or is greater than or equal to*. The inverse of the partial order relation  $<$  is the partial order relation  $>$  and vice versa. It follows therefore that for the complete lattices  $(\mathcal{L}, <)$  and  $(\mathcal{L}, >)$ , for any  $\mathcal{X} \subseteq \mathcal{L}$ ,

$$\sup_{(\mathcal{L}, <)} \mathcal{X} = \inf_{(\mathcal{L}, >)} \mathcal{X} \quad \text{and} \quad \inf_{(\mathcal{L}, <)} \mathcal{X} = \sup_{(\mathcal{L}, >)} \mathcal{X}.$$

The lattice  $(\mathcal{L}, <)$  is said to be the *dual* of the lattice  $(\mathcal{L}, >)$  and vice versa. Furthermore, any statement of properties for  $(\mathcal{L}, <)$  can be restated in terms of  $(\mathcal{L}, >)$  by interchanging sup and inf, the universal element and the null element, and  $<$  and  $>$ .

Definitions 2 through 7 provide the most general framework in which the two rudimentary morphological operators, the *dilation* and *erosion*, can be defined. Within this framework and using these operators, the baser operations and concepts of mathematical morphology can be defined.

## 2.4.2. Dilations and erosions for complete lattices

Dilations and erosions are increasing mappings of a complete lattice into itself that commute with the supremum and the infimum respectively.

---

<sup>3</sup> Let  $f$  be a real function on a topological space. If  $\{x \mid f(x) \geq \alpha\}$  is closed (i.e. the set contains its boundary) for every real  $\alpha$ , then  $f$  is said to be *upper semi-continuous*. The sets  $\{x \mid f(x) \geq \alpha\}$ , for all real  $\alpha$ , are collectively called the *threshold sets* of  $f$ .

**Definition 8: Increasing mappings.**

Let  $(\mathcal{L}, R)$  be a complete lattice. A mapping  $\Gamma: \mathcal{L} \rightarrow \mathcal{L}$  is *increasing* or *monotone* if for all  $X, Y \in \mathcal{L}$ ,  $XRY \Rightarrow \Gamma(X)R\Gamma(Y)$ .

**Definition 9: The dilation and the erosion.**

Let  $(\mathcal{L}, R)$  be a complete lattice, and  $\Gamma$  be a mapping from  $\mathcal{L}$  into  $\mathcal{L}$ . For each subset of  $\mathcal{L}$  with elements  $X_i$ ,  $i \in \mathbf{I}$ ,

(i)  $\Gamma$  is called a *dilation* if it commutes with the supremum, that is

$$\Gamma\left(\bigvee_{i \in \mathbf{I}} X_i\right) = \bigvee_{i \in \mathbf{I}} \Gamma(X_i), \quad (1)$$

(ii)  $\Gamma$  is called an *erosion* if it commutes with the infimum, that is

$$\Gamma\left(\bigwedge_{i \in \mathbf{I}} X_i\right) = \bigwedge_{i \in \mathbf{I}} \Gamma(X_i). \quad (2)$$

**Remarks**

- (i) From the remarks accompanying Definition 7, it follows that  $\Gamma(O) = O$  for dilation, and  $\Gamma(U) = U$  for erosion.
- (ii) The dilation and the erosion are obviously increasing mappings.
- (iii) The dilation and the erosion are duals of each other in the sense that if the partial order relation of the lattice is reversed, a dilation becomes an erosion, and vice versa.

**Example**

Consider the mapping  $d: \overline{\mathbf{R}} \rightarrow \overline{\mathbf{R}}$  defined

$$d(x) = 3x + 1.$$

For the complete lattice  $(\overline{\mathbf{R}}, \leq)$ ,  $d$  defines a dilation (satisfying relation (1)), whilst for the dual lattice  $(\overline{\mathbf{R}}, \geq)$ ,  $d$  defines an erosion (satisfying relation (2)). For example, consider the subset  $\{2, 3\}$  of  $\overline{\mathbf{R}}$  for which

$$d\left(\sup_{(\overline{\mathbf{R}}, \leq)} \{2, 3\}\right) = \sup_{(\overline{\mathbf{R}}, \leq)} \{d(2), d(3)\} = 10, \text{ and}$$

$$d\left(\inf_{(\overline{\mathbf{R}}, \geq)} \{2, 3\}\right) = \inf_{(\overline{\mathbf{R}}, \geq)} \{d(2), d(3)\} = 10.$$

**Theorem 1:** (Serra, 1988, p. 17).

Let  $(\mathcal{L}, R)$  be a complete lattice. Then to each dilation  $\delta: \mathcal{L} \rightarrow \mathcal{L}$  there corresponds a unique erosion  $\varepsilon: \mathcal{L} \rightarrow \mathcal{L}$  and vice versa, such that

$$\delta(X)RY \Leftrightarrow XRe(Y), \quad X, Y \in \mathcal{L} \quad (3)$$

Furthermore,  $\varepsilon$  has the representation

$$\varepsilon(X) = \sup\{B \in \mathcal{L} \mid \delta(B)RX\}.$$

The pair  $(\varepsilon, \delta)$  is called an *adjunction* for which  $\varepsilon$  is called the *upper adjoint* and  $\delta$  is called the *lower adjoint* (Heijmans & Ronse, 1990, p. 264).

*Remarks*

- (i) If  $(\varepsilon, \delta)$  is an adjunction in the lattice  $(\mathcal{L}, <)$  then  $\delta(X) < Y \Leftrightarrow X < \varepsilon(Y)$ ,  $X, Y \in \mathcal{L}$ , where  $\delta$  and  $\varepsilon$  are dilation and erosion respectively on the lattice. By duality  $\varepsilon(X) > Y \Leftrightarrow X > \delta(Y)$ ,  $X, Y \in \mathcal{L}$  and hence  $(\delta, \varepsilon)$  is an adjunction in the lattice  $(\mathcal{L}, >)$ . This is still in agreement with relation (3) because from the point of view of the lattice  $(\mathcal{L}, >)$ ,  $\varepsilon$  is a dilation and  $\delta$  is an erosion.
- (ii) Setting  $X = \varepsilon\delta(Z)$  in relation (3) gives  $\delta\varepsilon\delta(Z)RY \Leftrightarrow \varepsilon\delta(Z)Re(Y)$ , where  $Z \in \mathcal{L}$ . For  $Y = \delta(Z)$ , the right-hand side of the double implication becomes an equality. Consequently the left-hand side must also be an equality, namely that  $\delta\varepsilon\delta(Z) = \delta(Z)$ . Similarly it can be shown that  $\varepsilon\delta\varepsilon(Z) = \varepsilon(Z)$ .
- (iii) The duality implicit in the adjunction  $(\varepsilon, \delta)$  is quite different from the duality w.r.t. order viz.

$$\begin{array}{cccc} \delta(X) & \Leftrightarrow & \varepsilon(X) & \text{and} & \varepsilon(X) & \Leftrightarrow & \delta(X). \\ (\mathcal{L}, <) & & (\mathcal{L}, >) & & (\mathcal{L}, <) & & (\mathcal{L}, >) \end{array}$$

*Example*

To the dilation on the complete lattice  $(\overline{\mathbf{R}}, \leq)$  defined in the previous example, there corresponds an erosion

$$e(x) = (x - 1) / 3$$

which clearly satisfies relation (3). For example,

$$d(x) \leq 13 \Leftrightarrow x \leq e(13).$$

### 2.4.3. A representation theorem for increasing mappings

The following theorem states that increasing mappings from a complete lattice into itself can be written as the supremum of erosions or by duality, the infimum of dilations. The theorem is a generalisation of a more specific result for Euclidean morphology known as the *Matheron representation theorem* for translation-invariant and increasing mappings – the theorem is discussed in the following chapter where it is used to express a median filter as the union of a finite set of erosions.

**Theorem 2. Representation theorem (Heijmans & Ronse, 1990, p. 262).**

The mapping  $\psi: \mathcal{L} \rightarrow \mathcal{L}$  defined on the complete lattice  $(\mathcal{L}, R)$  such that  $\psi(U) = U$  is increasing iff it can be expressed as the supremum of a non-empty set of erosions.

### 2.4.4. Morphological openings and closings for complete lattices

Under self composition a dilation  $\delta$  yields another dilation and an erosion  $\epsilon$  yields another erosion. But what can be said of the products  $\delta\epsilon$  and  $\epsilon\delta$ ? Well, if  $(\epsilon, \delta)$  is an *adjunction* then the products  $\delta\epsilon$  and  $\epsilon\delta$  define two very special operations – the *morphological opening* and the *morphological closing* respectively. Like the dilation and the erosion, the opening and closing are both increasing mappings. What makes them special, however, is their *extensivity* and *idempotence* properties.

**Definition 10. Extensivity of mappings.**

Let  $(\mathcal{L}, R)$  be a complete lattice. A mapping  $\psi: \mathcal{L} \rightarrow \mathcal{L}$  is

- (i) *extensive* if  $X R \psi(X) \forall X \in \mathcal{L}$  or
- (ii) *anti-extensive* if  $\psi(X) R X \forall X \in \mathcal{L}$ .

**Definition 11: Idempotent mappings.**

Given a complete lattice  $(\mathcal{L}, R)$ , any mapping  $\psi: \mathcal{L} \rightarrow \mathcal{L}$  for which  $\psi(\psi(X)) = \psi(X) \forall X \in \mathcal{L}$  is said to be *idempotent*.

In general the dilation and the erosion are non-invertible operators (in contrast to the previous example). That is to say, if  $(\mathcal{L}, R)$  is a complete lattice and  $X \in \mathcal{L}$ , then in general one cannot reconstruct  $X$  from  $\delta(X)$  or  $\epsilon(X)$ . By setting  $Y = \delta(X)$  in relation (3) it follows that  $X R \epsilon\delta(X) \forall X \in \mathcal{L}$  (extensivity). Similarly it follows that  $\delta\epsilon(X) R X \forall X \in \mathcal{L}$  (anti-extensivity).

**Definition 12: Morphological openings and closings.**

Let  $(\epsilon, \delta)$  be an adjunction in the complete lattice  $(\mathcal{L}, R)$ . The products  $\delta\epsilon$  and  $\epsilon\delta$  are idempotent operations called the morphological *opening* and the morphological *closing* respectively. Moreover (see Definition 10) the opening is anti-extensive, whilst the closing is extensive.

**Proof of idempotence**

Now  $X R \epsilon \delta(X)$  because the closing is extensive. This implies that  $\epsilon(X) R \epsilon \delta \epsilon(X)$ . It follows that (i)  $\delta \epsilon(X) R \delta \epsilon \delta \epsilon(X)$  (because  $\delta$  is increasing). However  $\delta \epsilon(X) R X$ , because the opening is anti-extensive, which implies that (ii)  $\delta \epsilon \delta \epsilon(X) R \delta \epsilon(X)$ . Comparing (i) and (ii) it follows that  $\delta \epsilon \delta \epsilon(X) = \delta \epsilon(X)$  and so the opening  $\delta \epsilon$  (and by duality, the closing  $\epsilon \delta$ ) is idempotent.

**Remarks**

- (i) The dilation  $\delta$  and the erosion  $\epsilon$  are, in general, neither extensive nor anti-extensive.
- (ii) Clearly because of the duality between dilation and erosion, each opening (resp. closing) w.r.t. the complete lattice  $(\mathcal{L}, <)$  is a closing (resp. opening) w.r.t. the complete lattice  $(\mathcal{L}, >)$ .

**2.4.5. Size distributions**

Size distributions (resp. anti-size distributions) are families of openings (resp. closings) "that are parameterized [sic] by a positive number (the size)" (Serra, 1988, p. 108). In the Euclidean context the resulting measures (e.g. area) from a series of such openings (or closings) provides "information on the relative representation of particles at different size scales" (Grivas & Skolnick, 1989, p. 214).

**Definition 13. Matheron's axioms for size distributions (Serra, 1988, p. 108).**

A family of mappings  $\{\gamma_\lambda\}$  in the lattice  $(\mathcal{L}, R)$  that depend upon a parameter  $\lambda \in \mathbf{R}^+$  is a *size distribution* if the following pair of axioms are satisfied:

$$\begin{cases} \text{(i) } \gamma_\lambda \text{ is an opening } \forall \lambda > 0 \\ \text{(ii) } \lambda, \mu > 0 \Rightarrow \gamma_\lambda \gamma_\mu = \gamma_\mu \gamma_\lambda = \gamma_{\sup(\lambda, \mu)} \end{cases}$$

**Remarks**

- (i) Axiom (ii) is equivalent to either  $\lambda \geq \mu > 0 \Rightarrow \gamma_\lambda(X) R \gamma_\mu(X) \forall X \in \mathcal{L}$ , or  $\lambda \geq \mu > 0 \Rightarrow \mathcal{B}_{\gamma_\lambda} \subseteq \mathcal{B}_{\gamma_\mu}$  where  $\mathcal{B}$  denotes *domain of invariance* (see Proposition 6).
- (ii) By duality a family of closings  $\{\varphi_\lambda\}$  that depend upon a parameter  $\lambda \in \mathbf{R}^+$  is an *anti-size distribution* if the following property is satisfied:  $\lambda, \mu > 0 \Rightarrow \varphi_\lambda \varphi_\mu = \varphi_\mu \varphi_\lambda = \varphi_{\sup(\lambda, \mu)}$ . This property is equivalent to either  $\lambda \geq \mu > 0 \Rightarrow \varphi_\mu(X) R \varphi_\lambda(X) \forall X \in \mathcal{L}$ , or  $\lambda \geq \mu > 0 \Rightarrow \mathcal{B}_{\varphi_\lambda} \subseteq \mathcal{B}_{\varphi_\mu}$ .

### 2.4.6. Monotone continuity

Before formally introducing the complete boolean lattice, and in preparation for the discussion on *alternating sequential filters* in Chapter 3, it is necessary to define *monotone continuity*. Let  $(\mathbf{I}, R)$  be a poset. Consider the family  $\{X_i\}$ ,  $i \in \mathbf{I}$ , of elements of the lattice  $(\mathcal{L}, R)$ . Note that for simplicity,  $R$  is used to denote the partial order relation of the lattice as well as that of the index set  $\mathbf{I}$ . The family  $\{X_i\}$  is said to converge monotonically to  $X$  from below, written  $X_i \uparrow X$ , if  $iRj \Rightarrow X_i R X_j$  and  $X = \bigvee X_i$ . Similarly, one can define monotonic convergence from above, written  $X_i \downarrow X$ . An increasing mapping  $\psi: \mathcal{L} \rightarrow \mathcal{L}$  is called  $\uparrow$ -continuous if  $X_i \uparrow X \Rightarrow \psi(X_i) \uparrow \psi(X)$  and is called  $\downarrow$ -continuous if  $X_i \downarrow X \Rightarrow \psi(X_i) \downarrow \psi(X)$ . A mapping that is both  $\uparrow$ - and  $\downarrow$ -continuous is called *continuous*. From Definition 9 it follows that dilations are  $\uparrow$ -continuous and erosions are  $\downarrow$ -continuous. The following theorem (Serra, 1988, p. 25) characterises monotone continuity for morphological openings and closings.

#### *Theorem 3.*

Let  $(\epsilon, \delta)$  be an adjunction in the lattice  $(\mathcal{L}, R)$ . The dilation  $\delta$  is a  $\uparrow$ -continuous mapping of  $\mathcal{L}$  into itself and the erosion  $\epsilon$  is a  $\downarrow$ -continuous mapping. Furthermore, if  $\delta$  is  $\downarrow$ -continuous (and therefore continuous) then the opening  $\delta\epsilon$  and the closing  $\epsilon\delta$  are also  $\downarrow$ -continuous. Similarly, if  $\epsilon$  is  $\uparrow$ -continuous then  $\delta\epsilon$  and  $\epsilon\delta$  are both  $\uparrow$ -continuous.

#### *Remarks*

- (i)  $\downarrow$ -continuity for  $\delta$  does not imply  $\uparrow$ -continuity for  $\epsilon$  (Serra, 1988, p. 25).
- (ii) "In general, neither openings nor closings are continuous" (Serra, 1988, p. 25) in the sense of monotone continuity.

### 2.4.7. Complete boolean lattices

Endowing a complete lattice with *distributivity* and *complementation* properties leads to a *complete boolean lattice*, or complete boolean algebra.



**Definition 14: The complete boolean lattice.**

If  $(\mathcal{L}, R)$  is a complete lattice and also satisfies the following properties, it is said to be a complete boolean lattice.

(i) *distributivity*:

$$\forall X, Y, Z \in \mathcal{L},$$

$$X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z), \text{ and by duality,}$$

$$X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z);$$

(ii) *complementation*:

$$\forall X \in \mathcal{L}, \exists X^c \in \mathcal{L} \text{ (called the complement of } X) \text{ such that } X \vee X^c = U, \text{ and by duality, } X \wedge X^c = O.$$

**Remark**

Any lattice  $(\mathcal{L}, R)$  is said to be *modular* if  $\forall X, Y, Z \in \mathcal{L}$ ,

$$Y \vee (X \wedge Z) = (Y \vee X) \wedge Z.$$

Consequently "any distributive lattice is modular, but the converse is not true in general" (Serra, 1988, p. 124).

The algebra of sets is a complete boolean algebra. Hence given a set  $S$ , the set of subsets (parts) of  $S$ , denoted  $\mathcal{P}(S)$ , is a complete boolean lattice, having the inclusion ( $\subseteq$ ) partial order relation, operators union  $\cup$  (supremum), intersection  $\cap$  (infimum), null element  $\emptyset$ , and universal element  $S$ . Binary (also called boolean) Euclidean morphology for two-dimensional digital images is founded on  $\mathcal{P}(\mathbb{Z}^2)$ ; i.e. sets of points in discrete space. As Serra (1988) points out, "new notions, such as connectivity and the skeleton, . . . now come to light" (p. 40). Vincent (1989) states that "one can show that the study of complete boolean lattices can be reduced by isomorphism<sup>4</sup> to the case of  $\mathcal{P}(S)$ " (p. 369).

The dilation  $\delta: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  is a mapping from the complete boolean lattice  $\mathcal{P}(S)$  into itself. However it can also be conceived as being generated from a mapping (called a *structuring function*) of the set  $S$  into  $\mathcal{P}(S)$ .

**Definition 15: The structuring function.**

Given an arbitrary set  $S$ , a structuring function is any mapping  $\Gamma: S \rightarrow \mathcal{P}(S)$ .

<sup>4</sup> Given two complete lattices  $(\mathcal{L}, R)$  and  $(\mathcal{L}', R')$  the mapping (bijection)  $\psi: \mathcal{L} \rightarrow \mathcal{L}'$  is an isomorphism if for any  $X, Y \in \mathcal{L}$  s.t.  $X R Y \Rightarrow \psi(X) R' \psi(Y)$ .

**Proposition 2: Equivalence between dilations and structuring functions.**

Let  $S$  be an arbitrary set. The structuring function  $\Gamma: S \rightarrow \mathcal{P}(S)$  uniquely determines a dilation  $\delta: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  as follows:

$$\delta(X) = \bigcup_{x \in X} \Gamma(x), \quad \forall X \in \mathcal{P}(S).$$

Conversely, to every dilation  $\delta: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  there corresponds a unique structuring function  $\Gamma: S \rightarrow \mathcal{P}(S)$ .

*Proof*

Now  $\delta(X) = \bigcup_{x \in X} \Gamma(x)$  is the set of points that *descend* from at least one point of the set  $X$  and is thus unique. It is a dilation because

$$\begin{aligned} \delta\left(\bigcup_i X_i\right) &= \bigcup_i \{\Gamma(x) \mid x \in X_i\} \\ &= \bigcup_i \delta(X_i). \end{aligned}$$

Conversely, any dilation  $\delta: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  induces a structuring function determined by all the correspondences  $\{x\} \rightarrow \delta(\{x\})$  (Serra, 1988, p. 41).

*Example*

Consider the Euclidean plane  $\mathbb{R}^2$ . Let  $D(x, \rho)$  denote the closed disk of radius  $\rho$  and centre  $x \in \mathbb{R}^2$ . Now define the structuring function  $\Gamma: x \rightarrow D(x, \rho)$ . From the above proposition it follows that

$$\delta(X) = \bigcup_{x \in X} \Gamma(x)$$

is a dilation. The reader already familiar with binary morphology will recognise this *swelling* operation as nothing more than the binary dilation of  $X$  by a closed disk structuring element, centred at the origin, and of radius  $\rho$ .

*Remark*

The converse part of proposition 2 is very important because it establishes the ubiquity of structuring functions.

**Proposition 3: Induced metrics.**

(Vincent, 1989, p. 369), (Serra, 1988, p. 47).

One can induce a metric on the space  $S$  by considering families of structuring functions  $\{\Gamma_\lambda\}$ , dependent on the parameter  $\lambda \in \mathbb{R}^+$ , that satisfy the following axioms:

- (i)  $\lambda \rightarrow 0^+ \Rightarrow \Gamma_\lambda \downarrow \Gamma_0$ ; where  $\Gamma_0$  is the identity mapping such that  $\Gamma_0(x) = \{x\} \quad \forall x \in S$ ;
- (ii)  $0 \leq \mu \leq \lambda \Rightarrow \Gamma_\mu \subseteq \Gamma_\lambda$ ;
- (iii)  $\Gamma_\lambda \Gamma_\mu \subseteq \Gamma_{\lambda+\mu}$ ,
- (iv)  $\forall x \in S, \exists \lambda_0 > 0$  such that  $\Gamma_{\lambda_0}(x) = S$ .

$S$  is then a metric space with distance relation

$$d(x, y) = \inf\{\lambda \mid x \in \Gamma_\lambda(y), y \in \Gamma_\lambda(x)\}, \quad x, y \in S.$$

**Example**

Consider the Euclidean distance between two points in  $\mathbb{R}^2$ . This metric can be induced on the space  $\mathbb{R}^2$  by considering families of closed disks. Let  $\Gamma_\lambda(x)$  denote the closed disk of radius  $\lambda \in \mathbb{R}, \lambda \geq 0$ , centred at  $x \in \mathbb{R}^2$ . (In Euclidean morphology,  $\Gamma_\lambda(x)$  is the binary dilation of  $x$  by a closed disk structuring element, centred at the origin, and of radius  $\lambda$ ). The distance relation of Proposition 3 is then clearly the Euclidean distance.

Endowing  $S$  with a metric makes it very easy to define *granulometries* (size distributions) and makes possible the formulation of *morphometric quantities* (feature parameters). Note "that size distributions do not depend directly on metric analyses, although it is sometimes wrongly stated in the literature that they do" (Serra, 1982, p. 165). The complete lattice structure is sufficient to define granulometries; however, with the investiture of a metric one can readily speak of measures such as area in relation to the granulometry. For instance, for the reader familiar with binary mathematical morphology in the Euclidean plane, one can conceive of generating a distribution of image area remaining after successive openings by disks of increasing radii.

**2.4.8. Morphological filters**

Two different definitions for the *morphological filter* can be found in the general literature on Euclidean morphology. These are considered in section 2.6. In the context of a complete lattice, however, a morphological filter is an increasing and idempotent (see Definition 16) mapping of the lattice into itself (Serra, 1988, p. 104). The reasons for considering this class of mappings are outlined at the end of the chapter.

**Definition 16: The morphological filter.**

Given a complete lattice  $(L, R)$ , every mapping  $\psi: L \rightarrow L$  that is both increasing and idempotent is called a *morphological filter*.

**Remarks**

- (i) Openings and closings are the most elementary morphological filters.
- (ii) Dilations and erosions do not constitute morphological filters because, although they are increasing mappings, they are not in general, idempotent.

## 2.5. Euclidean Morphology

Mathematical morphology was initially formulated for binary signals (images), viewed as sets of points in  $\mathbb{R}^n$ , by Matheron and Serra (see Serra, 1982). The extension of the theory to multilevel (grey-scale) signals is due mainly to Serra, who used cross-sections to generalise, and Sternberg, who used umbrae to generalise (see Serra, 1982; and Sternberg, 1986). A review of Euclidean morphology follows.

### 2.5.1. Image representation theory

Consider an ink pen drawing on a piece of white paper. The image can be modelled by a set of points in  $\mathbb{R}^2$  that locate the image foreground. "The concept of a set, however, is more general than needed to represent signals [binary images]" (Maragos & Schafer, 1987a, p. 1154). The topologically closed set, i.e. a set that contains its boundary, is sufficiently general to represent a binary image. Furthermore, when dealing with granulometries and image-functionals (morphometrics), it is the class of compact (i.e. closed and bounded) convex sets that are important (see Chapter 4). Now consider a *black-and-white* photograph. Black-and-white is a misnomer because the photograph actually consists of grey tones. This image can be modelled by a real valued function of two variables with domain in  $\mathbb{R}^2$ . The function describes the grey-level (brightness) surface of the image; i.e. the value of the function at any point  $(x, y)$  in its domain is a measure of intensity. Once again, the model is more general than is needed. Serra (1982), and Maragos and Schafer (1987a) deal with u.s.c. functions (see Footnote 3). "The notion of an upper semi-continuous function . . . defined on the plane corresponds to that of a closed set" (Serra, 1982, p. 425); more on this later. As far as the definition of morphological operators for binary and grey-scale images is concerned, one need not worry about closed sets and u.s.c. functions respectively. It is only when "one wishes to deal with topological or probabilistic aspects . . . [that] such functions [resp. sets] become very important" (Heijmans, 1991, p. 577). Moreover, when dealing with discrete (sampled) images, upper semi-continuity and closed sets are no longer an issue. "Discrete-domain signals are trivially upper semicontinuous because all their threshold sets [see Footnote 3] are subsets of  $\mathbb{Z}^m$  and hence closed" (Maragos, 1989, p. 591).

Mathematical morphology for  $n$ -dimensional ( $n$ -D) binary images is founded on the complete boolean lattice  $\mathcal{P}(\mathbf{E}^n)$ , where  $\mathbf{E} = \mathbf{R}$  for continuous images and  $\mathbf{E} = \mathbf{Z}$  for discrete (sampled) images. The lattice has the null element,  $\emptyset$ , universal element,  $\mathbf{E}^n$ , the inclusion  $\subseteq$  partial order relation, set union  $\cup$  for supremum, and set intersection  $\cap$  for infimum. Mathematical morphology for  $n$ -D grey-scale images is founded on the complete (but not boolean) lattice of all functions  $\mathcal{F}:\mathbf{E}^n \rightarrow \bar{\mathbf{E}}$ . The lattice has the null element,  $O$  s.t.  $O(x) = -\infty$ , universal element,  $U$  s.t.  $U(x) = +\infty$ , the usual supremum and infimum (which reduce to maximum and minimum respectively when  $\mathbf{E} = \mathbf{Z}$ ), and the partial order relation  $\ll$  defined ( $f \ll g \Leftrightarrow D_f \subseteq D_g$  and  $f(x) \leq g(x) \forall x \in D_f$ ), where  $D_f \subset \mathbf{E}^n$  is the domain of  $f$ . Consider the following definitions of translation-invariant operators for binary and grey-scale images.

**Definition 17. Translation-invariance for  $n$ -D binary images.**

A mapping  $\psi:\mathcal{P}(\mathbf{E}^n) \rightarrow \mathcal{P}(\mathbf{E}^n)$  is said to be translation-invariant if  $\psi(A_h) = [\psi(A)]_h$  for all  $A \in \mathcal{P}(\mathbf{E}^n)$ ,  $h \in \mathbf{E}^n$ , where  $A_h$  denotes the translate of  $A$  along the vector  $h$ ; i.e.  $A_h = \{a+h \mid a \in A\}$ .

**Definition 18. Translation-invariance for  $n$ -D grey-level images.**

A mapping  $\psi:\mathcal{F} \rightarrow \mathcal{F}$  is said to be *spatially* or *horizontally* translation-invariant if  $\psi(f_h(x)) = [\psi(f(x))]_h$  for all  $f \in \mathcal{F}$ ,  $h \in \mathbf{E}^n$ , where  $f_h$  denotes the translate of  $f(x)$  along the vector  $h$ ; i.e.  $f_h(x) = f(x-h)$ . The mapping is said to be *grey-level* or *vertically* translation-invariant if  $\psi(f(x)+i) = \psi(f(x)) + i$  for all  $f \in \mathcal{F}$ , and  $i \in \bar{\mathbf{E}}$ .

**Remarks**

- (i) In the general image processing literature the term *translation-invariance* usually refers to spatial translation-invariance only. This is precisely its meaning in Definition 1.
- (ii) Grey-scale dilation, erosion, opening, and closing are horizontally and vertically translation-invariant provided that the grey-level set (codomain) is closed under addition.
- (iii) Heijmans and Ronse (1990, p. 272) generalise the notion of translation-invariance by considering certain automorphisms<sup>5</sup> on arbitrary complete lattices.

### 2.5.1.1. A general representation theory

A "unifying theory for many concepts and operations encountered in or related to morphological image and signal analysis" was developed by Maragos (1989, p. 586). His general theory describes any system of signal-to-signal transformations that are "translation-invariant, increasing (preserve a signal ordering), and semicontinuous (insensitive to very fine signal details)" (p. 586). In particular, the theory establishes a classification of filters according to the type of input required and the

<sup>5</sup> An automorphism on the lattice  $(\mathcal{L}, R)$  is an isomorphism  $\mathcal{L} \rightarrow \mathcal{L}$ .

type of output produced. In the theory, binary images are exclusively represented by sets, and grey-scale images by functions. Filters can then be placed into one of three categories: SP (set processing), FP (function processing), and FSP (function and set processing). An SP filter is one whose input and output are both binary images (sets); e.g. the set complement operation. An FP filter is one whose input and output are both grey-scale images (functions); e.g. scalar multiplication of a function by negative one. FSP filters are a hybrid of FP and SP filters as they produce binary output given a binary input, and grey-scale output given a grey-scale input; e.g. a  $3 \times 3$  median filter for 2-D images. Now a binary image can be equally represented as a *flat* grey-scale image (see Figure 2). Hence Maragos and Schafer (1987a) refer to FSP filters as a subclass of FP filters. This is true in the sense that binary inputs and outputs are subsumed by grey-tone inputs and outputs respectively.

#### Example

A  $3 \times 3$  median filter is a FSP filter because its application to a binary image yields a binary image, whilst its application to a grey-tone image yields a grey-tone image. In contrast a  $3 \times 3$  moving average filter is not a FSP filter but rather a FP filter.

$$f = \begin{bmatrix} * & 0 & * \\ 0 & 0 & 0 \\ * & 0 & * \end{bmatrix}_{-1,1} \quad A = \begin{bmatrix} * & 1 & * \\ 1 & 1 & 1 \\ * & 1 & * \end{bmatrix}_{-1,1}$$

Figure 2. A flat grey-tone image  $f$  can be equally represented as a binary image  $A$  (see Appendix A).

### 2.5.2. Dilation, erosion, opening, and closing

Recall that a morphological filter is defined to be both increasing and idempotent (Definition 16). Furthermore, in accordance with Definition 1, in the Euclidean setting a morphological filter has to be translation-invariant. These requirements naturally lead to the following definitions for dilation and erosion, and by composition to opening and closing (see the remarks following Definition 19). Appendix B lists some of the properties of these morphological operations.

**Definition 19. Dilation.**

Given two binary images  $A, B \in \mathcal{P}(\mathbb{E}^n)$ , the dilation of the image  $A$  by the structuring element  $B$  is defined

$$\mathcal{B}(A, B) = A \oplus B = \bigcup_{b \in B} A_b. \quad (4)$$

Given two grey-level images  $f, g \in \mathcal{I}$  with domains  $D, E \subset \mathbb{E}^n$  respectively, the dilation of the image  $f$  by the structuring element  $g$  has domain  $\mathcal{B}(D, E)$  and is defined point-wise as follows

$$\begin{aligned} \mathcal{B}(f, g)(x) &= (f \oplus g)(x) \\ &= \bigvee_{\substack{z \in E \\ (x-z) \in D}} \{f_z(x) + g(z)\} \\ &= \bigvee_{\substack{z \in E \\ (x-z) \in D}} \{f(x-z) + g(z)\}. \end{aligned} \quad (5)$$

**Remarks**

- (i) The symbol  $\oplus$  denotes Minkowski addition which is defined to be the set of points formed by the vector addition of all possible pairs of points  $a \in A$ , and  $b \in B$ , where  $A, B \in \mathcal{P}(\mathbb{E}^n)$ ; i.e.

$$A \oplus B = \{a + b \mid a \in A, b \in B\} = \bigcup_{b \in B} A_b = \bigcup_{a \in A} B_a.$$

The symbol still remains valid for grey-level dilation because a grey-level dilation can be resolved into a set of binary dilations (see section 2.5.5.).

- (ii) With respect to relation (4), Serra (1988) states that "*the class of dilations . . . that are invariant under translation coincides with the Minkowski additions . . .  $X \oplus B$* " (p. 73). Moreover, "Minkowski addition and subtraction [see the following definition] are the only translation-invariant dilations and erosions of a Euclidean or digital space" (Heijmans & Ronse, 1990, p. 252).
- (iii) In relation (4) "the structuring function [see Proposition 2] at point  $h$  is deduced from that of the origin by translation  $h$ . The function is thus reduced to only one set, up to a translation, which we call a *structuring element*" (Serra, 1988, p. 73).
- (iv) Though both binary and grey-level dilation are commutative, i.e.  $\mathcal{B}(A, B) = \mathcal{B}(B, A)$  and  $\mathcal{B}(f, g) = \mathcal{B}(g, f)$ , the first argument is generally referred to as the *image*, and the second as the *structuring element*.
- (v) Maragos (1989, p. 589) defines grey-scale dilation (restricted to u.s.c. functions) as follows:

$$\mathcal{B}(f, g)(x) = \bigvee_{y \in (\text{Spt}(f) \cap \text{Spt}(\tilde{g}) + x)} \{f(y) + g(x - y)\}, \quad x \in \mathbb{E}^n,$$

where  $\tilde{g}(x) = g(-x)$ , and  $\text{Spt}(f)$  is the support of  $f$ ; i.e.  $\text{Spt}(f) = \{x \in \mathbb{E} \mid f(x) \neq -\infty\}$ .

This is essentially equivalent to relation (5), keeping in mind that in practice, "by convention, [we let]  $f(x) = g(x) = -\infty$  for  $x \notin \text{support of } f \text{ (resp. } g\text{)}$ " (Serra, 1982, p. 443). The partial order relation « inherent in (5) reduces to  $\leq$  if this convention is adopted (Dougherty, 1989, p. 174).

**Definition 20. Erosion.**

Given two binary images  $A, B \in \mathcal{P}(\mathbf{E}^n)$ , the erosion of the image  $A$  by the structuring element  $B$  is defined

$$\mathcal{E}(A, B) = A \ominus \tilde{B} = \bigcap_{b \in B} (A)_{\tilde{b}}, \quad (6)$$

where  $\tilde{B} = \{-b \mid b \in B\}$  is called the *transposed* or *symmetric* set of  $B$  w.r.t. the origin.

Given two grey-level images  $f, g \in \mathcal{I}$  with domains  $D, E \subset \mathbf{E}^n$  respectively, the erosion of the image  $f$  by the structuring element  $g$  has domain  $\mathcal{E}(D, E)$  and is defined point-wise as follows

$$\begin{aligned} \mathcal{E}(f, g)(x) &= (f \ominus \tilde{g})(x) \\ &= \bigwedge_{z \in E} \{f_z(x) - g(z)\} \\ &= \bigwedge_{z \in E} \{f(x+z) - g(z)\}, \end{aligned} \quad (7)$$

where  $\tilde{g}(x) = g(-x)$ .

**Remarks**

- (i) The symbol  $\ominus$  denotes Minkowski subtraction which is defined  $A \ominus B = \bigcap_{b \in B} A_b$ , where  $A, B \in \mathcal{P}(\mathbf{E}^n)$ .
- (ii) The binary erosion defined in relation (6) is equivalent to  $\mathcal{E}(A, B) = \{x \mid B_x \subseteq A\}$  (Serra, 1982, p.43; Giardina & Dougherty, 1988, p. 6)
- (iii) One needs to exercise discretion when consulting the literature as the symbol  $\ominus$  is often used to mean erosion rather than Minkowski subtraction (e.g. Sternberg, 1986; Haralick, Sternberg, & Zhuang, 1987; Heijmans, 1991).

Openings and closings follow from Definition 12 viz.



**Morphological openings and closings.**

Given two binary images  $A, B \in \mathcal{P}(\mathbf{E}^n)$ ,

- (i)  $\mathcal{O}(A, B) = \mathcal{B}(\mathcal{E}(A, B), B)$  is the opening, and
- (ii)  $\mathcal{C}(A, B) = \mathcal{E}(\mathcal{B}(A, B), B)$  is the closing

of the image  $A$  by the structuring element  $B$ .

Given two grey-level images  $f, g \in \mathcal{I}$  with domains  $D, E \subset \mathbf{E}^n$  respectively,

- (i)  $\mathcal{O}(f, g) = \mathcal{B}(\mathcal{E}(f, g), g)$  is the opening, and
- (ii)  $\mathcal{C}(f, g) = \mathcal{E}(\mathcal{B}(f, g), g)$  is the closing

of the image  $f$  by the structuring element  $g$ .  $\mathcal{O}(f, g)$  has domain  $\mathcal{O}(D, E)$  and  $\mathcal{C}(f, g)$  has domain  $\mathcal{C}(D, E)$ .

Binary dilations, erosions, openings, and closings are SP filters, whilst their grey-level counterparts are FP filters if the structuring elements are non-flat, and FSP filters otherwise. In the case of flat structuring elements, relations (5) and (7) reduce to

$$\begin{aligned}\mathcal{B}(f, B)(x) &= (f \oplus B)(x) \\ &= \bigvee_{\substack{b \in B \\ (x-b) \in D}} \{f(x-b)\} \\ &= \sup \{f(y) \mid y \in \tilde{B}_x\}\end{aligned}\tag{8}$$

$$\begin{aligned}\mathcal{E}(f, B)(x) &= (f \ominus \tilde{B})(x) \\ &= \bigwedge_{b \in B} \{f(x+b)\} \\ &= \inf \{f(y) \mid y \in B_x\}.\end{aligned}\tag{9}$$

The SP, FP, and FSP classification of binary and grey-scale dilation, erosion, opening, and closing is given in Table 1.

Table 1. SP, FP, and FSP filters.

SP Filters	FP Filters	FSP Filters
$A^c$	$-f$	
$\bar{A}$	$\bar{f}$	
$A - B$ (set difference)	$f - g$	
$A_h$	$f_h$	
$A \oplus B$	$f \oplus g$	$f \oplus B$
$A \ominus B$	$f \ominus g$	$f \ominus B$
$\mathcal{D}(A, B)$	$\mathcal{D}(f, g)$	$\mathcal{D}(f, B)$
$\mathcal{E}(A, B)$	$\mathcal{E}(f, g)$	$\mathcal{E}(f, B)$
$\mathcal{O}(A, B)$	$\mathcal{O}(f, g)$	$\mathcal{O}(f, B)$
$\mathcal{C}(A, B)$	$\mathcal{C}(f, g)$	$\mathcal{C}(f, B)$

### 2.5.3. Duality between dilations/erosions and openings/closings

It is easy to show that the duality relationships expressed in the following two propositions hold true.

**Proposition 4. Duality between dilation and erosion.**

Binary dilation and erosion satisfy the following duality w.r.t. complementation:

$$\begin{aligned}\mathcal{D}(A, B) &= \left[ \mathcal{E}(A^c, \bar{B}) \right]^c \\ \mathcal{E}(A, B) &= \left[ \mathcal{D}(A^c, \bar{B}) \right]^c\end{aligned}$$

where  $A, B \in \mathcal{P}(\mathbf{E}^n)$  and  $A^c = \{x \mid x \notin A\}$ .

Grey-scale dilation and erosion satisfy the following duality w.r.t. negation:

$$\begin{aligned}\mathcal{D}(f, g)(x) &= -\mathcal{E}(-f, \bar{g})(x), \quad \forall x \in \mathcal{D}(D, E) \cap \mathcal{E}(D, \bar{E}) \\ \mathcal{E}(f, g)(x) &= -\mathcal{D}(-f, \bar{g})(x), \quad \forall x \in \mathcal{E}(D, E) \cap \mathcal{D}(D, \bar{E})\end{aligned}$$

where  $f, g \in \mathcal{I}$  with domains  $D, E \subset \mathbf{E}^n$  respectively, and  $\bar{g}(x) = g(-x)$ .

#### Remarks

- (i) Comparing the binary and grey-scale dilation and erosion duality expressions, it might appear that arithmetic negation is the grey-level analogue of the set complement operator for binary images. However, it is clear that in general  $\bigvee \{f(x), -f(x)\} \neq U(x)$  and

$\wedge\{f(x), -f(x)\} \neq O(x)$  for all  $f \in \mathcal{F}$ ; hence (recall Definition 14) arithmetic negation is not a complement operator for the lattice  $(\mathcal{F}, \ll)$ , and furthermore, the lattice is not boolean.

- (ii) When dealing with discrete space  $\mathbf{Z}^2$ , one can encounter problems concerning duality if a the grid underlying the space is square (see section 2.5.9.).

**Proposition 5. Duality between opening and closing.**

Binary opening and closing satisfy the following duality w.r.t. complementation:

$$\mathcal{O}(A, B) = \left[ \mathcal{C}(A^c, \bar{B}) \right]^c$$

$$\mathcal{C}(A, B) = \left[ \mathcal{O}(A^c, \bar{B}) \right]^c$$

where  $A, B \in \mathcal{P}(\mathbf{E}^n)$ .

Grey-scale opening and closing satisfy the following duality w.r.t. negation:

$$\mathcal{O}(f, g)(x) = -\mathcal{C}(-f, \bar{g})(x), \quad \forall x \in \mathcal{O}(D, E) \cap \mathcal{C}(D, \bar{E})$$

$$\mathcal{C}(f, g)(x) = -\mathcal{O}(-f, \bar{g})(x), \quad \forall x \in \mathcal{C}(D, E) \cap \mathcal{O}(D, \bar{E})$$

where  $f, g \in \mathcal{F}$  with domains  $D, E \subset \mathbf{E}^n$  respectively.

## 2.5.4. Anamorphoses

When dealing with a grey-scale image, one often applies radiometric (contrast) enhancement operations. The chosen transformation serves to either correct contrast deficiencies or to reveal details not apparent in the original image. The human vision system actually incorporates radiometric transformation; the relationship between the intensity of light incident upon the photoreceptors in the retina of the eye, and the intensity perceived by the brain is non-linear (perceived intensity turns out to be the logarithm of the incident intensity). As a consequence in the dark regions of an image, a small change in intensity is perceived as a large intensity change, whilst in bright regions it is perceived as a very small intensity change. This sensitivity to detail in the darker regions can be exploited. By darkening the bright areas of an image, previously undetectable detail can be revealed. The more usual contrast enhancements employed in image processing are:  $f(x) \rightarrow af(x) + b$  ( $a, b > 0$ );  $\log(f(x))$ ;  $[f(x)]^2$ ;  $\sqrt{f(x)}$ ;  $f(x)$  for  $f(x) \leq \lambda$ ,  $\lambda$  for  $f(x) \geq \lambda$ , and their various combinations (Serra, 1982, p. 435). These transformations are termed *anamorphoses*.

**Definition 21. An anamorphosis.**

An anamorphosis is any increasing and continuous mapping  $\psi: \mathcal{F} \rightarrow \mathcal{F}$ .

An important feature of FSP filters is that they commute with anamorphoses (Serra, 1988, p. 188; Heijmans, 1991, p. 573). So for example,  $\mathcal{O}(\log(f), B) = \log(\mathcal{O}(f, B))$ . In contrast, convolutions and tophat transformations (see section 6.4.3.) commute with linear anamorphoses only (Dougherty, 1993, p. 508).

### 2.5.5. Representing functions by sets

Two quite different, yet related methods of representing an  $n$ -dimensional ( $n$ -D) function by one or more sets, are pertinent to the extension of binary mathematical morphology to the grey-scale domain. Using *threshold decomposition* an  $n$ -D function can be resolved into a collection of  $n$ -D sets called *cross-sections* or *slices*. Via the *umbra transform* an  $n$ -D function can be represented by a single  $(n+1)$ -D set called the *umbra*. Heijmans (1991) is quick to point out that "in the literature the extension of dilations and erosions by means of the umbra transform has received disproportionately much attention" (p. 568).

#### 2.5.5.1. Threshold decomposition

##### *Definition 22.*

Consider a function  $f \in \mathcal{F}$  with domain  $D \subset \mathbb{E}^n$ . Its cross-section at grey-level  $t \in \bar{\mathbb{E}}$  (the threshold value) is defined

$$X_t(f) = \{x \in D \mid f(x) \geq t\}.$$

##### *Remarks*

- (i) When  $\mathbb{E} = \mathbb{Z}$  and the range of  $f$  consists of  $m$  grey-levels,  $f$  can be decomposed into  $m$  sets (each of which is a subset of the domain  $D$ ).
- (ii) The cross-section at level  $t$  is said to have been obtained by *thresholding*  $f$  at  $t$ .
- (iii) If  $f$  is a u.s.c. function then its cross-sections are topologically closed sets that are monotonically decreasing; i.e.  $s \leq t \Rightarrow X_s \supseteq X_t$  (Serra, 1982, p. 426).
- (iv) In the literature, cross-sections are also referred to as *threshold sets* (Heijmans, 1991) and *slices* (Shih & Mitchell, 1989).
- (v) The function  $f$  can be reconstructed from its cross-sections, point-wise, as follows:

$$f(x) = \bigvee \{t \mid x \in X_t(f)\}.$$

### 2.5.5.2. The umbra transform

*Umbra* is a Latin word meaning shade or shadow. If the grey-level dimension of a 2-D grey-level image  $f \in \mathcal{F}$  is viewed as a third spatial dimension, the image can be reinterpreted as a 3-D binary image (a surface). If a point light source is located at an infinite distance along the positive  $z$ -axis, then the light rays that strike the surface will necessarily be parallel. The umbra of the image is then all the points of the image (surface) as well as all the points in its shadow (which extends infinitely downward).

**Definition 23. The umbra transform.**

Consider a function  $f \in \mathcal{F}$  with domain  $D \subset \mathbf{E}^n$ . The umbra transform of  $f$ ,  $U(f) \subset D \times \mathbf{E}$ , is defined

$$U(f) = \{(x, t) \mid t \leq f(x)\}.$$

The function  $f$  can be reconstructed from its umbra  $A \subset \mathbf{E}^{n+1}$ , by taking the *top-surface* of the set,  $T(A)$ , point-wise as follows

$$f(x) = T(A)(x) = \bigvee \{t \mid (x, t) \in A\}.$$

*Remark*

"To any real-valued [resp. integer-valued] u.s.c. function  $f(x)$ ,  $x \in \mathbf{R}^m$  [resp.  $\mathbf{Z}^m$ ], there corresponds a unique umbra  $U(f)$ . This umbra  $U$  is a closed set in  $\mathbf{R}^{m+1}$  [resp.  $\mathbf{Z}^{m+1}$ ]" (Maragos & Schafer, 1987a, p. 1155).

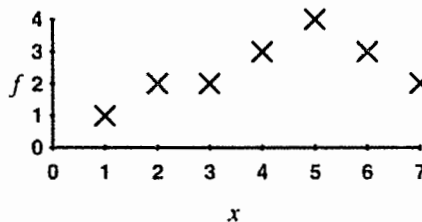
### 2.5.5.3. Relating the umbra to the cross-sections of a function

The following relationship exists between the umbra of a u.s.c. function and its cross-sections:

$$(x, t) \in U(f) \Leftrightarrow t \leq f(x) \Leftrightarrow x \in X_t(f).$$

*Example*

Consider the following discrete function of one variable:



Its umbra can be represented by the following binary bound matrix (see Appendix A):

$$U(f) = \begin{bmatrix} * & * & * & * & 1 & * & * \\ * & * & * & 1 & 1 & 1 & * \\ * & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{1,4}.$$

The umbra matrix is said to be minimal because each column contains at least one 1. Strictly speaking, the umbra should extend infinitely downward. The cross-sections of  $f$  are:

$$X_1(f) = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]_{1,1}$$

$$X_2(f) = [\cdot \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]_{1,2}$$

$$X_3(f) = [\cdot \ \cdot \ \cdot \ 1 \ 1 \ 1 \ \cdot]_{1,3}$$

$$X_4(f) = [\cdot \ \cdot \ \cdot \ \cdot \ 1 \ \cdot \ \cdot]_{1,4}$$

Hence,

$$U(f) = X_4(f) \cup X_3(f) \cup X_2(f) \cup X_1(f).$$

### 2.5.6. The umbra homomorphism theorem

This theorem establishes the relationship between binary morphology and grey-scale morphology via the umbra transform. The umbra transform is a homomorphism from grey-scale to binary morphology (Haralick, Sternberg, & Zhuang, 1987, p. 546).

#### *Theorem 4. Umbra homomorphism theorem.*

Let  $f, g \in \mathcal{F}$  be u.s.c. functions so that their umbrae are closed sets in  $\mathbf{E}^{n+1}$ . Then

$$U(\mathcal{B}(f, g)) = \mathcal{B}(U(f), U(g)) \text{ and}$$

$$U(\mathcal{E}(f, g)) = \mathcal{E}(U(f), U(g)).$$

#### *Remarks*

- (i) It follows that  $\mathcal{B}(f, g) = T(\mathcal{B}(U(f), U(g)))$  and  $\mathcal{E}(f, g) = T(\mathcal{E}(U(f), U(g)))$ . Hence grey-scale dilation (resp. erosion) can be expressed in terms of binary dilation (resp. erosion).
- (ii) "A pair  $\langle S, \circ \rangle$  where  $S$  is a set and  $\circ$  is an associative binary operation on  $S$  is called a **semigroup**" (Allenby, 1983, p. 194). Hungerford (1974, p. 30) states that if  $G$  and  $H$  are semigroups then a function  $f: G \rightarrow H$  is a homomorphism provided that  $f(ab) = f(a)f(b)$  for all  $a, b \in G$  (the products  $ab$  and  $f(a)f(b)$  denote the binary operators of  $G$  and  $H$  respectively). Now, for example, consider the semigroups  $\langle \mathcal{F}, \oplus \rangle$  and  $\langle \rho(\mathbf{E}^{n+1}), \oplus \rangle$  (note that  $\oplus$  is used to denote both grey-scale dilation and binary dilation). Hence  $U: \langle \mathcal{F}, \oplus \rangle \rightarrow \langle \rho(\mathbf{E}^{n+1}), \oplus \rangle$  is a homomorphism because  $U(f \oplus g) = U(f) \oplus U(g)$ .

### 2.5.7. Implementing dilation and erosion for digital grey-level images

Digital binary images are sets in  $\mathbf{Z}^2$  and digital grey-level images are discrete functions  $f: \mathbf{Z}^2 \rightarrow \bar{\mathbf{Z}}$ . The implementation of binary dilation and erosion in software is straight forward. To date, three methods for implementing their grey-scale analogues have been devised. Grey-level dilation for instance can be implemented in any of the following ways:

- (i) as a maximum of a set of sums, (Definition 19);
- (ii) as binary dilation of umbrae; i.e.  $\mathcal{D}(f, g) = \mathcal{T}(\mathcal{D}(U(f), U(g)))$ ;
- (iii) as binary dilation of cross-sections; i.e.

$$\mathcal{D}(f, g) = \max \left\{ \left[ \sum_{i=0}^N \{ \mathcal{D}(X_0(f), X_i(g)) \} \right] - \vec{1}, \right. \\ \left[ \sum_{i=0}^N \{ \mathcal{D}(X_1(f), X_i(g)) \} \right], \\ \left[ \sum_{i=0}^N \{ \mathcal{D}(X_2(f), X_i(g)) \} \right] + \vec{1}, \\ \dots, \\ \left[ \sum_{i=0}^N \{ \mathcal{D}(X_M(f), X_i(g)) \} \right] + (M-1) \vec{1} \left. \right\},$$

where  $f$  has maximum grey-level  $M$  and  $g$  has maximum grey-level  $N$  and  $\vec{1}$  denotes the sequence  $i, i, \dots, i$  of arbitrary length (Shih & Mitchell, 1989, p.35).

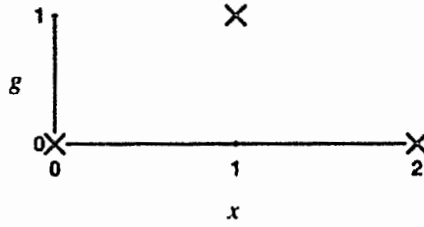
#### Remarks

- (i) Algorithms for grey-scale dilation and erosion based on the point-wise formulae can be found in Mehnert (1990).
- (ii) Though the umbra transform has been used to implement grey-scale dilation and erosion on a computer (e.g. Sternberg's cytocomputer), the fact that the umbra of a two-dimensional image is a three-dimensional binary image means that such implementations are not efficient both in terms of storage and execution time.
- (iii) Though the method of cross-sections retains dimensionality, one must perform as many binary dilations or erosions as there are grey-levels. Shih and Mitchell (1989) have proposed a VLSI architecture for implementing grey-scale dilation and erosion using threshold decomposition.

It is worthwhile to look at an example to compare each of the methods.

#### Example

Consider the one-dimensional image (actually a signal)  $f$  defined in section 2.5.5.3. Also consider the structuring element  $g$  defined as follows, along with its umbra and cross-sections:



$$U(g) = \begin{bmatrix} * & 1 & * \\ 1 & 1 & 1 \end{bmatrix}_{0,1}$$

$$X_0 = [1 \quad 1 \quad 1]_{0,0}$$

$$X_1 = [* \quad 1 \quad *]_{0,1}$$

Now  $f$  and  $g$  can be represented by the bound vectors  $f = [1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 3 \quad 2]_1$  and  $g = [0 \quad 1 \quad 0]_0$  respectively.

(i) Using the point-wise formula,

$$f_0 + g(0) = [1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 3 \quad 2 \quad * \quad *]_1$$

$$f_1 + g(1) = [* \quad 2 \quad 3 \quad 3 \quad 4 \quad 5 \quad 4 \quad 3 \quad *]_1$$

$$f_2 + g(2) = [* \quad * \quad 1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 3 \quad 2]_1,$$

and the column maxima give

$$\mathcal{B}(f, g) = [1 \quad 2 \quad 3 \quad 3 \quad 4 \quad 5 \quad 4 \quad 3 \quad 2]_1.$$

(ii) Binary dilation of the umbrae yields the following minimal umbra matrix:

$$\mathcal{B}(U(f), U(g)) = \begin{bmatrix} * & * & * & * & * & 1 & * & * & * \\ * & * & * & * & 1 & 1 & 1 & * & * \\ * & * & 1 & 1 & 1 & 1 & 1 & 1 & * \\ * & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{1,5}.$$

The top surface gives

$$\mathcal{B}(f, g) = T(\mathcal{B}(U(f), U(g))) = [1 \quad 2 \quad 3 \quad 3 \quad 4 \quad 5 \quad 4 \quad 3 \quad 2]_1.$$

(iii) Finally, binary dilation of cross-sections, using the Shih and Mitchell formula, gives



$$\begin{aligned}
\mathcal{B}(X_1(f), X_0(g)) &= [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\
\mathcal{B}(X_1(f), X_1(g)) &= [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \\
\mathcal{B}(X_2(f), X_0(g)) &= [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\
\mathcal{B}(X_2(f), X_1(g)) &= [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \\
\mathcal{B}(X_3(f), X_0(g)) &= [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \\
\mathcal{B}(X_3(f), X_1(g)) &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0] \\
\mathcal{B}(X_4(f), X_0(g)) &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0] \\
\mathcal{B}(X_4(f), X_1(g)) &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]
\end{aligned}$$

$$\begin{aligned}
\mathcal{B}(X_1(f), X_0(g)) + \mathcal{B}(X_1(f), X_1(g)) + \vec{0} &= [1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1] \\
\mathcal{B}(X_2(f), X_0(g)) + \mathcal{B}(X_2(f), X_1(g)) + \vec{1} &= [0 \ 2 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 2] \\
\mathcal{B}(X_3(f), X_0(g)) + \mathcal{B}(X_3(f), X_1(g)) + \vec{2} &= [0 \ 0 \ 0 \ 3 \ 4 \ 4 \ 4 \ 3 \ 0] \\
\mathcal{B}(X_4(f), X_0(g)) + \mathcal{B}(X_4(f), X_1(g)) + \vec{3} &= [0 \ 0 \ 0 \ 0 \ 4 \ 5 \ 4 \ 0 \ 0]
\end{aligned}$$

The column maxima give

$$\mathcal{B}(f, g) = [1 \ 2 \ 3 \ 3 \ 4 \ 5 \ 4 \ 3 \ 2]_1.$$

### 2.5.8. The genesis of mathematical morphology is more than just dilations and erosions

Historically (Euclidean) mathematical morphology referred to all those mappings and methods built up from the dilation and the erosion. Their various combinations give rise to "an infinite world of *new* mappings, which can then be concatenated with each other: openings, closings, size distributions, morphological filters, ultimate erosion, skeletons, conditional bisector, and many others" (Serra, 1988, p. 5). In binary mathematical morphology Euclidean distance is inextricably linked with these mappings; the distance transform can be used to obtain dilations, erosions, skeletons and so on (see Mehnert, 1990, p. 81). The distance transform brings to the fore topological concepts such as convexity and connectivity. From a practical standpoint, the main advantage of its use is that "the complexity of the corresponding algorithms is reduced to a constant independent of the size of the structuring element" (Preteux & Merlet, 1991, p. 66). Unfortunately the interpretation of the binary operators in terms of the distance transform cannot be directly transposed to the operators of grey-scale morphology. However, Preteux and Merlet (1991) introduced the notions of topographical and differential distance functions to establish "the equivalence between the two fundamental notions of skeleton by influence zones [SKIZ] and watershed [WS]" (p. 66); these are "two basic and fundamental operators in segmentation, respectively in binary and grey-level images" (Dougherty, 1993, p. 324) (see section 4.1.) It is important

to realise that mathematical morphology embodies not only those operators and methods constructed from dilations and erosions, but also a host of other methods including distance transforms, watershed transformation, tophat transforms, and homotopy modification.

### 2.5.9. Topology for digital spaces

Imagine a two-dimensional binary image manifested in the Euclidean plane and modelled by a set of points that locate its foreground in  $\mathbf{R}^2$ . To obtain a digital image (suitable for processing by computer) it must be sampled at a discrete number of points using typically a square or hexagonal sampling grid (though other tessellations can be used). The sampled (*digitised*) image now resides in the space  $\mathbf{Z}^2$ . It is an "unfortunate fact that the two spaces  $\mathbf{R}^2$  and  $\mathbf{Z}^2$  are not isomorphic, since there are several different ways to interpret the same module [i.e.  $\mathbf{Z}^2$ ] in Euclidean space" (Serra, 1982, p. 207). Consider, for instance, rotations in  $\mathbf{Z}^2$  (Figure 3). In contrast to  $\mathbf{R}^2$ , the square grid admits only four rotations; successive rotations through  $90^\circ$  (rotations 1, 3, 5, and 7). The hexagonal grid permits six rotations; successive rotations through  $60^\circ$ . A more liberal interpretation of rotations on the square grid leads to the eight rotations shown in Figure 3; however rotational symmetry is compromised. In essence there exist two distinct sets of similar images; rotations 1, 3, 5, and 7, and rotations 2, 4, 6, and 8. The number of permissible rotations is important when one tries to interpret say Crofton's formula for perimeter estimation (see Chapter 4).

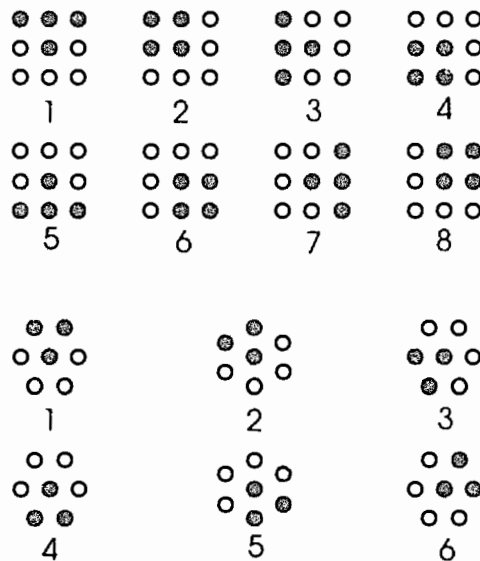


Figure 3. Rotations on the square and hexagonal grids.

Now imagine a two-dimensional continuous grey-tone image that is modelled by a function  $f: \mathbf{R}^2 \rightarrow \mathbf{R}$  (which describes the grey-level surface of the image). Once again, a sampling grid is used to obtain a digital image. The brightness values sampled at the grid points are constrained to a finite number of

levels. Hence the digitised grey-scale image is the discrete-valued function  $\hat{f}: \mathbb{Z}^2 \rightarrow \mathbb{Z}$ . As in the binary case, the grid underlying  $\mathbb{Z}^2$  affects Euclidean interpretations; e.g. the brightness gradient of a grey-level image at a particular point in  $\mathbb{Z}^2$ .

Serra and his colleagues deal predominantly with the hexagonal grid. Clearly "the main advantage of the hexagonal grid over the square grid is that it has more rotational symmetry" (Heijmans, 1989, p. 20). In contrast, the chapters that follow deal exclusively with the square grid. Apart from being the most widely used tessellation in image processing, images sampled on this grid are amenable to representation by bound matrices (see Appendix A). Furthermore, the square grid mirrors the pixel arrangement associated with computer raster graphics displays.

Topological notions such as connectivity and convexity also have several interpretations for  $\mathbb{Z}^2$ , depending on the underlying grid. For the square grid, every pixel has the  $3 \times 3$  neighbourhood illustrated in Figure 4. Pixels  $p_1, p_3, p_5$ , and  $p_7$  are the *direct* neighbours of  $p$  and are considered to be connected to it. The question arises – are  $p_2, p_4, p_6$ , and  $p_8$ , the *indirect* neighbours of  $p$ , connected to  $p$ ? Consider Figure 5: is this one object or several disparate particles? Even if one concludes that the image is a square, how does one interpret the background? Assigning the same connectivity to the background induces the paradox that the hole within the square is connected to the rest of the background.

$p_6$	$p_7$	$p_8$
$p_5$	$p$	$p_1$
$p_4$	$p_3$	$p_2$

Figure 4. The  $3 \times 3$  neighbourhood of  $p$  for the square grid.

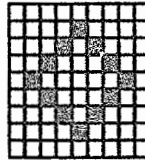


Figure 5. Is this a square or several disparate particles?

Conventionally, connectivity for the square grid is defined as follows. The direct neighbours of  $p$  are termed its *4-neighbours*. Collectively, the direct and indirect neighbours of  $p$  are called its *8-neighbours*. A pair of foreground pixels  $p, q \in \mathbb{Z}^2$  are 4-connected (resp. 8-connected) if there exists a path  $p_1, p_2, \dots, p_n$ , where  $p = p_1$  and  $q = p_n$ , such that  $p_i$  and  $p_{i+1}$  are 4-neighbours (resp. 8-neighbours), and the  $p_i$  are also foreground pixels, for  $i = 1, 2, \dots, n-1$ . A binary image  $X \in \mathcal{P}(\mathbb{Z}^2)$  is

said to be a 4-connected (resp. 8-connected) region if every pair of pixels in  $X$  are 4-connected (resp. 8-connected). Clearly if Figure 5 is considered to be an 8-connected square, then to avoid any ambiguity, the background must have 4-connectivity. As a result of having to adopt opposite types of connectivity for the foreground and background, " $X$  and  $X^c$  play asymmetric roles" (Serra, 1982, p. 180). Figure 5 represents either a single object or several particles depending on whether it is interpreted as  $X$  or  $X^c$  respectively. It is worth noting that for the hexagonal grid,  $X$  and  $X^c$  play symmetric roles because there is no inherent ambiguity of connectivity between pixels.

In binary Euclidean morphology, set (image) convexity is a property that is invariant under the basic morphological transformations; dilation, erosion, opening, and closing. Furthermore, "convex sets play a significant role in Euclidean morphology, from both a granulometric and an image-functional point of view" (Giardina & Dougherty, 1988, p. 27). Giardina and Dougherty define convexity for the square grid as follows. A binary image is said to be *vertically convex* if for any two foreground pixels in a given column, all of the pixels vertically between them are also foreground pixels. The image is said to be *horizontally convex* if for any two foreground pixels in a given row, all of the pixels horizontally between them are also foreground pixels. This leads to the following pair of definitions.

**Definition 24.**

A binary digital image  $A \in \mathcal{P}(\mathbb{Z}^2)$  is said to be *strongly grid convex* if the following conditions are satisfied:

- (1)  $A$  is 4-connected,
- (2)  $A$  is vertically convex, and
- (3)  $A$  is horizontally convex.

**Definition 25.**

A binary digital image  $A \in \mathcal{P}(\mathbb{Z}^2)$  is said to be *weakly grid convex* if the following conditions are satisfied:

- (1)  $A$  is 8-connected but not 4-connected,
- (2)  $A$  is vertically convex and
- (3)  $A$  is horizontally convex.

Although digital convexity is not, strictly speaking, the same thing as Euclidean convexity, "there is one sense in which they are the same. If  $P$  and  $Q$  are any two activated [foreground] pixels of a grid convex image that lie in the same column or row, then all pixels between them are activated" (Giardina & Dougherty, 1988, p.115).

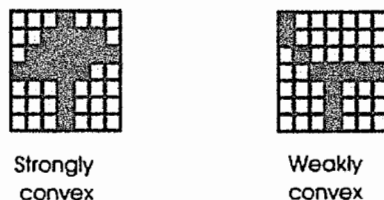


Figure 6. Digital convexity.

## 2.6. Linear Versus Morphological Filtering

The perceived geometric detail in an image is inherent in the spatial interdependence of neighbouring pixels. A rapid change in brightness in an image corresponds to high spatial frequency and is synonymous with noise and object edges within the image. Gradual intensity changes within an image correspond to low spatial frequency. The low-pass filter attenuates high spatial frequency, leaving low frequencies intact. The high-pass filter has the opposite effect. Band-pass filters remove all but a selected band of spatial frequencies from an image. Every linear "filter turns out to be the convolution product  $f * \phi$  of a signal [image]  $f$  by a (generalized) [sic] function  $\phi$ " (Serra, 1986, p. 288). In practice, linear filters are often considered to be "band-pass devices, even if this is not exactly true" (Serra, 1988, p. 102). "A high frequency signal cut off above 400 kHz by a first filter should theoretically not be modified by a second filter identical to the first" (Serra, 1986, p. 288). Linear filters that are truly idempotent are termed *ideal*. Early in the history of image filtering it was supposed "that the global structure of an image would be derived from a low-pass filtering and the finer details from a high-pass filtering" (Heijmans & Ronse, 1990, p. 247). The predisposition to this way of thinking can be attributed to the theory of filtering already established for sound signals. Sound signals combine in a linear fashion. A sound reproduction system, such as a stereo or radio, must reproduce individual sounds in proportion to their original intensities. When listening to a broadcast or reproduction of an orchestral performance, the human ear sums the sound intensities, or their logarithms (Serra, 1986, p. 288), emanating from the individual instruments. Many distortions in images are attributed to physical phenomena that are linear in nature. Camera movement manifests itself in a photograph as essentially the sum of several pictures (Serra 1988, p. 102). NASA's Ranger 7 lunar probe was equipped with the most technologically advanced video cameras available at the time. The Vidicon cameras used, however, induced interference patterns in the transmitted images because of the oscillation inherent in the electronics. This additive noise was successfully removed using linear methods. Lack of focus in an image can also be improved using linear filters. Clearly such situations justify employing linear filters.

Serra (1988) points out that "although acoustic signals are summed, visual signals are not compounded in this manner" (p. 102). He argues the case for morphological filtering along the following lines. The world around us is composed of opaque objects that hide one another. When one object is placed in front of another, the light rays reflected from both objects do not sum. On the

contrary, the covering object blocks the light emanating from the object behind it. This notion of inclusion justifies the requirement that a morphological filter be an increasing mapping (Definition 8). The second requirement of a morphological filter is that it be idempotent (Definition 11). This can be justified as follows. Linearity can imply that there is no loss of information. For instance, imagine a two-dimensional grey-level image, described by the real-valued function  $g: \mathbf{R}^2 \rightarrow \mathbf{R}$ , that exhibits blurriness. The blur might be modelled as a convolution with some function  $h$  as follows (in the absence of noise):

$$g(x, y) = \iint h(x - x', y - y') f(x', y') dx' dy',$$

where  $g$  is the blurred image,  $h$  is the blur, and  $f$  is the original image. Taking the Fourier transform of both sides and applying the convolution theorem (Rosenfeld & Kak, 1982, p. 269) yields

$$\begin{aligned} G(u, v) &= H(u, v) F(u, v) \\ \Rightarrow F(u, v) &= G(u, v) / H(u, v), \end{aligned}$$

where  $G(u, v)$ ,  $H(u, v)$ , and  $F(u, v)$  are the Fourier transforms of  $g(x, y)$ ,  $h(x, y)$ , and  $f(x, y)$  respectively. Provided that the nature of the blur is known, and hence  $H(u, v)$  is known (e.g. motion blur),  $f(x, y)$  can be reconstructed via the inverse Fourier transform. Clearly, the underlying assumption is that the convolution that produced the blur did not result in a loss of information. In comparison, "an increasing transformation generally produces a loss of information" (Serra, 1988, p. 104). It is for this very reason that a morphological filter needs to be idempotent; to control this loss of information.

Heijmans and Ronse (1990) acknowledge Serra's arguments but believe that the reason why non-linear methods in image analysis are generally better suited than linear methods is fundamentally because "the two human senses of vision and audition have not the same purpose" (p. 247). It is a question of indirection. Audition is the analysis of sound waves. Vision is not the analysis of light waves; rather it is the recognition of objects within a three-dimensional space from the light waves they reflect.

A number of authors choose to define morphological filters in a broader sense than Serra. Most notably, Giardina and Dougherty (1988) state that "mappings that are both increasing and translation invariant are called *morphological filters*" (p. 134). This definition then admits dilations and erosions as morphological filters. Furthermore a host of other operations are admitted, including "order statistics and [spatial] convolutions with nonnegative [sic] mask weights summing to unity" (Dougherty, 1990, p. 179). The case for this characterisation of a morphological filter can be argued on theoretical grounds in respect of the Matheron representation theorem for translation-invariant and increasing mappings (see Chapter 3). The theorem essentially states that any increasing and translation-invariant mapping can be expressed as the supremum of erosions or equivalently as the infimum of dilations. In practice,

however, one seldom applies a dilation or an erosion on its own; to the contrary, they are usually applied in pairs producing openings and closings. For this reason, and more importantly because of the desirability of idempotence, the author prefers Serra's definition of a morphological filter. From the practitioner's standpoint, idempotence ensures that the filtering process can be carefully controlled. After the application of a morphological filter, one has necessarily to decide upon some other course of action. This is in stark contrast to the application of non-idempotent filters such as the mean and median filters, whereupon one has to decide whether or not to repeat the application. Figure 7 illustrates a simple hierarchical classification of traditional image filters and morphological operations, according to linearity and idempotence.

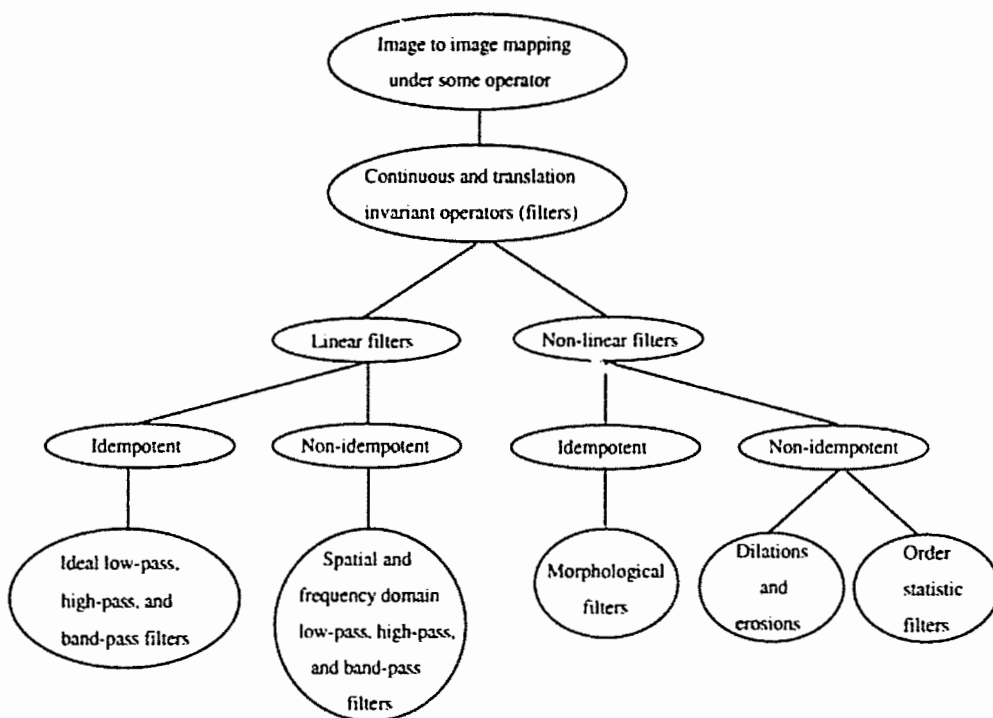


Figure 7. A classification of traditional filters and morphological operators based on linearity and idempotence.

---

---

# **CHAPTER 3.**

## **A TAXONOMY OF MORPHOLOGICAL FILTERS**

---

---

### **3.1. About This Chapter**

This chapter begins with a brief history of morphological filters – section 3.2. Section 3.3 then introduces algebraic openings and closings. These operations, which generalise morphological openings and closings respectively, constitute the building blocks for all morphological filters. The section also examines the various methods used to construct (algebraic) openings and closings. In this regard products of dilations and erosions represent but one way of constructing them. The remaining sections establish the taxonomy of morphological filters that begins with the (algebraic) openings and closings, and encompasses open-closings, close-openings, general composition of filters, self-dual filters, alternating sequential filters, multiple structuring element filters, the generalised morphological filter, soft morphological filters, and Dolby morphological filters.

### **3.2. A Brief History of Morphological Filters**

Matheron originally devised the opening for the purpose of generating size distributions (granulometries) for binary images. The opening, however, found much wider use as a noise suppression filter. Research done in the late 1970s, predominantly by Serra and Sternberg, the latter conceiving of the idea of combining openings and algebraic differences, led to the extension of mathematical morphology to grey-tone functions (Serra, 1988, p. 101). "Around the same time, morphologists realized [sic] that combinatory effects of an opening followed by a closing led to an idempotent operation" (p. 101). Note that in general the composition of two morphological filters is not a morphological filter. Sternberg (1986) introduced the iterative morphological filter – sequences of alternating openings and closings, which employ successively larger spherical structuring elements (SEs), used for noise suppression. Stevenson and Arce (1986) introduced the 2D CO filter – combinations of several openings and closings employing multiple SEs. Matheron and Serra (in Serra, 1988) defined and studied morphological filtering for complete lattices; Sternberg's iterative filter turns out to be just one of a class of filters called alternating sequential filters. Song and Delp (1989) introduced the generalised morphological filter; the 2D CO filter is just a particular case of this more general filter. More recently Song and Delp (1990) have introduced noise suppression filters consisting

---



of suprema of openings and infima of closings employing multiple SEs. Ronse and Heijmans (1991) made an algebraic study of openings and closings on the complete lattice; they generalised the Euclidean notion of operators invariant under translations by considering operators invariant under certain abelian (i.e. commutative) groups of automorphisms on the lattice. Koskinen, Astola, and Neuvo (1991) generalised the definitions for digital dilation and erosion by replacing the maximum and minimum operations with general weighted order statistics. They refer to these new operators as soft dilation and erosion respectively. This leads to soft openings and closings and more generally to soft morphological filters. Unfortunately soft morphological filters are not necessarily idempotent. Dougherty (1992a, 1992b) placed morphological operators into the framework of statistical estimation to develop a theory of mean-square optimisation. This leads to the possibility of determining from amongst all possible filters the one that optimally restores a noise corrupted image. A discussion of this approach and related studies is provided in section 7.2. Serra (in Dougherty, 1993) introduced the Dolby morphological opening and derived morphological filters. Though these filters are not based on the actual Dolby algorithm for audio signals, they have a similar purpose.

### 3.3. Algebraic Openings and Closings

To begin with let the complete lattice  $(\mathcal{L}, \prec)$  be a model for the object space; this might be the set of all grey-tone digital images for instance. If  $\mathcal{O}$  denotes the set of all increasing mappings from  $\mathcal{L}$  into itself then  $(\mathcal{O}, \prec)$  is also a complete lattice (Serra, 1988, p. 104) with order relation

$$\psi' \prec \psi \text{ if } \psi'(X) \prec \psi(X) \quad \forall X \in \mathcal{L},$$

and supremum and infimum defined as

$$\begin{aligned} (\bigvee_i \psi_i)(X) &= \bigvee_i [\psi_i(X)] & \text{and} & & (\bigwedge_i \psi_i)(X) &= \bigwedge_i [\psi_i(X)] & \text{respectively, where } \psi_i \in \mathcal{O}. \\ \sup \text{ in } (\mathcal{O}, \prec) & & \sup \text{ in } (\mathcal{L}, \prec) & & \inf \text{ in } (\mathcal{O}, \prec) & & \inf \text{ in } (\mathcal{L}, \prec) \end{aligned}$$

For simplicity the symbols denoting supremum, infimum, and the order relation for the lattice of operators  $(\mathcal{O}, \prec)$  remain the same as those for the lattice of operands  $(\mathcal{L}, \prec)$ .

Recall that for the complete lattice  $(\mathcal{L}, \prec)$  the morphological filter is any increasing and idempotent mapping  $\psi: \mathcal{L} \rightarrow \mathcal{L}$  (Definition 16). Now if  $\psi, \zeta \in \mathcal{O}$  are morphological filters it is an unfortunate fact that in general  $\psi \vee \zeta$ ,  $\psi \wedge \zeta$ ,  $\psi \circ \zeta$ , and  $\zeta \circ \psi$  are not. This motivates the introduction of two classes of mappings: *underfilters* and *overfilters* (Serra, 1988, p. 105).

**Definition 26. Underfilters and overfilters.**

A mapping  $\psi \in \mathcal{O}$  such that  $\psi\psi \prec \psi$  is called an *underfilter*. A mapping  $\psi \in \mathcal{O}$  such that  $\psi \prec \psi\psi$  is called an *overfilter*.

**Remarks**

- (i) The inequality  $\psi\psi \prec \psi$  is equivalent to writing  $\psi = (I \vee \psi) \circ \psi$ . Likewise, the inequality  $\psi \prec \psi\psi$  is equivalent to writing  $\psi = (I \wedge \psi) \circ \psi$ .
- (ii) In general neither dilations nor erosions are underfilters or overfilters. However, (conditionally) extensive dilations and (conditionally) anti-extensive erosions are overfilters and underfilters respectively. In Euclidean morphology, such dilations and erosions correspond to dilation and erosion by origin-containing SEs.

**Example**

The identity operator  $I \in \mathcal{O}$  is defined  $I: X \rightarrow X$  for all  $X \in \mathcal{L}$ . Given any  $\psi \in \mathcal{O}$  then clearly,  $I \vee \psi$  is an overfilter and  $I \wedge \psi$  is an underfilter. For instance, consider  $I \vee \psi$ . If  $\psi$  is anti-extensive, i.e.  $\psi \prec I$ , then  $I \vee \psi = I$  and hence  $(I \vee \psi) \prec (I \vee \psi) \circ (I \vee \psi)$ . If on the other hand,  $\psi$  is extensive, i.e.  $I \prec \psi$ , then  $I \vee \psi = \psi$  and hence once again  $(I \vee \psi) \prec (I \vee \psi) \circ (I \vee \psi) = \psi\psi$ . Therefore  $I \vee \psi$  must be an overfilter; this is true regardless of whether  $\psi$  is extensive, anti-extensive, or neither.

As the example above shows under- and overfilters are very common. Now if  $\psi \in \mathcal{O}$  is both an underfilter and overfilter then  $\psi\psi \prec \psi$  and  $\psi \prec \psi\psi$  which implies that  $\psi\psi = \psi$ ; i.e.  $\psi$  is idempotent. Therefore a morphological filter is necessarily both an underfilter and overfilter.

Like their morphological counterparts algebraic openings and closings are morphological filters because they are increasing and idempotent. They are defined as follows.

**Definition 27**

Let  $\psi \in \mathcal{O}$  be a morphological filter. If  $\psi \prec I$  then it is called an *opening* (anti-extensive). If  $I \prec \psi$  then it is called a *closing* (extensive).

**Remarks**

- (i) Haralick (1989) refers to these operators as *generalised openings* and *closings*.
- (ii) The morphological openings and closings of Chapter 2 clearly satisfy Definition 27.

Hereinafter the terms *opening* and *closing* refer to generalised openings and closings respectively. The openings  $\delta \epsilon$  and closings  $\epsilon \delta$  for adjunctions  $(\epsilon, \delta)$  in the lattice  $(\mathcal{L}, \prec)$  are qualified as *morphological openings* and *closings*. Heijmans (1989), and Heijmans and Ronse (1990) qualify the translation-

invariant morphological openings and closings, which are constructed from Minkowski addition and subtraction (and their grey-level extension) employing a SE, as *structural* openings and closings. Some important results concerning openings and closings follow.

**Proposition 6.** (Serra, 1988, p. 105)

Associated with every opening  $\gamma: \mathcal{L} \rightarrow \mathcal{L}$  (resp. closing  $\varphi: \mathcal{L} \rightarrow \mathcal{L}$ ) there exists a *domain of invariance*  $\mathcal{B} \subset \mathcal{L}$  such that  $\gamma(B) = B$  (resp.  $\varphi(B) = B$ ) for all  $B \in \mathcal{B}$ . Furthermore  $\mathcal{B}$  is the image of  $\mathcal{L}$  under  $\gamma$  (resp.  $\varphi$ ).

**Theorem 5.** (Serra, 1988, p. 106)

The supremum (resp. infimum) of a family of openings  $\gamma_i$  (resp. closings  $\varphi_i$ ), with corresponding domains of invariance  $\mathcal{B}_i$ , is again an opening (resp. closing) whose domain of invariance is given by the union (resp. intersection) of the  $\mathcal{B}_i$ .

**Theorem 6.** (Serra, 1988, p. 22).

Any opening  $\gamma \in \mathcal{O}$  (resp. closing  $\varphi \in \mathcal{C}$ ) can be decomposed into a supremum of morphological openings (resp. infimum of morphological closings).

### 3.3.1. The Matheron representation theorem for translation-invariant openings and closings

Theorem 6 is a generalisation of the following result, due to Matheron (1975), for the complete lattice  $\mathcal{P}(\mathbf{R}^n)$  (cited in Serra, 1988, p. 106).

**Theorem 7. Matheron representation theorem for translation-invariant openings.**

The mapping  $\psi: \mathcal{P}(\mathbf{E}^n) \rightarrow \mathcal{P}(\mathbf{E}^n)$  is a translation-invariant opening if and only if there exists a class of sets  $\mathcal{B} \subset \mathcal{P}(\mathbf{E}^n)$  such that

$$\psi(A) = \bigcup_{B \in \mathcal{B}} \mathcal{O}(A, B).$$

The domain of invariance of  $\psi$  is then all those sets that are generated as the unions and translations of the Bs.

*Remark*

Clearly, by duality any translation-invariant closing can be represented as the intersection of morphological closings.

Giardina and Dougherty (1988) extended the theorem to grey-tone images.

**Theorem 8. Grey-scale Matheron representation theorem for translation-invariant openings.**

The mapping  $\psi \in \mathcal{F}$  is a translation-invariant opening if and only if there exists a class of mappings  $\mathcal{B} \subset \mathcal{F}$  such that

$$\psi(f) = \bigvee_{g \in \mathcal{B}} \mathcal{O}(f, g).$$

*Remark*

Clearly, by duality any translation-invariant closing can be represented as the infimum of morphological closings.

### 3.3.2. How to construct openings and closings

"Pedagogically speaking, openings and closings represent the basic material that is constantly used to generate all other [morphological] filters" (Serra, 1988, p. 105). Therefore it is necessary to know how to construct them. In this regard morphological openings (resp. closings) and suprema (resp. infima) thereof are but one way of constructing openings (resp. closings). Figure 8 shows an inventory of several techniques detailed in the literature.

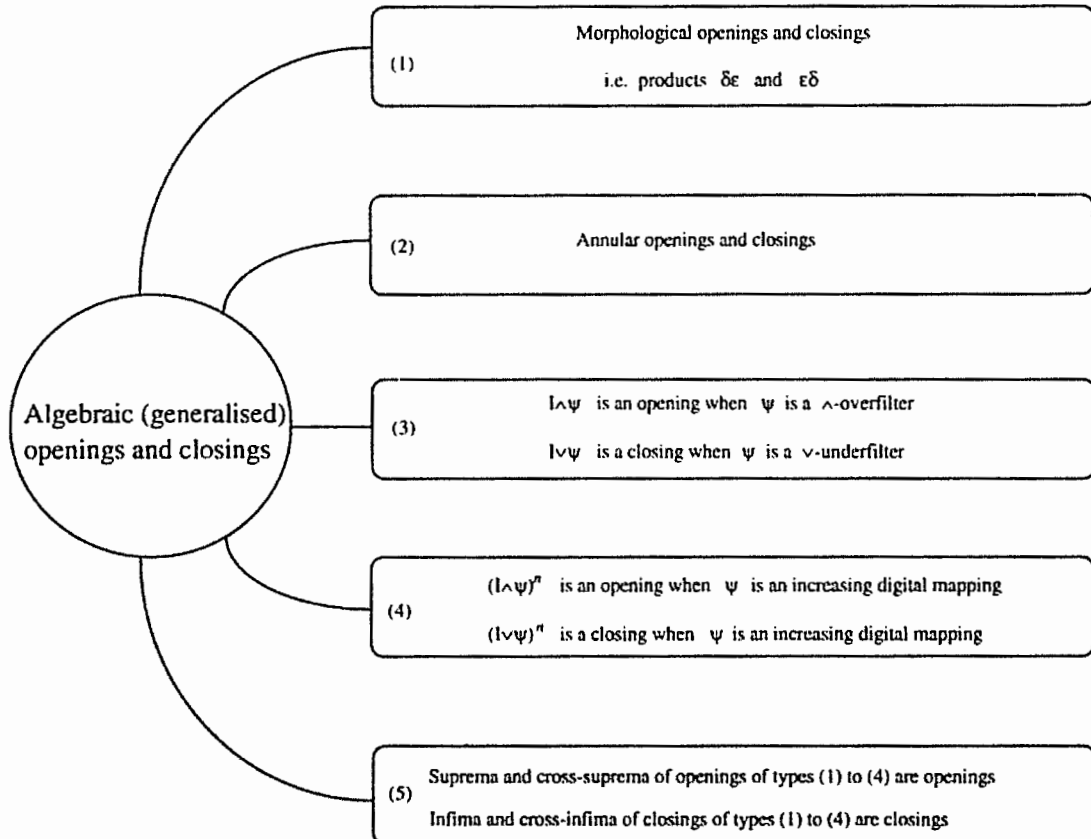


Figure 8. An inventory of methods for generating openings and closings.

### 3.3.2.1. Annular openings and closings

Serra (1988, p. 107) introduced another type of Euclidean binary opening called the *annular opening*. It is defined as follows.

**Definition 28. The annular opening**

Given  $A, B \in \mathcal{P}(\mathbb{E}^n)$  such that  $B$  is a symmetric SE, i.e.  $B = \tilde{B}$ , the mapping

$$\gamma: A \rightarrow (A \oplus B) \cap A$$

is an opening with domain of invariance  $\mathcal{B} = \{A \mid A \subseteq A \oplus B\}$ .

To prove that  $\gamma: A \rightarrow (A \oplus B) \cap A$  is indeed an opening it is necessary to show that it is increasing, anti-extensive and idempotent (Definition 27). Serra's proof (1988, p. 107), however, is not correct. In particular his arguments supporting idempotence are in error. Consider the following proof.

*Proof*

- (i)  $\gamma$  is increasing because it is the intersection of  $\oplus$  and  $\cap$  which are both increasing operators.
- (ii)  $\gamma$  is anti-extensive by design

$$\begin{aligned} o \in B &\Rightarrow \gamma(A) = A \\ o \notin B &\Rightarrow \gamma(A) \subset A \end{aligned} \Bigg\} \gamma(A) \subseteq A$$

where  $o$  denotes the origin.

- (iii) For idempotence consider  $\gamma \circ \gamma$

$$\gamma\gamma(A) = \underbrace{[(A \oplus B) \cap A] \oplus B}_{(1)} \cap \underbrace{[(A \oplus B) \cap A]}_{(2)}$$

(1) has representation

$$\left( \bigcup_{b \in B} A_b \cap A \right) \oplus B \Leftrightarrow \bigcup_{b' \in B} \bigcup_{b \in B} (A_b \cap A)_{b'} \Leftrightarrow \bigcup_{b' \in B} \bigcup_{b \in B} (A_{b+b'} \cap A_{b'});$$

(2) has representation  $\bigcup_{b' \in B} (A_{b'} \cap A);$

For each  $b' \in B$  there exists a  $b \in B$  s.t.  $b' + b = o$  (because  $B$  is symmetric). Hence the family  $\{A \cap A_{b'}, b' \in B\}$  is a subfamily of  $\{A_{b+b'} \cap A_{b'}, b' \in B, b \in B\}$ ; i.e. (2) is a subset of (1).

Hence  $\gamma\gamma(A) = \gamma(A)$ ; i.e.  $\gamma$  is idempotent.

$\therefore \gamma$  must be an opening.

*Remarks*

- (i) When  $B$  is a ring-shaped (annular) SE centred about the origin "this opening removes isolated particles in a set" (Ronse & Heijmans, 1991, p.82).
- (ii) The annular opening removes particles from  $A$  "as a function of their environment, and does not consider their size or shape [in contrast to the structural opening]" (Serra, 1988, p. 108).

Ronse and Heijmans (1991, p. 84) extended the annular opening to grey-scale images and more generally to complete *distributive* lattices (see section 2.4.7.). For a flat symmetric SE  $g$  centred about the origin  $f \wedge B(f, g)$  is an opening. Oddly enough neither Ronse and Heijmans (1991) nor Serra (1988) mention the dual operator; the annular closing. For that matter, the notion of an annular closing does not appear anywhere in the literature. Consider the following definition for the complete lattice  $\mathcal{P}(\mathbb{E}^n)$ .

**Definition 29. The annular closing**

Given  $A, B \in \mathcal{P}(\mathbb{E}^n)$  such that  $B$  is a symmetric SE, i.e.  $B = \tilde{B}$ , the mapping

$$\varphi: A \rightarrow (A \ominus B) \cup A$$

is a closing with domain of invariance  $\mathcal{B} = \{A \mid A \ominus B \subseteq A\}$ .

To prove that  $\varphi: A \rightarrow (A \ominus B) \cup A$  is a closing it is necessary to show that it is increasing, extensive and idempotent. This necessarily follows by duality from the preceding proof.

*Proof*

- (i)  $\varphi$  is increasing because it is the union of two increasing operators.
- (ii)  $\varphi$  is extensive by design

$$\left. \begin{array}{l} o \in B \Rightarrow \varphi(A) = A \\ o \in B \Rightarrow \varphi(A) \supseteq A \end{array} \right\} \varphi(A) \supseteq A.$$

- (iii) For idempotence consider  $\varphi \circ \varphi$

$$\varphi\varphi(A) = \underbrace{\left[ ((A \ominus B) \cup A) \ominus B \right]}_{(1)} \cup \underbrace{\left[ (A \ominus B) \cup A \right]}_{(2)}$$

(1) has representation

$$\left( \bigcap_{b \in B} A_b \cup A \right) \ominus B \Leftrightarrow \bigcap_{b' \in B} \bigcap_{b \in B} (A_b \cup A)_{b'} \Leftrightarrow \bigcap_{b' \in B} \bigcap_{b \in B} (A_{b+b'} \cup A_{b'});$$

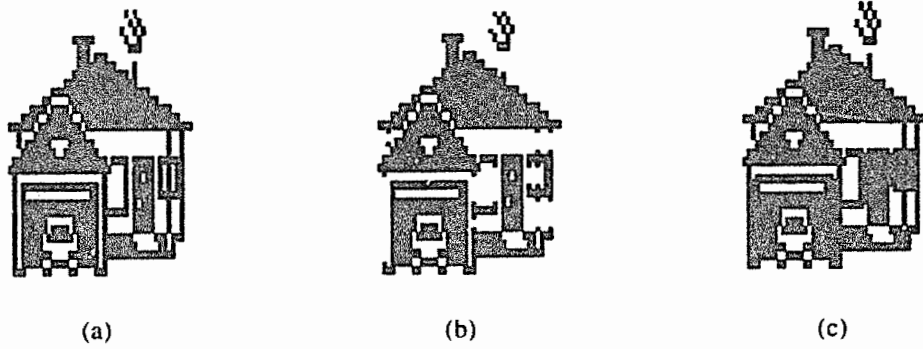
(2) has representation  $\bigcap_{b' \in B} (A_{b'} \cup A);$

For each  $b' \in B$  there exists a  $b \in B$  s.t.  $b' + b = 0$  (because  $B$  is symmetric). Hence the family  $\{A \cup A_{b'}, b' \in B\}$  is a subfamily of  $\{A_{b+b'} \cup A_{b'}, b' \in B, b \in B\}$ . Thus the intersection of all the translates in (1) must equal the intersection of all the translates in (2). Hence  $\varphi\varphi(A) = \varphi(A)$ ; i.e.  $\varphi$  is idempotent.

$\therefore \varphi$  must be a closing.

#### Remark

This closing fills holes in the foreground of an image as a function of their environment and not size or shape (see Figure 9).



$$B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}_{-1,1}$$

Figure 9. (a) Original image  $X$ . (b) Annular opening by  $B$ . (c) Annular closing by  $B$ .

### 3.3.2.2. Generating openings from inf-overfilters and closings from sup-underfilters

In general the operator  $I \wedge \psi$  is an opening whenever  $\psi$  is an *inf-overfilter* and the operator  $I \vee \psi$  is a closing whenever  $\psi$  is a *sup-underfilter*.

#### Definition 30. Inf-overfilters and sup-underfilters (Matheron).

A mapping  $\psi \in \mathcal{O}$  is an inf-overfilter ( $\wedge$ -overfilter) if  $\psi \circ (I \wedge \psi) = \psi$ . A mapping  $\zeta \in \mathcal{O}$  is a sup-underfilter ( $\vee$ -underfilter) if  $\zeta \circ (I \vee \zeta) = \zeta$ .

### Remark

"Any  $\vee$ -underfilter is an underfilter, any  $\wedge$ -overfilter is an overfilter . . . . But the converse is not true: there are underfilters that are not  $\vee$ -underfilters" (Serra, 1988, p. 123).

Though the annular opening has the form  $I \wedge \psi$ ,  $\psi = (A \oplus B)$  is not an inf-overfilter in general. When  $B$  contains the origin though,  $\psi$  is a conditionally extensive dilation and is both an inf-overfilter (because  $[A \cap (A \oplus B)] \oplus B = A \oplus B$ ,  $o \in B$ ) and an overfilter (see the remarks under Definition 26). Thus annular openings are not examples of openings constructed from inf-overfilters. Furthermore annular closings are not examples of closings constructed from sup-underfilters. The following result, due to Matheron (in Serra, 1988, p. 129), characterises under- and overfilters, and inf-overfilters and sup-underfilters.

### Proposition 7.

Given  $\eta, \varphi, \gamma \in \mathcal{O}$  where  $\varphi$  is a closing such that  $\eta \prec \varphi$ , and  $\gamma$  is an opening such that  $\gamma \prec \eta$  then

- $\psi$  is an *underfilter* if and only if  $\psi = \varphi \eta$ ,
- $\psi$  is a *sup-underfilter* if and only if  $\psi = \eta \varphi$ ,
- $\psi$  is an *overfilter* if and only if  $\psi = \gamma \eta$ , and
- $\psi$  is an *inf-overfilter* if and only if  $\psi = \eta \gamma$ .

### Remark

Trivially when  $\eta$  is the identity operator  $I$  then  $\gamma \prec I \prec \varphi$  (recall Definition 27) and hence  $\gamma$  is an inf-overfilter and  $\varphi$  is a sup-underfilter. In fact  $I \wedge \gamma = \gamma$  and  $I \vee \varphi = \varphi$ .

### Example

Consider the lattice  $\mathcal{P}(\mathbb{Z}^2)$ . Let  $\delta_B = \mathcal{D}(\cdot, B)$  and  $\varepsilon_B = \mathcal{E}(\cdot, B)$ . The composition  $\delta_A \varepsilon_B$  for  $B \subseteq A$  and  $A, B \in \mathcal{P}(\mathbb{Z}^2)$  is an inf-overfilter (Ronse & Heijmans, 1991, p. 87). To prove this put  $\gamma = \delta_B \varepsilon_B$  and  $\eta = \delta_A \varepsilon_B$ . According to Proposition 7 the product  $\eta \gamma = \delta_A \varepsilon_B \delta_B \varepsilon_B = \delta_A \varepsilon_B$  (see the remarks under Theorem 1) is an inf-overfilter if and only if  $\gamma \prec \eta$ . This inequality is satisfied because  $\delta_B \subseteq \delta_A \Rightarrow \delta_B \varepsilon_B \subseteq \delta_A \varepsilon_B$  for  $B \subseteq A$ . Therefore  $\delta_A \varepsilon_B$  is an inf-overfilter and it follows that the operator  $I \wedge \delta_A \varepsilon_B$  must be an opening. By duality  $\varepsilon_A \delta_B$  is a sup-underfilter and so  $I \vee \varepsilon_A \delta_B$  must be a closing.



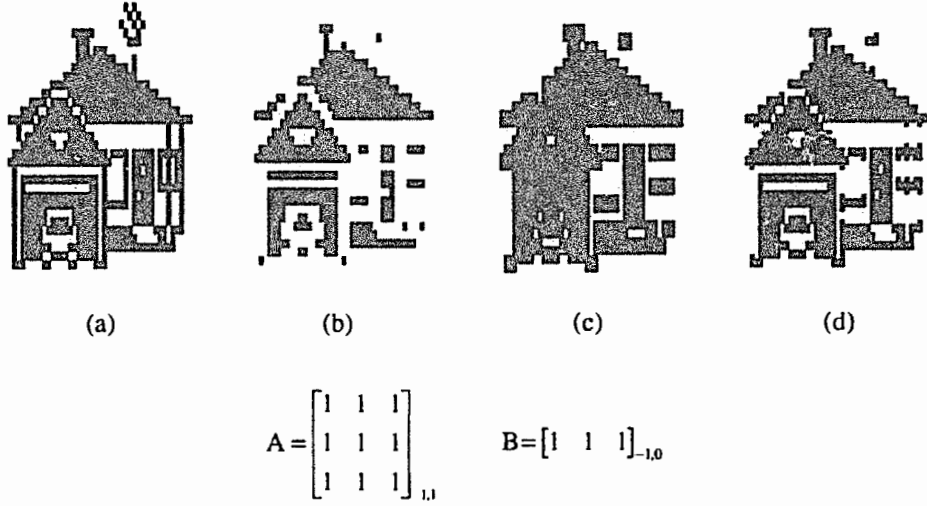


Figure 10. An opening constructed from an inf-overfilter. (a) Original image  $X$ . (b)  $\varepsilon_B$ . (c)  $\delta_A \varepsilon_B$ . (d)  $X \cap \delta_A \varepsilon_B$ .

A formal definition of the domain of invariance of openings generated from inf-overfilters on complete distributive lattices can be found in Ronse and Heijmans (1991, p. 87). In the case of  $X \cap \delta_A \varepsilon_B$  the domain of invariance is the family of translates of the sets enclosed between  $B$  and  $A$  (p. 88). In particular the family consists of unions of translates of  $B$ , and the subsets of  $A$  each generated by adding a single point from the set difference  $A - B$  to the set  $B$  (see Figure 11).

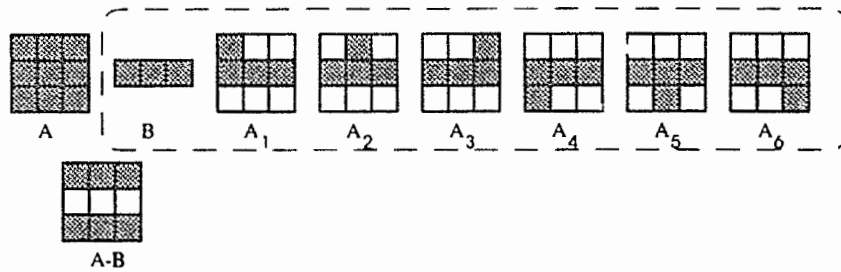


Figure 11. The domain of invariance of  $X \cap \delta_A \varepsilon_B$  consists of all the sets generated by unions of translates of the sets in the family  $\{B, A_i\}$ .

Consequently the opening depicted in Figure 10 can be decomposed into the union of seven morphological openings; one for each SE in the family  $\{B, A_i\}$  (see Theorem 7).

### 3.3.2.3. The Matheron representation theorem for increasing and translation-invariant mappings

In the Euclidean setting there exists a more specific result than Theorem 2 concerning increasing mappings. Matheron (1975) proved that all mappings  $\rho(\mathbf{E}^n) \rightarrow \rho(\mathbf{E}^n)$  that are translation-invariant and increasing can be represented as a union of (structural) erosions or alternatively as an intersection of (structural) dilations (cited in Heijmans, 1991, p. 569). This class of mappings includes not only dilations, erosions, openings, and closings, but also order statistic filters and spatial convolutions with non-negative mask weights (see the Giardina and Dougherty definition of a morphological filter – last paragraph of section 2.6.). Though the Matheron theorem does not offer a direct means of generating openings or closings it is pertinent to the theory of rank-openings (section 3.3.2.5.) and is also relevant to the generation of openings and closings from increasing digital mappings (section 3.3.2.4.).

**Theorem 9. Matheron representation theorem for increasing translation-invariant mappings.**

If  $\psi: \rho(\mathbf{E}^n) \rightarrow \rho(\mathbf{E}^n)$  is an increasing and translation-invariant mapping, then for any  $A \in \rho(\mathbf{E}^n)$ ,

$$\psi(A) = \bigcup_{B \in \text{Ker}[\psi]} \mathcal{E}(A, B) = \bigcap_{B \in \text{Ker}[\psi^*]} \mathcal{D}(A, B)$$

where  $\text{Ker}[\psi] = \{B \in \rho(\mathbf{E}^n) \mid o \in \psi(B)\}$  is the *kernel* of  $\psi$ ,  $\psi^*$  is the dual of  $\psi$  defined  $\psi^*(A) = [\psi(A^c)]^c$  and  $o$  is the origin.

#### Remarks

- (i)  $\text{Ker}[\psi]$  is the collection of all subsets  $B$  of  $\mathbf{E}^n$  such that  $\psi(B)$  contains the origin.
- (ii) Giardina and Dougherty (1988) point out that "while the kernel is in general uncountably infinite, there is a great redundancy in the expansion as given by Matheron" (p. 136).

"As it stands, the Matheron representation is computationally intractable" (Dougherty & Kraus, 1990b, p. 15). It turns out that the Matheron expansion "need only be taken over a subcollection of the kernel, called the *basis* [italics added]" (Dougherty & Kraus, 1990a, p. 161.). "The . . . basis [representation] was introduced independently by Maragos . . . and Dougherty and Giardina" (Dougherty & Kraus, 1990b, p. 15). The basis is a subset of the kernel for which

- (i) given any  $K \in \text{Ker}[\psi]$  there exists a  $B \in \text{Bas}[\psi]$  such that  $B \subseteq K$ , and
- (ii)  $B_i, B_j \in \text{Bas}[\psi] \Leftrightarrow B_i \not\subset B_j$ , for all pairs  $i, j$ .

"If the underlying space is discrete [i.e.  $E = Z$ ] and if the transformation  $\psi$  is locally defined . . . then . . . [the basis] can be chosen finite" (Heijmans, 1989, p. 23). Note that a transformation  $\psi$  (e.g. a 3×3 median filter) is said to be *locally defined* (or to have *local knowledge*) if for any bounded set  $Z$  (e.g. a video frame) within which one wishes to know  $\psi(X)$ ,  $\psi(X)$  can be determined at each point in  $X \cap Z$  from some mask  $M$  (e.g. a 3×3 moving window) strictly within  $Z$  (Serra, 1982, p. 11).

### Example



Figure 12. (a) Original image  $X$ . (b) 4-median of  $X$ .

The 4-median filter is a variant of the 3×3 moving window median filter. It differs in that a cross-shaped window (the central pixel and its direct neighbours in the 3×3 window) is employed rather than a square window. At each new window position the five image pixels are sorted and the median value replaces the central pixel in the filtered image. Figure 12 shows a binary image  $X$  and the result after applying the 4-median filter. This is the same example used by Heijmans (1989, p. 24). However although Heijmans proves that the 4-median filter has a basis representation consisting of 10 erosions, he does not derive them. For any given position the sequence of pixels in the cross-window, after sorting, must be one of: 00000, 00001, 00011, 00111, 01111, 11111. The last three configurations lead to an output value of 1. The number of possible window configurations for which the median is a 1 is as follows:

$$C_3^5 = 10 \text{ possible configurations of three 1s and two 0s,}$$

$$C_4^5 = 5 \text{ possible configurations of four 1s and one 0, and}$$

$$C_5^5 = 1 \text{ possible configuration of five 1s and no 0s.}$$

Figure 13 shows each of the 16 configurations. Now binary erosion can be expressed as  $\mathcal{E}(A, B) = \{x \mid B_x \subseteq A\}$  (see the remarks under Definition 20). If the configurations in Figure 13 are interpreted to be SEs with the origin at the centre, then the 4-median filter  $\psi(A)$  has the representation

$$\psi(A) = \bigcup_{i=1}^{16} \mathcal{E}(A, B_i), \text{ where the } B_i \text{ are the SEs in Figure 13.}$$

In fact because the last 6 configurations (SEs) are supersets of the first 10 configurations, the 4-median filter has the basis representation

$$\psi(A) = \bigcup_{i=1}^{10} \mathcal{E}(A, B_i), \text{ where the } B_i \text{ are the first ten SEs in Figure 13.}$$

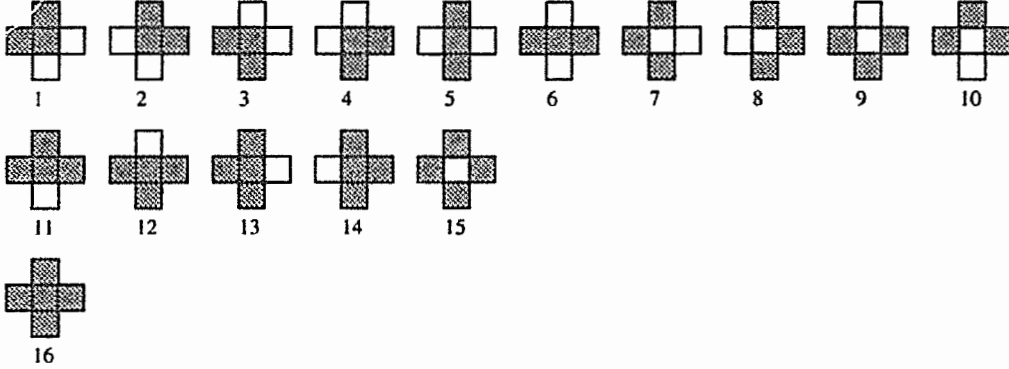


Figure 13. The 16 possible pixel configurations for the 4-median filter that lead to a binary 1.

The first ten configurations are unique and satisfy the basis requirements that (i) no element of the basis is a subset of any other and that (ii) the 4-median of each configuration contains the origin. Figure 14 illustrates the ten erosions of the original image, depicted in Figure 12, by the SEs in the basis set. Clearly their union gives the 4-median depicted in Figure 12. In comparison to the 4-median filter, the basis set of the more familiar  $3 \times 3$  median filter consists of  $C_3^9 = 126$  SEs.



Figure 14. The 10 erosions whose union yields the 4-median filter.

The grey-scale version of Matheron's theorem "was first discovered by Maragos . . . under the condition that  $f$  [in  $\psi(f)$ ] is upper semicontinuous" (Dougherty & Kraus, 1990b, p. 16). Giardina and Dougherty (1988) arrived at the same representation without the u.s.c. requirement (although like Maragos they employ umbrae in their proof). The most general proof is that proffered by Crombez (1990) – the proof doesn't use the umbra transform or the binary representation theorem.

**Theorem 10. Grey-scale Matheron representation theorem.**

If  $\psi$  is a translation-invariant and increasing mapping between bounded, real-valued functions with domains in  $\mathbf{R}^2$  then for any such function  $f$ ,

$$\text{Domain}[\psi(f)] = \bigcup_{g \in \text{Ker}[\psi]} \text{Domain}[\mathcal{E}(f, g)], \text{ and}$$

$$\psi(f)(x) = \bigvee_{\substack{g \in \text{Ker}[\psi] \\ x \in \text{Domain}[\mathcal{E}(f, g)]}} \mathcal{E}(f, g)(x).$$

**Remarks**

- (i) The Matheron representation theorem for binary images is a special case of the grey-scale theorem; the case where functions only take on the value 0.
- (ii) By duality a grey-scale translation-invariant and increasing mapping can be expressed as the infimum of dilations.

- (iii) Giardina and Dougherty (1988) extended the theorem to the digital setting. Dougherty (1989) established the dual result.

Once again the representation has theoretical relevance "but its practicality is rather small because, in general, it is not a simple task to analytically find and describe all the (infinite in number) kernel functions" (Maragos & Schafer, 1987a, p. 1166). Studies apropos to the basis representation have been done by Dougherty and Giardina, and Maragos and Schafer; e.g. Dougherty and Giardina (1986), Maragos and Schafer (1987a, 1987b), Giardina and Dougherty (1988), and Maragos (1989).

### 3.3.2.4. Generating openings and closings from increasing digital mappings

For any increasing mapping  $\psi$  defined on a finite complete lattice  $(\mathcal{L}, \prec)$ , the iteration of the operator  $I \wedge \psi$  a certain number of times leads to an opening (Serra, 1988, p.110).

#### *Proposition 8*

If  $(\mathcal{L}, \prec)$  is a finite complete lattice then for every  $\psi \in \mathcal{O}$  there exists an opening  $\gamma$ , such that  $\gamma \prec \psi$ , given by

$$\gamma = (I \wedge \psi)^n$$

for some large  $n$ . Similarly there exists a closing  $\varphi \prec \psi$  given by

$$\varphi = (I \vee \psi)^m$$

for some large  $m$ .

#### *Remarks*

- (i) Digital grey-tone images (and by subsumption digital binary images) have finite spatial and radiometric (brightness) extent (resolution). The set of all digital images therefore constitutes a finite lattice.
- (ii) Matheron (cited in Serra, 1988, p. 113) has shown that Proposition 8 does not hold true in the infinite case.

#### *Example*

Heijmans (1989, p. 35) introduced the median opening  $\gamma(X) = (X \cap \psi(X))^n$  where  $\psi$  is a median filter. By Proposition 8,  $\varphi(X) = (X \cup \psi(X))^m$  is a (median) closing. Figures 15 and 16 illustrate the 4-median opening and the 4-median closing respectively.



Figure 15. From left to right, top to bottom: original image  $X$ ,  $(X \cap \psi(X))$ ,  $(X \cap \psi(X))^2$ , ...,  $(X \cap \psi(X))^{10}$ , where  $\psi$  is the 4-median filter. In this case,  $\gamma(X) = (X \cap \psi(X))^n$  is an opening for  $n=10$  (i.e. sequence converges after 10 iterations).



Figure 16. For the original image  $X$  depicted in Figure 15,  $\phi(X) = (X \cup \psi(X))^m$  is a closing for  $m=3$ .

### 3.3.2.5. Rank-openings

The median filter belongs to the more general class of order statistic filters otherwise known as ranked order filters. These filters "are a class of nonlinear and translation-invariant discrete filters that have become popular in digital speech and image processing, and also in statistical or econometric time series analysis" (Maragos & Schafer, 1987b, p. 1170). Furthermore they are easy to implement and have the desirable property that "they very effectively reduce high frequency and impulsive noise in digital images without the extensive blurring and edge destruction associated with linear filters" (Fitch, Coyle

& Gallagher, 1985, p. 445). Ranked order filters are not morphological filters because, although they are increasing, they are not, in general, idempotent. A median filter, for example, has the undesirable property that its iteration may produce oscillations (Serra, 1988, p.160).

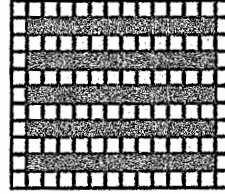


Figure 17. This image oscillates when a 3x3 median filter is applied to it iteratively (adapted from Serra, 1988, p. 160).

For a discrete function  $f: \mathbb{Z}^n \rightarrow \mathbb{Z}$  and a finite window  $W \subset \mathbb{Z}^n$  containing the origin, the rank operator of rank  $k$ , also called the  $k$ -th order statistic filter, is defined pointwise as follows:

$$[\text{Rank}_k(f, W)](x) = k \text{ th order statistic of } \{f(y) \mid y \in W_x\} \text{ where } x \in \mathbb{Z}^n.$$

Effectively the origin of the window  $W$  is moved to each point  $x$  of  $f(x)$ , the points of  $f(x)$  within the window are sorted, and  $f(x)$  is replaced by the  $k$ -th order statistic. If  $W$  comprises  $n$  points then for

- (i)  $k=n$  the operator is identically  $f \oplus \tilde{W}$  (see relation (8) of section 2.5.2.),
- (ii)  $k=1$  the operator is identically  $f \ominus \tilde{W}$  (see relation (9) of section 2.5.2.), and
- (iii)  $k=(n+1)/2$ , and  $n$  is odd, the operator is the median filter.

The window  $W$  is actually a flat (binary) SE that can be of any shape as long as it has finite extent. Serra (1988, p. 193) points out that the rank operator is increasing because it can be represented as the supremum of erosions (see Theorem 10) viz.

$$\text{Rank}_k(f, W) = \sup \{ f \ominus \tilde{W}_i \mid W_i \subseteq W, \text{Card } W_i \geq k \}.$$

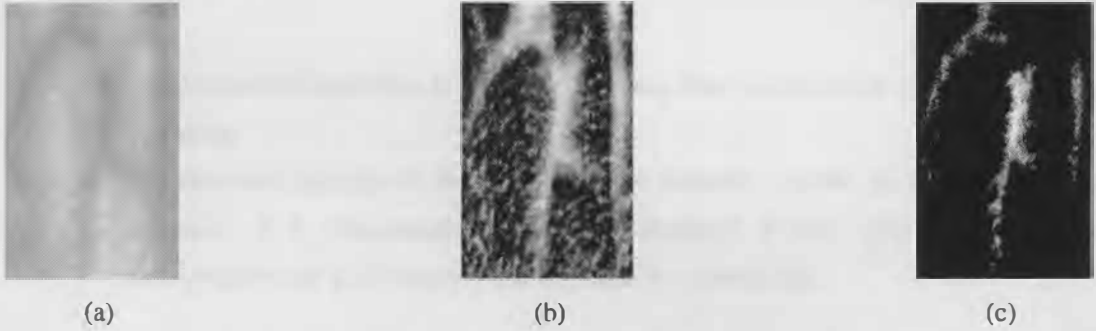
(See the example of section 3.3.2.3.)

The operator  $\delta_W \text{Rank}_k(f, W) = \delta_W \left( \bigvee_i \epsilon_{W_i} \right)$ , where  $W_i \subseteq W$ , is an inf-overfilter. Therefore  $I \wedge \delta_W \text{Rank}_k(f, W)$  is an opening (called a *rank-opening*; see section 3.3.2.2.). To show this recall that the dilation commutes with the supremum (Definition 9) so that  $\delta_W \left( \bigvee_i \epsilon_{W_i} \right) = \bigvee_i (\delta_W \epsilon_{W_i})$ . Therefore  $I \wedge \delta_W \text{Rank}_k(f, W) = I \wedge \left( \bigvee_i (\delta_W \epsilon_{W_i}) \right)$ . This can be written as  $\bigvee_i (I \wedge \delta_W \epsilon_{W_i})$ . The expression in the



brackets is the opening  $I \wedge \delta_A \epsilon_B$ , for  $B \subseteq A$ , introduced in section 3.3.2.2. It is possible to replace the  $W$  in  $\delta_W$  with any set  $C$  that contains  $W$ , and "to obtain another opening that is less active [i.e. closer to the identity mapping] as  $C$  is large" (Serra, 1988, p. 193).

The arithmetic difference  $I - \gamma$ , where  $\gamma$  is a morphological opening by a structuring element, is known as the *tophat transform* (see section 6.4.3.). Replacing the structural opening with a rank-opening reduces the sensitivity of the difference  $I - \gamma$  to noise and artefacts (Ronse & Heijmans, 1991, p. 89). Figure 18 illustrates this type of tophat transform. The aim is to enhance the surface blood vessels in a digitised infrared image of the back of a hand (the original image in its entirety is depicted in Figure 20). Here the transform is actually applied to the negative of the image  $f$  (thereby converting valleys to peaks and vice versa) so that the subcutaneous vascular network (which is darker than the background) is highlighted.



**Figure 18.** The sensitivity of the tophat transform to noise and artefacts can be reduced by replacing the structural opening with a rank-opening.

- (a) A thermographic image,  $f$ , of a section of the subcutaneous vascular network on the back of a hand (see Figure 20).
- (b) Histogram equalisation of the tophat transform  $(-f) - \min[-f, \mathcal{B}(\text{Rank}_1(-f, B), B)]$ , where  $B$  is a  $45 \times 45$  square SE. Here the rank operator is identically an erosion by the symmetric SE  $B$  (origin at centre) and hence the tophat transform simplifies to  $(-f) - \mathcal{O}(-f, B)$ .
- (c) Histogram equalisation of the tophat transform  $(-f) - \min[-f, \mathcal{B}(\text{Rank}_{23}(-f, B), B)]$ . Here the rank operator is identically the  $45 \times 45$  median filter.

### 3.4. Composite Morphological Filters

The fact that, in general, the composition of two morphological filters is not a morphological filter motivated the introduction of two classes of mappings: overfilters and underfilters (Definition 26). Recall that a morphological filter  $\psi \in \mathcal{O}$  is necessarily both an under- and overfilter (section 3.3.) so that  $(I \vee \psi) \circ \psi = \psi$  and  $(I \wedge \psi) \circ \psi = \psi$ . Reversing the order of composition, however, leads only to the

inequalities  $\psi \prec \psi \circ (I \vee \psi)$  and  $\psi \circ (I \wedge \psi) \prec \psi$  (Serra, 1988, p. 115). These two inequalities motivate the definition of sup- and inf-filters.

**Definition 31. Sup- and inf-filters.**

Let  $\psi \in \mathcal{O}$  be a morphological filter. If  $\psi \circ (I \vee \psi) = \psi$  then  $\psi$  is called a *sup-filter*. If  $\psi \circ (I \wedge \psi) = \psi$  then  $\psi$  is called an *inf-filter*.

**Remark**

An inf-filter must be an inf-overfilter and at the same time an underfilter. Likewise, a sup-filter must be a sup-underfilter and at the same time an overfilter (Serra, 1988, p. 123).

**Definition 32. Strong filters.**

A morphological filter that is both an inf-filter and a sup-filter is called a *strong filter*.

**Remarks**

- (i) The definition is equivalent to saying that a strong filter is both an inf-overfilter and a sup-underfilter.
- (ii) Openings and closings are strong filters. For example, consider an opening  $\gamma \in \mathcal{O}$ . By definition,  $\gamma$  is anti-extensive; i.e.  $\gamma \prec I$ . Therefore  $(I \vee \gamma) = I$  and  $(I \wedge \gamma) = \gamma$ . Hence  $\gamma \circ (I \vee \gamma) = \gamma \circ I = \gamma$  and  $\gamma \circ (I \wedge \gamma) = \gamma \circ \gamma = \gamma$  and so  $\gamma$  must be a strong filter.

### 3.4.1. Open-closings and close-openings

"In contrast to the case of dilations [and by duality erosions], the composition of two openings is generally not an opening. This composition is anti-extensive and increasing, but not necessarily idempotent" (Ronse & Heijmans, 1991, p. 77). By duality the product of two closings does not necessarily yield another closing. But what can be said of the compositions  $\gamma\phi$  and  $\phi\gamma$  for an arbitrary opening  $\gamma$  and closing  $\phi$ ? In his 1986 paper on grey-scale morphology (originally submitted for publication in 1983) Sternberg states "the product of an opening and a closing is increasing and idempotent. Similarly, a transformation consisting of a closing followed by an opening is increasing and idempotent" (p. 346). A formal introduction to *open-closings* and *close-openings* [sic] for sets and u.s.c. functions can be found in Maragos' PhD thesis (cited in Maragos & Schafer, 1987a, p. 1161). In Maragos' terminology an open-closing is an opening followed by a closing using the same SE. Other authors (e.g. Chmúrny & Lehtoský, 1990) refer to an open-closing as the opening of a closing; i.e. the product  $\gamma\phi$  using a single SE, which is a close-opening in Maragos' terminology. It turns out that in the more general setting of the lattice of increasing operators  $\mathcal{O}$ ,  $\gamma\phi$  is a sup-filter and  $\phi\gamma$  is an inf-filter (Note:  $\gamma$  need not be the dual of  $\phi$ ). In fact Serra (1988, p. 116) proves that the converse (see the following proposition) is true.

### Proposition 9

The transform  $\psi \in \mathcal{O}$  is a sup-filter (resp. inf-filter) if and only if it has the representation  $\psi = \gamma\phi$  (resp.  $\psi = \phi\gamma$ ), for an opening  $\gamma$  and a closing  $\phi$ .

### Remark

In the literature, the abbreviations OC and CO are used by some authors (e.g. Chmúrny & Lehtoský, 1990) to denote opening of a closing and closing of an opening respectively. Unfortunately the same abbreviations are also used to denote the multiple SE filters (section 3.8.) of Stevenson and Arce (1986, 1987).

An open-closing generally produces a result different to that of a close-opening when applied to the same grey-scale image. The extensivity of the closing and the anti-extensivity of the opening induces grey-scale bias.

## 3.4.2. General composition

The following two propositions, due to Matheron (in Serra, 1988, p. 119), characterise compositions of morphological filters.

### Proposition 10.

Given any two morphological filters  $\psi, \zeta \in \mathcal{O}$  such that  $\psi \prec \zeta$ , then

- (i)  $\psi \prec \psi\zeta\psi \prec \psi\zeta \wedge \zeta\psi \prec \psi\zeta \vee \zeta\psi \prec \zeta\psi\zeta \prec \zeta$ ;
- (ii)  $\zeta\psi, \zeta\psi\zeta$  are morphological filters with the same domain of invariance  $\mathcal{B}_{\zeta\psi}$ , and  $\psi\zeta, \psi\zeta\psi$  are morphological filters with the same domain of invariance  $\mathcal{B}_{\psi\zeta}$ ;
- (iii)  $\zeta\psi\zeta$  is the smallest morphological filter greater than  $\psi\zeta \vee \zeta\psi$ , and  $\psi\zeta\psi$  is the greatest morphological filter smaller than  $\psi\zeta \wedge \zeta\psi$ ;
- (iv) the following equivalences hold:

$$\begin{aligned} \mathcal{B}_{\zeta\psi} = \mathcal{B}_{\psi\zeta} &\Leftrightarrow \mathcal{B}_{\zeta\psi} = \mathcal{B}_{\zeta} \cap \mathcal{B}_{\psi} \Leftrightarrow \mathcal{B}_{\psi\zeta} = \mathcal{B}_{\zeta} \cap \mathcal{B}_{\psi} \\ &\Leftrightarrow \zeta\psi\zeta = \psi\zeta \Leftrightarrow \psi\zeta\psi = \zeta\psi \\ &\Leftrightarrow \zeta\psi \prec \psi\zeta. \end{aligned}$$

### Proposition 11.

Given any two morphological filters  $\psi, \zeta \in \mathcal{O}$  such that  $\psi \prec \zeta$ , then

- (i) if  $\zeta$  is a sup-filter,  $\psi\zeta$  and  $\zeta\psi\zeta$  are sup-filters;
- (ii) if  $\psi$  is an inf-filter,  $\zeta\psi$  and  $\psi\zeta\psi$  are inf-filters;
- (iii) if  $\psi\zeta$  is an inf-filter,  $\zeta\psi\zeta$  is an inf-filter;
- (iv) if  $\zeta\psi$  is a sup-filter,  $\psi\zeta\psi$  is a sup-filter.

### 3.5. The Middle Filter

If the lattice of operands  $(\mathcal{L}, \prec)$  is modular (see the remark under Definition 14) then for an arbitrary pair comprising an inf-overfilter and a larger (w.r.t. the ordering on the lattice of operators  $(\mathcal{O}, \prec)$ ) sup-underfilter there exists a strong filter, called the *middle element*, between them. From a practical standpoint the modularity requirement is inconsequential because the complete lattice of all u.s.c. functions  $\mathcal{F}: \mathbf{E}^n \rightarrow \overline{\mathbf{E}}$  and the complete lattice  $\mathcal{P}(\mathbf{E}^n)$  are both distributive and hence modular.

**Theorem 11.** *The middle filter (Serra, 1988, p. 133).*

If  $\xi$  is a sup-underfilter and  $\zeta$  is an inf-overfilter on the complete modular lattice  $(\mathcal{L}, \prec)$  and  $\zeta \prec \xi$  then there exists a strong filter  $\alpha$ , called the middle element of  $\xi$  and  $\zeta$ , such that

$$\zeta \prec \alpha \prec \xi.$$

The domain of invariance of  $\alpha$  is given by  $\mathcal{B}_\alpha = \mathcal{B}_\xi \cap \mathcal{B}_\zeta$  where  $\mathcal{B}_\psi = \{X \mid X \in \mathcal{L}, X \prec \psi(X)\}$  denotes the domain of extensivity of  $\psi$  and  $\mathcal{B}_\psi = \{X \mid X \in \mathcal{L}, \psi(X) \prec X\}$  denotes the domain of anti-extensivity of  $\psi$ .

**Corollary**

The middle element  $\alpha$  is the only strong filter  $\psi$ , with domain of invariance  $\mathcal{B}_\alpha$ , such that  $\zeta \prec \psi \prec \xi$ .

It turns out that the middle element is the idempotent limit of the successive powers of a mapping called the morphological centre which is defined in the next section.

### 3.6. Self Dual Filtering - The Morphological Centre

The most striking difference between morphological filters and more conventional filters, such as convolutions (weighted moving averages with positive weights) and order statistic filters, is that they have the ability to treat the positive and negative features of an image differently. "In practice this selectivity is often found to be an advantage; it allows us to extract light details on a dark background (or the opposite), and to fit the filter very accurately to the type of image under study" (Serra, 1988, p. 159). Convolutions and order statistic filters are *self-dual*; i.e.  $\psi(-f) = -\psi(f)$  where  $\psi$  represents the filter and  $f$  is a real-valued function with domain in  $\mathbf{R}^n$ . Self-duality is not an intrinsic property of morphological filters however. For example the grey-scale opening is clearly not self-dual, i.e.  $\mathcal{O}(-f, g) \neq -\mathcal{O}(f, g)$ , because in fact  $\mathcal{O}(-f, g) = -\mathcal{C}(f, \tilde{g})$  (Proposition 5). In situations where image features are sometimes darker than the background and sometimes lighter, self-dual filtering is desirable (e.g. to attenuate salt-and-pepper noise). The *morphological centre* is a self-dual mapping that retains

the advantages of morphological filtering, such as compatibility with anamorphoses (for FSP morphological filters - see section 2.5.4.); a property not shared by convolution (Serra, 1988, p. 159). Definition of the morphological centre requires that the underlying object space  $(\mathcal{L}, \prec)$  be a complete distributive lattice. It follows that if the lattice of operands  $(\mathcal{L}, \prec)$  is a complete distributive lattice then so is the lattice of operators  $(\mathcal{O}, \prec)$  (Serra, 1988, p. 164). The following proposition establishes a new kind of partial order relation, denoted  $\prec_I$ , that permits the comparison of two morphological filters on the basis of *closeness* (w.r.t. the ordering  $\prec$ ) to the identity mapping. The relation  $\psi_1 \prec_I \psi_2$  indicates that the mapping  $\psi_1$  is closer to the identity mapping than  $\psi_2$ . The poset  $(\mathcal{O}, \prec_I)$  defines an inf semilattice<sup>6</sup> with the identity operator  $I$  as the null element. For a family of mappings in  $\mathcal{O}$  the morphological centre is defined to be the infimum w.r.t. the ordering  $\prec_I$ .

**Proposition 12. The morphological centre.**

If  $(\mathcal{O}, \prec)$  is a complete distributive lattice then for the ordering  $\prec_I$  defined

$$\psi_1 \prec_I \psi_2 \Leftrightarrow \begin{cases} I \wedge \psi_2 \prec I \wedge \psi_1 \\ I \vee \psi_1 \prec I \vee \psi_2 \end{cases} \quad \text{for all pairs } \psi_1, \psi_2 \in \mathcal{O}.$$

$(\mathcal{O}, \prec_I)$  is a complete inf semilattice. For a family  $\{\psi_i\}$ ,  $\psi_i \in \mathcal{O}$ , in this semilattice the infimum  $\beta$  is called the *morphological centre* and is given by

$$\begin{aligned} \beta &= \inf_{(\mathcal{O}, \prec_I)} \psi_i \\ &= \left[ I \wedge \left( \bigvee_i \psi_i \right) \right] \vee \left( \bigwedge_i \psi_i \right) \\ &= \left[ I \vee \left( \bigwedge_i \psi_i \right) \right] \wedge \left( \bigvee_i \psi_i \right) \end{aligned} \quad \text{for sup and inf in the lattice } (\mathcal{O}, \prec).$$

**Definition 33. The morphological centre (Serra, 1988, p. 164).**

If  $(\mathcal{O}, \prec)$  is a complete distributive lattice and  $\xi', \zeta' \in \mathcal{O}$  such that  $\xi' \prec I \prec \zeta'$  then there exists a unique mapping  $\beta \in \mathcal{O}$  called the *morphological centre* determined by

$$\beta = (I \wedge \xi) \vee \zeta = (I \vee \zeta) \wedge \xi,$$

provided there exist  $\xi, \zeta \in \mathcal{O}$  with

$$\zeta \prec \xi, \quad \xi' = (I \wedge \xi), \quad \zeta' = (I \vee \zeta).$$

<sup>6</sup> An inf (resp. sup) *semilattice* is a poset  $(\mathcal{L}, R)$  for which the set  $\{X, Y\}$  has an infimum (resp. supremum) for all  $X, Y \in \mathcal{L}$ . A poset that is both an inf and sup semilattice is therefore a lattice.

The domain of invariance of  $\beta$  is given by  $\mathcal{D}_\beta = \mathcal{D}_\xi \cap \mathcal{D}_\zeta$ .

#### Remarks

- (i) In proposition 12,  $\xi = \vee \psi_i$  and  $\zeta = \wedge \psi_i$ .
- (ii) The following system of inequalities can be deduced from Definition 33:

$$\begin{cases} I \wedge \xi \prec \beta \prec \xi, \\ \zeta \prec \beta \prec I \vee \zeta. \end{cases}$$

Hence  $\beta$  is a central mapping in the sense that it is between  $\zeta$  and  $\xi$ ; i.e.  $\zeta \prec \beta \prec \xi$ .

Though  $\beta$  is a self-dual mapping it is not necessarily idempotent. The quest for idempotence leads to the following criterion (Serra, 1988, p. 166) that ensures that as  $n$  increases the successive powers  $\beta^2 = \beta \circ \beta, \dots, \beta^n = \beta^{n-1} \circ \beta$  become more and more active<sup>7</sup>; i.e. they form an increasing sequence w.r.t. the ordering  $\prec$  viz.  $\beta^2 \prec \beta^3 \prec \dots \prec \beta^n$ .

#### Proposition 13.

Let  $\{\psi_i\}$  be a family of filters in  $\mathcal{O}$ ,  $\xi = \vee \psi_i$ ,  $\zeta = \wedge \psi_i$ , and  $\beta$  be the centre of the  $\psi_i$ , then the sequence of successive powers  $\beta^n$  is increasing for the ordering  $\prec$  if for all integers  $n$ , such that  $0 < n < \infty$ , we have

$$\zeta_n = \bigvee_{k=0}^{n-1} \zeta \circ \beta^k \prec \xi_n = \bigwedge_{k=0}^{n-1} \xi \circ \beta^k.$$

$\beta^n$  is then given by

$$\beta^n = (I \wedge \xi_n) \vee \zeta_n.$$

In addition the domain of invariance of  $\beta^n$  is the same as that for  $\beta$ .

#### Remark

"There always exists an  $n_0 < \infty$  such that  $\beta^{n_0+1} = \beta^{n_0}$ , and this maximal iteration is a [morphological] filter" (Serra, 1988, p. 168).

The preceding theory can be generalised by considering an arbitrary mapping  $\theta \in \mathcal{O}$  in lieu of the identity mapping  $I$ . This leads to the ordering  $\prec_\theta$  defined

$$\psi_1 \prec_\theta \psi_2 \Leftrightarrow \begin{cases} \theta \wedge \psi_2 \prec \theta \wedge \psi_1 \\ \theta \vee \psi_1 \prec \theta \vee \psi_2 \end{cases} \quad \text{for all pairs } \psi_1, \psi_2 \in \mathcal{O}.$$

<sup>7</sup> Given  $\psi, \zeta \in \mathcal{O}$  the mapping  $\psi$  is said to be *more active* than  $\zeta$  if  $\zeta \prec \psi$ .

The preceding definition and propositions remain valid provided  $\theta$  is substituted for  $I$ . So for instance the morphological centre of the family  $\{\psi_i\}$ ,  $\psi_i \in \mathcal{O}$ , w.r.t. the mapping  $\theta$ , is given by

$$\beta_\theta = (\theta \wedge \xi) \vee \zeta = (\theta \vee \zeta) \wedge \xi, \text{ where } \xi = \vee \psi_i \text{ and } \zeta = \wedge \psi_i.$$

Note, however, that the domain of invariance of  $\beta_\theta$  must be expressed differently viz.

$$\mathcal{D}(\beta_\theta) = \mathcal{D}_\theta(\xi) \cap \mathcal{D}_\theta(\zeta),$$

where  $\mathcal{D}_\theta(\psi) = \{X \mid X \in \mathcal{L}, \theta(X) \prec \psi(X)\}$  and  $\mathcal{D}_\theta(\psi) = \{X \mid X \in \mathcal{L}, \psi(X) \prec \theta(X)\}$ .

### Examples

(i) Let  $\gamma$  be an arbitrary opening and  $\phi$  be an arbitrary closing on the complete distributive lattice  $(\mathcal{L}, \prec)$ .

Letting  $\xi = \phi$  and  $\zeta = \gamma$ , then  $\zeta \prec \xi$  is true (because  $\gamma \prec I \prec \phi$  by definition). Furthermore  $\xi' = (I \wedge \phi)$  and  $\zeta' = (I \vee \gamma)$  and so  $\xi' \prec I \prec \zeta'$  is satisfied. Hence Definition 33 is applicable and the morphological centre is given by  $\beta = (I \wedge \xi) \vee \zeta = (I \wedge \phi) \vee \gamma = I \vee \gamma = I$ . So trivially the identity mapping is the morphological centre between  $\gamma$  and  $\phi$ .

(ii) Let  $\xi$  be an inf-overfilter so that  $\xi = \xi(I \wedge \xi)$  implying that  $\xi \prec \xi\xi$ . Let  $\zeta$  be a sup-underfilter so that  $\zeta = \zeta(I \vee \zeta)$  implying that  $\zeta\zeta \prec \zeta$ . Recall that  $I \wedge \xi \prec \beta \prec I \vee \zeta$  (see the remarks under Definition 33) which implies (Serra, 1988, p. 170) that

$$\dots \zeta\zeta \prec \zeta\beta \prec \zeta = \zeta(I \vee \zeta) \prec \xi = \xi(I \wedge \xi) \prec \xi\beta \prec \xi\xi \dots$$

This satisfies the requirements of Proposition 13. For  $n=1$ ,

$$\beta = (I \wedge \xi) \vee \zeta, \text{ because } \xi_1 = \wedge \{\xi\beta^0\} = \xi I = \xi, \text{ and } \zeta_1 = \vee \{\zeta\beta^0\} = \zeta I = \zeta.$$

For  $n=2$ ,

$$\xi_2 = \bigwedge_{k=0}^1 \xi\beta^k = \wedge \{\xi, \xi\beta\} = \xi, \text{ and } \zeta_2 = \bigvee_{k=0}^1 \xi\beta^k = \vee \{\zeta, \zeta\beta\} = \zeta. \text{ So } \beta^2 = (I \wedge \xi_2) \vee \zeta_2 = (I \wedge \xi) \vee \zeta.$$

in this case  $\beta$  is idempotent and is therefore a morphological filter. As an illustration (see Figure 19), let  $\xi = \phi\gamma\phi$  and  $\zeta = \gamma\phi\gamma$  where  $\gamma$  is a grey-scale opening by the unit square SE (i.e.  $3 \times 3$  pixels), and  $\phi$  is the dual closing. Now because openings and closings are strong filters they are at the same time inf-filters and sup-filters. By way of Propositions 10 and 11 the composition  $\phi\gamma\phi$  is a sup-filter and the composition  $\gamma\phi\gamma$  is an inf-filter, and furthermore  $\gamma\phi\gamma \prec \phi\gamma\phi$ ; i.e.  $\zeta \prec \xi$ . Now an inf-filter is

also an inf-overfilter and a sup-filter is also a sup-underfilter (see the remark under Definition 31). Consequently the centre is given by  $\beta = (I \wedge \phi \gamma \phi) \vee \gamma \phi \gamma$ . Moreover the middle filter  $\alpha$  between  $\xi$  and  $\zeta$  is the idempotent limit of  $\beta^n$ . In this case idempotence is reached at the first step so that  $\alpha = \beta = (I \wedge \phi \gamma \phi) \vee \gamma \phi \gamma$ . In Figure 19,  $\beta$  is compared with a  $3 \times 3$  moving average, and a  $3 \times 3$  median filter. To facilitate visual comparison the morphological gradient is displayed for each of the methods of filtering. The gradient is defined to be  $[d(f, B) - \varepsilon(f, B)]$ , where  $B$  is the unit cross SE (i.e. consisting of 5 pixels). (Note: to produce grey on a white background the grey-maps for each of the gradient images have been reversed). For the gradient image of the moving average, contours are blurred. The gradient image of the median filter is much better though some noise is still evident top right. The gradient image of the morphological centre, however, is the best; it exhibits sharply defined contours and the best noise attenuation.





(a)



(b)



(c)



(d)



(e)

Figure 19. The morphological centre.

(a) Original image  $f$  of the Mona Lisa.

(b) Morphological gradient of  $f$ .

(c) Morphological gradient of the  $3 \times 3$  mean filtering of  $f$ .

(d) Morphological gradient of the  $3 \times 3$  median filtering of  $f$ .

(e) Morphological gradient of the morphological centre  $\beta = (I \wedge \varphi \gamma) \vee \gamma \varphi \gamma$ , where  $\gamma = \mathcal{O}(f, B)$  and  $\varphi = \mathcal{C}(f, B)$  and  $B$  is the unit square (i.e.  $3 \times 3$  pixels).

### 3.7. Alternating Sequential Filters

Sternberg (1986) introduced the *iterative morphological filter* for the purpose of attenuating "image noise without adding grayscale [sic] bias" (Sternberg, 1986, p. 345). It is an unfortunate fact that the application of an opening followed by a closing generally produces different results to the product of a closing followed by an opening. The iterative morphological filter consists of iterations of grey-scale open-closings or close-openings using an increasing family of homothetic convex SEs such as spheres of increasing radii. Sternberg's iterative morphological filter has been generalised by Serra (1988) in his theory of *alternating sequential filters*. The alternating sequential filter (ASF) is defined as follows.

**Definition 34.** *The alternating sequential filter.*

Let  $\{\gamma_\lambda\} \subset \mathcal{O}$  be a family of openings and  $\{\varphi_\lambda\} \subset \mathcal{C}$  be a family of closings dependent on the parameter  $\lambda \in \mathbb{R}^+$  such that  $\lambda \geq \mu \Rightarrow \gamma_\lambda \prec \gamma_\mu$  and  $\varphi_\mu \prec \varphi_\lambda$ . Furthermore, assume that both  $\gamma_\lambda$  and  $\varphi_\lambda$  are  $\downarrow$ -continuous for all  $\lambda$ . Next, define

$$M_k(\lambda, \lambda') = m_\lambda \dots m_{\lambda+i(\lambda'-\lambda)/2^k} \dots m_{\lambda'},$$

where  $\lambda' \geq \lambda \geq 0$ ,  $m_\lambda = \gamma_\lambda \varphi_\lambda$  and  $0 \leq i \leq 2^k$ . The operator

$$M = M_{\lambda'}^{\lambda} = \bigwedge_k M_k(\lambda, \lambda'),$$

is a morphological filter called an *alternating sequential filter* of primitives  $\gamma$  and  $\varphi$  and with bounds  $\lambda$  and  $\lambda'$ .

*Remarks*

- (i)  $\{\gamma_\lambda\}$  is a size distribution and  $\{\varphi_\lambda\}$  is an anti-size distribution (see section 2.4.5.). The two families are chosen independently of each other and hence  $\gamma_\lambda$  and  $\varphi_\lambda$  are not necessarily duals;
- (ii) The ASF  $M$  is  $\downarrow$ -continuous (Serra, 1988, p. 206) (see Theorem 3);
- (iii) For  $\lambda > \mu \Rightarrow m_\lambda m_\mu \prec m_\lambda$  and  $m_\lambda \prec m_\mu m_\lambda$ .

*proof*

$\gamma_\mu \prec I$  because any opening is anti-extensive

$$\Rightarrow \gamma_\mu \varphi_\mu \prec \varphi_\mu$$

$$\Rightarrow \varphi_\lambda \gamma_\mu \varphi_\mu \prec \varphi_\lambda \varphi_\mu \text{ because } \varphi_\lambda \text{ is increasing}$$

$$\Rightarrow \varphi_\lambda \gamma_\mu \varphi_\mu \prec \varphi_\lambda \text{ because } \varphi_\lambda \varphi_\mu = \varphi_\lambda \text{ (property of anti-size distributions)}$$

$$\Rightarrow \gamma_\lambda \varphi_\lambda \gamma_\mu \varphi_\mu \prec \gamma_\lambda \varphi_\lambda \text{ because } \gamma_\lambda \text{ is increasing}$$

$$\therefore m_\lambda m_\mu \prec m_\lambda.$$

$I \prec \varphi_\mu$  because any closing is extensive

$$\begin{aligned}
&\Rightarrow \gamma_\mu \prec \gamma_\mu \phi_\mu \text{ because } \gamma_\mu \text{ is increasing} \\
&\Rightarrow \gamma_\mu \gamma_\lambda \prec \gamma_\mu \phi_\mu \gamma_\lambda \\
&\Rightarrow \gamma_\lambda \prec \gamma_\mu \phi_\mu \gamma_\lambda \text{ because } \gamma_\mu \gamma_\lambda = \gamma_\lambda \text{ (property of size distributions)} \\
&\Rightarrow \gamma_\lambda \phi_\lambda \prec \gamma_\mu \phi_\mu \gamma_\lambda \phi_\lambda \\
&\therefore m_\lambda \prec m_\mu m_\lambda.
\end{aligned}$$

It follows therefore that  $k' \geq k \Rightarrow M_{k'} \prec M_k$ . Hence for increasing  $k$ ,  $M_k$  defines a decreasing sequence of morphological filters; "therefore we naturally introduce the infimum w.r.t.  $k$  of the  $M_k$ " in the definition of  $M_\lambda^{\lambda'}$  (Serra, 1988, p. 205).

(iv) Using the relations in (iii) it is easy to show that  $M_k M_k = M_k$ ; i.e. idempotence.

(v) When  $\lambda' = \lambda$  then trivially  $M_\lambda^{\lambda'} = m_\lambda$ .

(vi) In the Euclidean context if the primitives  $\gamma$  and  $\phi$  are u.s.c. then the ASF  $M$  is also u.s.c. provided that  $M_k$  is the product of a finite number of factors  $m$  (Serra, 1988, p. 206). In addition "any division procedure performed on the segment  $[\lambda, \lambda']$  leads to the same ASF  $M_\lambda^{\lambda'}$  . . . provided that the set of indices associated with the procedure is dense on  $[\lambda, \lambda']$ " (Serra, 1988, p. 207).

#### Properties of ASFs (Serra, 1988, p. 208)

Provided the primitives  $\gamma_\lambda$  and  $\phi_\lambda$  are u.s.c. it follows that

(i) If  $\lambda'' \geq \lambda' \geq \lambda > 0$  then  $M_\lambda^{\lambda''} M_\lambda^{\lambda'} = M_\lambda^{\lambda''} M_\lambda^{\lambda'} = M_\lambda^{\lambda''} M_\lambda^{\lambda'} = M_\lambda^{\lambda''}$  (absorption laws);

The absorption laws are generally not commutative so that  $M_\lambda^{\lambda'} M_\lambda^{\lambda''} \neq M_\lambda^{\lambda''}$ ; it is only true that  $M_\lambda^{\lambda''} \prec M_\lambda^{\lambda'} M_\lambda^{\lambda''}$ ;

(ii) In general the product  $M_{\lambda_1}^{\lambda_2} M_{\lambda_3}^{\lambda_4}$  is not a morphological filter unless two of the four values of  $\lambda$  are fixed as in (i);

(iii)  $\lambda'' \geq \lambda' \geq \lambda > 0 \Rightarrow M_\lambda^{\lambda''} M_\lambda^{\lambda'} = M_\lambda^{\lambda''}$  (transitivity law);

(iv) For any  $\lambda$  between zero and  $\lambda'$ ,  $\gamma_\lambda \prec M_\lambda^{\lambda'} \prec \gamma_\lambda \phi_\lambda$ . Furthermore the ASF converges to  $\gamma_\lambda \phi_\lambda$  as  $\lambda \rightarrow \lambda'$  (Serra, 1988, p. 210).

### 3.7.1. Derivates of the ASF

Replacing  $m_\lambda$  by one of the other three elementary products  $n_\lambda = \phi_\lambda \gamma_\lambda$ ,  $r_\lambda = \phi_\lambda \gamma_\lambda \phi_\lambda$ , or  $s_\lambda = \gamma_\lambda \phi_\lambda \gamma_\lambda$  in the definition for  $M_k(\lambda, \lambda')$  (Definition 34) leads to the derivative morphological filters  $N_\lambda^{\lambda'}$ ,  $R_\lambda^{\lambda'}$ , and  $S_\lambda^{\lambda'}$  respectively. Serra (1988, p. 209) has shown that as  $k \rightarrow \infty$  the factors in the expressions for these derived filters can regrouped so as to reveal a direct relationship with  $M_\lambda^{\lambda'}$ . For instance as  $k \rightarrow \infty$  the filter  $N_\lambda^{\lambda'}$  converges to  $\phi_\lambda M_\lambda^{\lambda'} \gamma_\lambda$ . Consequently

$$N_\lambda^{\lambda'} = \phi_\lambda M_\lambda^{\lambda'} \gamma_\lambda, \quad R_\lambda^{\lambda'} = \phi_\lambda M_\lambda^{\lambda'}, \quad \text{and} \quad S_\lambda^{\lambda'} = M_\lambda^{\lambda'} \gamma_\lambda.$$

Other types of derived filters are obtained by reversing the order of the factors  $m_\lambda$  in  $M_k(\lambda, \lambda')$ ; i.e. let  $\lambda' \leq \lambda$  and hence the indices now decrease. This produces a new type of ASF with different properties to those of the ASF of Definition 34.  $M_k(\lambda, \lambda')$  no longer defines a decreasing sequence of morphological filters. However Serra (1988, p. 209) shows that if one begins with the factors  $n_\lambda$  then  $\lambda > \mu \Rightarrow n_\lambda \prec n_\lambda n_\mu$  and  $n_\mu n_\lambda \prec n_\lambda$  and so  $k' \geq k \Rightarrow N_{k'} \prec N_k$  (the derivation is similar to that described in the previous remarks). Definition 34 is now applicable and hence for  $0 < \lambda' \leq \lambda$ ,  $N_\lambda^{\lambda'}$  is an ASF. Furthermore the definition also extends to the derived filters  $M_\lambda^{\lambda'} = \gamma_\lambda \cdot N_\lambda^{\lambda'} \phi_\lambda$ ,  $R_\lambda^{\lambda'} = N_\lambda^{\lambda'} \phi_\lambda$ , and  $S_\lambda^{\lambda'} = \gamma_\lambda \cdot N_\lambda^{\lambda'}$  (as before). The properties accompanying Definition 34 still hold true; however the inequalities for  $\lambda$  must be reversed; i.e.  $\lambda'' \geq \lambda' \geq \lambda > 0$  becomes  $0 < \lambda'' \leq \lambda' \leq \lambda$ .

Consider now the composition  $M_\lambda^{\lambda'} M_\lambda^{\lambda''}$  of two u.s.c. ASFs when either  $\lambda$  or  $\lambda'$  is fixed; i.e. the product of two ASFs such that one is defined for increasing indices and the other for decreasing indices. Serra (1988, p. 210) shows that whilst it is only true that  $m_\lambda \prec M_0^\lambda M_\lambda^0$  (by convention he sets  $\lambda'=0$ ), the reverse composition  $\tilde{M}_\lambda = M_\lambda^0 M_0^\lambda$  is in fact idempotent and a morphological filter. Moreover  $\langle \tilde{M}_\lambda, \circ \rangle$  is a commutative semigroup (see the remarks following Theorem 4) of morphological filters. In summary:

**Theorem 12**

For the u.s.c. primitives  $\gamma_\lambda$  and  $\phi_\lambda$  the product  $\tilde{M}_\lambda = M_\lambda^{\lambda_0} M_{\lambda_0}^\lambda$ ,  $\lambda$  in  $[\lambda_0, \infty)$ , defines a commutative semigroup of morphological filters that satisfies the following law of composition:

$$\tilde{M}_\lambda \tilde{M}_\mu = \tilde{M}_{\sup(\lambda, \mu)}, \lambda, \mu \text{ in } [\lambda_0, \infty).$$

Moreover  $\tilde{M}_\lambda$  is also u.s.c.

### 3.7.2. Digital ASFs

For digitisation purposes consider the ASF  $M_\lambda^{\lambda'}$  for which one of the bounds is zero and the primitives are the digital openings and closings of Chapter 2. The digital versions of the ASFs  $M_\lambda^0$  and  $M_0^\lambda$  are the ASFs  $M_i = m_1 \dots m_i$  and  $M^i = m_i \dots m_1$  respectively. Following the notation of Serra (1988) the ASF  $M_i(\Delta) = m_1 m_{1+\Delta} m_{1+2\Delta} \dots m_i$  and  $M^i(\Delta) = m_i m_{i-\Delta} m_{i-2\Delta} \dots m_1$ .

*Examples:*

- $M_i(2) = m_1 m_3 \dots m_{i-2} m_i$  and  $M^i(2) = m_i m_{i-2} \dots m_3 m_1$   
 $\tilde{M}_7(2) = M_7(2) M^7(2)$   
 $= m_1 m_3 m_5 m_7 m_7 m_5 m_3 m_1$   
 $= m_1 m_3 m_5 m_7 m_5 m_3 m_1$   
 $= \gamma_1 \phi_1 \gamma_3 \phi_3 \gamma_5 \phi_5 \gamma_7 \phi_7 \gamma_5 \phi_5 \gamma_3 \phi_3 \gamma_1 \phi_1$
- $R_i(3) = N_i(3) \phi_i$  and  $R^i(3) = \phi_i M^i(3)$   
 $= n_1 n_4 \dots n_{i-3} n_i \phi_i$   $= \phi_i m_i m_{i-3} \dots m_4 m_1$

$$\begin{aligned}
\bar{R}_7(3) &= R_7(3)R^7(3) \\
&= n_1 n_4 n_7 \phi_7 \phi_7 m_7 m_4 m_1 \\
&= \phi_1 \gamma_1 \phi_4 \gamma_4 \phi_7 \gamma_7 (\phi_7 \phi_7) \gamma_7 \phi_7 \gamma_4 \phi_4 \gamma_1 \phi_1 \\
&= \phi_1 \gamma_1 \phi_4 \gamma_4 (\phi_7 \gamma_7 \phi_7 \gamma_7) \phi_7 \gamma_4 \phi_4 \gamma_1 \phi_1 \text{ (idempotence of closing)} \\
&= \phi_1 \gamma_1 \phi_4 \gamma_4 \phi_7 \gamma_7 \phi_7 \gamma_4 \phi_4 \gamma_1 \phi_1 \text{ (idempotence of open-closing)}
\end{aligned}$$

### Remarks

- (i) Sternberg's iterative filters have the form

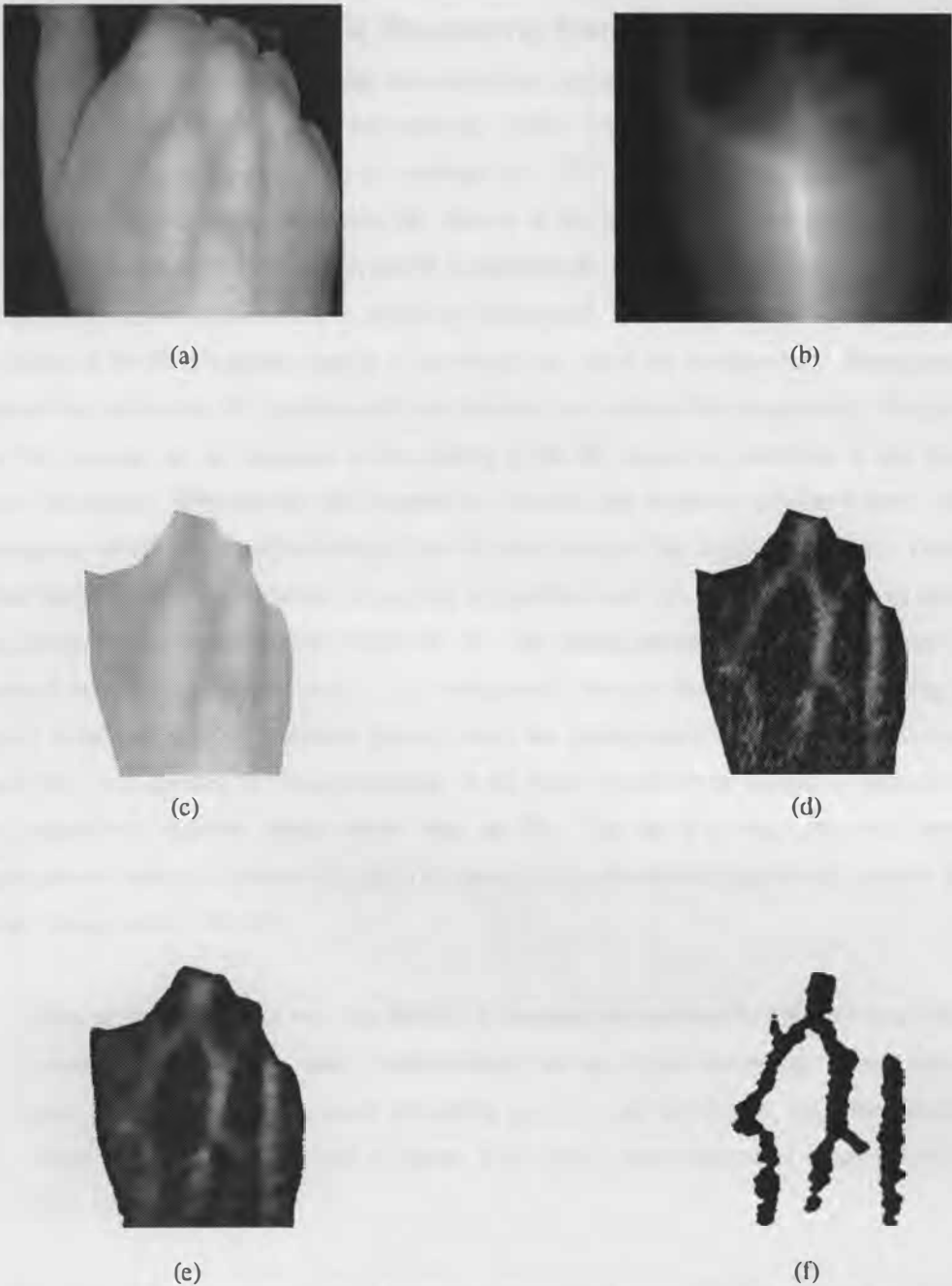
$$M^i(1) = \gamma_i \phi_i \dots \gamma_2 \phi_2 \gamma_1 \phi_1 \quad \text{and} \quad N^i(1) = \phi_i \gamma_i \dots \phi_2 \gamma_2 \phi_1 \gamma_1$$

where  $\gamma_i$  is an opening by sphere of radius  $i$  and  $\phi_i$  is a closing by a sphere of radius  $i$ .

- (ii) Schonfeld and Goutsias (1991, p. 22) have developed a procedure for choosing the optimal parameterisation of an ASF for removing noise from a corrupted binary image.

### Example

Mehnert, Cross, and Smith (1993) employ a simple ASF in an algorithm designed to extract (segment) the subcutaneous vascular network from a digitised thermograph of the back of a hand. The first stage of the algorithm (see Figure 20) involves clipping the original image (to isolate the region on the back of the hand containing the venous network) and normalising its background (using the tophat transform). The second stage of the algorithm employs the ASF  $M^i(1) = \gamma_i \phi_i \dots \gamma_2 \phi_2 \gamma_1 \phi_1$ , where  $\gamma_i$  is an opening and  $\phi_i$  is a closing by a sphere of radius  $i$ , to eliminate noise and artefacts including hair, skin pores, and pigmentation. The final stage of the algorithm involves thresholding the ASF filtered image to obtain a binary image of the vascular network.



**Figure 20.** Segmentation of the vascular network on the back of the hand.

- (a) Digitised thermograph of the back of a hand.
- (b) Distance map (measuring distance from a foreground point, i.e. part of the hand, to the nearest background point) of (a).
- (c) Image (a) clipped according to distance.
- (d) Tophat transform of (c).
- (e) The result after applying a close-opening alternating sequential filter (ASF), employing spherical structuring elements, to the image in (d).
- (f) Grey-level thresholding of (e).

### 3.8. Multiple Structuring Element Filters

Though "morphological openings were originally devised for size distributions . . . at first their main use was in image cleaning [i.e. noise removal]" (Serra, 1988, p. 101). A geometric interpretation can be given to the opening or closing of an image by a SE. For binary images the opening can be visualised as the sliding of the SE around the interior of the image (i.e. within foreground features). Those parts of the foreground into which the SE is unable to fit, e.g. sharp corners and particles smaller in size than the SE, are removed (i.e. replaced by background). Similarly the closing can be visualised as the sliding of the SE around the exterior of the image (i.e. within the background). Those parts of the background into which the SE is unable to fit are removed (i.e. replaced by foreground). For grey-scale images the opening can be visualised as the sliding of the SE along the underside of the brightness surface of the image. Wherever the SE is unable to penetrate, the surface is smoothed over. Similarly the closing can be visualised as the sliding of the SE over the top of the brightness surface. Once again wherever the SE is unable to penetrate, the surface is smoothed over. In summary, for binary images the opening removes features too small to contain the SE. The closing performs in exactly the same way but acts instead on the image complement; i.e. the background. For grey-scale images the opening has the propensity to remove positive structures (peaks) whilst the closing tends to remove negative structures (pits and ruts). The opening or closing employed in the tophat transform for instance, suppresses all the positive respectively negative details smaller than the SE. The use of a single structural opening or closing to remove noise is effective only when the noise to be attenuated is significantly smaller than the important image details. However

when an image contains very fine detail it is necessary to suppress the noise without blurring or removing the fine image detail. Unfortunately, the use of one structuring element will seldom meet this requirement. A small structuring element will not remove the noise effectively, a larger structuring element will eliminate fine details along with noise (Song, Stevenson, & Delp, 1990, p. 68).

If the final goal of the image processing is pattern recognition then this loss of detail is intolerable. Song and Delp (1990, p. 308) state that "combinations of morphological operators with other set operations as well as the utilization [sic] of multiple structuring elements can enhance the performance of morphological-based filters". Sternberg's iterative filter is one of the earliest examples of multiple-SE filtering. Stevenson and Arce (1986) introduced the 2D CO filter consisting of "a combination of openings and closing [sic] with one dimensional lines as its structuring elements" (p. 358). Specifically the 2D CO filter of order  $n$  is defined:

$$\psi^n(A) = \bigcup_{B_j^n \in J^n} \left( \bigcap_{B_i^n \in J^n} \mathcal{C}(A, B_i^n) \right) \cdot B_j^n \text{ for binary images, and}$$

$$\psi^n(f)(x) = \max_{B_j^n \in J^n} \left( \left( \min_{B_i^n \in J^n} \mathcal{C}(f, B_i^n) \right), B_j^n \right)(x) \text{ for grey-scale images,}$$

where  $A \in \mathcal{P}(\mathbb{Z}^2)$ ,  $f: \mathbb{Z}^2 \rightarrow \bar{\mathbb{Z}}$ , and the  $B_i^n \in \mathcal{P}(\mathbb{Z}^2)$  are the linear SEs, comprising  $n+1$  adjacent and identically valued points, depicted in Figure 21.

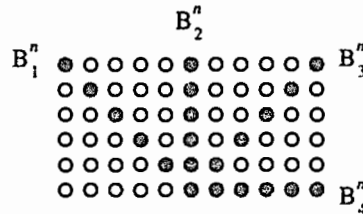


Figure 21. The SEs (for the square grid) used by the 2D CO filter.

Stevenson and Arce (1986, 1987) have analytically derived the output density function for the first-order 2D CO filter and have used Monte Carlo trials to derive output distributions for higher order 2D CO filters. The authors have compared the 2D CO filter with equivalent-order median-type filters and have concluded that it does "not have as much noise suppression as any of the median-type filters . . . examined [square median, cross median, max/median]" (1987, p. 1304). However "since the structure [i.e. edges and geometry] preserving characteristics of the 2D CO filter are much better than any of the other filters, it is possible to use higher order 2D CO filters, to obtain higher noise suppression, without significant structure loss" (1987, p. 1304).

Recall that the supremum of several openings is yet another opening and that the infimum of several closings is yet another closing (Theorem 5). The 2D CO filter thus has the form  $\gamma\phi$  (see section 3.4.1.). The filter generalises to what shall be called the *generalised CO filter* (Figure 22). From Theorem 5 it is evident that a generalised OC filter can also be defined (Figure 23). Note that Stevenson and Arce (1986, 1987) define only a 1D OC filter. Even a later paper by Song, Stevenson, and Delp (1990) fails to acknowledge let alone define a 2D OC filter. The generalised OC and CO filters are defined as follows:



**Definition 35. The generalised OC and CO filters.**

Let  $\{\gamma_i\} \subset \mathcal{O}$  be an indexed family of openings and  $\{\varphi_j\} \subset \mathcal{C}$  be an indexed family of closings, where  $i, j \in \mathbf{I}$ . Then

- (i)  $\zeta(X) = \bigwedge_j \varphi_j \left( \bigvee_i \gamma_i(X) \right)$  is a morphological filter called the *generalised OC filter*, and
- (ii)  $\psi(X) = \bigvee_i \gamma_i \left( \bigwedge_j \varphi_j(X) \right)$  is a morphological filter called the *generalised CO filter*,

where  $X \in \mathcal{L}$ .

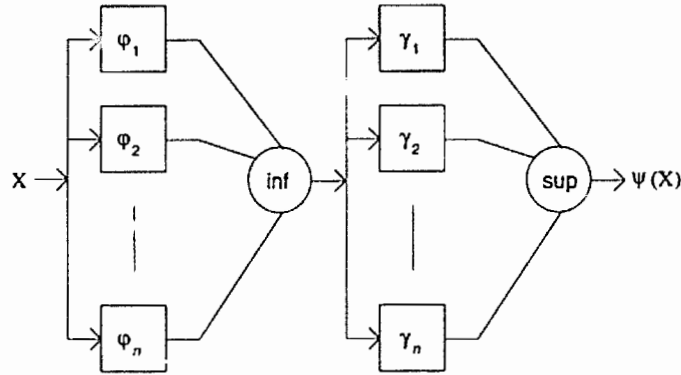


Figure 22. The generalised CO filter.

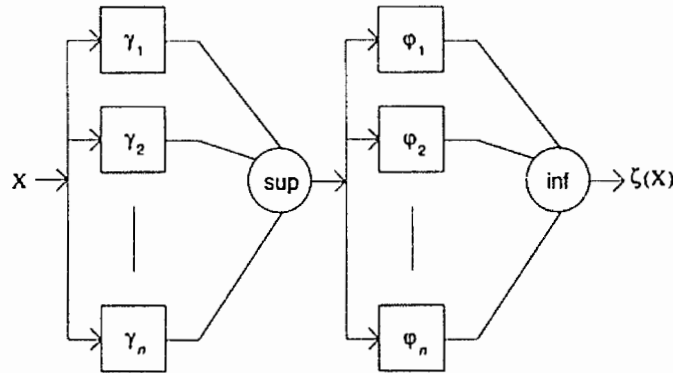


Figure 23. The generalised OC filter.

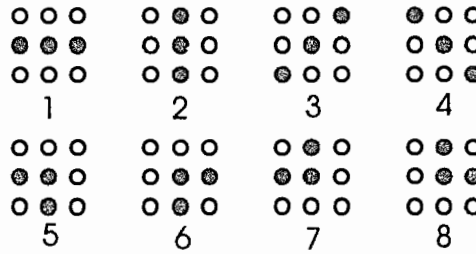
A generalised OC filter is used in the segmentation algorithm detailed in the Alzheimer's disease case study of Chapter 6.

Several other filters introduced by Song and Delp (1990) are just particular cases of Theorem 5, or the generalised OC filter. For example they have proposed the following two filters for noise reduction:

$$\psi(A) = \bigcup_i \mathcal{O}(A, K_i), \quad \zeta(A) = \bigcap_i \mathcal{C}(A, K_i), \quad \text{and the composition } \zeta \circ \psi \text{ for binary images, and}$$

$$\Psi(f) = \max_i \mathcal{O}(f, K_i), \quad Z(f) = \min_i \mathcal{C}(f, K_i), \quad \text{and the composition } Z \circ \Psi \text{ for grey-scale images,}$$

where the  $\{K_i\}$  are the eight SEs:



### 3.9. Generalised Morphological Filters

Openings, closings, open-closings, close-openings, and the generalised OC and CO filters very effectively remove impulse noise. Impulse noise is characterised by "isolated clusters of pixels whose values are either much higher or lower than values of neighboring [sic] image pixels" (Song, Stevenson, and Delp, 1990, p. 68). When it comes to removing non-impulse noise these filters do not perform very well. Linear filters on the other hand are effective though they tend to blur edges and contours. Motivated by the desire to combine the noise attenuation properties of linear filters and the geometry preserving properties of morphological filters, Song and Delp (1989) devised the *generalised morphological filter*. The filter "consists of a cascade of two stages, each of which involves linear combinations of the outputs from one type of basic morphological operator using multiple structuring elements" (p. 992).

#### Definition 36. The generalised morphological filter

Given an image  $f: \mathbb{Z}^2 \rightarrow \bar{\mathbb{Z}}$ , a family of structuring elements  $\{g_1, g_2, \dots, g_n\}$  s.t.  $g_i: \mathbb{Z}^2 \rightarrow \bar{\mathbb{Z}}$ , and a set of real constants  $\{\alpha_1, \alpha_2, \dots, \alpha_n \mid \sum_{i=1}^n \alpha_i = 1\}$ , the output  $y$  from the first stage of the filter is given pointwise by

$$y(x) = \sum_{i=1}^n \alpha_i y_{(i)}(x), \quad \text{where } y_{(i)}(x) \text{ is } i\text{-th largest value of the set } \{\mathcal{C}(f, g_j)(x) \mid j = 1, 2, \dots, n\}.$$

For another set of real constants  $\{\beta_1, \beta_2, \dots, \beta_n \mid \sum_{i=1}^n \beta_i = 1\}$ , and the output  $y$  from the first stage, the output of the second stage of the filter is given pointwise by

$$z(x) = \sum_{i=1}^n \beta_i z_{(i)}(x), \text{ where } z_{(i)}(x) \text{ is } i\text{-th largest value of the set } \{ \theta(y, g_j)(x) \mid j = 1, 2, \dots, n \}.$$

A block diagram of the filter is shown in Figure 24.

#### Remarks

- (i) The generalised morphological filter is not a morphological filter because, in general, it is not idempotent.
- (ii) The 2D CO filter of Stevenson and Arce (1986, 1987) corresponds to the case where  $\alpha_i = \begin{cases} 1 & \text{for } i = n \\ 0 & \text{otherwise} \end{cases}$ ,  $\beta_i = \begin{cases} 1 & \text{for } i = 1 \\ 0 & \text{otherwise} \end{cases}$ . In this instance the filter is idempotent and is therefore a morphological filter.
- (iii) Setting  $\alpha_i = \beta_i = \frac{1}{n}$ ,  $\forall i$ , "an averaging version of the filter is obtained. Since the outputs of morphological operators contain structural information, the averaging process applied to these outputs will result in reduced blurring of geometrical image features" (Song, Stevenson, & Delp, 1990, p. 68).
- (iv) Another type of generalised morphological filter is obtained by changing the order of the two stages; i.e. performing openings in the first stage, and closings in the second. "The choice of the order of the two stages is problem dependent" (Song & Delp, 1989, p. 992). Naturally the 2D OC filter turns out to be a particular case of this new filter.
- (v) The result of an opening followed by a closing is generally not the same as that of a closing followed by an opening. Grey-scale bias exists in both these compositions as a result of the anti-extensivity and extensivity properties of the opening and the closing respectively. Consequently the generalised morphological filter is also biased. However, "since the sizes of the multiple structuring elements used are usually smaller than the single structuring element used in traditional morphological filters, the bias problem for the filters is not as significant as with the traditional [close-opening or open-closing] filter" (Song, Stevenson, & Delp, 1990, p. 68).

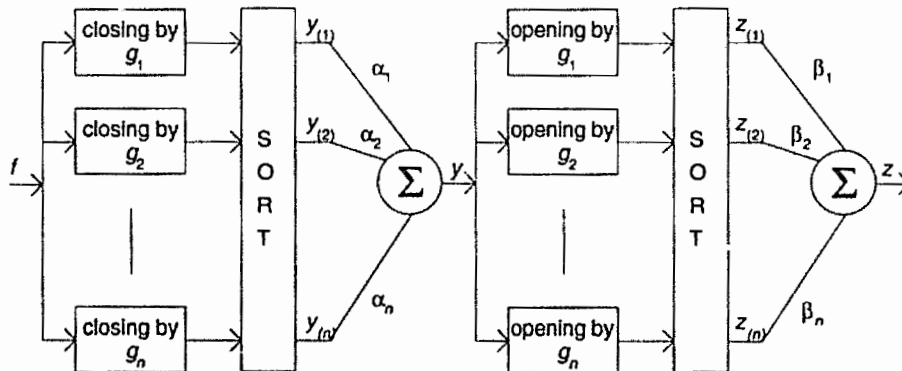


Figure 24. The generalised morphological filter.

### 3.10. Soft Morphological Filters

Koskinen et al. (1991) have modified the definitions for (discrete) FSP morphological operators to create a more general class of operators, called soft morphological operators, that includes the standard definitions as a special case. These new operators maintain most of the properties of their standard counterparts but "are less sensitive to additive noise and to small variations in the shape of the objects to be filtered" (p. 262). The behaviour of the soft operators can best be described as that of the standard operators where the usual SE has been replaced by a *soft* SE comprising a solid core and a soft or yielding boundary. "The key idea of soft morphological filters is that the structuring set is divided into two parts: the 'hard centre' which behaves like the ordinary structuring set and the 'soft boundary' where maximum or minimum are replaced by other order statistics" (Koskinen et al., 1991, p. 263). Soft dilation and erosion are defined as follows.

**Definition 37. Soft dilation and erosion.**

Let  $\diamond$  denote the repetition operator defined  $k \diamond \lambda = \underbrace{\lambda, \dots, \lambda}_{k\text{-times}}$ . For  $A, B \in \mathcal{P}(\mathbb{Z}^2)$ ,  $A \subseteq B$ , and  $f: \mathbb{Z}^2 \rightarrow \bar{\mathbb{Z}}$  soft dilation and erosion are defined pointwise:

soft dilation

$$\mathcal{D}(f, [B, A, k])(x) = k\text{-th largest value of } \{k \diamond f(y) \mid y \in \tilde{A}_x\} \cup \{f(z) \mid z \in (B - \tilde{A})_x\}$$

soft erosion

$$\mathcal{E}(f, [B, A, k])(x) = k\text{-th smallest value of } \{k \diamond f(y) \mid y \in A_x\} \cup \{f(z) \mid z \in (B - A)_x\}$$

where  $1 \leq k \leq \min\{|B|/2, |B - A|\}$ ,  $B - A$  denotes set difference (i.e. points belonging to  $B$  but not to  $A$ ), and  $|B|$  is the cardinality of  $B$ .

**Remarks**

- (i) When  $k=1$  and either  $A=B$  or  $A=\emptyset$  then the definitions for soft dilation and erosion are identically FSP dilation and erosion (see Relations (8) and (9) of Chapter 2) respectively; i.e.

$$\begin{aligned} \mathcal{D}(f, B) &= \mathcal{D}(f, [B, B, 1]) = \mathcal{D}(f, [B, \emptyset, 1]) \text{ and} \\ \mathcal{E}(f, B) &= \mathcal{E}(f, [B, B, 1]) = \mathcal{E}(f, [B, \emptyset, 1]). \end{aligned}$$

- (ii) Koskinen et al. (1991) actually defined soft dilation as follows:

$$\mathcal{D}(f, [B, A, k])(x) = k\text{-th largest value of } \{k \diamond f(y) \mid y \in A_x\} \cup \{f(z) \mid z \in (B - A)_x\}.$$

When  $k=1$  and either  $A=B$  or  $A=\emptyset$  this definition reduces to the Minkowski addition  $f \oplus \tilde{B}$ . It is this operator that Serra (1982) refers to as dilation by the SE  $B$ . Other authors, including Haralick et al. (1987), Giardina and Dougherty (1988), and Heijmans

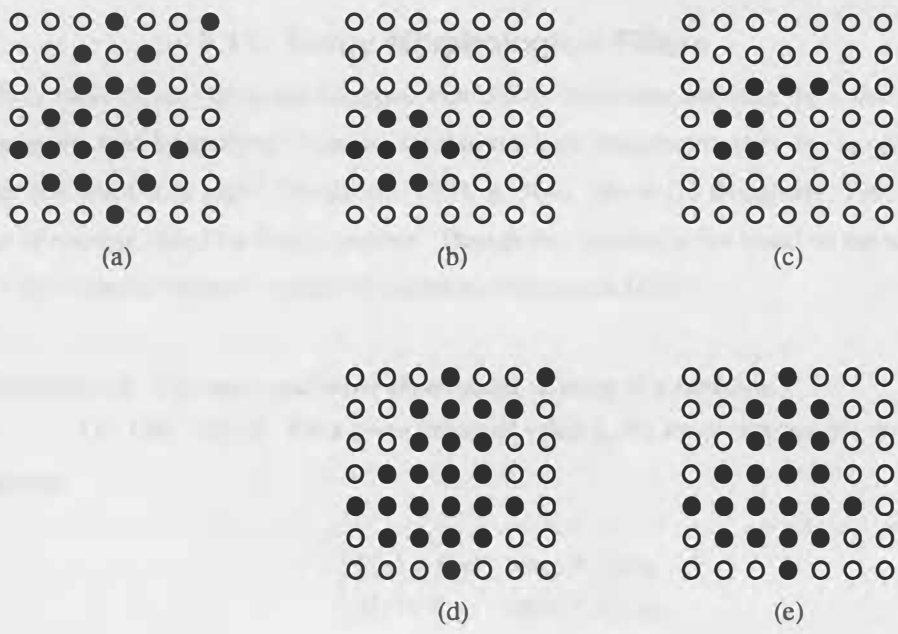
(1991) define the dilation of  $f$  by the SE  $B$  to be exactly the Minkowski addition  $f \oplus B$  (Relation (8)). Obviously for symmetric SEs  $B$ , i.e.  $B = \check{B}$ , the two definitions are equivalent. Note that Koskinen et al. (1991, p. 264) incorrectly state that soft dilation (their definition) by  $[B, A, k]$  is the morphological dual of soft erosion by  $[\check{B}, \check{A}, k]$ ; this is does hold true for the definition of dilation given in Definition 37.

### Properties

- (i) Like the standard dilation and erosion soft dilation and erosion are duals viz.  $\mathcal{D}(f, [B, A, k]) = -\mathcal{E}(-f, [\check{B}, \check{A}, k])$ .
- (ii) When  $A$  contains the origin soft dilation is extensive and soft erosion is anti-extensive and the following is true:  $\mathcal{E}(f, B) \leq \mathcal{E}(f, [B, A, k]) \leq \mathcal{D}(f, [B, A, k]) \leq \mathcal{D}(f, B)$  (Koskinen et al., 1991, p. 264).
- (iii) Soft opening and closing are defined in the usual way. Soft openings and closings are neither extensive nor anti-extensive; in this respect they are more akin to open-closings and close-openings (p. 265).
- (iv) The soft operators are increasing and translation-invariant (p. 264).
- (v) Soft openings and closings are less sensitive to impulse noise than the standard structural openings and closings (p. 266).
- (vi) Soft openings and closings are not in general idempotent (p. 268).

### Examples

- (i) Figure 25 illustrates the difference between soft and structural opening and closing respectively. It is the example given by Koskinen et al. (1991, p. 270) but with a minor correction. In their example the authors correctly derived the soft opening and closing, and the structural opening, but incorrectly derived the structural closing.
- (ii) Figure 26 compares the result of a standard close-opening with that of a soft close-opening when applied to an image corrupted with additive noise. The reduced sensitivity of the soft operators to impulse noise leads to improved noise attenuation.



$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}_{1,1} \quad A = [1]_{0,0}$$

Figure 25. A comparison of soft opening and closing with SP opening and closing respectively. (a) Original image  $X$ . (b)  $\mathcal{O}(X, B)$ . (c)  $\mathcal{O}(X, [B, A, 2])$ . (d)  $\mathcal{C}(X, B)$ . (e)  $\mathcal{C}(X, [B, A, 2])$ .

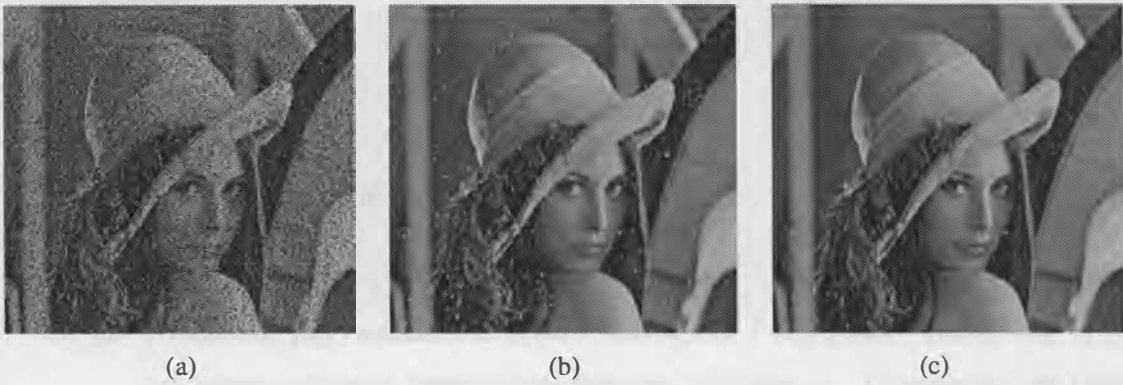


Figure 26. Soft openings and closings are less sensitive to impulse noise than FSP openings and closings. (a) An image  $f$  corrupted with additive noise. (b)  $\mathcal{O}(\mathcal{C}(f, B), B)$ . (c)  $\mathcal{O}(\mathcal{C}(f, [B, A, 2]), [B, A, 2])$ .

The structuring elements  $A$  and  $B$  are those of Figure 25.

### 3.11. Dolby Morphological Filters

Many audio cassette decks are equipped with Dolby circuits that attenuate the sound reproduced from the magnetic tape by applying "a severe filter to the high frequencies when the amplitude is low and a weak one when it is high" (Dougherty, 1993, p. 509). Serra (in Dougherty, 1993) defined a special type of opening called the Dolby opening. Though this opening is not based on the actual Dolby algorithm it has a similar purpose – hence the adoption of the name Dolby.

**Definition 38.** *The upper and lower thresholded versions of a function.*

Let  $f: \mathbb{R}^n \rightarrow [0, 1]$ . For a given threshold value  $t_0$  the lower-thresholded version of  $f$  is defined

$$\begin{cases} f_l(x) = f(x) & \text{when } f(x) \geq t_0 \\ f_l(x) = 0 & \text{when } f(x) < t_0, \end{cases}$$

and the upper-thresholded version of  $f$  is defined

$$\begin{cases} f_u(x) = 1 & \text{when } f(x) > t_0 \\ f_u(x) = f(x) & \text{when } f(x) \leq t_0. \end{cases}$$

**Definition 39.** *Dolby opening.*

Let  $f: \mathbb{R}^n \rightarrow [0, 1]$  and  $f_l$  denote its lower-thresholded version. Also let  $\gamma_1$  and  $\gamma_2$  be openings by flat SEs such that  $\gamma_1 \prec \gamma_2$  (i.e. the former is more severe than the latter). The operator

$$\gamma(f) = \gamma_1(f) \vee \gamma_2(f_l)$$

is an opening (Theorem 5) called a Dolby opening.

*Remarks*

- (i) The opening  $\gamma_1(f)$  attenuates the positive structures of  $f$  (e.g. peaks). The opening  $\gamma_2(f_l)$  less severely attenuates those positive structures of  $f$  above  $t_0$ . Consequently the Dolby opening (i.e. the maximum of the two aforementioned openings) more severely attenuates values below  $t_0$  than those above. In other words detail darker than  $t_0$  is removed whilst detail lighter than  $t_0$  (and possessing the same geometric characteristics) is retained.
- (ii) The Dolby opening and closing (following definition) commute with all anamorphoses  $\psi$  for which  $\psi(t_0) = t_0$  (Dougherty, 1993, p. 509).

**Definition 40. Dolby closing.**

Let  $f: \mathbb{R}^n \rightarrow [0,1]$  and  $f_u$  denote its upper-thresholded version. Also let  $\varphi_1$  and  $\varphi_2$  be closings by flat SEs such that  $\varphi_2 \prec \varphi_1$ . The operator

$$\varphi(f) = \varphi_1(f) \wedge \varphi_2(f_u)$$

is a closing called a Dolby closing.

*Remark*

The closing  $\varphi_1(f)$  attenuates negative structures of  $f$  (e.g. pits and ruts). The closing  $\varphi_2(f_u)$  less severely attenuates those negative structures of  $f$  below  $t_0$ . Consequently the Dolby closing (i.e. the minimum of the two aforementioned closings) more severely attenuates values above  $t_0$  than those below. Hence if  $\varphi$  modifies geometric detail lighter than  $t_0$ , the same detail below  $t_0$  will be modified to a lesser extent. An example of this type of closing is illustrated in Figure 27.

Two other variants of the thresholds defined in Definition 38 are:

$$\begin{cases} f'_l(x) = f(x) & \text{when } f(x) > t_0 \\ f'_l(x) = t_0 & \text{when } f(x) \leq t_0 \end{cases}, \text{ and}$$

$$\begin{cases} f'_u(x) = t_0 & \text{when } f(x) \leq t_0 \\ f'_u(x) = f(x) & \text{when } f(x) > t_0. \end{cases}$$

This gives rise to the opening  $\gamma'(f) = \gamma_1(f) \vee \gamma_2(f'_l)$ , and the closing  $\varphi'(f) = \varphi_1(f) \wedge \varphi_2(f'_u)$ . Openings and closings of this type act in an opposite manner to the Dolby openings and closings, respectively, already defined. "For example, the action of  $\varphi'$  is fine above  $t_0$  and coarse below" (Dougherty, 1993, p. 510).



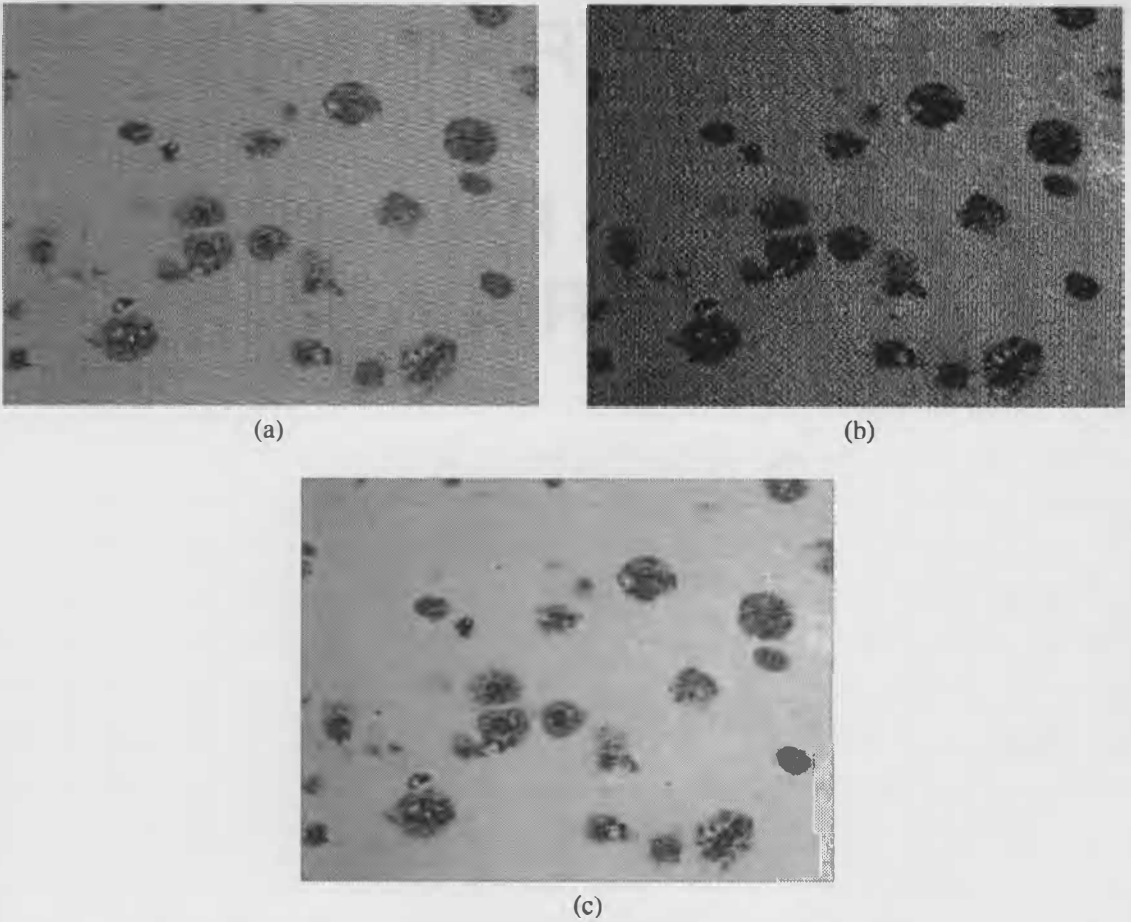


Figure 27. Dolby closing.

- (a) Original image  $f$  – a digitised light micrograph of post mortem brain tissue from an Alzheimer's patient (8-bit grey-scale). Electronic noise introduced during image acquisition is clearly evident.
- (b) The threshold set  $\{x | f(x) \leq t_0\}$  (black values) where  $t_0 = 120$ . This value was determined manually from the intensity histogram of  $f$ .
- (c) The Dolby closing  $\varphi = \min[C(f, B), C(f_u, C)]$ , where
- $$\begin{cases} f_u(x) = 255 & \text{when } f(x) > 120 \\ f_u(x) = f(x) & \text{when } f(x) \leq 120 \end{cases}$$
- $B$  is a disk of radius 3 pixels, and  $C$  is a disk of radius 1 pixel (see Figure 44).

---

## PART II.

# PLANAR DIGITAL MEASUREMENT

---

---

## CHAPTER 4.

# DIGITAL MORPHOMETRY

---

---

### 4.1. About This Chapter

The quantitative measurement of image features, digital morphometry, is in fact a two-step process: "geometrical transformations and then measurements" (Serra, 1986, p. 292). The goal of the first step is to partition the domain of the (grey-scale) image into subsets representing the features to be measured. This is classically referred to as *segmentation*. When the grey-levels of the objects to be measured are quite different from those present in the rest of the image, *thresholding* is used. Thresholding is the process of obtaining the threshold sets (see Footnote 3) of the image  $f$  for several chosen values of  $\alpha$ . Often these values are chosen manually using the histogram of the grey-levels in  $f$ . The minima of the histogram correspond to possible threshold values. In situations where the accurate location of minima proves difficult, the contrast of  $f$  can be improved by morphological filtering and/or the application of traditional radiometric and spatial enhancement techniques. Additional information can be used in conjunction with the grey-level histogram to assist with the choice of threshold values. For instance the grey-level histogram of the gradient of  $f$  (see section 6.4.4.) can be superimposed on the grey-level histogram of  $f$ . Another more elaborate technique, applicable when the histogram is bimodal (one peak representing the objects of interest and the other the background), involves choosing the threshold value for which the entropy (measure of information content) of the grey-level histograms for the foreground and background is maximised (Abutaleb, 1989, p. 23). "Unfortunately, all these methods fail when the same phase [feature] exhibits different [grey] levels at different places" (Serra, 1982, p. 458). One then has to resort to topological features of  $f$  – this leads to morphological segmentation using the *watershed transform* and homotopy modification (see Meyer & Beucher, 1990; and Dougherty, 1993).

After segmentation the connected components (objects or features) of the resulting binary image can be counted and measured. This chapter examines the elementary feature parameters (morphometrics) that can be derived using binary morphological operators. In particular the connectivity number, area, and perimetric measures are defined for images digitised on the square grid. We most naturally associate these quantities with convex shapes. Hence the chapter begins with an examination of the convex set as a model for isolated particles or objects within an image.

---

## 4.2. The Convex Set Model

It is important to realise that we do not "want to force convexity on natural objects (they are often too complicated)" (Serra, 1982, p. 93). We choose to work with the class of compact convex sets, called *ovoids*, because if  $X$  and  $B$  are ovoids then  $\lambda X$  for  $\lambda \in \mathbf{R}$ ,  $\lambda \cap B$ , and  $X \oplus B$  are ovoids. Indeed set convexity is preserved for all of the basic morphological operations: dilation, erosion, opening, and closing. Though in general the union of two ovoids does not yield another ovoid, their dilation does (Serra, 1982, p. 96). The failure of the union operator to uphold set convexity turns out to be a most useful property. The *convex ring*, i.e. the class of sets that can be decomposed into the union of a finite number of ovoids, is an archetype binary image (set) for random collections of particles.

### 4.2.1. Minkowski functionals

The morphometric quantities that can be associated with ovoids are termed *ovoid functionals*. A non-negative ovoid functional  $m: \mathbf{R}^n \rightarrow \mathbf{R}$  has the following properties:

- (i) *isometry invariance*:  $m(\tau(X)) = m(X)$  where  $\tau: \mathbf{R}^n \rightarrow \mathbf{R}^n$  is an isometry<sup>8</sup>,
- (ii) *increasing*: if  $X \subseteq Y$  then  $m(X) \leq m(Y)$ ,
- (iii) *C-additivity*:  $m(X) + m(Y) = m(X \cup Y) + m(X \cap Y)$ ,

where  $X$  and  $Y$  are ovoids in  $\mathbf{R}^n$ .

Hadwiger (1957) showed that all ovoid functionals can in fact be written as linear combinations of only a small subset of them called *Minkowski functionals* (cited in Serra, 1982, p. 102). Every ovoid  $X \in \mathbf{R}^n$  has  $n+1$  Minkowski functionals. The  $i$ -th order Minkowski functional of  $\mathbf{R}^n$  is denoted  $W_i^{(n)}$ . The functionals are defined according to a recurrence relation on sub-dimensions of the space (see Appendix C). Table 2 lists the morphometrics generated by the Minkowski functionals for  $\mathbf{R}^0$  to  $\mathbf{R}^3$ .

---

<sup>8</sup> An isometry of  $\mathbf{R}^n$  is a distance preserving mapping of  $\mathbf{R}^n$  into itself. Translations, rotations, and reflections (in lines) are examples of isometries in  $\mathbf{R}^2$  (Allenby, 1983, p. 233).

Table 2. The morphometric quantities generated by the Minkowski functionals  $W_i^{(n)}$ .

Minkowski functional $W_i^{(n)}$ of order $i$	$n$ -dimensional space			
	0	1	2	3
0	$N^{(0)}(X)$	$L(X)$	$A(X)$	$V(X)$
1		$N^{(1)}(X)$	$U(X)$	$S(X)$
2			$N^{(2)}(X)$	$M(X)$
3				$N^{(3)}(X)$

$N$  denotes connectivity number,  $L$  length,  $A$  area,  $U$  perimeter,  $V$  volume,  $S$  surface area, and  $M$  norm.

### 4.3. The Hit-or-Miss Transform

All of the algorithms, transformations, and feature parameters (morphometrics) stemming from binary mathematical morphology can be traced back to a single ancestor: the *hit-or-miss* transform of Serra (1982). The following definition characterises the transform.

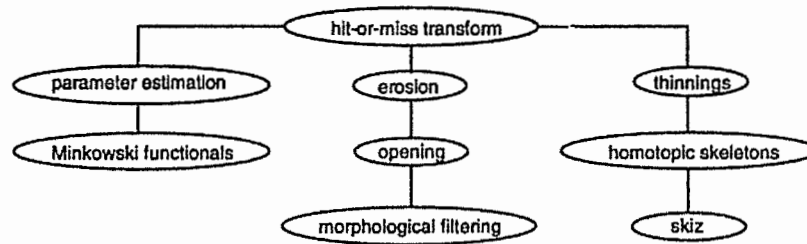
**Definition 41.** *The hit-or-miss transform.*

Given  $A, B_1, B_2 \in \mathcal{P}(\mathbf{E}^n)$ , where  $\mathbf{E} = \mathbf{R}$  or  $\mathbf{E} = \mathbf{Z}$ , the hit-or-miss transform of  $A$  by the disjoint structuring elements  $B_1$  and  $B_2$  is defined

$$HMT(A; B_1, B_2) = \mathcal{E}(A, B_1) \cap \mathcal{E}(A^c, B_2).$$

#### Remarks

- (i) Recall that  $\mathcal{E}(A, B) = \{x | B_x \subseteq A\}$  (see remarks following Definition 20). Consequently the  $HMT(A; B_1, B_2)$  is the set of all pixels  $\{x\}$  such that the translates  $(B_1)_x$  are subsets of  $A$  (i.e. they hit  $A$ ) and the translates  $(B_2)_x$  are subsets of  $A^c$  (i.e. they miss  $A$ ).
- (ii) Serra (1982) introduced the transform as the *hit-or-miss* transform though perhaps it should have been called more appropriately the *hit-and-miss* transform; indeed Giardina and Dougherty (1988) refer to it as such.



*Figure 28.* The hit-or-miss transform is the common ancestor of all of the algorithms and criteria of mathematical morphology. The dual operators are not shown (adapted from Serra, 1986, p. 290).

As Figure 28 shows, the definition of the hit-or-miss transform leads directly to sequential thinning algorithms, the erosion operation, and parameter estimation. The thinning of  $A$  by the disjoint structuring elements  $B_1$  and  $B_2$  is defined:  $THIN(A; B_1, B_2) = A \cap (HMT(A; B_1, B_2))^c$ ; i.e. the set difference between the original image and its hit-or-miss transform. Sequential thinning using a sequence of pairs of disjoint structuring elements can be used to obtain homotopic (connectivity relations are preserved) *skeletons* (see Mehnert, 1990, p. 86). The erosion operation is a special case of the hit-or-miss transform; the case when  $B_2$  is the empty set, i.e.  $\mathcal{E}(A, B) = HMT(A; B, \emptyset)$ . This follows from the fact that the empty set is a subset of every set and hence  $\mathcal{E}(A, \emptyset) = \{x | \emptyset_x \subseteq A\} = E^n$ . By duality w.r.t. complementation, one obtains thickenings from thinnings and the dilation from the erosion. Finally the hit-or-miss transform leads to the estimation of the parameters defined by the Minkowski functionals.

#### 4.4. Connectivity Number

The number of connected components in a binary image constitutes the simplest morphometric quantity. Connected component labelling algorithms (e.g. Manohar & Ramapriyan, 1989) individually label each connected component with a unique value. Effectively this transforms the binary image into a grey-tone image such that each connected component of the original binary image is uniquely identified by a grey value. This not only yields the number of connected components but also makes it possible to deal with each component separately. Generally such labelling algorithms are iterative and are thus computationally expensive. Mehnert (1990, p. 142) introduced a two-pass algorithm; numeric labelling of 4- or 8-connected components and the construction of equivalence trees takes place in the first pass, and a relabelling (using the equivalences established in the first pass) of the components takes place in the second pass. Though this algorithm is relatively fast, it requires additional memory and overhead to support and maintain the equivalence trees. If a particle count is all that is required then it is more sensible to calculate the connectivity number (requires only a single pass).

**Proposition 14. Connectivity number**

The connectivity number  $N$  of the binary image  $A \subset \mathbb{Z}^2$  is given by

$$N(A) = n\left\{\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right\} + n\left\{\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right\} - n\left\{\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right\} \text{ for 4-connectivity, and}$$

$$N(A) = n\left\{\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right\} - n\left\{\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right\} \text{ for 8-connectivity,}$$

where  $n\{*\}$  denotes the number of configurations of type  $*$  in the image  $A$  (blank entry matches both 0 and 1).

These expressions (Serra, 1982, p. 201) derive from Euler's formula for finite planar graphs. A "graph is planar if it can be embedded in the plane; that is if it can be drawn in the plane so that edges intersect only at endpoints" (Townsend, 1987, p. 216). Euler's formula is

$$V - E + F = 1 + N$$

where  $V$  is the number of vertices,

$E$  is the number of edges,

$F$  is the number of faces (i.e. the number of regions the plane is divided into by the graph)

$N$  is the number of connected components.

A finite planar graph can be constructed for a binary digital image by joining foreground pixels (vertices) with edges to denote connectivity.

**Example**

Consider Figure 29. The original image, (a), represents either two 8-connected regions or five 4-connected regions. The connectivity graphs for these two interpretations are (b) and (c) respectively. For (b)  $V=38, E=57, F=22 \Rightarrow N=2$ , and for (c)  $V=38, E=42, F=10 \Rightarrow N=5$ .

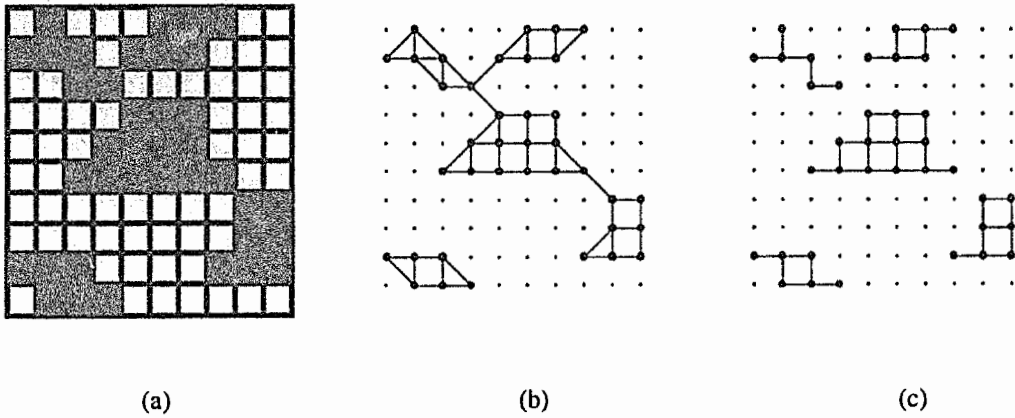


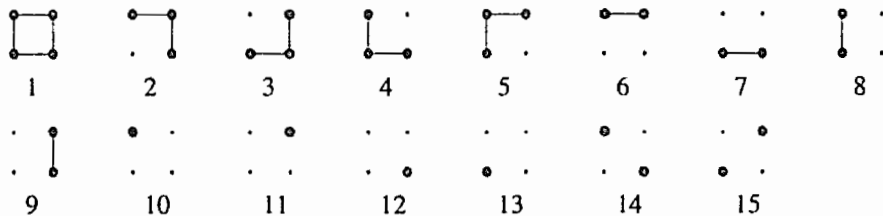
Figure 29. (a) Original image X; (b) 8-connected graph of X; (c) 4-connected graph of X.

*Proof of Proposition 14 (4-connected case)*

For a 4-connected binary image Euler's formula has the form

$$N(A) = \underbrace{n\{1\}}_V - \underbrace{\left\{ n\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + n\begin{Bmatrix} 1 & 1 \end{Bmatrix} \right\}}_E + \underbrace{n\begin{Bmatrix} 1 & 1 \\ 1 & 1 \end{Bmatrix}}_{F-1}.$$

The fifteen elementary configurations for a 4-connected graph are:



First, note that for configurations 1 to 13,  $V - E + F - 1 = 1$ . For configurations 14 and 15,  $V - E + F - 1 = 2$ . Now  $n\begin{Bmatrix} 1 & 0 \\ 0 & 0 \end{Bmatrix}$  will match configurations 1 to 4 and 6 to 14 exactly once, and configurations 5 and 15, twice. The overcount for configuration 5 can be corrected by subtracting  $n\begin{Bmatrix} 1 & 1 \\ 1 & 0 \end{Bmatrix}$  (which only matches configuration 5). The undercount for configuration 14 can be corrected by adding  $n\begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix}$  (which only matches configuration 14). Hence Serra's formula  $N(A) = n\begin{Bmatrix} 1 & 0 \\ 0 & 0 \end{Bmatrix} + n\begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix} - n\begin{Bmatrix} 1 & 1 \\ 1 & 0 \end{Bmatrix}$  is obtained. The proof for the 8-connected case is left to the reader.



Proposition 14 can be expressed in terms of the hit-or-miss transform as follows:

(i) for 4-connectivity,

$$N(A) = \left| HMT \left\{ A; \begin{bmatrix} 1 & * \\ * & * \end{bmatrix}_{0,0}, \begin{bmatrix} * & 1 \\ 1 & 1 \end{bmatrix}_{0,0} \right\} \right| + \left| HMT \left\{ A; \begin{bmatrix} 1 & * \\ * & 1 \end{bmatrix}_{0,0}, \begin{bmatrix} * & 1 \\ 1 & * \end{bmatrix}_{0,0} \right\} \right| - \left| HMT \left\{ A; \begin{bmatrix} 1 & 1 \\ 1 & * \end{bmatrix}_{0,0}, \begin{bmatrix} * & * \\ * & 1 \end{bmatrix}_{0,0} \right\} \right|$$

(ii) for 8-connectivity,

$$N(A) = \left| HMT \left\{ A; \begin{bmatrix} 1 & * \\ * & * \end{bmatrix}_{0,0}, \begin{bmatrix} * & 1 \\ 1 & 1 \end{bmatrix}_{0,0} \right\} \right| - \left| HMT \left\{ A; \begin{bmatrix} * & 1 \\ 1 & * \end{bmatrix}_{0,0}, \begin{bmatrix} * & * \\ * & 1 \end{bmatrix}_{0,0} \right\} \right|$$

where  $|\star|$  denotes the cardinality of  $\star$ ; i.e. the number of foreground pixels (1s) in  $\star$ .

#### 4.5. Area

The digitisation process can be visualised as the superimposition of a sampling grid on an image in  $\mathbb{R}^2$  and the subsequent sampling of intensities at grid points (Chapter 2). In essence, each grid point constitutes the centre of a square pixel in the digitised image. This implies therefore that a sampled pixel must necessarily have measurable area. For a square sampling grid with spacing  $a$  units between grid points (Figure 30), each pixel has area  $a^2$ . Therefore the area of an image  $A \subset \mathbb{Z}^2$  is given by the total number of pixels in  $A$  multiplied by  $a^2$ . In terms of the hit-or-miss transform this can be expressed as  $A = n\{1\} \cdot a^2 = \left| HMT \left\{ A; [1]_{0,0}, \emptyset \right\} \right| \cdot a^2$ . This estimator is unbiased (Serra, 1982, p. 220).

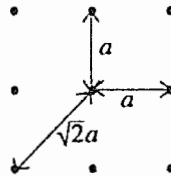


Figure 30. The spacing between direct and indirect neighbours on the square grid.

#### 4.6. Perimeter

Using results from integral geometry the Minkowski functionals of an ovoid can be related to either its projections or sections. The following result, due to Cauchy, shows that the perimeter of an ovoid  $X \subset \mathbb{R}^2$  is equal to  $\pi$  times its average projection length:

$$\frac{1}{\pi}U(X) = \frac{1}{2\pi} \int_0^{2\pi} L(X|\Delta(x, \alpha)) d\alpha \quad (\text{Cauchy relation})$$

where  $X|\Delta(x, \alpha)$  is the projection of  $X$  onto the straight line  $\Delta(x, \alpha)$  with direction  $\alpha$  and passing through some test point  $x$  (Serra, 1982, p. 105). An equivalent result, known as Crofton's formula, relates the perimeter of  $X$  to its sections:

$$\frac{1}{\pi}U(X) = \frac{1}{\pi} \int_0^\pi d\alpha \int_{-\infty}^{+\infty} N^{(1)}[X \cap \Delta(x, \alpha)] dx \quad (\text{Crofton's formula})$$

where  $N^{(1)}$  is the connectivity number and  $\Delta(x, \alpha)$  is a test line in direction  $\alpha$  and passing through a point  $x$ . Crofton's formula proves to be conducive to digital interpretation. The first step in obtaining a digital interpretation of Crofton's formula is to translate the notions of line segments and intercepts in  $\mathbb{R}^2$  to the space  $\mathbb{Z}^2$ .

#### 4.6.1. Line segments and intercepts in $\mathbb{Z}^2$

Let  $\vec{p}$  denote the position vector of the point  $p \in \mathbb{Z}^2$ . This vector is the line segment consisting of the origin, the point  $p$ , and all those (discrete) points on the line joining the origin to  $p$ . Similarly, given  $p, q \in \mathbb{Z}^2$ , the vector  $\vec{pq}$  is the line segment comprising  $p$  and  $q$ , and all those (discrete) points on the line joining  $p$  to  $q$ . If  $\vec{pq}$  consists of the points  $p$  and  $q$  only then it is a unit vector. In Crofton's formula the intercepts of  $X \cap \Delta(x, \alpha)$ , for a given  $\alpha$ , are all those line segments and isolated points resulting from the superposition of  $X$  on the line  $\Delta$  (see Figure 31).

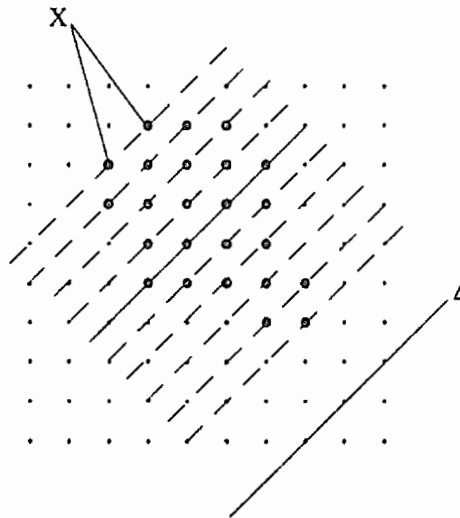


Figure 31. The 10 intercepts of  $X \cap \Delta$  for the line  $\Delta$  indicated.

### 4.6.2. Principal directions

In section 2.5.9, it was shown that the square grid admits only four rotations in  $\mathbf{Z}^2$  (i.e. successive rotations through  $90^\circ$ ) and hence there are only two cardinal directions. A more liberal interpretation of rotations leads to the three *principal directions* illustrated in Figure 32 (this is not the only possibility). The unit vectors  $\bar{\alpha}$  and  $\bar{\beta}$  constitute a basis for  $\mathbf{Z}^2$ . The unit vector  $\bar{\gamma}$ , as illustrated, is the linear combination  $\bar{\gamma} = -\bar{\alpha} - \bar{\beta}$ . The triple  $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$  defines a set of *principal directions* for  $\mathbf{Z}^2$ . Each pair  $(\bar{\alpha}, \bar{\beta})$ ,  $(\bar{\beta}, \bar{\gamma})$ , and  $(\bar{\alpha}, \bar{\gamma})$  is a basis for  $\mathbf{Z}^2$ . Moreover there can be no more than three principal directions. For instance if  $\bar{\delta} = \bar{\alpha} + \bar{\beta}$  were chosen as a possible fourth principal direction then  $(\bar{\gamma}, \bar{\delta})$  would not constitute a basis for  $\mathbf{Z}^2$  because the vectors are linearly dependent.

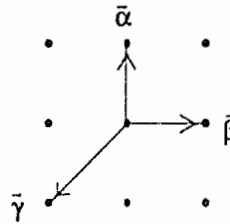


Figure 32. Principal directions  $\bar{\alpha}$ ,  $\bar{\beta}$  and  $\bar{\gamma}$  for the square grid.

### 4.6.3. A digital interpretation of Crofton's formula

Serra (1982, p. 221) proffers a digital interpretation of Crofton's formula for the hexagonal grid. The following is an interpretation for the square grid:

$$U(X) = \lim_{\substack{P \rightarrow \infty \\ k \rightarrow \infty}} \frac{\pi}{P} \sum_{p=1}^P a_k \left[ \eta_{\alpha_p} \{0 \ 1\} \right],$$

where  $P$  is the number of test directions,  $\alpha_p$  is the  $p$ -th test direction,  $\eta_{\alpha_p} \{*\}$  is the number of configurations of type  $*$  in the direction  $\alpha_p$ ,

$$a_k = \begin{cases} a.2^{-k} & \text{for test directions } \bar{\alpha} \text{ and } \bar{\beta} \\ \sqrt{2}.a.2^{-k} & \text{for the test direction } \bar{\gamma} \end{cases}$$

and  $a$  is the distance between direct neighbours of the square grid (see Figure 30).

#### Note

- (i) the  $\alpha_p$  must be uniformly distributed on the unit circle, and
- (ii) the  $\eta$  notation can be expressed using the hit-or-miss transform as before.

Restricting the  $\alpha_p$  to the principal directions  $\bar{\alpha}$ ,  $\bar{\beta}$  and  $\bar{\gamma}$ , it follows from above that

$$\begin{aligned} U^*(X) &= \frac{\pi}{3} \left[ a \left( n_{\alpha} \{0 \ 1\} + n_{\beta} \{0 \ 1\} \right) + \sqrt{2} a n_{\gamma} \{0 \ 1\} \right] \\ &= \frac{\pi}{3} \left[ a \left( n \{0 \ 1\} + n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right) + \sqrt{2} a n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right] \\ &= a \left[ n \{0 \ 1\} + n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right] + \sqrt{2} a n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}. \end{aligned}$$

A possible refinement to this estimate, suggested by Serra (1982, p.222), is to double the number of test directions by taking perpendiculars to  $\bar{\alpha}$ ,  $\bar{\beta}$  and  $\bar{\gamma}$ , viz.

$$\begin{aligned} U^{**}(X) &= \frac{\pi}{6} \left[ 2a \left( n \{0 \ 1\} + n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right) + \sqrt{2} a \left( n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + n \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \right) \right] \\ &= \frac{2\pi}{6} \left[ a \left( n \{0 \ 1\} + n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right) + \frac{a}{\sqrt{2}} \left( n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + n \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \right) \right] \\ &= a \left[ n \{0 \ 1\} + n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \right] + \frac{a}{\sqrt{2}} \left[ n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + n \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \right]. \end{aligned}$$

and to average  $U^*$  and  $U^{**}$ ; i.e.  $\frac{1}{2}(U^*(X) + U^{**}(X))$ . However "we cannot always be certain that the quality of estimation can be improved by adding  $U^{**}(X)$ ; the bias carried by  $U^{**}$  is worse than that of  $U^*$  (due to a larger elementary step)" (Serra, 1982, p. 222).

#### 4.7. Aspect Ratio Correction

When a rectangular sampling grid rather than a square grid is used to digitise an image it is necessary to adjust the area and perimetric formulae to accommodate the change in aspect ratio. The connectivity number formulae remain unchanged. If the intergrid spacing is as depicted in Figure 33 then an individual pixel in the digitised image is a rectangle with area  $ab$ . Consequently the area estimator of section 4.5. becomes  $A = n\{1\}.ab$ . The perimeter estimators  $U^*$  and  $U^{**}$  become

$$\begin{aligned} U^*(X) &= a n \{0 \ 1\} + b n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + \sqrt{a^2 + b^2} n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}, \\ U^{**}(X) &= a n \{0 \ 1\} + b n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + \frac{\sqrt{a^2 + b^2}}{2} \left[ n \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + n \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \right]. \end{aligned}$$

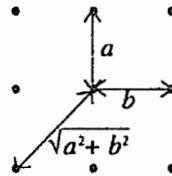


Figure 33. Intergrid spacing for the rectangular grid.

Directional bias is introduced into both the area and perimetric estimators because pixels are non-square. In particular, for the perimetric estimators the  $\alpha_p$  are no longer uniformly distributed on the unit circle.

---

## PART III.

# A DIGITAL IMAGE PROCESSING AND ANALYSIS LANGUAGE

---

---

# CHAPTER 5.

## D.I.M.P.A.L.

---

---

### 5.1. About This Chapter

DIMPAL is an acronym for *Digital Image Processing and Analysis Language*. DIMPAL was developed as a research tool for the Alzheimer's disease case study documented in the next chapter. Most of the images in the preceding chapters were produced using DIMPAL. This chapter describes DIMPAL, its construction, and grammar.

### 5.2. Introduction

DIMPAL fulfils the need for a PC-based image processing and analysis language suitable for researching and developing algorithms for a wide range of image processing applications. The typical shortcomings of commercially available PC-based image processing software are:

- an inability to fully exploit the 32-bit architecture of the PC's 80386 or 80486 microprocessor because of the underlying operating system, i.e. DOS , or DOS with Windows<sup>9</sup>;
- an inability to display multiple images and other graphics objects (e.g. histograms and intensity profiles) simultaneously;
- an inability to encode missing values needed to represent non-rectangular images;
- provision of only a limited set of binary and grey-scale morphological operators, e.g. only FSP morphological operators;
- limited or no support for multiple data types;
- limited or no support for user defined operations.

DIMPAL redresses each of these deficiencies.

DIMPAL is a general purpose image processing and analysis language. Strictly speaking it is a line interpreter capable of understanding and executing equation-like statements. Variables are used to represent images – or more precisely bound matrices (Appendix A). Functions are used to represent

---

<sup>9</sup> Windows is a trademark of Microsoft Corporation.

image operations such as histogram equalisation. DIMPAL provides a suite of functions for performing operations including  $m \times n$  window filtering, contrast stretching, connected component labelling and measurement, binary and grey-scale dilation and erosion, distance transformation, and skeletonisation (see Appendix D). Statements are entered at the keyboard and are executed one at a time. Alternatively a statement can be generated automatically by interacting with the *menu bar* and *dialogue boxes*. DIMPAL also accepts input from ASCII script files containing multiple statements. These files are created using a conventional text editor such as OS/2's system editor.

DIMPAL understands two types of statements: assignment statements and command statements. The right-hand side of an assignment statement, and the actual parameters of a function or command can be complex expressions consisting of variables, functions, constants, and arithmetic and logic operators. Assignment statements are used to create new variables or to overwrite old ones. DIMPAL offers a variety of commands (Appendix D) for displaying monochrome and RGB images, viewing image row and column intensity profiles, generating image histograms, calculating image statistics, creating GIF (CompuServe's *graphics interchange format*) and ILBM (Electronic Arts' *interleaved bitmap*) image files, and declaring, listing, merging, and discarding variables.

DIMPAL is written in ANSI C for IBM's Operating System/2 (OS/2) version 2.x. This operating system was chosen because it provides a flat linear 32-bit memory address space (4 gigabytes of virtual memory of which 512 megabytes are available for access by user applications), a 32-bit graphics engine, and pre-emptive multitasking. As DIMPAL is written in ANSI C it can be ported to other operating systems with relative ease; of course operating system dependent routines (e.g. image display functions) have to be rewritten. At the time of writing DIMPAL has been partially ported to AIX<sup>10</sup>.

DIMPAL comprises several distinct software components (objects). The relationship between each of these objects is illustrated in Figure 34.

- *console window* – the primary interface to DIMPAL – behaves like a teletype terminal (TTY) and accepts common ANSI TTY control sequences (see Petzold, 1989, p. 311);
- *file manager* – handles all image I/O between secondary storage and memory;
- *lexical analyser* – extracts tokens such as variable names, operators, and constants from user input;
- *parser* – processes both command and assignment statements (recursive descent parsing);
- *symbol table manager* – maintains a list of variable names and descriptions (implemented as a binary tree);

---

<sup>10</sup> ® AIX is a registered trademark of International Business Machines Corporation.



- *command dictionary* – list of commands and associated syntax;
- *function dictionary* – list of functions and associated syntax;
- *commands* – actual implementations of DIMPAL standard, and user defined commands (e.g. *display*);
- *functions* – actual implementations of DIMPAL standard, and user defined functions (e.g. *dilate*);

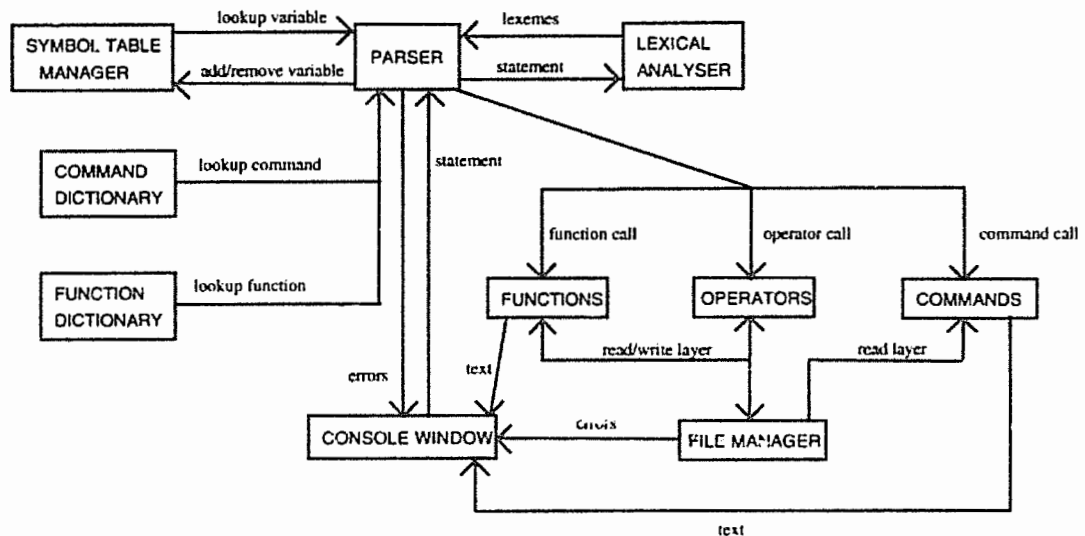


Figure 34. DIMPAL software components.

### 5.3. Lexical Conventions

A DIMPAL statement is fundamentally a collection of small syntactic units called *tokens*. In DIMPAL there are four types of tokens: identifiers, constants, operators, and separators. Particular instances of tokens are called *lexemes*. For example, *image* and *new\_picture* are instances of the token *identifier*. It is the task of the lexical analyser to resolve a statement into its constituent tokens. It ignores any spaces and tabs (with the exception of spaces between double quotes) in a statement, as well as anything following a semicolon (which is deemed to be a comment). A formal description of each type of token follows.

#### 5.3.1. Identifiers

An identifier is defined to be a letter followed by a mixed sequence of letters, numbers, and underscores (`_`). Upper and lower case letters constitute different letters. Some examples of valid identifiers are: *Plaques*, *Senile\_Plaques*, and *plaque\_23*. There are no restrictions on the length of an identifier. However the length of the identifier (name) for a variable is governed by the underlying file system because each DIMPAL variable is stored as a file with the same name as the variable and the

extension *.lay* (meaning layers). Consequently, under OS/2 variable names are limited to 8 characters for FAT (file allocation table) partitions, and 254 characters for HPFS (high performance file system) partitions.

### 5.3.2. Constants

An integer (constant) is defined to be a sequence of one or more digits (no sign). A real (constant) is defined to be an integer followed by a period followed by another integer. Alternatively a real constant is defined as an integer, optionally followed by a period and another integer, followed by the letter *e* (upper or lower case), followed by an integer (possibly signed). In extended Backus-Naur form (EBNF), a real constant has the production:

```
<real>::=<integer> . <integer> |
        <integer> [.<integer>] (E | e) [+ | -] <integer>
```

Some examples of real constants are: *123.2*, *5.2e-2*, and *17e+2*. A string (constant) is defined to be any sequence of alphanumeric characters surrounded by double quotes. The string may not contain the " character itself.

### 5.3.3. Operators

All of the DIMPAL operators, except *^*, are a subset of the *C* language operators. In DIMPAL the symbol *^* represents exponentiation. Table 3 lists the DIMPAL operators in order of decreasing precedence; e.g. when evaluating an expression DIMPAL will perform multiplication before addition. Parentheses can of course be used to override precedence.

Table 3. DIMPAL operators in decreasing order of precedence (top to bottom).

Operator	Associativity	Description
- !	right to left	unary minus, logical negation
* / %	left to right	multiplication, division, modulus
+ -	left to right	addition, subtraction
< > <= >=	left to right	relational operators
== !=	left to right	equal to, not equal to
^	left to right	exponentiation
&&	left to right	logical AND
	left to right	logical OR

### 5.3.4. Separators

When an identifier is followed by a left-hand round bracket it is deemed to be a function name. Double quotes delimit string constants. The comma separates actual parameters in functions and commands that require arguments. The semicolon marks the beginning of a comment.

## 5.4. Language Grammar

DIMPAL statements are parsed using a method known as *recursive descent parsing* (see Hunter, 1981; Cockshott, 1990). A collection of recursive procedures, which in total constitute the parser, determine the syntactic validity of a statement based upon the sequence of lexemes generated by the lexical analyser. For instance the sequence *result*, =, *image1*, +, *image2* determines a syntactically correct assignment statement. If the statement is also semantically correct (see section 5.6.) then it is evaluated as the recursion *unwinds*. DIMPAL statements can be described by a Class 2, also called context-free, grammar (see Cockshott, 1990, p. 53). Such grammars are amenable to representation in EBNF notation. Appendix E lists the production rules for a DIMPAL statement.

## 5.5. Variables

A variable is a collection of one or more bound matrices (Appendix A). The matrices can be of differing sizes and data types. Typically the bound matrices represent binary and grey-scale images sampled on the square grid; but they can be used to represent any spatial data arranged on a square grid. In DIMPAL these bound matrices are referred to as layers. The *describe* command prints the details of each layer of a variable including data type, size, and location in  $\mathbf{Z}^2$ . DIMPAL supports the following data types: *byte*, *boolean*, *integer*, *long*, *float*, and *double*. These data types reflect those of the C language. The data types *byte* and *boolean* are actually *unsigned char*. A variable of type *byte* can store values in the range [0,255]. A *boolean* variable can store only 0 and 1. The sizes of the other data types are implementation dependent. For example under OS/2 1.3, integers are stored as 16-bits whilst under OS/2 2.1 they are stored as 32-bits. Missing value sentinels are provided for integer, long, float, and double variables. This makes it possible to encode non-rectangular images such as disks and grey-level spheres.

During a DIMPAL session variables may be introduced in any of the following ways:

- *declaration* – a variable defined in a previous session may be declared to the system using the *declare* command, e.g. `declare("plaques.lay");`
- *importation* – several functions permit data stored in other formats to be imported into DIMPAL and assigned to a variable, e.g. `new_variable=ASCII_file("stelts.txt");`
- *assignment*  
e.g. `tophat=image[-]dilate(erode(image,disk(5)),disk(5)).`

The command *list* prints all of the variables known to DIMPAL. The bound matrices represented by a variable are stored in a file on disk (secondary storage); e.g. the variable *tophat* would be physically stored in the file *tophat.lay*. The symbol table manager maintains a binary tree of descriptors for each variable including associated file paths and names. A disadvantage of this arrangement is that files can be erased by other processes without DIMPAL knowing. There are two significant advantages however. The first is that data is not lost when a DIMPAL session is terminated. The second is that a layer of a variable is loaded into memory only when an operation or function is applied to it; this minimises memory usage (many DIMPAL functions load layers into memory one line at a time – hence very large images can be processed).

The layers (bound matrices) of a variable are sequentially numbered from 1 to *n*. In addition each layer has its own unique identifier. In an expression or function call, a single layer can be referenced by appending the variable name with a period and then the layer identifier or number.

#### *Example*

Suppose that *RGB\_picture* is a variable that has three layers: *red*, *green*, and *blue*. The command

```
describe( RGB_picture )
```

would then provide information about each of the layers. The command

```
describe( RGB_picture.2 ) or equivalently describe( RGB_picture.green )
```

would print information for only the second layer. Commands and functions deal with multiple layers in different ways. For instance, if the layers of *RGB\_picture* are all of type byte and have the same dimensions and spatial location, then the command

```
display( RGB_picture, colour_map( "grey" ) )
```

would display three separate grey-scale images. On the other hand the command

```
composite( RGB_picture )
```

would display only a single colour composite picture.

## 5.6. Expression Evaluation

A DIMPAL expression (i.e. <disjunction> – see Appendix E) is a combination of variables, layer references, functions, and arithmetic and logic operators. When DIMPAL evaluates an expression it evaluates functions and anything in parentheses first, and then performs operations in order of precedence – highest to lowest (see Table 3).

When applying operators to scalars (i.e. integer and real constants) the data type of the result is the highest (i.e. the most general) data type of all of the operands. For example, the result of `!3` is of type integer, and the result of `3 + 2.2` is of type double. When the result of an integer constant or the result of an integer expression exceeds the range of the integer data type then it is promoted to type long.

When applying operators to layers DIMPAL first determines the number of layers that will be produced. For the monadic operators `!` and `-` the number of layers produced is equal to the number of layers in the operand. For example, if `picture` is a variable with three layers then the result of `!picture` also has three layers. In the case of dyadic operators only the following combinations are permissible:

*n layers <operator> 1 layer*  
*1 layer <operator> n layers*  
*n layers <operator> n layers.*

In each case DIMPAL determines that the result also has *n* layers. For the first case, the second operand (single layer) is applied to each of the layers of the first operand in sequence thus producing an *n* layer result. For the second case, the first operand is applied to each of the layers of the second operand in sequence also producing an *n* layer result. For the last case, the operator is applied pairwise: *layer 1 <operator> layer 1*, *layer 2 <operator> layer 2*, and so on. Next DIMPAL determines the data types of the layers of the result. For logical operators the result always has layers of type boolean. For example if `image1` and `image2` are single layer variables then the expression `image1 < image2` yields a variable with a single layer of type boolean that contains 1s wherever `image1` is less than `image2` and 0s elsewhere. For arithmetic operators the data types of the layers of the result are determined as follows:

*unary minus*

boolean and byte layers are promoted to integer  
 all other types are preserved

*layer <operator> integer\_constant or integer\_constant <operator> layer*  
 result inherits the highest of the data types of *layer* and *integer\_constant*

*layer <operator> real\_constant or real\_constant <operator> layer*  
 result has type double

*layer <operator> layer*  
 result inherits the highest of the data types of the two layers

When two layers are added, multiplied etc., the operation is performed entry-wise. Hence multiplication is not matrix multiplication. A dyadic operation involving a scalar (i.e. a numeric constant or expression) and a layer implies that the scalar is applied to each element of the layer. For example if the variable *image* has one layer then the operation *image* + 3 \* 2 results in the value 6 being added to each element of *image*. Whenever a missing value is encountered the result is always a missing value.

An operation on a pair of layers is only valid if they have the same dimensions and spatial location. However DIMPAL can be forced to perform the operation on the intersection of the layers if the *clipping* operator is invoked. This is an operator that can be applied to dyadic operators and to functions. The clipped version of addition, for example, is denoted [+]. A clipped function has the syntax *[function](. . .)*. At the time of writing none of the DIMPAL functions (Appendix D) utilise clipping. An example of a function that might utilise clipping is a function that performs matrix multiplication.

---

## PART IV.

# ALZHEIMER'S DISEASE CASE STUDY

---

---

## CHAPTER 6.

# MATHEMATICAL MORPHOLOGY AS A TOOL FOR THE MORPHOMETRIC INVESTIGATION OF NEURITIC PLAQUES ASSOCIATED WITH ALZHEIMER'S DISEASE

---

---

### 6.1. About This Chapter

This chapter details an investigation of the application of mathematical morphology to the study of Alzheimer's disease. In particular an image processing algorithm is proffered for the automatic segmentation and measurement of neuritic plaques (microscopic lesions associated with Alzheimer's disease) from light micrographs of post mortem brain tissue. The chapter begins with an introduction to Alzheimer's disease and a description of the hallmark neuropathologic lesions. This is followed by a review of a previous investigation undertaken by Bartoo, Kim, Haralick, Nochlin, and Sumi (1988), into the application of mathematical morphology to the segmentation of these lesions from digitised light micrographs. Finally the proposed digital image processing algorithm is described and evaluated.

### 6.2. Alzheimer's Disease

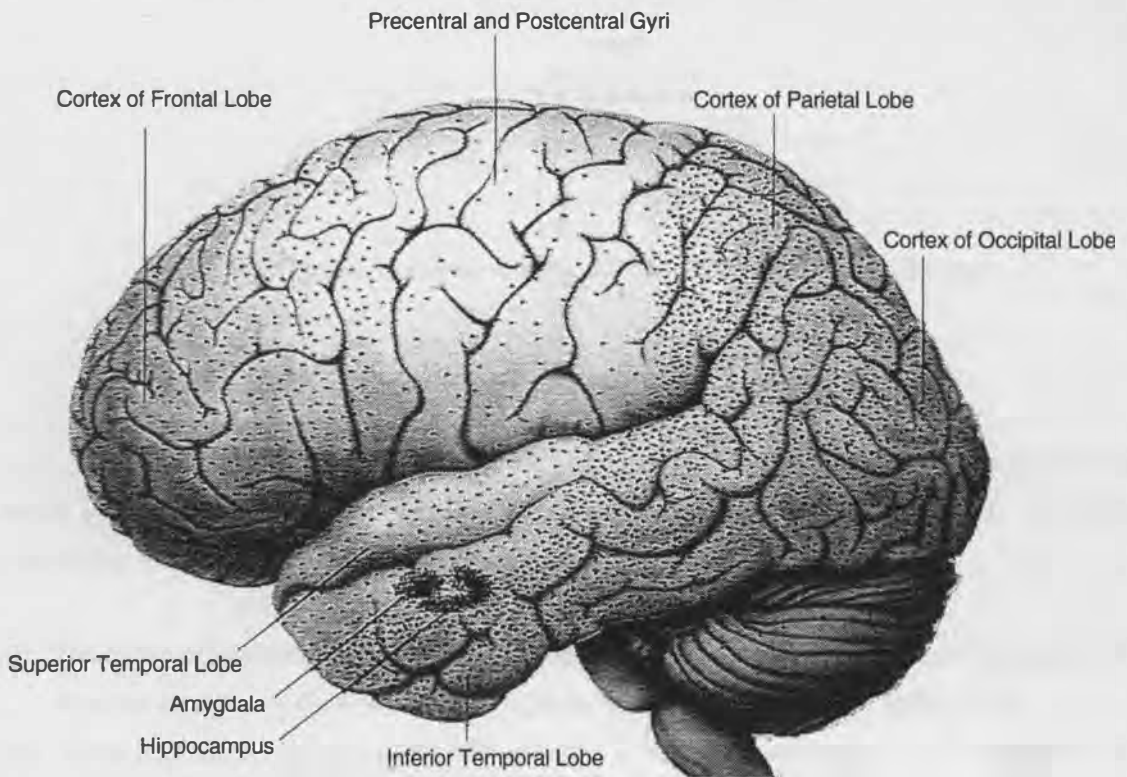
Special nerve cells called *neurones* "are the fundamental units or building blocks of the brain" (Kuffler, Nicholls, & Martin, 1984, p. 2). The human brain contains some  $10^{11}$  neurones. The body of a neurone is approximately pyramidal or spherical in shape. A single long fibre, called the *axon*, extends away from the cell body. Numerous smaller fibres, called *dendrites*, branch out from the surface of the neurone. If the axon is imagined to be the trunk of a tree, with the neurone sitting atop, then the dendrites constitute the branches. A single neurone transmits nerve impulses over the axon and receives them over the dendrites (Stevens, 1979, p. 15). In this way a neurone is able to communicate with other neurones. Special chemicals called *neurotransmitters* facilitate the transmission of these electrical signals across the intervening space between neurones. The neurotransmitter *acetylcholine* is known to be deficient in the brains of people with Alzheimer's disease (Forsythe, 1990, p. 32). Acetylcholine is thought to play a role in the formation of memory. The underlying process that manages the production of this neurotransmitter is very complex and is not yet completely understood.



Alzheimer's disease is a form of progressive dementia associated with the degeneration of the brain. Confirmation of a diagnosis of Alzheimer's disease can only be made post mortem. Microscopic neuropathologic lesions called *neurofibrillary tangles* (NFT) and *neuritic (senile) plaques* (NP) are found in abundance in the Alzheimer brain. These lesions were "first described by the Bavarian psychiatrist Alois Alzheimer in 1907—senile plaques and neurofibrillary tangles—are the most common basis for late-life dementia in many developed countries" (Selkoe, 1991, p. 40). NFT are accumulations of abnormal fibres (arranged as paired helical filaments) within affected neurones. NP are circular clusters of abnormal nerve tissue that replace normal neurones. NP contain "a protein fragment, approximately 40 amino acids long, referred to as the amyloid beta-protein [also referred to as the  $\beta$ A4 amyloid protein]" (Selkoe, 1991, p. 40). Both of these lesions occur primarily in the *cerebral cortex* and in the *hippocampus* of the brain (see Figure 35). Their presence is not indicative of the disease because they may also be found in normally aged brains in smaller numbers. However, "the number of NFT and NP is strongly [positively] correlated with the degree of dementia . . . and therefore may be of value as a quantitative measure of the dementia" (Adams & Jones, 1987, p. 51). "Most pathoanatomic studies of Alzheimer's disease have been done on advanced, severely demented cases, [and consequently] there is relatively little published information on the distribution of tangles or plaques in normal aging or in mild dementia" (Price, Davis, Morris, & White, 1991, p. 295). Current research is hampered by the enormous amount of human effort and time required to perform counts and measure the area and other morphometric quantities of these lesions. As a result, to date there have been relatively few quantitative studies. Clearly there is a need for an image analysis tool to automate the counting and measurement procedures.

### 6.3. Previous Research

Bartoo et al. (1988) investigated the application of mathematical morphology to the segmentation of NP and NFT associated with Alzheimer's disease. They dealt with coronal tissue sections of the hippocampus and amygdala (see Figure 35) stained with thioflavin-S. Thioflavin-S is a fluorescent stain that highlights NP, NFT, and blood vessels containing the  $\beta$ A4 amyloid protein. In addition the stain detects *lipofuscin* particles. These particles accumulate in the cytoplasm of neurones; their "size, complexity, and distribution within the neuronal cytoplasm are age-dependent" (Adams & Jones, 1987, p. 26). When viewed through a conventional light microscope the NP and NFT appear yellow, and the lipofuscin particles appear orange, against a dark green background. Bartoo et al. (1988) used a monochrome frame-grabber to capture images, at 200 $\times$  magnification, from a RS170 video camera mounted on an Olympus BH-2 microscope equipped with a fluorescent attachment (exciting wavelength of 490 nm and barrier filter at 455 nm). Digitised images had a spatial resolution of 512  $\times$  512 pixels and a grey-scale resolution of 8-bits (i.e. 256 shades of grey).



*Figure 35.* Brain of an Alzheimer patient. Neuritic plaques are indicated by stippling. (Adapted from Selkoe, 1991, p. 42.).

The NP and NFT detection algorithm developed by Bartoo et al. (1988) employs grey-level thresholding and binary morphological openings and closings. The algorithm is as follows:

- (i) Acquire a grey-tone image from the microscope.
- (ii) Globally threshold this image to produce a binary image. The threshold point is determined manually.
- (iii) Close the binary image with the *gamma* SE depicted in Figure 36. This operation joins together objects (particles) within a senile plaque to form one large region. The dimensions of the SE are chosen to minimise the number of occurrences of two plaques, in close proximity, being joined. Bartoo et al. stated that they had devised "a routine based on the Bayesian decision making method" (1988, p. 472) to determine the optimal dimensions of the *gamma* SE for each image – no details were given.
- (iv) Perform a closing by a disk of radius 2 to fill in any remaining holes and gaps.
- (v) Perform an opening by a disk of radius 7 to remove all objects "too small" to be NP. This leaves only the NP.
- (vi) Subtract (set difference) the image containing the NP from the original binary image in (ii).
- (vii) Open this new image by a suitably sized disk to remove all objects "too small" to be NFT. This leaves only the NFT.

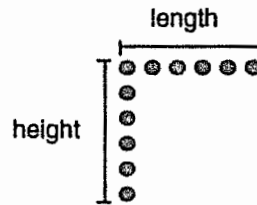


Figure 36. Gamma SE employed in the NP and NFT segmentation algorithm devised by Bartoo et al. (1988).

Bartoo et al. (1988) compared the results of their algorithm with independent manual counts of senile plaques in fifty images. The algorithm achieved 81% correlation. They gave no similar comparison for NFT detection. The algorithm has several major drawbacks:

- (i) The choice of an initial threshold value is determined manually. This value is critical because "the detection algorithm is fairly sensitive to the binary threshold" (Bartoo et al., 1988, p. 474).
- (ii) "There is overlap in the sizes of the NFT and SP [i.e. NP]. . . . [such that] a large globoid NFT can be mistaken for a very small SP" (Bartoo et al., 1988, p. 474).
- (iii) As a consequence of using a monochrome camera, large accumulations of lipofuscin particles in normal neurones can be mistaken for a NFT. (Recall that lipofuscin particles stain orange whilst NP and NFT stain yellow).

Though Bartoo et al. stated that "further improvements such as local adaptive thresholding, border count correction, and perhaps different staining techniques are currently being investigated" (1988, p. 475), they do not appear to have published any further results.

## 6.4. The Investigation

The approach taken by Bartoo et al. (1988) is fundamentally flawed because of their choice of stain. Thioflavin-S is a non-specific stain that highlights features in addition to NP and NFT. The more conventional Bielschowsky silver method is also non-specific. Although the method is very successful in staining NP and NFT (Price et al., 1991, p. 305) it may also stain normal neurones and neuropil<sup>11</sup> just as darkly (Bartoo et al., 1988, p. 471). To specifically stain amyloid deposits (NP) or paired helical filaments (NFT) antibody staining techniques must be used. "Neurofibrillary tangles are a relatively common finding in several unrelated disorders in which senile plaques do not occur" (Pappolla, 1989, p. 869). Consequently in this investigation only NP were dealt with. This section describes the algorithm

<sup>11</sup> *Neuropil* is an old-fashioned term used to describe the nerve tissue structure containing specialised points of contact (synapses) between nerve cells (Kuffler et al., 1984, p. 469).

developed to segment the NP from digitised photomicrographs of tissue sections stained with monoclonal or polyclonal antibodies against  $\beta$ A4 amyloid (i.e. NP). The results of an experiment that compared manual and computer measurements of NP are also presented. Finally a method for comparing the spatial distributions of plaques in different images is described.

#### 6.4.1. Classifying neuritic plaques

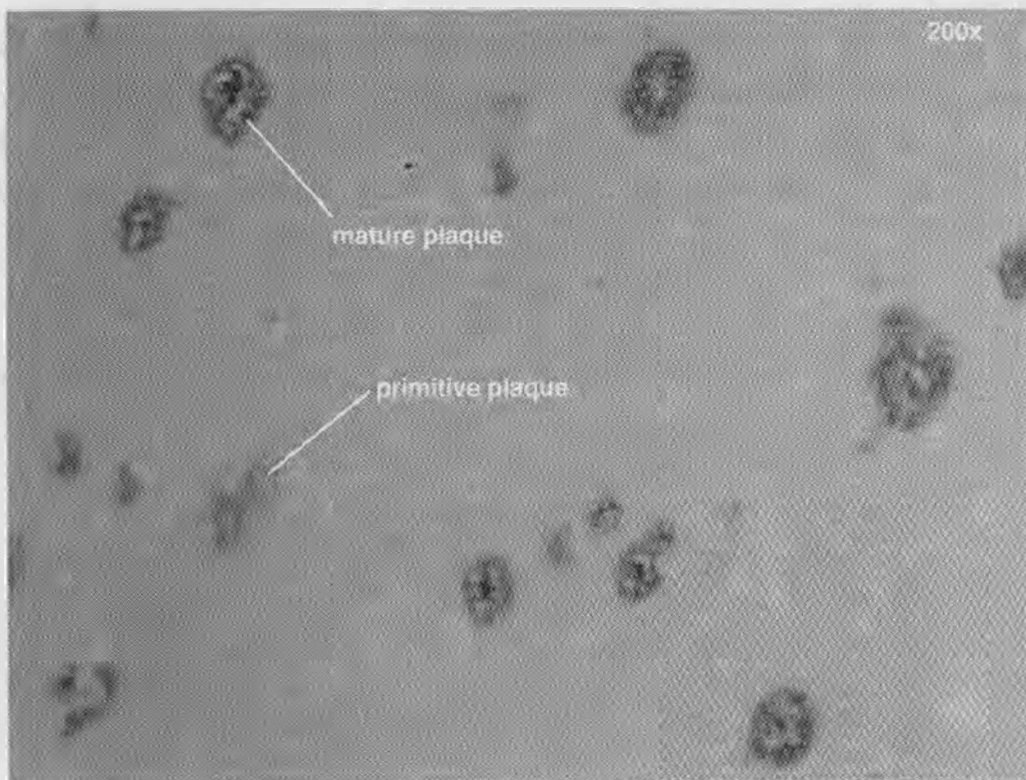
It is possible to classify the maturity of NP according to the relative concentrations of  $\beta$ A4 amyloid protein they contain. For example, Adams and Jones (1987, p.55) classify plaques as: *primitive plaques* containing only small wisps of amyloid; *mature (classical) plaques* displaying a central, often star-shaped mass of amyloid; and *compact* or *burnt out plaques* consisting almost entirely of amyloid. "The senile plaque is a complex, slowly evolving structure, and the time required to generate fully formed, 'mature' plaques may be years or even decades" (Selkoe, 1991, p. 42).

#### 6.4.2. Staining and image acquisition

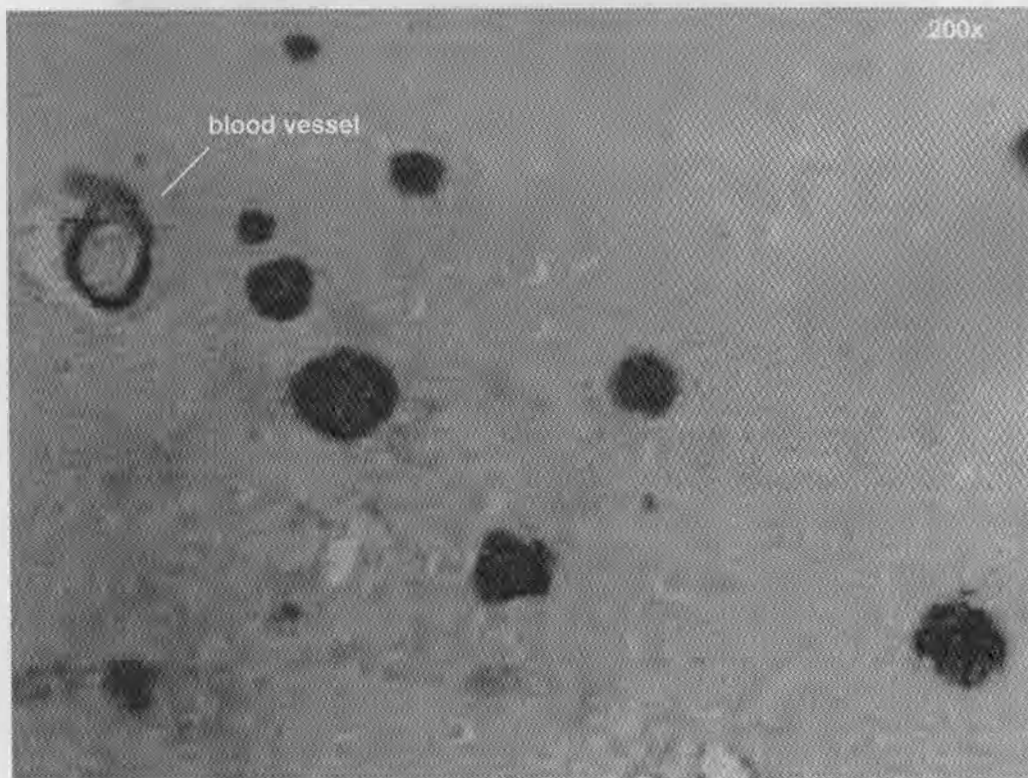
Bartoo et al. (1988, p. 471) argued that thioflavin-S is preferable to antibody staining for quantitative analysis because for antibody staining the number of objects highlighted depends upon the concentration and level of exposure to the stain. This may be true for the primitive plaques but because thioflavin-S itself lacks sensitivity to primitive plaques and amorphous amyloid deposits (Price et al., 1991, p. 305) the argument is not very sound. The works of Majocha, Benes, Reifel, Rodenrys, and Marotta (1988), and Price et al. (1991) are examples of quantitative studies that have used antibody staining.

Tissue used in this study was obtained from several post mortem brains; each with a neuropathologically confirmed diagnosis of Alzheimer's disease. All tissue sections originate from the cortex of the *inferior temporal lobe* (see Figure 35) and are twenty to thirty micrometres ( $\mu$ m) thick. The sections were immunochemically stained with polyclonal antibodies against  $\beta$ A4 amyloid. The stain is specific to  $\beta$ A4 amyloid and consequently does not stain neurones or NFT. A Leitz Laborlux S optical microscope fitted with a JVC colour CCD camera was used to image the tissue sections at 200 $\times$  magnification. When viewed through the microscope the NP appear as yellow-brown circular clusters of amyloid fibrils against a yellow background (see Appendix F). A monochrome frame-grabber (chrominance filter enabled) was used to digitise the images at a spatial resolution of 640H  $\times$  480V pixels and at a grey-scale resolution of 8-bits. Figures 37 and 38 are examples of digitised micrographs of tissue sections. There can be a lot of variability in the range of grey-tones present in different images. This is primarily influenced by the maturity of the plaques present although some variation in concentration and length of exposure to the stain may also contribute. Other factors include the age of the brain and the post mortem time that the tissue is processed (in this case, approximately 12 hours).

---



*Figure 37.* Light micrograph of a tissue section stained with polyclonal antibodies against  $\beta$ A4 amyloid.



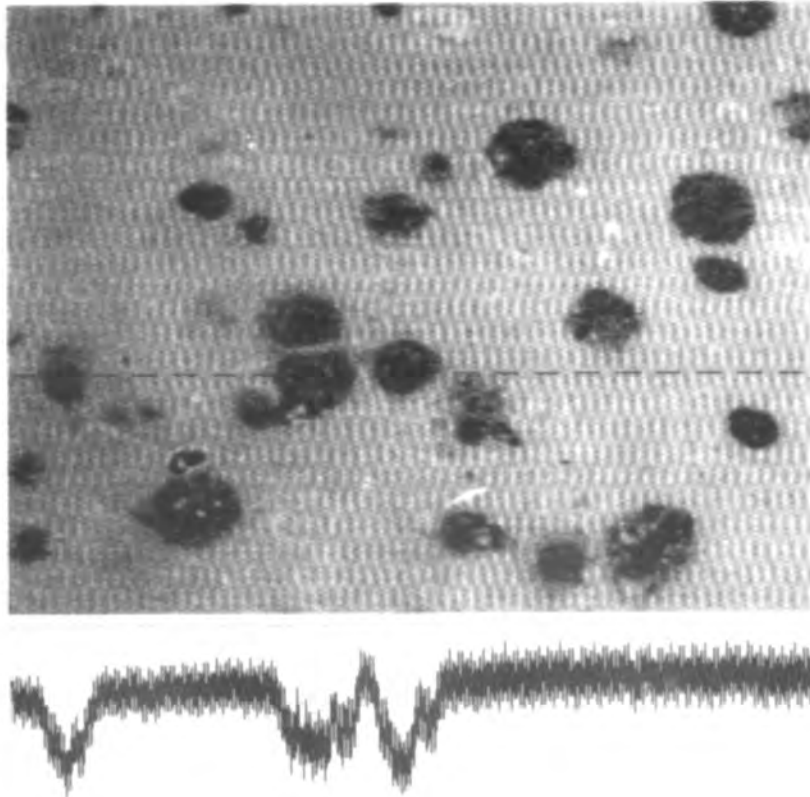
*Figure 38.* Amyloid can also be found surrounding blood vessels of the cerebral cortex as this light micrograph shows.

### 6.4.3. Noise removal and background normalisation

The images acquired from the frame-grabber exhibited electronic noise as Figure 39 attests. In addition some of the images exhibited wave-like patterns as a result of the vibrotome used to cut the tissue sections. The following ASF was used to attenuate this background noise:

$$N^3(I) = \varphi_3 \gamma_3 \varphi_2 \gamma_2 \varphi_1 \gamma_1,$$

where  $\gamma_i = \mathcal{O}(\cdot, g_i)$ ,  $\varphi_i = \mathcal{C}(\cdot, g_i)$ , and  $g_i$  is a sphere of radius  $i$  (see Figure 40).



*Figure 39.* A digitised micrograph (not to scale) along with a brightness profile for the row indicated. Electronic noise introduced during digitisation is clearly evident.

$$\begin{bmatrix} * & 0 & * \\ 0 & 1 & 0 \\ * & 0 & * \end{bmatrix}_{-1,1} ; \begin{bmatrix} * & * & 0 & * & * \\ * & 1 & 1 & 1 & * \\ 0 & 1 & 2 & 1 & 0 \\ * & 1 & 1 & 1 & * \\ * & * & 0 & * & * \end{bmatrix}_{-2,2} ; \begin{bmatrix} * & * & * & 0 & * & * & * \\ * & 1 & 2 & 2 & 2 & 1 & * \\ * & 2 & 2 & 2 & 2 & 2 & * \\ 0 & 2 & 2 & 3 & 2 & 2 & 0 \\ * & 2 & 2 & 2 & 2 & 2 & * \\ * & 1 & 2 & 2 & 2 & 1 & * \\ * & * & * & 0 & * & * & * \end{bmatrix}_{-3,3}$$

Figure 40. Bound matrices representing digital spheres, satisfying  $x^2 + y^2 + z^2 = r^2$ , for  $r=1, 2$ , and 3 respectively.

The ASF comprises three consecutive open-closings by spheres of radii 1 through 3. A geometrical interpretation of this filter is obtained by visualising the brightness surface of the digitised image,  $f$ , as a topographic landscape (Sternberg, 1986). A single iteration (i.e. an open-closing) of the ASF can be likened to the rolling of a sphere across the underside of the surface of the landscape and then across the top of the landscape. Wherever the sphere is unable to penetrate, e.g. small crevices, the brightness surface is smoothed over (see Figure 41).

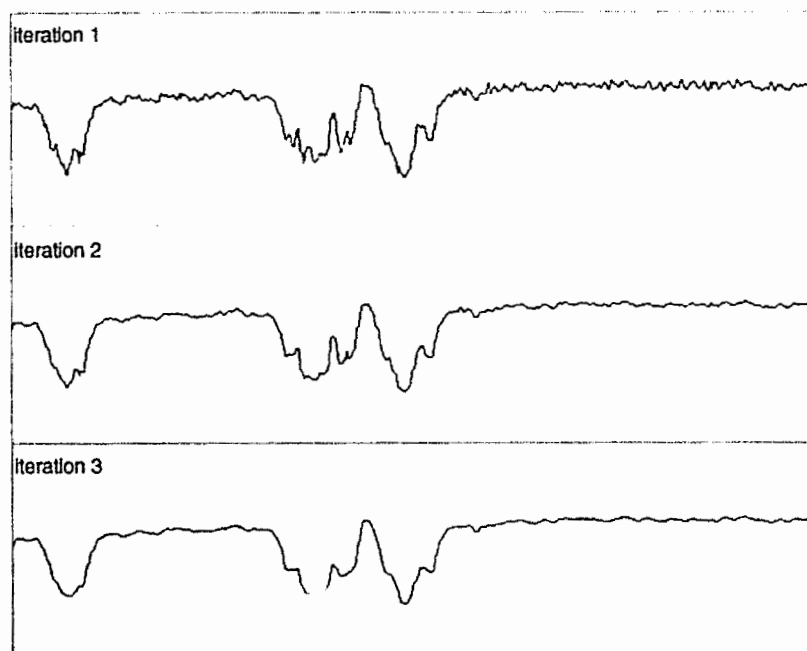


Figure 41. The effect of the ASF  $N^3(l) = \gamma_3 \gamma_3 \phi_2 \gamma_2 \phi_1 \gamma_1$  upon the brightness profile depicted in Figure 39.

Figure 39 shows that individual plaques are not simple homogeneous regions. They often resemble agglomerations of smaller particles. In the case of the more mature plaques the particles appear to

accumulate around a darker mass (the core). The following generalised OC filter was used to improve connectivity within plaques and to suppress artefacts larger than those attenuated by the ASF:

$$f'' = \min_{j=1}^4 \left[ \mathcal{C} \left( \max_{i=1}^4 [\mathcal{O}(f', B_i^{10})], B_i^{10} \right) \right],$$

where  $f'$  is the ASF filtered image and the  $B_i^n$  are the linear SEs depicted in Figure 21. The choice of value for  $n$  dictates the size of the smallest plaques that can be detected. For this study a value of 10 was chosen based upon size criteria established in conjunction with Dr Inta Adams, a neuroanatomist in the Department of Science, Edith Cowan University. Linear rather than disk SEs were chosen so that the irregular boundaries of the plaques could be accurately detected. Furthermore the linear SEs preserve any blood vessels that may be present. One might think that a disk SE could be used to eliminate blood vessels. However, as Figure 38 shows, blood vessels can be associated with amorphous amyloid deposits that are larger in size than the vessel walls. Therefore it is necessary to segment the blood vessels along with the plaques and to reject them forthwith.

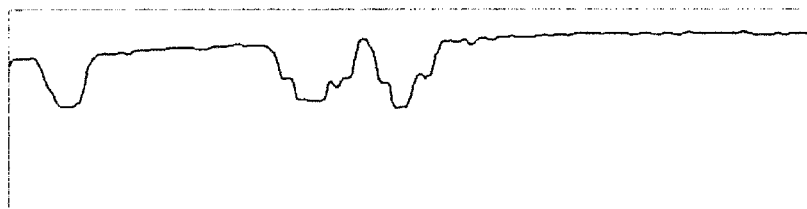


Figure 42. The brightness profile for the row in the OC filtered image corresponding to the row originally depicted in Figure 39.

The trend apparent in the brightness profile shown in Figure 42 is a result of both the non-uniform staining of the tissue section and the uneven illumination across the microscope stage. This background non-uniformity was a feature of all of the images digitised (in varying degrees). This meant that plaques could not be segmented by simple global thresholding. The following tophat transform (Meyer, 1986) was used to *normalise* the background after OC-filtering:

$$\mathcal{C}(f'', B) - f'',$$

where  $f''$  is the OC filtered image, and  $B$  is a flat octagonal SE larger than the largest plaque. Of the images analysed in this study the micrograph depicted in Figure 43 contained the largest plaque. Consequently an octagon of width 145 pixels was used in the tophat transform. An octagonal rather than a disk-shaped SE was chosen because it can be decomposed into a sequence of simple dilations. "The problem with a sequence of digital disks is that, unlike digital octagons, squares, diamonds, lines,



and even Euclidean [i.e.  $\mathbf{R}^2$ ] disks, such a sequence does not satisfy the condition of similarity up to a dilation for all successive pairs" (Vogt, 1988, p. 390). Consider the digital disks of radii 1 to 3 shown in Figure 44.  $D_2$  can be obtained from the dilation of  $D_1$  by itself. However  $D_3$  cannot be obtained from the dilation of  $D_1$  by  $D_2$ . Similarly  $D_4$  cannot be obtained from the dilation of  $D_2$  by itself though it can be obtained from the dilation of  $D_3$  by  $D_1$ .

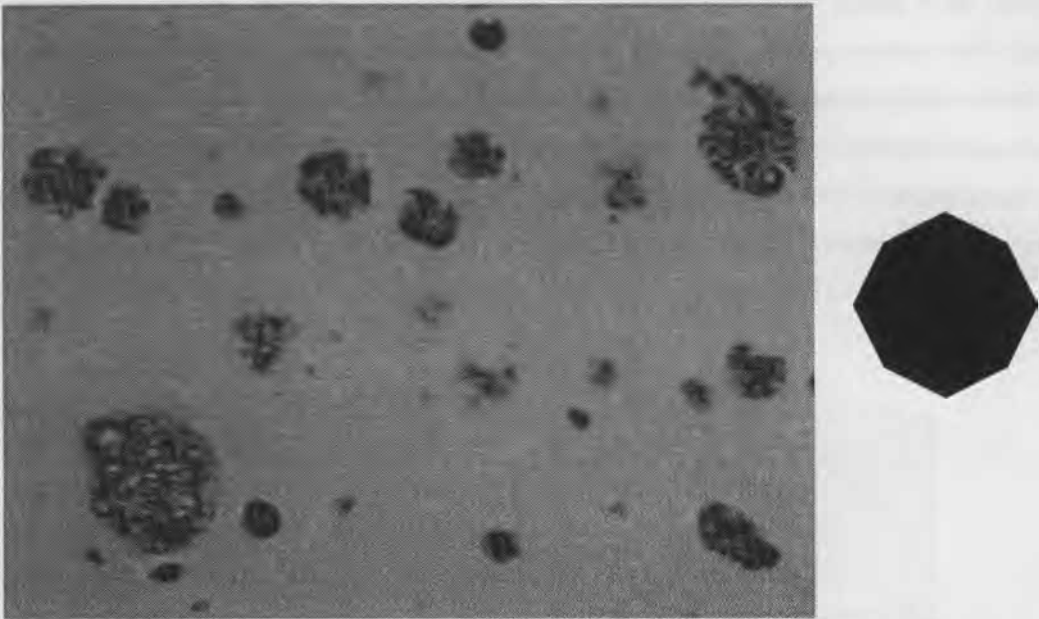


Figure 43. Light micrograph (not to scale) and the octagonal SE (actual size relative to the micrograph) used in the tophat transform for background normalisation. The SE must be larger than the largest plaque.

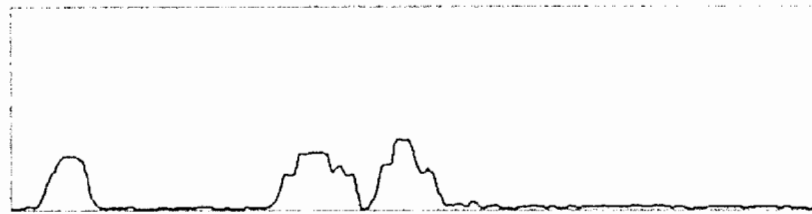
$$D_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}_{-1,1} ; D_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}_{-2,2} ; D_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{-3,3}$$

Figure 44. Bound matrices representing digital disks, satisfying  $x^2 + y^2 \leq r^2$ , for  $r=1, 2$ , and 3 respectively.

The octagonal SE depicted in Figure 43 has the dilation decomposition:

$$\underbrace{D_3 \oplus D_3 \oplus D_3 \oplus D_3 \oplus \dots \oplus D_3}_{24 \text{ times}},$$

It is a crude approximation to a disk of radius 75 pixels. A brute force closing using this large SE would require 15101 translations and a maximum operation (dilation) followed by 15101 translations and a minimum operation (erosion). Using dilation decomposition the closing reduces to 120 translations (i.e.  $24 \times 5$ ) and 24 maximum operations followed by 120 translations and 24 minimum operations. The closing of an OC-filtered image by the octagonal SE can be visualised as the sliding of an octagonal prism (or more colloquially, an octagonal cylinder) over the top of the brightness surface of the image. Wherever the prism is unable to penetrate, e.g. troughs corresponding to plaques and blood vessels, the surface is smoothed over. Effectively only artefacts larger than the plaques, such as large accumulations of stain and areas of constant background brightness, remain. Subtraction of the OC-filtered image from the closing (which must yield a positive image because the closing is extensive) leaves only the plaques and blood vessels; the background has been removed, i.e. normalised (Figure 45).



*Figure 45.* The brightness profile for the row in the tophat transformed image corresponding to the row depicted in Figure 39.

#### 6.4.4. Segmentation

The plaques can now be segmented by simple grey-level thresholding. After tophat transformation the image histogram looks much like that shown Figure 46. By inspection a suitable threshold value, i.e. an intensity value separating foreground (plaques) and background, lies somewhere in the valley between the two peaks. An algorithm to automatically determine the threshold point was developed based on Beucher's gradient.

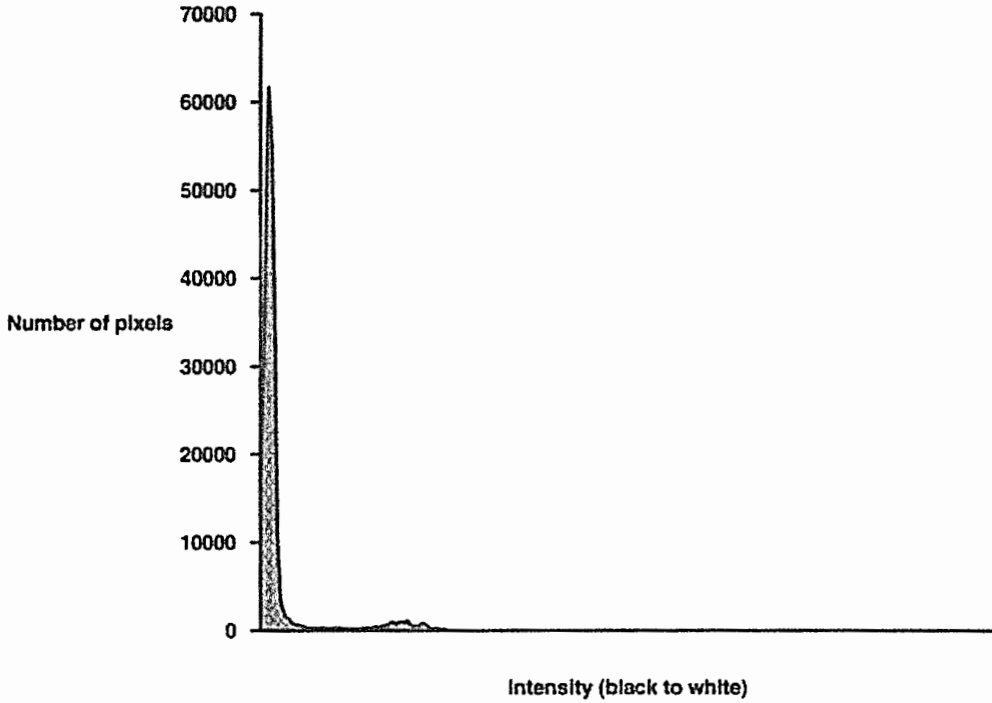


Figure 46. Typical image histogram after tophat transformation of an OC-filtered image.

#### 6.4.4.1. Beucher's gradient

By definition the gradient of a function  $f(x, y)$  is the vector  $\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$ . For a given point  $P(x, y)$ , the norm of the vector gives the value of the maximal directional derivative of  $f$  at  $P$ :

$$|\nabla f(P)| = \sqrt{\left( \frac{\partial f}{\partial x} \Big|_P \right)^2 + \left( \frac{\partial f}{\partial y} \Big|_P \right)^2}.$$

Beucher (1978) proposed the following algorithm for calculating the norm of  $\nabla f$  (cited in Serra, 1982, p. 441):

$$|\nabla f| = \lim_{\lambda \rightarrow 0^+} \frac{(f \oplus \lambda B) - (f \ominus \lambda B)}{2\lambda},$$

where  $B$  is the unit disk. The digital version of Beucher's gradient (Serra, 1988, p. 312) is given by:

$$\frac{(f \oplus B) - (f \ominus B)}{2}$$

where, for the square grid,  $B$  is either the unit square, or  $D_1$  of Figure 44.

#### 6.4.4.2. Selection of the threshold value

After tophat transformation the image background is very nearly zero (see Figure 45). Consequently the maximum rate of change of brightness, which occurs at the edges of the plaques (and possibly blood vessels), provides the required threshold value for grey-level thresholding. Specifically the threshold value  $t$  is given by

$$t = \frac{1}{2} \left[ \max_{(x,y)} (f \oplus B) - (f \ominus B) \right], \text{ where } f \text{ represents the tophat-transformed image.}$$

The NP and possibly blood vessels are then given by the threshold set  $\{(x, y) | f(x, y) \geq t\}$ .

#### 6.4.4.3. Connected component labelling

The next step involves individually labelling each of the connected components in the threshold set with a unique numeric label. The segmentation algorithm implemented in DIMPAL (Appendix G) utilises the connected component labelling algorithm of Manohar and Ramapriyan (1989). The algorithm individually labels each of the 8-connected components, in scan order (i.e. left to right and top to bottom).

#### 6.4.4.4. Border correction

A digitised micrograph represents only part of a larger tissue section that is in turn part of a brain. If the set  $X$  denotes the domain of this larger image (be it the tissue section or brain) then the digitised micrograph represents that part of  $X$  seen through a rectangular mask  $Z$  corresponding to the video frame captured by the frame-grabber, i.e.  $X \cap Z$ . One must determine, however, the size of the mask in which all of the transformations used to obtain the threshold set are known without error. From Chapter 2 recall that dilation commutes with the supremum and that erosion commutes with the infimum (Definition 9). Consequently

$$\begin{aligned} \mathcal{D}\left(\bigcup_i X_i, B\right) &= \bigcup_i \mathcal{D}(X_i, B), \text{ and} \\ \mathcal{E}\left(\bigcap_i X_i, B\right) &= \bigcap_i \mathcal{E}(X_i, B). \end{aligned}$$

Therefore  $\mathcal{E}(X \cap Z, B) = \mathcal{E}(X, B) \cap \mathcal{E}(Z, B)$ , i.e. the erosion of a digitised tissue section represents what is seen of the eroded image  $f$  through the eroded mask  $Z$ . To obtain a similar relation for dilation first consider that  $X \cup Z^c = (X \cap Z) \cup Z^c$ . It follows then that

$$\mathcal{D}(X \cup Z^c, B) = \mathcal{D}(X, B) \cup \mathcal{D}(Z^c, B) = \mathcal{D}(X, B) \cup [\mathcal{E}(Z, \bar{B})]^c.$$

Now the largest mask whose intersection with  $\left[\mathcal{E}(Z, \tilde{B})\right]^c$  is the empty set is  $\mathcal{E}(Z, \tilde{B})$ . Thus (Serra, 1982, p. 49)

$$\begin{aligned} \mathcal{A}(X \cap Z) \cup Z^c, B) \cap \mathcal{E}(Z, \tilde{B}) &= \left[ \mathcal{A}(X, B) \cup \left[ \mathcal{E}(Z, \tilde{B}) \right]^c \right] \cap \mathcal{E}(Z, \tilde{B}) \\ \left[ \mathcal{A}(X \cap Z, B) \cup \mathcal{A}(Z^c, B) \right] \cap \mathcal{E}(Z, \tilde{B}) &= \mathcal{A}(X, B) \cap \mathcal{E}(Z, \tilde{B}) \\ \left[ \mathcal{A}(X \cap Z, B) \cup \left[ \mathcal{E}(Z, \tilde{B}) \right]^c \right] \cap \mathcal{E}(Z, \tilde{B}) &= \mathcal{A}(X, B) \cap \mathcal{E}(Z, \tilde{B}) \\ \mathcal{A}(X \cap Z, B) \cap \mathcal{E}(Z, \tilde{B}) &= \mathcal{A}(X, B) \cap \mathcal{E}(Z, \tilde{B}). \end{aligned}$$

This last relation states that the size of the mask in which both the dilation on the set  $X$  and that on  $X \cap Z$  are known is  $\mathcal{E}(Z, \tilde{B}) = Z \ominus B$ . In summary " $Z \ominus B$  is the zone in which  $X \ominus B$ , and . . .  $X \oplus B$ , are known without error" (Serra, 1986, p. 295). In particular if  $X \cap Z$  is successively dilated or eroded by the symmetric SEs  $B_1, B_2, \dots, B_n$  then the final mask in which all of the transformations are known without error is  $Z \ominus (B_1 \oplus B_2 \oplus \dots \oplus B_n)$ .

For the proposed segmentation algorithm the zone in the threshold set for which the transformations are known without error is that defined by removing 44 rows from the top, 44 from the bottom, 44 columns from the left, and 44 columns from the right. The value 44 derives from:

ASF:  $4 \times (1 + 2 + 3) = 24 - 3$  open-closings with spheres of radii (in pixels) 1 to 3,

OC filter:  $4 \times 5 = 20 - 1$  open-closing; union of linear SEs is approximately a  $11 \times 11$  square.

"Note that the eroded masks  $Z \ominus B$  are not the largest possible in general. In the case of dilation for example, it may happen that some points  $x$ , located out of  $Z$ , . . . belong to the dilate (similarly for erosion). Nevertheless the mask  $Z \ominus B$  is the largest that is *independent of*  $X$  and in which the transform is known for each point" (Serra, 1982, p. 49).

In practice though, an edge correction of 11 pixels suffices. The segmentation algorithm already described involves four open-closings. For a given open-closing the initial erosion reduces  $Z$  to its smallest dimensions and the subsequent dilations and final erosion increase its size again. If the mask were to remain at the size dictated by the first erosion, then at the end of the ASF filtering the mask would be diminished by 6 pixels along all borders. After applying the OC filter the mask would be further diminished by 5 pixels along all borders. The segmentation algorithm implemented in DIMPAL (Appendix G) incorporates the following edge correction strategy:

- (i) the threshold set is stripped of 11 rows top and bottom, and 11 columns left and right;
- (ii) any connected components touching the border of this reduced mask are discarded.

#### 6.4.4.5. Eliminating primitive plaques, and blood vessels

Primitive plaques are characterised by lower concentrations of amyloid protein and thus are composed of lighter grey-tones than the more mature plaques. The method implemented for eliminating primitive plaques is as follows:

- (i) the maximum grey-value of the tophat transform is determined (this must correspond to a mature plaque or a blood vessel containing amyloid);
- (ii) all plaques containing grey-values (w.r.t. the tophat transform) greater than or equal to 75% (heuristic) of the maximum value are retained.

The method employed to eliminate blood vessels is as follows:

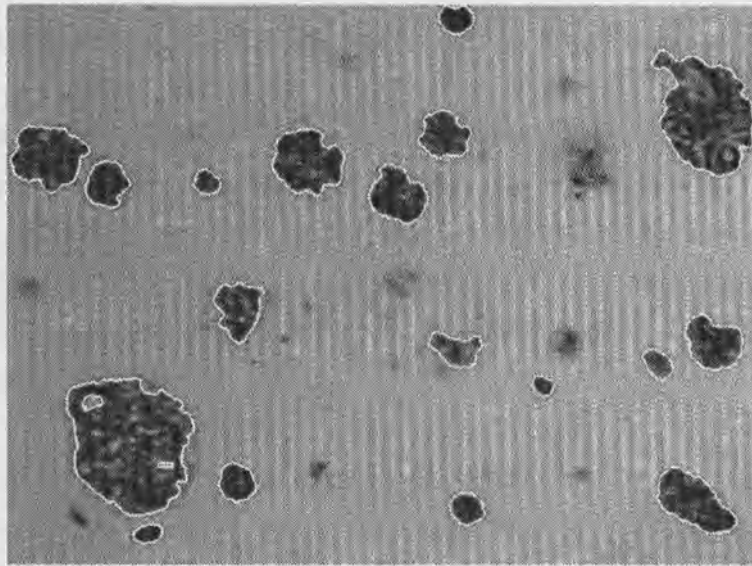
- (i) for each of the remaining connected components calculate the connectivity number (using the 8-connectivity relation given in Proposition 14), area, and perimeter (relations of sections 4.5., 4.6., and 4.7.);
- (ii) for each connected component calculate the  $P2A$  shape factor (see Danielsson, 1978, p. 292) viz.

$$P2A = \frac{P^2}{4\pi A}, \text{ where } P = \text{perimeter, and } A = \text{area:}$$

- (iii) discard those connected components that have connectivity number 0 (i.e. contain a hole) and a shape factor (which is 1 for a circle) greater than 2 (which is invariably the case for blood vessels because they have a total perimeter that is the sum of the inner and outer perimeters).

#### 6.4.4.6. Boundary smoothing, and elimination of small plaques

After the primitive plaques and blood vessels have been removed only the mature plaques remain. As Figure 47 illustrates, the linear SEs used in the generalised OC filter preserve the irregular plaque boundaries. A binary opening by a suitably sized disk (a disk of radius 10 pixels was used in this study) smooths these boundaries and eliminates plaques considered to be too small (see Appendix H, image (k)). This smoothing is analogous to that performed manually when plaque boundaries are traced by hand (plaques are generally modelled by disks and ellipses).



*Figure 47.* Segmentation of the mature plaques in the micrograph depicted in Figure 43. Boundaries have been accurately detected.

#### 6.4.5. Counts, area, and perimeter

The segmentation algorithm (Appendix G) was applied to 15 images arbitrarily selected from two tissue sections acquired from the frame-grabber and microscope (four of these are shown in Figure 51). The algorithm successfully segmented the mature plaques in all of these images (Appendix H).

In the initial stages of this study only half a dozen colour prints, reproduced from 35 mm slides, of tissue sections (originating from the cerebral cortical region) stained with monoclonal antibodies against  $\beta$ A4 amyloid, were available for processing. These prints were scanned, using a 300dpi colour scanner, for processing by DIMPAL. In contrast to the later images acquired from the frame-grabber and microscope these digitised images did not have a 1:1 pixel aspect ratio. Dr Adams arranged for a research assistant to count and measure the mature plaques present in the colour prints. For each print this involved

- (i) manually tracing the outlines of the plaques onto tracing paper and then determining the radius of a circle of best fit, with the aid of a ruler, for each plaque – thereby yielding area and perimeter estimates, and
- (ii) using a semi-automatic measuring device to provide area and perimeter estimates – this entailed tracing around each plaque on a graphics tablet using a stylus – after tracing the boundary of an individual plaque its area and perimeter estimates appear in a liquid crystal display.

Each RGB colour image was transformed into a grey-tone image in preparation for segmentation and measurement by DIMPAL. In DIMPAL this transformation was achieved using the statement

```
image=integer((plaques.red^2+plaques.green^2+plaques.blue^2)^0.5),
```

where *plaques* is a three layer variable – R layer, G layer, and B layer, i.e. each grey-tone pixel is the norm of the corresponding RGB colour vector. (Note that this represents only one possible approach. Pei and Chen (1991), for instance, transform an RGB image into the HLS colour space and then apply morphological operations to the L (lightness) component). To exactly match the size criterion used by the research assistant and to adjust for the lower resolution of these images (i.e. 494H × 385V), the following two statements:

```
plaques=dilate(erode(components>0,disk(10)),disk(10)),and
```

```
components=border_correct(separate_components(clip(threshold(tophat,threshold_value),11,-11,628,-468),8))
```

of the script file *segment.prg* (Appendix G) had to be changed to

```
plaques=dilate(erode(components>0,disk(7)),disk(7)),and
```

```
components=border_correct(separate_components(clip(threshold(tophat,threshold_value),11,-11,482,-373),8))
```

respectively. Manual registration with the original colour print revealed that a single pixel had width 0.36 mm and height 0.30 mm (to two decimal places) at 200× magnification. After segmentation the plaques were measured using

```
show(measure_components(separate_components(plaques,8),0.36,0.30)).
```

Table 4 lists the measurements made manually and those made by DIMPAL for the plaques shown in Figure 48.



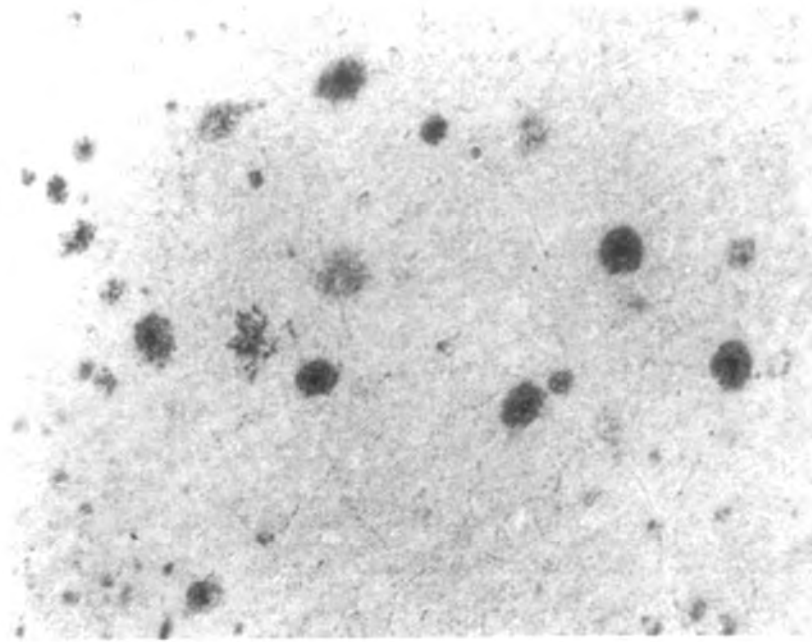
Table 4. Measurements of the plaques shown in Figure 48.

Plaque	Tablet and Stylus		Circle of Best Fit		DIMPAL Measurements			
	$\hat{A}$ (mm <sup>2</sup> )	$\hat{U}$ (mm)	$\hat{A}$ (mm <sup>2</sup> )	$\hat{U}$ (mm)	$\hat{A}$ (mm <sup>2</sup> )	$U^*$ (mm)	$U^{**}$ (mm)	$\hat{U} = \frac{1}{2}(U^* + U^{**})$ (mm)
1	71.27	30.86	63.59	28.26	64.80	37.73	37.96	37.84
2	69.40	29.90	70.85	29.83	68.04	37.78	38.24	38.01
3	51.24	26.89	50.24	25.12	58.21	33.43	36.00	34.72
4	17.15	14.68	19.63	15.70	19.66	20.29	20.76	20.53
5	45.22	24.52	44.16	23.55	49.90	32.86	33.57	33.22
6	48.44	26.95	56.72	26.69	59.72	34.56	36.53	35.59
7	57.26	26.58	56.72	26.69	60.16	38.06	37.13	37.60
8	22.88	17.58	23.75	17.27	26.24	22.85	23.79	23.32

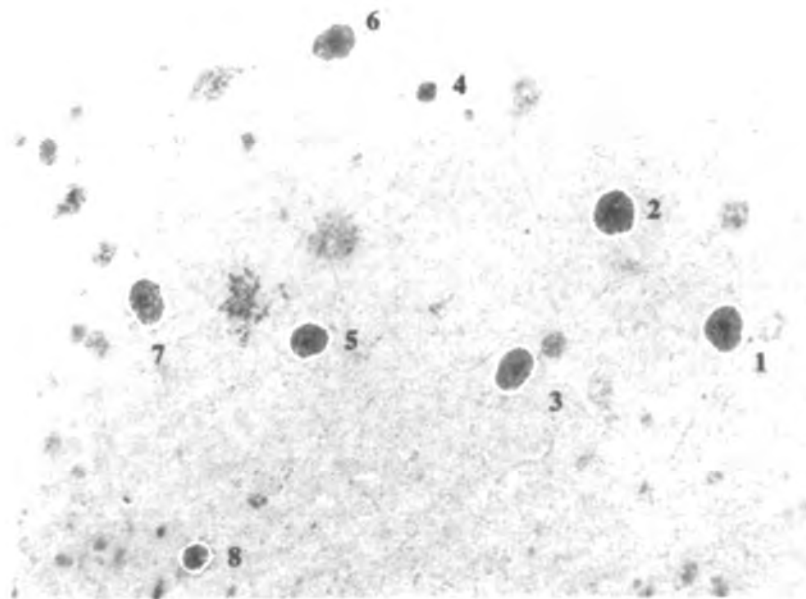
The results clearly show that in this instance the perimeter estimator  $\hat{U} = \frac{1}{2}(U^* + U^{**})$  (using the formulae with aspect ratio correction – section 4.7.) is positively biased. An unbiased estimator (heuristic) is  $\frac{3}{4} \cdot \frac{1}{2}(U^* + U^{**}) = \frac{3}{8}(U^* + U^{**})$  (see Figures 49 and 50). If the pixel aspect ratio was 1:1 then the area estimator (section 4.7.) would be unbiased. Even so, for the aspect ratio 6:5 (i.e. 0.36:0.3), Figures 49 and 50 show no significant bias. Moreover because the aspect ratio is very close to 1:1 it is difficult to distinguish estimator bias from discrepancies due to human subjectivity – the boundaries of plaques are not distinct and so subjective error is introduced when the plaques are manually traced.

#### Remarks

- (i) The positive bias associated with the perimeter estimator leads to positive bias in the *P2A* shape factor. Effectively the factor is  $(1.25)^2 = 1.5625$  for a circle rather than 1. Consequently in the DIMPAL algorithm (Appendix G) a value of 3 (approximately  $1.5625 \times 2$ ) is used in place of 2 in step (iii) of the blood vessel elimination procedure described in section 6.4.4.5.
- (ii) The bias exhibited by the perimeter estimator is influenced by the degree of smoothing of the boundaries of the plaques. The estimator is in essence a measure of the length of the smoothed boundary of a given plaque. Boundary smoothness therefore influences the quality of the estimator. In the extreme case, a fractal boundary would yield infinite perimeter.



(a)



(b)

*Figure 48.* (a) Original image. (b) Neuritic plaques detected by the DIMPAL algorithm. These were also the plaques classified manually.

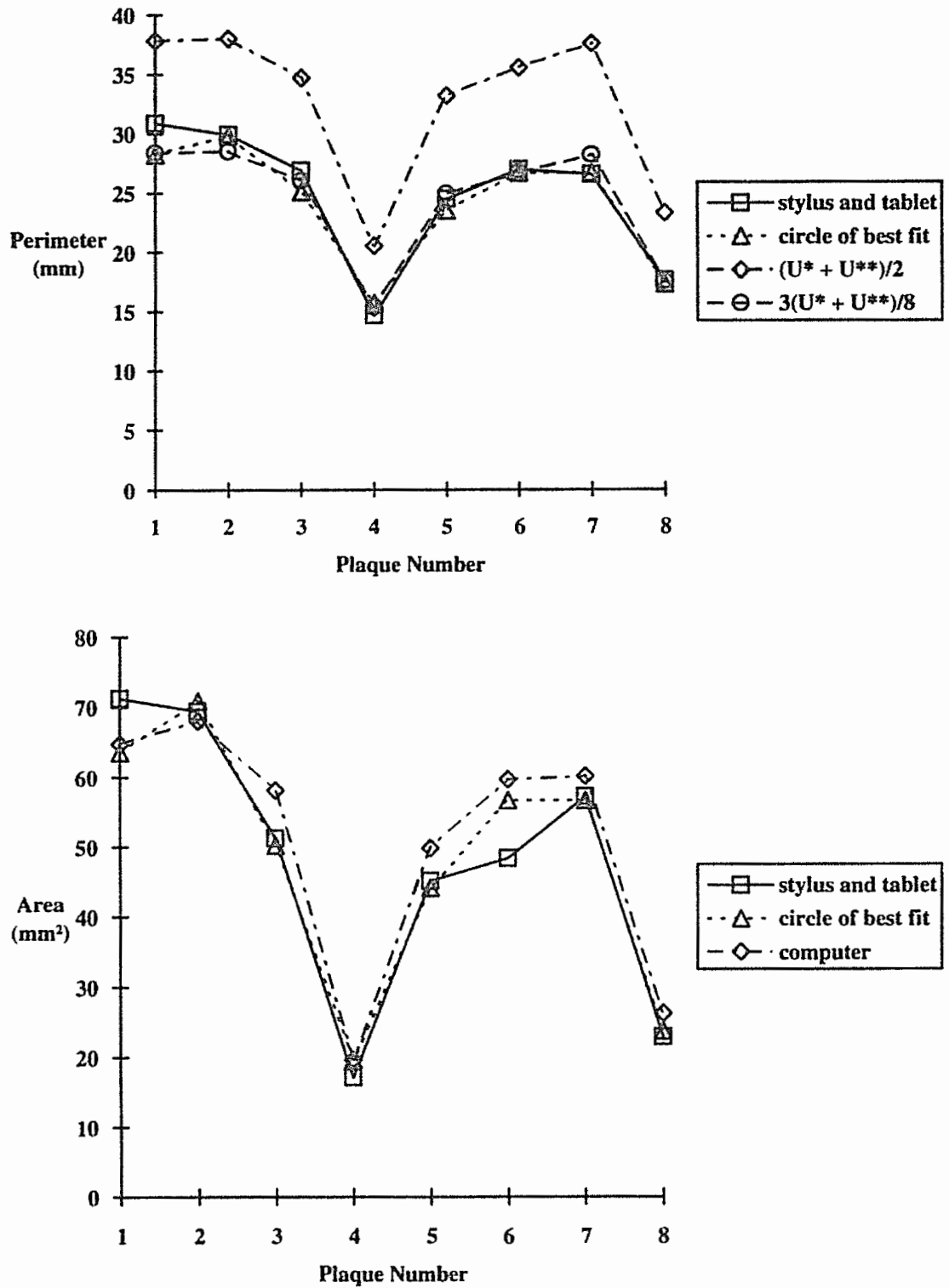


Figure 49. Measurements of the plaques shown in Figure 48.

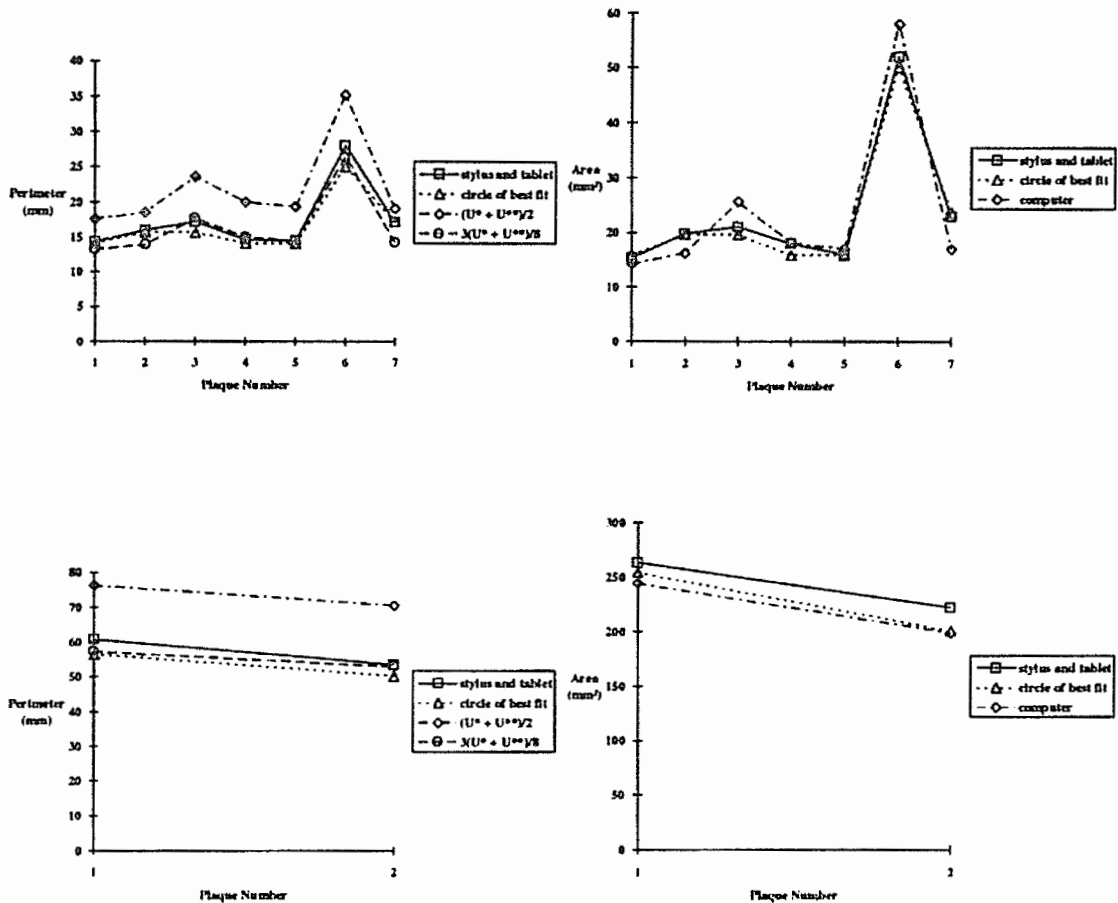


Figure 50. Measurements of plaques in two other images; one containing 7 plaques and the other 2.

#### 6.4.6. Cluster analysis

This section briefly describes a method for characterising the spatial distribution of plaques in a given micrograph. This makes it possible, for example, to compare the clustering of NP in tissue sections obtained from different parts of the cerebral cortex and hippocampus or in different cortical layers. One way of characterising the spatial distribution of plaques in a micrograph is to record the spatial coordinates of the centroid of each plaque. The frequencies of the nearest neighbour centre-to-centre distances then produce a characteristic distribution. Russ (1992, p. 336) points out that this approach "requires measuring individual features . . . [and] is also based on center-to-center [sic] distances, and not the distance between feature boundaries". An alternative approach, described by Russ, involves finding the Euclidean distance map (EDM) of the background surrounding the segmented plaques. The EDM is a binary to grey-level image mapping. Finding the EDM of the background entails assigning each background pixel (binary 0) a value that is the distance from that pixel to the nearest foreground pixel. The cumulative number of pixels at each brightness (distance) level is recorded. A signature of the spatial distribution of the plaques is then given by a plot of the cumulative number of pixels (or percent of background) against brightness (distance from the boundary). Figure 51

shows four such curves for images that have been segmented using DIMPAL (Appendix G). Note that to produce these curves a digital approximation to the EDM was used (see Borgefors, 1986). A straight line fitted to the straight portion of this curve will have slope units/length. This length "is a measure of how far from a feature boundary a randomly placed point on the image is expected to lie" (Russ, 1992, p. 337). Russ' procedure is a rough-and-ready method of characterising plaque spatial distribution because it implicitly assumes that there are no objects (plaques) beyond the frame of the image<sup>12</sup>.

## 6.5. Discussion

The proposed segmentation algorithm represents a vast improvement over that devised by Bartoo et al. (1988). Firstly, the algorithm is completely automated – thresholds are determined automatically. Secondly, because this study focused on NP and dealt with tissue sections specifically stained with antibodies against  $\beta$ A4 amyloid, the problems Bartoo et al. (1988) encountered with regard to the misclassification of NFT as NP were avoided. Finally the algorithm incorporates border count correction. In an experiment consisting of 18 arbitrary images (including three colour prints) the algorithm was able to successfully segment the mature NP in each image. The proposed algorithm essentially requires two parameters:

- (i) the minimum grey-level needed for a plaque to be classified as mature – currently 75% of the maximum value in the tophat image, and
- (ii) the radius of the disk SE used to eliminate plaques that are too small – the radius must be greater than or equal to half the length of the digital line segments used in the generalised OC filter.

As it stands, the algorithm implicitly assumes the existence of at least one dark object (i.e. a mature plaque or blood vessel) in an image. A fixed threshold value could be substituted for the first parameter for sets of photomicrographs stained at the same time (equal exposure). Possible refinements to the algorithm include:

- (i) the classification of plaques as either primitive, mature, or burnt out by way of the textural information contained in the grey levels for each plaque;
- (ii) implementing the van Herk (1992) algorithm for octagonal dilations and erosions – this reduces the computational complexity of the closing in the tophat transform used for background normalisation to only 12 maximum and 12 minimum operations per pixel;
- (iii) the automatic separation of overlapping plaques – either via *cluster fast segmentation* (Serra, 1982, p. 413) or the application of the watershed transform to the inverted distance map of the segmented plaques (Dougherty, 1993, p. 467).

---

<sup>12</sup> The *K-functional* method (Diggle, 1983) offers a more informative and reliable way of characterising plaque spatial distribution.

Further research is required to develop a similar segmentation algorithm for NFT.

In an experiment that compared manual and computer measurements for 17 segmented plaques (from three images), the area and perimeter measurements obtained using the estimators of Chapter 4, attained strong positive correlation with the corresponding manual measurements (see section 7.1.). Whilst the area estimator did not exhibit any significant bias, the perimeter estimator was consistently positively biased. A heuristic correction factor was proposed for the perimeter estimator and was experimentally shown to produce unbiased estimates.

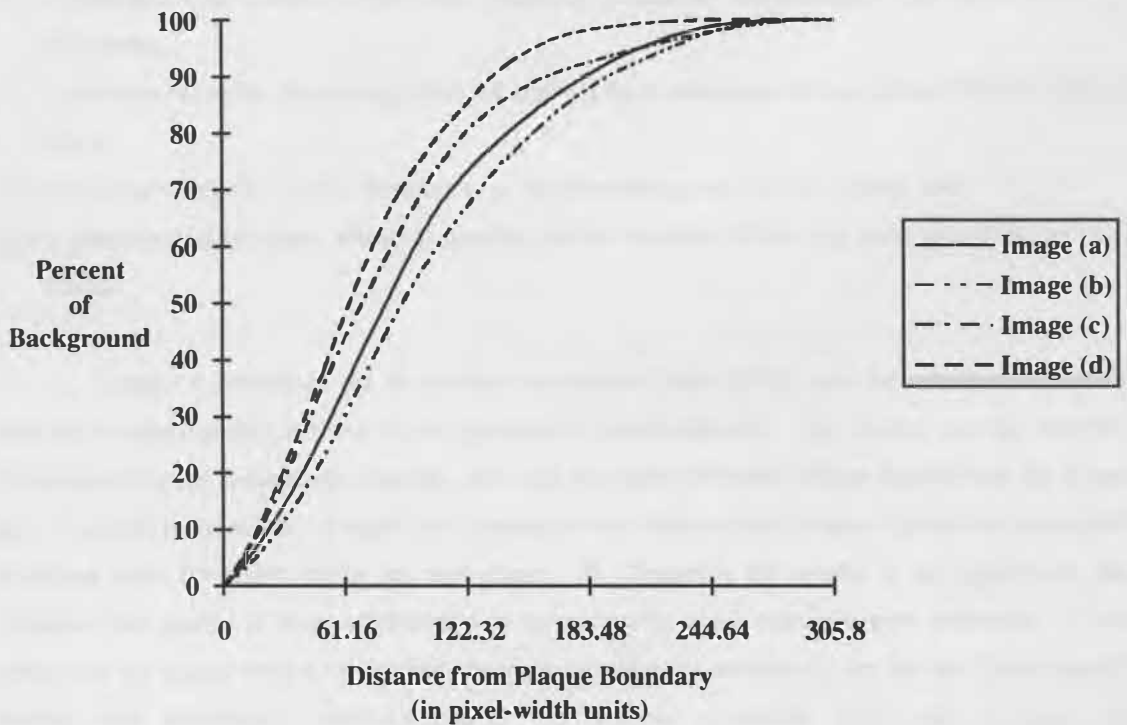
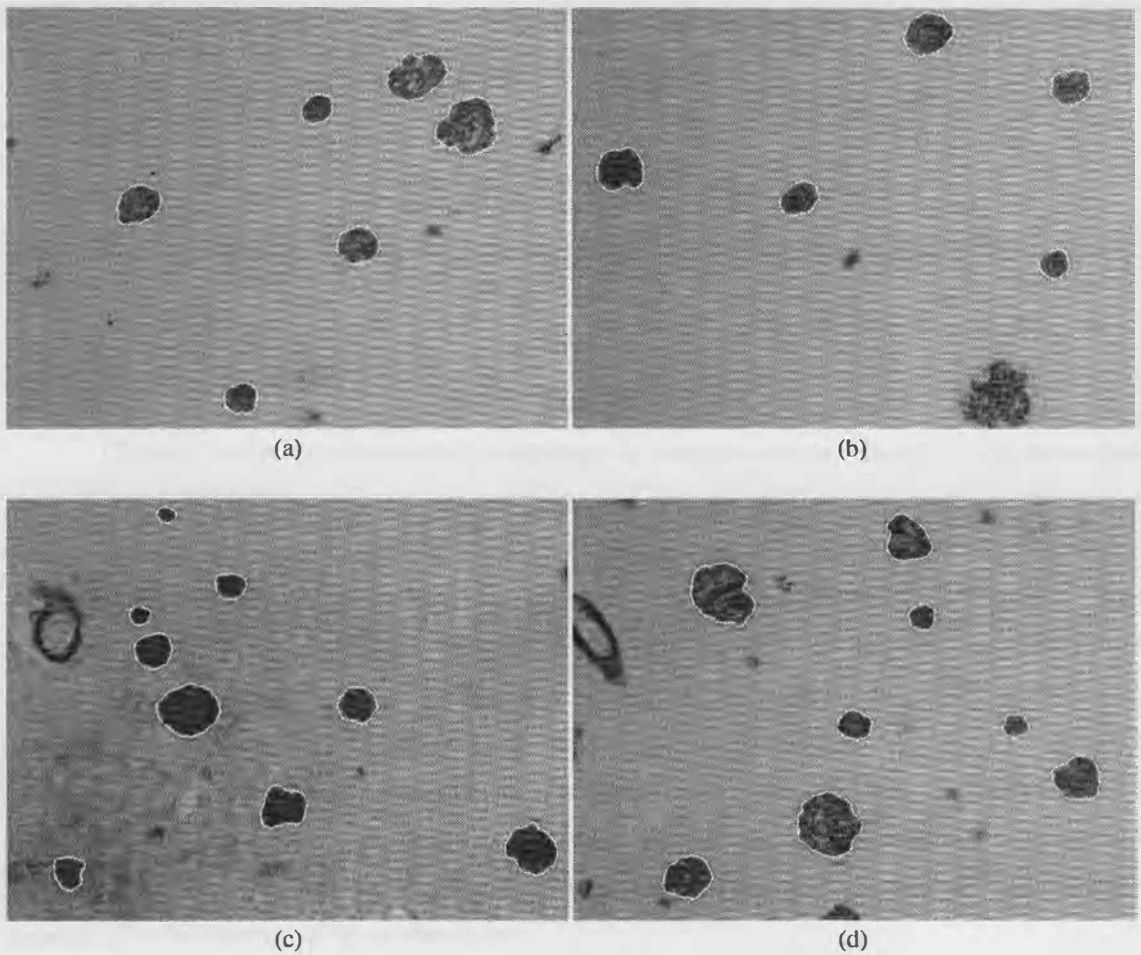


Figure 51. Spatial distributions of the plaques in the images (a), (b), (c), and (d) shown.

---

---

# CHAPTER 7.

## CONCLUSION

---

---

### 7.1. Summary

Chapter 2 reviewed mathematical morphology for complete lattices. Therein it was shown that classical binary and grey-scale Euclidean morphology are instances of this more general morphology – complete boolean lattices, and complete function lattices respectively. The complete lattice framework facilitated the establishment of the taxonomy of morphological filters presented in Chapter 3. The discussion at the end of Chapter 2 provided a rationale for focussing on all increasing and idempotent (and translation invariant in the case of Euclidean morphology) mappings from a complete lattice into itself. Whilst the list of such mappings – morphological filters – is infinite they can be grouped into various classes. To this end the taxonomy of morphological filters was established. The need for such a summary is justified when one considers the substantial theoretical advances in mathematical morphology in recent years and the proliferation of papers presented at conferences and published in journals. The key features of the taxonomy are:

- (i) it highlights links between related filter types (e.g. generalised morphological filter and the OC and CO filters),
- (ii) it provides examples illustrating either the application or behaviour of each of the different types of filters,
- (iii) it addresses omissions in the literature (e.g. annular closings and 2D OC filters), and
- (iv) it generalises filter types, wherever possible, to the complete lattice (e.g. generalised OC and CO filters).

Chapter 4 introduced the hit-or-miss transform of Serra (1982) and the role that it plays in defining morphologically derived feature parameters (morphometrics). The chapter derived formulae for determining the connectivity number, area, and perimeter of binary images digitised on the square grid. In addition the subject of aspect ratio correction was addressed for images digitised on rectangular sampling grids for which pixels are non-square. In Chapter 6 the results of an experiment that compared the quality of these estimators with measurements made manually were presented. It was noted that the digital version of Crofton's formula for perimeter estimation, for the set of micrographs studied, was consistently positively biased. A heuristic correction factor was proposed and demonstrated, experimentally, to provide unbiased perimeter estimates.



Chapter 5 introduced DIMPAL; a general purpose digital image processing and analysis language developed as a research tool for the Alzheimer's disease case study documented in Chapter 6. Moreover DIMPAL was used to create or verify the examples presented in earlier chapters and in particular Chapter 3.

Finally in Chapter 6 the ideas introduced in the preceding chapters were applied to the problem of segmenting and measuring neuritic plaques, a microscopic hallmark of Alzheimer's disease, from digitised light micrographs of post mortem brain tissue. The chapter described an automated segmentation algorithm, developed in DIMPAL, that incorporates an ASF and a generalised OC filter (Chapter 3). Using DIMPAL the algorithm was applied to a sample of 15 images acquired from two tissue sections stained with polyclonal antibodies against  $\beta$ A4 amyloid. The algorithm successfully segmented the mature plaques. In addition the algorithm was applied to another three images obtained from colour prints made from 35 mm slides of tissue sections stained with monoclonal antibodies against  $\beta$ A4 amyloid. Once again the algorithm correctly segmented the neuritic plaques. For these latter images, independent manual measurements (stylus and tablet, and circle of best fit) of plaques were compared with measurements made by DIMPAL (which uses the estimators of Chapter 4). The correlation matrices for the area and perimeter measurements respectively are:

area	stylus and tablet	circle of best fit
circle of best fit	0.998	1
DIMPAL	0.997	0.998

perimeter	stylus and tablet	circle of best fit
circle of best fit	0.997	1
DIMPAL	0.995	0.995

It was noted that, in this instance, the perimeter estimator is positively biased and a heuristic correction factor was proposed and experimentally validated. The segmentation algorithm and underlying methodology presented in Chapter 6 represents a vast improvement over those devised by Bartoo et al. (1988).

In short each of the research objectives outlined in Chapter 1 have been fulfilled.

## 7.2. Discussion

Serra's classic treatise *Image Analysis and Mathematical Morphology*, 1982, introduced the world to an image processing paradigm that had evolved out research at the Paris School of Mines by Matheron and Serra in the mid to late 60s. Though Matheron originally devised the (binary) opening to

generate size distributions practitioners ostensibly used it for noise filtering. A number of novel filtering techniques, most notably the iterative filter of Sternberg (1986), were developed by morphological practitioners in lieu of any theoretical foundations. "The clarification of the theory of morphological filtering opened the way to new types of filters that could never have been discovered by experimentation" (Serra, 1988, p. 101).

Dougherty (1992a, 1992b) placed morphological operators (both binary and grey-scale) into the framework of statistical estimation with the aim of developing a theory of mean-square (MS) optimisation. In image processing one often seeks to restore an image that has been corrupted or degraded in some manner (e.g. by impulse noise). A statistical analysis of the underlying signal (image) and noise processes leads to the development of a statistical estimator (an image filter) that takes the observed image as an input and produces an estimate of the uncorrupted image as an output (Loce & Dougherty, 1992, p. 412). The uncorrupted image, the corrupted image, and the noise are modelled as random processes and thus the statistical estimator is a function of random inputs. In reality the estimator (filter) is applied to a realisation of the image-noise process. Some criterion is then used to measure the effectiveness of the restoration. This gives rise to the possibility of finding amongst all possible filters the one that optimally restores the corrupted image. "Historically, the most commonly employed filter has been the optimal linear MS estimator" (Dougherty, 1992a, p. 37). This filter is defined in terms of linear combinations of the observed variables. Dougherty (1992a) interprets the classical binary morphological erosion as a statistical estimator that is a function of a vector of  $n$  binary observation random variables (corresponding to the pixels of the input binary image); "optimization [sic] is achieved by finding the structuring element that minimizes [sic] MS error" (p. 36). Using the Matheron representation theorem for increasing and translation invariant mappings, Dougherty generalised the procedure to morphological filters given by unions of erosions. In an analogous fashion Dougherty (1992b) defined optimal mean-square  $n$ -observation grey-scale morphological filters. Wang and Neuvo (1992) developed "an adaptive algorithm . . . for determining, from a given class of grayscale [sic] morphological filters, a filter which minimizes [sic] the mean square error between its output and a desired process" (p. 588). Loce and Dougherty (in Dougherty, 1993) have investigated a "suboptimal design methodology for binary filters [i.e. morphological operators] based on the imposition of various constraints and small (but well-chosen) libraries [of SEs], the goal being to facilitate computationally tractable design" (p. 44).

The formulation of mathematical morphology in terms of complete lattices (Serra, 1988) represented an extension of classical Euclidean morphology through a generalisation of the underlying object space. Heijmans and Ronse (1990) generalised Euclidean translation invariance to complete lattices by considering certain abelian groups of automorphisms on the lattice. This gives rise to another method of generalising Euclidean morphology – by replacing the translation group with another commutative group of automorphisms. In Serra's book (1982, p. 17) "a photograph is shown of trees in

a forest, taken by putting the camera at ground level and aiming toward the sky. Such photographs are used to measure the amount of sunshine in the woods. The resulting image shows clear radial symmetry with intrinsic origin (the projection point of the zenith)" (Dougherty, 1993, p. 207). In this case the image points should be indexed by polar co-ordinates and then the "group generated by rotations and multiplications with respect to the origin" (Dougherty, 1993, p. 207) constitutes the commutative group of automorphisms (see Heijmans and Ronse, 1990, p. 283). Roerdink (in Dougherty, 1993) has generalised mathematical morphology in still another way by dropping the assumption that the invariance group is commutative.

### 7.3. Future Directions

The underlying theory of morphological filtering presented in Chapters 2 and 3 (and more extensively covered in Maragos & Schafer 1987a, 1987b; Serra, 1988; Heijmans & Ronse, 1990; Ronse & Heijmans, 1991) establishes the "good" mappings but "does not give any indication as to how we should manage a sequence of morphological operations with a given aim in view" (Serra, 1988, p. 5). Algorithm development still relies heavily on heuristic methods and geometrical interpretations – particularly in relation to the choice of transformations and the types of SEs used. Whilst an algorithm such as that described in Chapter 6 follows a logical sequence of steps, there does not exist any theory for synthesising them. To quote Vogt (1989, preface):

the process of [morphological] image algorithm development is more time consuming and less motivating than it perhaps could be, because it typically requires repeating the same algorithm sequences over many different images. It is also mentally taxing because the developer has to visually evaluate and mentally integrate how well each algorithm performs over the entire image set, and from this propose changes in parameters of operator sequences which will improve the overall performance.

With regard to the "good" mappings it is sobering to note that "there are many practical consequences and useful algorithms that remain to be discovered" (Serra, 1988, p. 101).

My ongoing research interests include:

- (i) the application of mathematical morphology to the segmentation of the vascular network from the back of a hand for identification purposes (Mehnert, Cross, & Smith, 1993)\*.

---

\* Research project funded by the Institute of Security and Applied Technology (ISAT), Edith Cowan University.

- (ii) the development of imaging tools to enhance and analyse markings on bullet casings – these tools will assist ballistic experts in identifying the firearm from which a bullet was fired\* ,
- (iii) the continued development of DIMPAL, and
- (iv) the continuation of the Alzheimer's disease research with the possibility of commercial development.

---

---

## ***BIBLIOGRAPHY***

---

---

- Abutaleb, A.S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing*, 47, 22-32.
- Adams, I. & Jones, D. (1987). Effects of normal pathological aging on brain morphology: neurons and synapses. *Current Topics in Research on Synapses*, 4, 1-84.
- Aherne, W.A. & Dunhill, M.S. (1982). *Morphometry*. London: Edward Arnold.
- Allenby, R.B. (1983). *Rings, fields and groups*. London: Edward Arnold.
- Bachman, G. & Narici, L. (1966). *Functional analysis*. New York: Academic Press.
- Bartoo, G.T., Kim, Y., Haralick, R.M., Noehlin, D. & Sumi, S.M. (1988). Mathematical morphology techniques for image processing applications in biomedical imaging. *Proceedings of the SPIE - The International Society for Optical Engineering*, 914, 466-475.
- Bookstein, F.L. (1978). *The measurement of biological shape and shape change* (lecture notes in biomathematics, vol. 24). Berlin: Springer-Verlag.
- Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34, 344-371.
- Chmúrny, J. & Lehotský, M. (1990). Morphological image filtering. *Electrotechnický časopis*, 41(4), 249-257.
- Cockshott, W.P. (1990). *A compiler writer's toolbox*. New York: Ellis Horwood.
- Cook, D. (1983). Pixel averaging smooths graphics effects. *Creative Computing*, 9(1), 146-158.
- Crombez, G. (1990). The Matheron representation theorem for gray-scale morphological filters. *Proceedings of the American Mathematical Society*, 108(3), 795-799.
- Danielsson, P. (1978). A new shape factor. *Computer Graphics and Image Processing*, 7, 292-299.

- Diggle, P.J. (1983). *Statistical analysis of spatial point patterns*. London: Academic Press.
- Dougherty, E.R. (1989). The dual representation of gray-scale morphological filters. In, *Proceedings CVPR '89 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 172-177). Washington: IEEE Computer Society Press.
- Dougherty, E.R. (1990). Optimization of gray-scale morphological statistical estimators. In, *Fifth ASSP Workshop on Spectrum Estimation and Modeling* (pp. 178-181). New York: IEEE.
- Dougherty, E.R. (1992a). Optimal mean-square  $N$ -observation digital morphological filters: I. Optimal binary filters. *CVGIP: Image Understanding*, 55(1), 36-54.
- Dougherty, E.R. (1992b). Optimal mean-square  $N$ -observation digital morphological filters: II. Optimal gray-scale filters. *CVGIP: Image Understanding*, 55(1), 55-72.
- Dougherty, E.R. (Ed.). (1993). *Mathematical morphology in image processing*. New York: Marcel Dekker, Inc.
- Dougherty, E.R. & Giardina, C.R. (1986). An investigation of the relationship between morphological and linear filters. In, *Electronic Imaging '86, Boston* (pp. 270-274).
- Dougherty, E.R. & Kraus, E. (1990a). Filtering by morphological pseudoconvolutions. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1247, 157-168.
- Dougherty, E.R. & Kraus, E. (1990b). The role of shape in morphological bases for digital moving-average filters. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1350, 14-24.
- Fitch, J.P., Coyle, J.C. & Gallagher, N.C. (1985). Threshold decomposition of multidimensional ranked order operations. *IEEE Transactions on Circuits and Systems*, 32(5), 445-450.
- Forsythe, E. (1990). *Alzheimer's disease: the long bereavement*. London: Faber.
- Giardina, C.R. & Dougherty, E.R. (1988). *Morphological methods in image and signal processing*. New Jersey: Prentice-Hall.

- Grivas, D.A. & Skolnick, M.M. (1989). Morphological algorithms for the analysis of pavement structure. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1199, 211-219.
- Haralick, R.M. (1989). Mathematical morphology and computer vision. In, *Conference Record. Twenty-second Asilomar Conference on Signals, Systems and Computers* (vol. 1, pp. 479). Maple Press.
- Haralick, R.M., Sternberg, S.R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4), 532-550.
- Heijmans, H.J. (1989). Iteration of morphological transformations. *CWI Quarterly*, 2, 19-36.
- Heijmans, H.J. (1991). Theoretical aspects of gray-level morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), 568-582.
- Heijmans, H.J. & Ronse, C. (1990). The algebraic basis of mathematical morphology 1. Dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50, 245-295.
- Hungerford, T.W. (1974). *Algebra*. New York: Holt, Rinehart and Winston.
- Hunter, R. (1981). *The design and construction of compilers*. Chichester, New York: Wiley.
- Koskinen, L., Astola, J., & Neuvo, Y. (1991). Soft morphological filters. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1568, 262-270.
- Kuffler, S.W., Nicholls, J.G., & Martin, A.R. (1984). *From neuron to brain*. Sunderland, Massachusetts: Sinauer Associates Inc. Publishers.
- Loce, R.P. & Dougherty, E.R. (1992). Optimal morphological restoration: the morphological filter mean-absolute-error theorem. *Journal of Visual Communication and Image Representation*, 3(4), 412-432.
- Majocha, R.E., Benes, F.M., Reifel, J.L., Rodenrys, A.M., & Marotta, C.A. (1988). Laminar-specific distribution and infrastructural detail of amyloid in the Alzheimer disease cortex visualized by computer-enhanced imaging of epitopes recognized by monoclonal antibodies. In, *Proc. Natl. Sci. USA* (vol. 85, pp. 6182-6186).

- Manohar, M. & Ramapriyan, H.K. (1989). Connected component labeling of binary images on a mesh connected massively parallel processor. *Computer Vision, Graphics, and Image Processing*, 45(2), 133-149.
- Maragos, P. (1989). A representation theory for morphological image and signal processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), 586-599.
- Maragos, P. & Schafer, R.W. (1987a). Morphological filters - Part I: their set-theoretic analysis and relations to linear shift-invariant filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(8), 1153-1169.
- Maragos, P. & Schafer, R.W. (1987b). Morphological filters - Part II: their relations to median, order-statistic, and stack filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(8), 1170-1184.
- Matheron, G. (1975). *Random sets and integral geometry*. New York: Wiley.
- Mehnert, A.J. (1990). *Digital image processing using mathematical morphology*. Unpublished honours dissertation, Edith Cowan University, Perth, WA.
- Mehnert, A.J. (1993, February). *Mathematical morphology as a tool for the morphometric investigation of neuritic plaques associated with Alzheimer's disease*. Paper presented at the 29th Australian Applied Mathematics Conference, Höchstens, Hahndorf, SA.
- Mehnert, A.J. (1994). *DIMPAL: Digital image processing and analysis language* [Computer program]. Perth, WA: A.J. Mehnert, c/o Department of Mathematics, Edith Cowan University.
- Mehnert, A.J., Cross, J.M., & Smith, C.L. (1993). *Thermographic imaging: segmentation of the subcutaneous vascular network of the back of the hand* (Research Report). Perth, Western Australia: Edith Cowan University, Institute of Security and Applied Technology.
- Meyer, F. (1986). Automatic screening of cytological specimens. *Computer Vision, Graphics, and Image Processing*, 35, 356-369.
- Meyer, F. & Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1), 21-46.



- Pappolla, M.A. (1989). Image analysis microspectroscopy of senile plaque capillary amyloid in Alzheimer's disease. *Arch Pathol Lab Med*, 113, 866-871.
- Pei, S. & Chen, F. (1991). Subband decomposition of monochrome and color images by mathematical morphology. *Optical Engineering*, 30(7), 921-933.
- Petzold, C. (1989). *Programming the OS/2 presentation manager*. Microsoft Press.
- Preteux, F. & Merlet, N. (1991). New concepts in mathematical morphology: the topographical and differential distance functions. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1568, 66-77.
- Price, J.L., Davis, P.B., Morris, J.C., & White, D.L. (1991). The distribution of tangles, plaques and related immunohistochemical markers in healthy aging and Alzheimer's disease. *Neurobiology of Aging*, 12, 295-312.
- Ronse, C. & Heijmans, H.J. (1991). The algebraic basis of mathematical morphology II. Openings and closings. *CVGIP: Image Understanding*, 54(1), 74-97.
- Rosenfeld, A. & Kak, A.C. (1982). *Digital picture processing* (2nd ed.)(vol. 1). New York: Academic Press.
- Russ, J.C. (1992). *The image processing handbook*. Florida: CRC Press Inc.
- Schonfeld, D. & Goutsias, J. (1991). On the morphological representation of binary images in a noisy environment. *Journal of Visual Communication and Image Representation*, 2(1), 17-30.
- Sebastia, R.W. (1989). *Concepts of programming languages*. Redwood City, California: Benjamin/Cummings Publishing Company, Inc.
- Selkoe, D. (1991). Amyloid protein and Alzheimer's disease. *Scientific American*, 265(5), 40-47.
- Serra, J. (1982). *Image analysis and mathematical morphology*. London: Academic Press.
- Serra, J. (1986). Introduction to mathematical morphology. *Computer Vision, Graphics, and Image Processing*, 35, 283-305.

- Serra, J. (Ed.). (1988). *Image analysis and mathematical morphology. Volume 2: Theoretical advances*. London: Academic Press.
- Shih, F.Y. & Mitchell, O.R. (1989). Threshold decomposition of gray-scale morphology into binary morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1), 31-42.
- Song, J. & Delp, E.J. (1989). A generalization of morphological filters using multiple structuring elements. In, *Proceedings, 1989 IEEE International Symposium on Circuits and Systems, Portland* (pp. 991-994). New York: IEEE.
- Song, J. & Delp, E.J. (1990). The analysis of morphological filters with multiple structuring elements. *Computer Vision, Graphics, and Image Processing*, 50, 308-328.
- Song, J., Stevenson, R.L. & Delp, E.J. (1990). The use of mathematical morphology in image enhancement. In *Proceedings of the 32nd Midwest Symposium on Circuits and Systems* (Vol. 1, pp. 67-70). New York: IEEE.
- Sternberg, S.R. (1986). Grayscale morphology. *Computer Vision, Graphics, and Image Processing*, 35, 333-355.
- Stevens, C.F. (1979). The neuron. In, *The brain* (pp. 15-25). San Francisco: W.H. Freeman and Company.
- Stevenson, R.L. & Arce, G.R. (1986, October). Theoretical analysis of morphological filters. *Proceedings, 24th Annual Allerton Conference on Communication, Control, and Computing* (pp. 353-362).
- Stevenson, R.L. & Arce, G.R. (1987). Morphological filters: statistics and further syntactic properties. *IEEE Transactions on Circuits and Systems*, 34(11), 1292-1305.
- Townsend, M. (1987). *Discrete mathematics: applied combinatorics and graph theory*. Menlo Park, California: Benjamin/Cummings Inc.
- Underwood, E.E. (1970). *Quantitative stereology*. Reading, Mass.: Addison-Wesley.
- Van Herk, M. (1992). A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13, 517-521.

- Vincent, L. (1989). Graphs and mathematical morphology. *Signal Processing*, 16(4), 365-388.
- Vogt, R.C. (1988). Morphological operator distributions based on monotonicity and the problem posed by digital disk-shaped structuring elements. *Proceedings of the SPIE - The International Society for Optical Engineering*, 938, 384-392.
- Vogt, R.C. (1989). *Automatic generation of morphological set recognition algorithms*. New York: Springer-Verlag.
- Wang, Q & Neuvo, Y. (1992). Adaptive grayscale morphological filters for image noise smoothing. In, *IEEE International Conference on Systems Engineering* (pp. 588-591). New York: IEEE.
- Weibull, E.R. (1980). *Stereological methods* (vol. 2). London: Academic Press.
- Zhang, T.Y. & Suen, C.Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236-239.

---



---

## Appendix A

---



---

### The Bound Matrix

The *bound matrix*, introduced by Giardina and Dougherty (1988), is a special kind of matrix used to represent two-dimensional digital images that have been sampled on a rectangular grid. The elements of the matrix correspond to pixel values in the digital image. The term *bound* derives from the fact that the matrix has a fixed location within the discrete plane  $\mathbf{Z}^2$ . For example, the bound matrix

$$f = \begin{bmatrix} 34 & 12 & 1 \\ 12 & 233 & 127 \\ 255 & 255 & 28 \end{bmatrix}_{3,-2}$$

represents a grey-scale digital image comprising nine pixels. The top left pixel of the matrix is located at  $(3,-2)$  in  $\mathbf{Z}^2$ ; i.e.  $f(3,-2) = 34$  (in functional notation). The locations of the remaining pixels are determined relative to the top left element of the bound matrix; hence using a conventional x-y coordinate system, the pixel with value 127 has spatial location  $(5,-3)$ .

Finite digital images with rectangular domains in  $\mathbf{Z}^2$  are readily expressed as bound matrices. Pixel values outside the bound matrix are said to be *undefined* and are represented by asterisks. The previous bound matrix can also be written as

$$f = \begin{bmatrix} * & * & * & * & * \\ * & 34 & 12 & 1 & * \\ * & 12 & 233 & 127 & * \\ * & 255 & 255 & 28 & * \\ * & * & * & * & * \end{bmatrix}_{2,-1}$$

The former bound matrix is said to be *minimal* because it contains no extraneous asterisks. For digital images with non-rectangular domains or with missing values (e.g. due to a sensor malfunction during digitisation), the asterisk is used to *pad out* the bound matrix. For instance, if the pixel with value 127 in the original bound matrix was actually undefined, then the minimal bound matrix would be written

$$f = \begin{bmatrix} 34 & 12 & 1 \\ 12 & 233 & * \\ 255 & 255 & 28 \end{bmatrix}_{3,-2}$$

In practice, the asterisk is treated as though it is a pixel with value negative infinity. This simplifies the definitions of the basic morphological operations expressed using bound matrices (see Mehnert, 1990, section 4).

Conventionally, binary digital images are represented by 1s and 0s. The 1s represent the foreground points, and the 0s the background. However for grey-scale images, 0 actually represents a grey-level value. To resolve this conflict, 1s and asterisks are used to define the bound matrix of a binary image.

## Appendix B

*Table B-1.* Properties of the basic morphological operators.

	Dilation	Erosion	Opening	Closing
Commutative	✓			
Associative	✓	✓	✓	✓
Extensive	*			✓
Antiextensive		*	✓	
Increasing w.r.t. image	✓	✓	✓	✓
Decreasing w.r.t. SE		✓		
Idempotent			✓	✓

\* only if the domain of the SE includes the origin

---

---

## Appendix C

---

---

### Minkowski Functionals in $\mathbf{R}^n$

The Minkowski functionals in  $\mathbf{R}^n$  are defined by a recurrence relation on the sub-dimensions of the space as follows (Serra, 1982, p. 104):

$$\begin{aligned} \text{for } n=1 \quad W_0^{(1)} &= L(X); \quad W_1^{(1)}(X) = 2 \\ \text{for } n > 1 \quad W_0^{(n)} &= V^{(n)}(X); \\ W_k^{(n)}(X) &= \frac{1}{nb_{n-1}} \int_{\Omega_n} W_{k-1}^{(n-1)}[X \cap \Pi_\omega^{(n-1)}] d\omega \quad 1 \leq k \leq n \end{aligned}$$

where  $X$  is a compact convex set (an ovoid) in  $\mathbf{R}^n$ ;  $L$  means length;  $V^{(n)}(X)$  is the  $n$ -volume of  $X$ ,  $b_n$  is the  $n$ -volume of the unit ball;  $\omega$  is a direction and  $\Omega_n$  is a set of directions (i.e. unit sphere) in  $\mathbf{R}^n$ ; and  $\Pi_\omega^n$  is an  $n$ -dimensional test plane normal to  $\omega$ .

---

---

## Appendix D

---

---

### DIMPAL Commands

#### **composite**

`composite(variable)`

*variable* must have three layers each of type byte and all of the same dimensions and spatial location

Displays a colour composite picture in a window. Layer 1 corresponds to red, layer 2 to green, and layer 3 to blue. The window can be resized using the mouse. Clicking the right mouse button anywhere within the window causes it to return to its original size. When the window is at normal size and it gains focus, cross-hairs are displayed; the coordinates of the pixel at the centre of the cross-hairs are displayed above the menu bar of the DIMPAL parent window. As the mouse is moved the coordinates are updated. When the left mouse button is clicked anywhere within the window, the coordinates and RGB values of the pixel at the centre of the cross-hairs are written to the console window.

#### **declare**

`declare(string)`

*string* is the filename and path (delimited by double quotes) of a *.lay* file

Adds a new variable to the symbol table. In subsequent statements the variable is referenced by its identifier which is identically the filename without the extension *.lay*.

e.g.     `declare("c:\variables\pictures.lay")`

#### **describe**

`describe(variable)`

*variable* is any variable or valid layer reference; layers can be of any data type



Displays the data type, dimensions, identifier, and spatial location in  $Z^2$  for each layer of *variable*. DIMPAL uses a coordinate system for which values of the horizontal (or *x*) axis increase to the right and values of the vertical (or *y*) axis increase going up.

**discard**

`discard(variable)`

*variable* must be a variable and not a layer reference

Removes the specified variable from the symbol table. This command does not affect the file *variable.lay* however.

**display**

`display(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers must be of type byte or integer

*variable2* is a single layer variable or a valid layer reference; the layer must define a colour-map consistent with that produced by the function *colour\_map*

Displays each layer in a separate window. Each window responds to the mouse in exactly the same manner as a window generated by the *composite* command (see the comments under the *composite* command).

e.g.     `grey=colour_map("grey")`  
          `display(pictures,greyscale)`  
          `display(pictures.2,colour_map("heat"))`

**end**

`end`

Terminates a DIMPAL session. This is equivalent to choosing *exit* from the *file* menu.

**execute**

`execute(string)`

*string* is the filename and path (delimited by double quotes) of an ASCII text file containing DIMPAL statements

Executes the statements in batch.

e.g. `execute("c:\programs\segment.prg")`

## histogram

`histogram(variable)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type byte

Displays an intensity histogram, i.e. the percentage of pixels at each intensity level, in a window. When the window gains focus a vertical line cursor is activated; the intensity value for the histogram bar beneath the cursor is displayed above the menu bar of the DIMPAL parent window. The cursor can be moved along the horizontal axis (intensities 0 to 255) using the mouse.

## list

`list`

Writes the list of variables known to DIMPAL to the console window.

## overlay

`overlay(variable1,variable2,integer1,integer2,integer3)`

*variable1* must be a single layer variable or a valid layer reference; layer must be of type byte

*variable2* must be a single layer variable or a valid layer reference; layer must be of type boolean

*integer1* must be in the interval [0,255]

*integer2* must be in the interval [0,255]

*integer3* must be in the interval [0,255]

Displays *variable1*, using a grey-scale colour-map, with the binary image *variable2* superimposed. The two layers need only intersect. Boolean 1s are displayed as pixels of the colour specified by the integer parameters (RGB). Boolean 0s are transparent. An overlay window responds to the mouse in exactly the same manner as a window generated by the *composite* command (see the comments under the *composite* command).

**profiles**

`profiles(variable, integer1, integer2)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type byte

*integer1* must be a valid row number

*integer2* must be a valid column number

Displays two separate windows; one showing the profile of the brightness surface along the chosen row; the other showing the profile of the brightness surface down the chosen column.

e.g. `profiles(pictures.Mona_Lisa,23,44)`

**save\_ASCII**

`save_ASCII(variable, string)`

*variable* must be a single layer variable or a valid layer reference; layer can be of any data type

*string* is the filename and path (delimited by double quotes) of the destination file

Writes the elements of the bound matrix to an ASCII file.

**save\_GIF**

`save_GIF(variable, string)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type byte

*string* is the filename and path (delimited by double quotes) of the destination file

Saves *variable* as a GIF (Compuserve's graphics interchange format) file.

**save\_IFF**

`save_IFF(variable, string)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type byte

*string* is the filename and path (delimited by double quotes) of the destination file

Saves *variable* as an IFF ILBM (Electronic Arts' interleaved bitmap) file.

e.g. `save_IFF(pictures.Mona_Lisa, "c:\mona.iff")`

**show**

`show(variable)`

*variable* must be a single layer variable or a valid layer reference; layer can be of any data type

Displays the layer in the console window as a matrix of numbers. Missing values are indicated by asterisks.

**show\_value**

`show_value(variable,integer1,integer2)`

*variable* must be a single layer variable or a valid layer reference; layer can be of any data type

*integer1* must be a valid row number

*integer2* must be a valid column number

Displays entry (*integer1*, *integer2*) of the layer (bound matrix) in the console window.

**statistics**

`statistics(variable)`

*variable* must be a single layer variable or a valid layer reference; layer can be of any data type

Displays statistics for *variable* in the console window. These include the number of missing values, mean, sample standard deviation, and maximum and minimum values.

**text**

`text(string)`

*string* is any sequence of characters delimited by double quotes; the string cannot contain the double quote character " itself

Prints *string* to the console window.

## DIMPAL Functions

### **add\_noise**

`add_noise(variable, integer1, integer2, integer3)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type integer

*integer1* an arbitrary value

*integer2* must be greater than *integer1*

*integer3* a value in the interval [0,10]

Corrupts the layer with additive noise from a discrete uniform distribution on [*integer1*,*integer2*]. Only *integer3* × 10 percent of values in the image are affected.

e.g.     `corrupted_image=add_noise(integer(pictures.1),-10,10,3)`

### **ASCII\_file**

`ASCII_file(string)`

*string* is the filename and path (delimited by double quotes) of an ASCII text file containing descriptions of several layers followed by the actual data

Converts the specified ASCII file into a DIMPAL variable.

e.g.     `stelts=ASCII_file("c:\templates\cubeoct.txt")`  
          `dilation_by_cubeoctahedron=max(dilate(image,stelts.1),dilate(image,stelts.2))`

**FILE: cubeoct.txt**

```

LAYERS: 2
IDENTIFIER: first_plane
TYPE: INTEGER
LOCATION: (-1,1)
SIZE: 3 x 3

IDENTIFIER: second_plane
TYPE: INTEGER
LOCATION: (-1,1)
SIZE: 3 x 3

0 * 0
* * *
0 * 0

* 0 *
0 0 0
* 0 *

```

**bmp\_file**

`bmp_file(string, integer1, integer2)`

*string* is the filename and path (delimited by double quotes) of an uncompressed monochrome .bmp file

*integer1* number of rows

*integer2* number of columns

Imports the specified .bmp file as a DIMPAL variable.

**border\_correct**

`border_correct(variable)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type long

Eliminates connected components along the borders of the layer (image); the layer is assumed to contain connected components each with a separate numeric label, e.g. first connected component consists of all 1s, the second all 2s etc. (as produced by the function *separate\_components*).

e.g. `result=border_correct(separate_components(binary_image,8))`

**byte**

`byte(variable)`

*variable* is any variable or valid layer reference; layers can be of any data type

Recasts the layers to type byte. Missing values are mapped to zero.

e.g. `histogram(byte(pictures/2))`

**clip**

`clip(variable, integer1, integer2, integer3, integer4)`

*variable* is any variable or valid layer reference; layers can be of any data type

*integer1* top left x coordinate

*integer2* top left y coordinate

*integer3* bottom right x coordinate

*integer4* bottom right y coordinate

Clips *variable* to the rectangular region defined by the coordinates (*integer1, integer2*) and (*integer3, integer4*). Individual layers need only intersect with the clipping region.

**colour\_map**

`colour_map(string)`

*string* must be either "grey" or "heat"

The specified colour-map is displayed in an edit window. Individual colours can be changed using RGB sliders. When the window is closed the function returns the colour-map as a DIMPAL variable.

e.g.     `heat_map=colour_map("heat")`

**cone**

`cone(integer)`

*integer* must be in the interval [0,255]

Generates a conical structuring element of specified radius. Missing values are used to encode those elements of the resulting layer that are not part of the circular domain of the structuring element. The new layer is of type integer.

**convolve**

`convolve(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers can be of any data type

*variable2* must be a single layer variable or a valid layer reference; layer must be of type integer; layer must have odd dimensions and be smaller than each of the layers of *variable1*

This function moves a  $dy \times dx$  (dimensions of *variable2*) window across the layer(s) of *variable1* in scan order and applies the specified kernel (*variable2*). At each window position template entries are multiplied by the corresponding layer entries and the sum of products then replaces the central entry in the output layer. To counter edge effects the output layers are reduced by  $(dy-1)$  rows and  $(dx-1)$  columns. For example the application of a  $3 \times 5$  template to a particular layer produces a new layer with one row removed from both the top and bottom, and two columns removed from both the left and right.



**cylinder**

`cylinder(integer1, integer2)`

*integer1* must be in the interval [0,255]

*integer2* an arbitrary value

Generates a cylindrical structuring element of specified radius and height. Missing values are used to encode those elements of the resulting layer that are not part of the circular domain of the structuring element. The new layer is of type integer.

**dilate**

`dilate(variable1, variable2)`

*variable1* is any variable or valid layer reference; layers must all be of type integer, or all of type boolean

*variable2* is any variable or valid layer reference; if the layers of *variable1* are all of type boolean then the layers of *variable2* must all be of type boolean (i.e. SP dilation); if the layers of *variable1* are all of type integer then the layers of *variable2* can be either of type integer or type boolean (i.e. FP dilation or FSP dilation respectively)

This function conforms to the multiple layer processing conventions implemented for the dyadic operators in DIMPAL.

**disk**

`disk(integer)`

*integer* must be in the interval [0,255]

Generates a disk structuring element of specified radius. Boolean 0s are used to encode those elements of the resulting layer that are not part of the circular domain of the structuring element. The new layer is of type boolean.

**distance\_map**

`distance_map(variable,string)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type boolean  
*string* must be one of "cityblock" or "chamfer57"

Produces a layer of type long such that entry  $(i, j)$  is the value of the shortest distance between entry  $(i, j)$  of *variable* and a boolean 1 in *variable*. For the city-block metric the direct neighbours of a pixel  $p$  are deemed to be a distance of 1 unit from  $p$ , and the indirect neighbours of  $p$  are deemed to be a distance of 2 units from  $p$ . For the chamfer 5-7 metric the direct neighbours of  $p$  are deemed to be a distance of 5 units from  $p$ , and the indirect neighbours of  $p$  are deemed to be a distance of 7 units from  $p$ . This metric approximates five times the Euclidean distance because  $5^2 + 5^2 \approx 7^2$ .

**equalise**

`equalise(variable)`

*variable* is any variable or valid layer reference; layers must all be of type byte

Applies histogram equalisation to each of the layers.

**erode**

`crode(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers must all be of type integer or all of type boolean

*variable2* is any variable or valid layer reference; if the layers of *variable1* are all of type boolean then the layers of *variable2* must all be of type boolean (i.e. SP erosion); if the layers of *variable1* are all of type integer then the layers of *variable2* can be either of type integer or type boolean (i.e. FP erosion or FSP erosion respectively)

This function conforms to the multiple layer processing convention implemented for the dyadic operators in DIMPAL. The images to be eroded correspond to *variable1*. The structuring elements correspond to *variable2*.

**filter**

`filter(variable,string,integer1,integer2)`

*variable* is any variable or valid layer reference; layers can be of any data type

*string* must be one of "mean", "median", "max", or "min"

*integer1* must be odd and positive (number of window rows)

*integer2* must be odd and positive (number of window columns)

This function moves an  $integer1 \times integer2$  window across the layer(s) of *variable* in scan order. At each window position the specified operation is applied to the entries within the window and the result then replaces the central entry in the output layer. To counter edge effects the output layers are reduced by ( $integer1-1$ ) rows and ( $integer2-1$ ) columns.

**Gaussian\_stretch**

`Gaussian_stretch(variable,real)`

*variable* is any variable or valid layer reference; layers must all be of type byte

*real* specifies the number of standard deviations from the mean

Modifies the shape of the intensity histogram to match as closely as possible that of the standard normal curve defined by  $\pm real$  standard deviations.

e.g. `contrast_enhanced=Gaussian_stretch(pictures,2.5)`

**group**

`group(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers can be of any data type

*variable2* is any variable or valid layer reference; layers can be of any data type

Appends the layers of *variable2* to those of *variable1*.

e.g. `new_variable=group(pictures,equalise(pictures))`

**histogram**

`histogram(variable)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type byte

Produces a row vector with 256 elements of type long. The first element represents the number of pixels of *variable* that have grey-level 0, the second grey-level 1, etc.

c.g. `save_ASCII(histogram(pictures.1),"c:\minitab.dat")`

**integer**

`integer(variable)`

*variable* is any variable or valid layer reference; layers can be of any data type

Recasts the layers to type integer. Missing values are unaffected.

**mask**

`mask(variable1,variable2)`

*variable1* must be a single layer variable or a valid layer reference; layers must be of type byte

*variable2* must be a single layer variable or a valid layer reference; layers must be of type boolean

The boolean layer *variable2* is superimposed on the layer *variable1* (the two layers need only intersect). A new layer is created that contains the elements of *variable1* wherever *variable2* is a boolean 1, and missing values otherwise.

**max**

`max(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers must be of type integer

*variable2* is any variable or valid layer reference; layers must be of type integer; *variable2* must have as many layers as *variable1*; corresponding layers must have the same dimensions

Produces a new variable such that for a particular layer, entry  $(i,j)$  is the maximum of the corresponding entries in the corresponding layers of *variable1* and *variable2*.

**max\_value**

`max_value(variable)`

*variable* must be a single layer variable or a valid layer reference; layer must be of type integer

Produces an integer layer containing a single value; the maximum of all of the elements of *variable*.

e.g.     `result=threshold(pictures.1,maximum(gradient))`

**measure\_components**

`measure_components(variable,real1,real2)`

*variable* is any variable or valid layer reference; layers must be of type long

*real1* is the width of a pixel (e.g. 0.15 mm)

*real2* is the height of a pixel

Produces a variable with layers of type double. Each layer has 6 rows and as many columns as there are connected components in the corresponding layer of *variable*; each layer of *variable* is assumed to contain connected components with separate numeric labels, e.g. first connected component consists of all 1s, the second all 2s etc. (as produced by the function *separate\_components*). For each new layer, the first row contains area measurements; the second perimeter measurements calculated using the  $U^*$  estimator of Section 4.7.; the third perimeter measurements calculated using the  $U^{**}$  estimator of Section 4.7.; the fourth perimeter measurements calculated using the estimator  $\frac{1}{2}(U^*(X)+U^{**}(X))$ ; the fifth connectivity numbers; and the sixth  $P2A$  shape numbers.

**min**

`min(variable1,variable2)`

*variable1* is any variable or valid layer reference; layers must be of type integer

*variable2* is any variable or valid layer reference; layers must be of type integer; *variable2* must have as many layers as *variable1*; corresponding layers must have the same dimensions

Produces a new variable such that for a particular layer, entry  $(i,j)$  is the minimum of the corresponding entries in the corresponding layers of *variable1* and *variable2*.

**retain\_components**

`retain_components(variable1,variable2)`

*variable1* must be a single layer variable or a valid layer reference; layer must be of type long  
*variable2* must be a single layer variable or a valid layer reference; layer must be of type boolean

The layer *variable1* is assumed to contain connected components each with a separate numeric label, e.g. first connected component consists of all 1s, the second all 2s etc. (as produced by the function *separate\_components*). The layer of *variable2* is assumed to be a row vector (even if this is not the case only the first row is used by the function). The first element of the vector corresponds to component 1, the second component 2, etc. If an element is a boolean 1 then the corresponding connected component is retained in the output layer.

**saturate**

`saturate(variable)`

*variable* is any variable or valid layer reference; layers can be of any data type

Fits the values of *variable* to the interval [0,255] using linear interpolation.

e.g. `display(saturate(pictures),grey)`

**separate\_components**

`separate_components(variable,integer)`

*variable* is any variable or valid layer reference; layers must be of type boolean

*integer* must be either 4 or 8

Used to individually label the connected components (*integer* determines the type of connectivity to be used) in each layer of *variable*. The resulting variable has layers of type long. Numeric labels begin at 1 for each layer.

**skeleton**

`skeleton(variable, integer)`

*variable* is any variable or valid layer reference; layers must be of type boolean

*integer* must be either 0 or 1

Thins each of the layers to produce 8-connected homotopic skeletons. This function utilises the parallel thinning algorithm devised by Zhang and Suen (1984). The value of *integer* indicates whether values exterior to the layers are to be treated as binary 1 or 0.

**sphere**

`sphere(integer)`

*integer* must be in the interval [0,255]

Generates a spherical structuring element of specified radius. Missing values are used to encode those elements of the resulting layer that are not part of the circular domain of the structuring element. The new layer is of type integer.

**threshold**

`threshold(variable1, variable2)`

*variable1* must be a single layer variable or a valid layer reference; layer must be of type integer

*variable2* must be a single layer variable or a valid layer reference; layer must be of type integer; layer must contain only one value

Produces a boolean image with 1s everywhere *variable1* is greater than or equal to the value in *variable2*.

c.g.     `result=threshold(image,max_value(gradient))`

**translate**

`translate(variable, integer1, integer2)`

*variable* is any variable or valid layer reference; layers can be of any data type

*integer1* an arbitrary value

*integer2* an arbitrary value

Translates each layer by *integer1* pixels in the *x* direction and *integer2* pixels in the *y* direction. This function only adjusts the spatial coordinates of each bound matrix. If the top left entry of a particular bound matrix has coordinates (*x*,*y*) then this function returns a layer with coordinates (*x* + *integer1*, *y* + *integer2*).



---

---

## Appendix E

---

---

### The Syntax of a DIMPAL Statement

```
<statement>::=<identifier> = <disjunction> |  
           <identifier> |  
           <identifier>({<parameter>}{,<parameter>})
```

```
<disjunction>::=<conjunction> |  
              <conjunction> || <disjunction> |  
              <conjunction> [||] <disjunction>
```

```
<conjunction>::=<expression> |  
              <expression> && <conjunction> |  
              <expression> [&&] <conjunction>
```

```
<expression>::=<simple expression> |  
              <simple expression> < <expression> |  
              <simple expression> [<] <expression> |  
              <simple expression> <= <expression> |  
              <simple expression> [<=] <expression> |  
              <simple expression> == <expression> |  
              <simple expression> [==] <expression> |  
              <simple expression> != <expression> |  
              <simple expression> [!=] <expression> |  
              <simple expression> >= <expression> |  
              <simple expression> [>=] <expression> |  
              <simple expression> > <expression> |  
              <simple expression> [>] <expression>
```

```
<simple expression>::=<term> |  
                  <term> + <simple_expression> |  
                  <term> [+] <simple_expression> |  
                  <term> - <simple_expression> |  
                  <term> [-] <simple_expression>
```

---

```

<term>::=<factor> |
        <factor> * <term> |
        <factor> [*] <term> |
        <factor> / <term> |
        <factor> [/] <term>

```

```

<factor>::=<subfactor> |
        <subfactor> ^ <factor> |
        <subfactor> [^] <factor>

```

```

<subfactor>::=<constant> |
        <variable> |
        <function> |
        (<disjunction>) |
        -<subfactor> |
        !<subfactor>

```

```

<constant>::=<integer> |
        <real>

```

```

<variable>::=<identifier> |
        <identifier> . <identifier> |
        <identifier> . <integer>

```

```

<function>::=<identifier> ({<parameter>},{<parameter>}) |
        [<identifier>] ({<parameter>},{<parameter>})

```

```

<parameter>::=<string> |
        <disjunction>

```

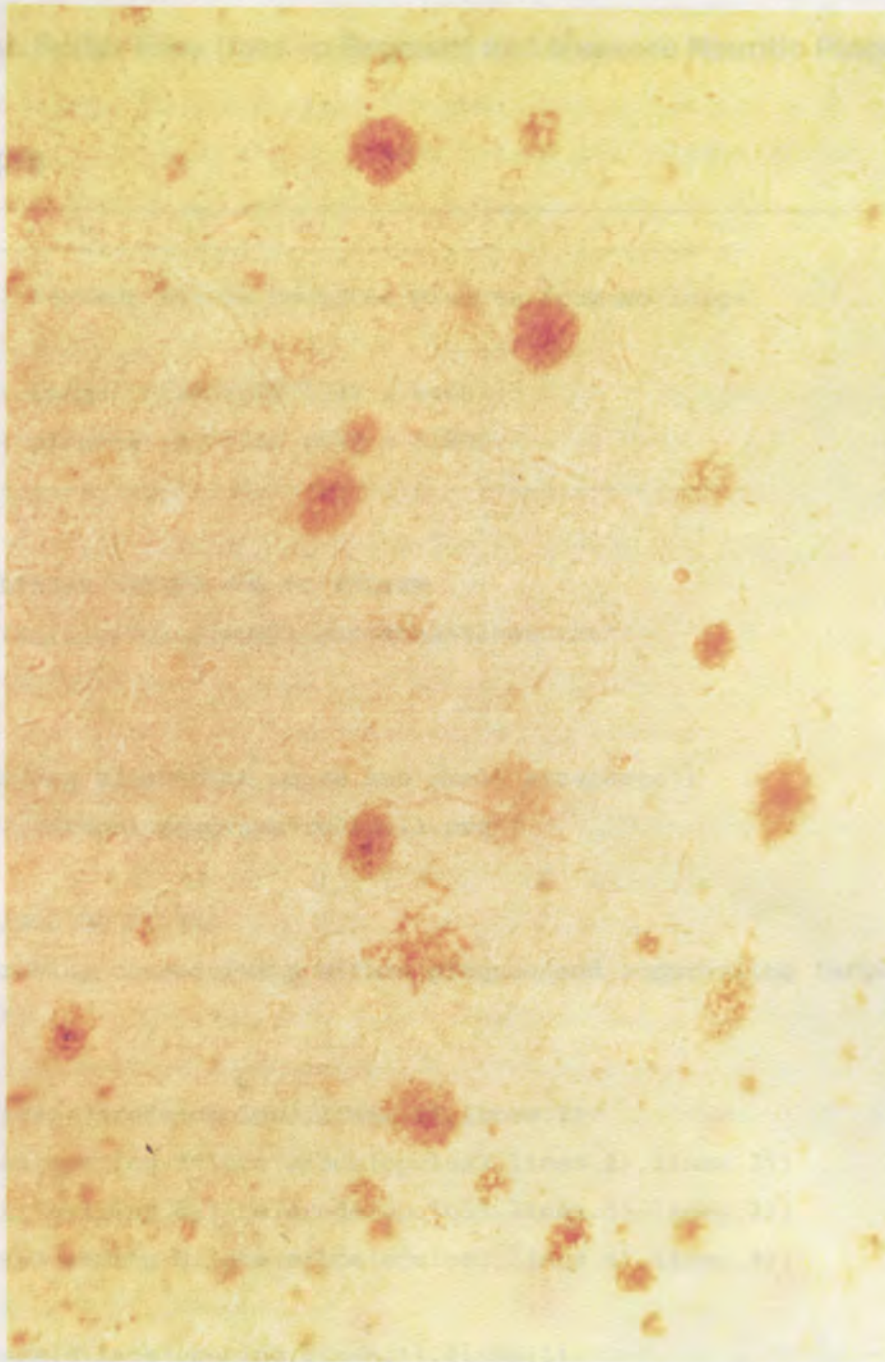
---

---

## Appendix F

---

---



*Figure F-1.* Light micrograph of Alzheimer brain tissue, at 200 $\times$  magnification, stained with monoclonal antibodies against  $\beta$ A4 amyloid. The section is about 20 to 30  $\mu$ m thick, and originates in the cerebral cortex.

---

---

## Appendix G

---

---

### DIMPAL Script Files Used to Segment and Measure Neuritic Plaques

#### segment.prg

```
-----  
;SYNOPSIS: Segment mature neuritic plaques from an image  
;  
;   INPUT: image    (INTEGER 480V x 640H)  
;   OUTPUT: plaques (BOOLEAN 458V x 618H)  
-----  
  
; DEFINE LINEAR STELTS FOR OC FILTER  
lines=ASCII_file("c:\dimpal\templates\lines.txt")  
  
; ASF  
text("Removing electronic noise and small artefacts")  
execute("c:\dimpal\programs\rollball.prg")  
  
; GENERALISED OC FILTER  
text("Improving connectivity within plaques and suppressing larger  
artefacts")  
  
opening=dilate(erode(opclos3,lines.1),lines.1)  
opening=max(opening,dilate(erode(opclos3,lines.2),lines.2))  
opening=max(opening,dilate(erode(opclos3,lines.3),lines.3))  
opening=max(opening,dilate(erode(opclos3,lines.4),lines.4))  
  
opclos=erode(dilate(opening,lines.1),lines.1)  
opclos=min(opclos,erode(dilate(opening,lines.2),lines.2))  
opclos=min(opclos,erode(dilate(opening,lines.3),lines.3))  
opclos=min(opclos,erode(dilate(opening,lines.4),lines.4))  
  
; TOPHAT TRANSFORM
```

```

text("Normalising the background")
stelt=disk(3)

closing=dilate(dilate(dilate(dilate(dilate(dilate(dilate(dilate(opclos
,stelt),stelt),stelt),stelt),stelt),stelt),stelt),stelt)
closing=dilate(dilate(dilate(dilate(dilate(dilate(dilate(dilate(closin
g,stelt),stelt),stelt),stelt),stelt),stelt),stelt),stelt)
closing=dilate(dilate(dilate(dilate(dilate(dilate(dilate(dilate(closin
g,stelt),stelt),stelt),stelt),stelt),stelt),stelt),stelt)

closing=erode(erode(erode(erode(erode(erode(erode(erode(closing,stelt)
,stelt),stelt),stelt),stelt),stelt),stelt),stelt)
closing=erode(erode(erode(erode(erode(erode(erode(erode(closing,stelt)
,stelt),stelt),stelt),stelt),stelt),stelt),stelt)
closing=erode(erode(erode(erode(erode(erode(erode(erode(closing,stelt)
,stelt),stelt),stelt),stelt),stelt),stelt),stelt)

text("Performing the tophat transform")
tophat=closing[-]opclos

; MORPHOLOGICAL GRADIENT
text("Calculating morphological gradient for automatic thresholding")
gradient=dilate(tophat,disk(1))[-]erode(tophat,disk(1))

; LABEL CONNECTED COMPONENTS AND PERFORM BORDER CORRECTION
text("Find connected components and perform border correction")
threshold_value=max_value(gradient)/2
components=border_correct(separate_components(clip(threshold(tophat,th
reshold_value),11,-11,628,-468),8))

; RETAIN ONLY MATURE PLAQUES
text("Keep only mature plaques")
to_be_retained=threshold(tophat,integer(max_value(tophat)*0.75))[*]com
ponents
to_be_retained=clip(histogram(byte(to_be_retained)),1,0,640,0)>0
components=retain_components(components,to_be_retained)

text("Remove components with connectivity # = 0 and shape factor > 3")

```

```
measure=measure_components(components,1,1)
to_be_retained=clip(measure,0,-4,640,-4)!=0
to_be_retained=(translate(clip(measure,0,-5,640,-5),0,1)<=3)||to_be_retained
components=retain_components(components,to_be_retained)

; REJECT SMALL PLAQUES
text("Eliminate plaques that are too small")
plaques=dilate(erode(components>0,disk(10)),disk(10))

; OUTLINE MATURE PLAQUES
text("Overlay outline of plaques on original image")
outline=dilate(plaques,disk(1))&&!erode(plaques,disk(1))
image=saturate(byte(image))
overlay(image,outline,255,255,255)
```

**lines.txt**

```
LAYERS: 4
IDENTIFIER: line_135
TYPE: BOOLEAN
LOCATION: (-5,5)
SIZE: 11 x 11
```

```
IDENTIFIER: line_45
TYPE: BOOLEAN
LOCATION: (-5,5)
SIZE: 11 x 11
```

```
IDENTIFIER: line_90
TYPE: BOOLEAN
LOCATION: (-5,5)
SIZE: 11 x 11
```

```
IDENTIFIER: line_0
TYPE: BOOLEAN
LOCATION: (-5,5)
SIZE: 11 x 11
```

```
10000000000
01000000000
00100000000
00010000000
00001000000
00000100000
00000010000
00000001000
00000000100
00000000010
00000000001
```

```
00000000001
00000000010
00000000100
```

00000001000  
00000010000  
00000100000  
00001000000  
00010000000  
00100000000  
01000000000  
10000000000

00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000  
00000100000

00000000000  
00000000000  
00000000000  
00000000000  
00000000000  
11111111111  
00000000000  
00000000000  
00000000000  
00000000000  
00000000000



**rollball.prg**

```
;-----  
;SYNOPSIS: ASF -- sequence of open-closings using  
;           spheres of radii 1 to 3  
;  
;   INPUT: image   (INTEGER)  
;   OUTPUT: opclos3 (INTEGER)  
;-----  
  
text("Performing open-closing with sphere of radius 1")  
stelt=sphere(1)  
opening=dilate(erode(integer(image),stelt),stelt)  
opclos1=erode(dilate(opening,stelt),stelt)  
  
text("Performing open-closing with sphere of radius 2")  
stelt=sphere(2)  
opening=dilate(erode(opclos1,stelt),stelt)  
opclos2=erode(dilate(opening,stelt),stelt)  
  
text("Performing open-closing with sphere of radius 3")  
stelt=sphere(3)  
opening=dilate(erode(opclos2,stelt),stelt)  
opclos3=erode(dilate(opening,stelt),stelt)
```

---

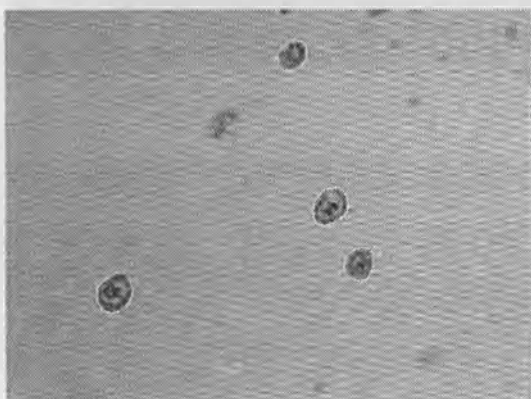
---

## *Appendix H*

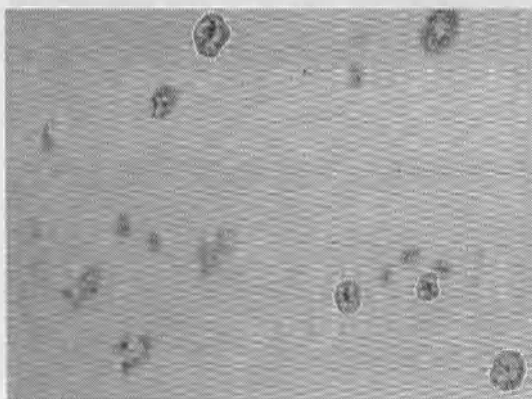
---

---

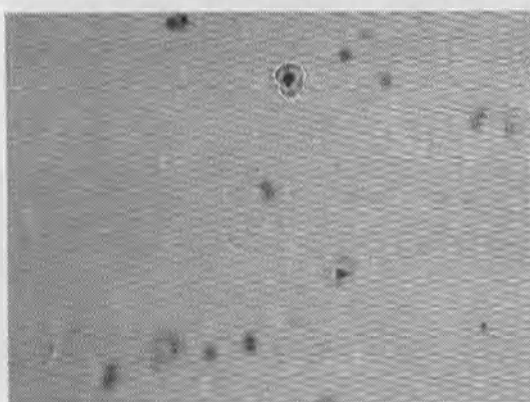
### **Segmented Images**



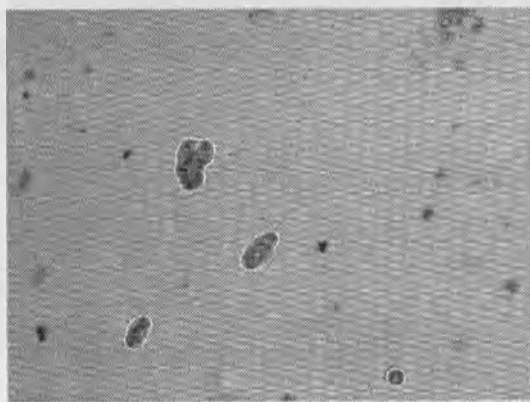
(a)



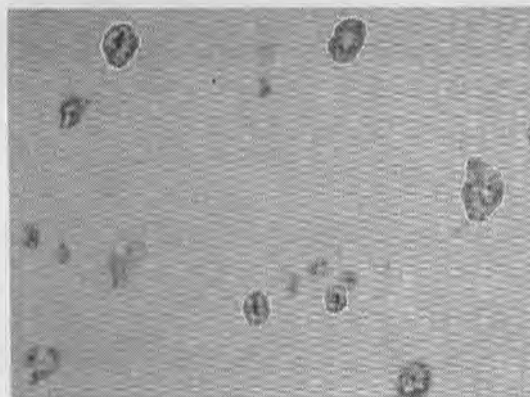
(b)



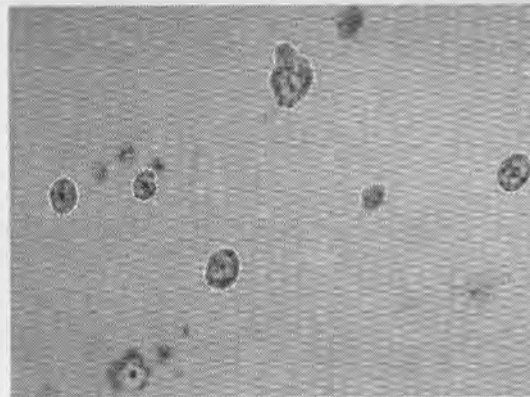
(c)



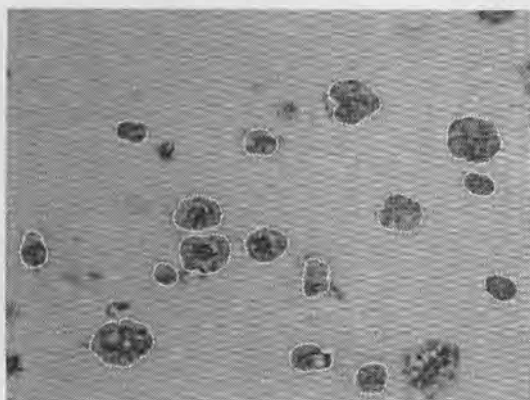
(d)



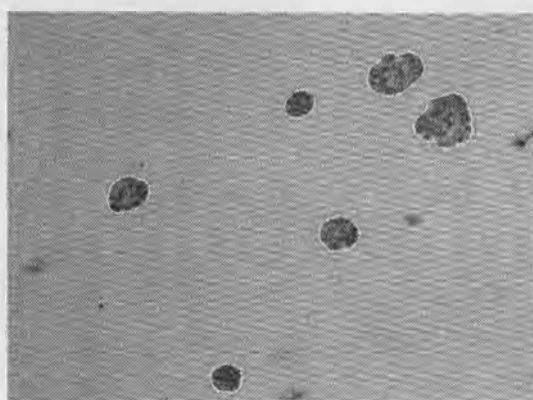
(e)



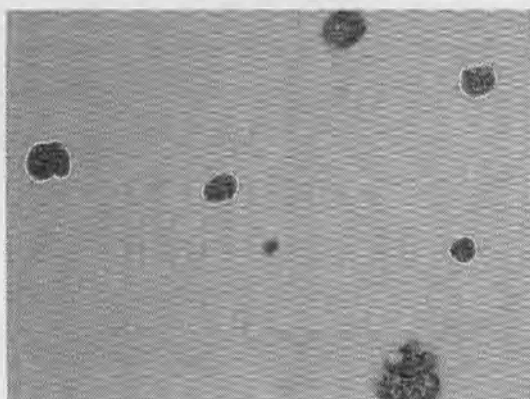
(f)



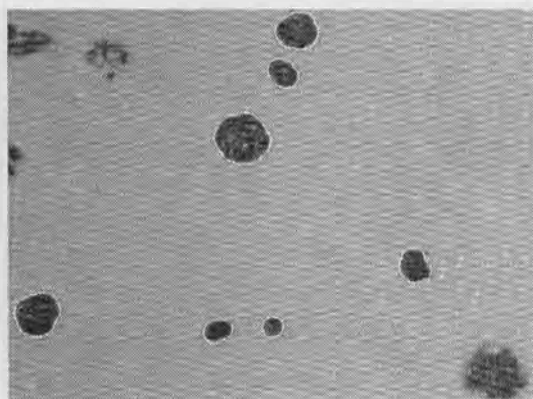
(g)



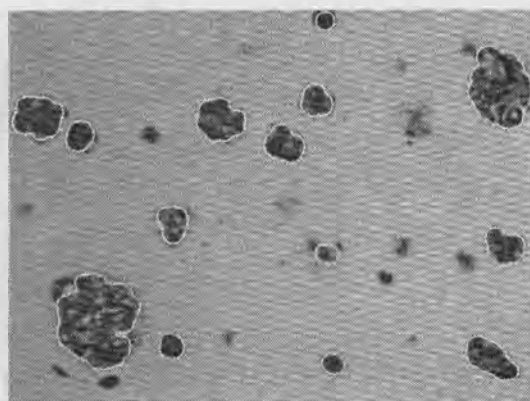
(h)



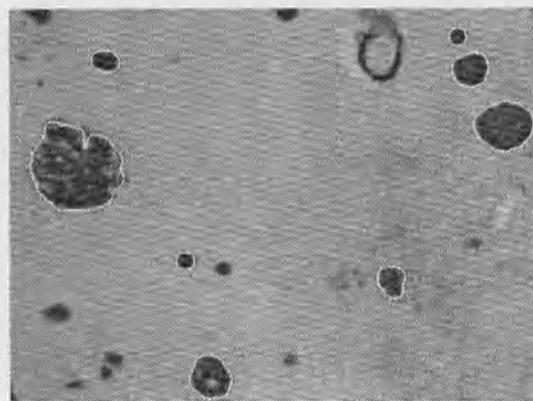
(i)



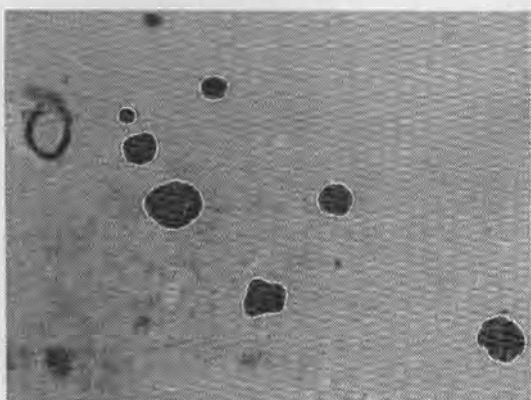
(j)



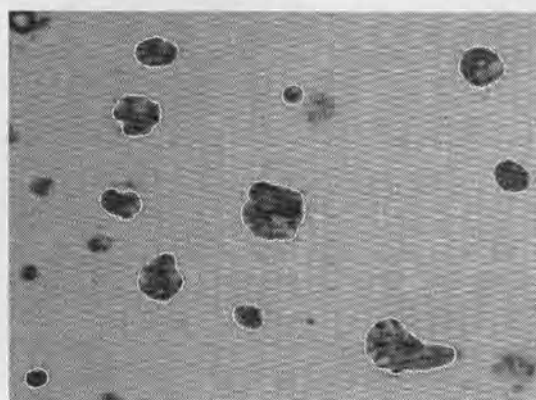
(k)



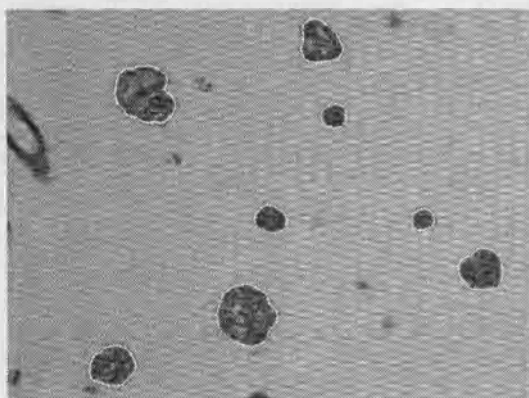
(l)



(m)



(n)



(o)