

Cooperative Research Centre for Sensor Signal and Information Processing



Image Analysis for the Study of Chromatin Distribution in Cell Nuclei

with Application to Cervical Cancer Screening

Andrew James Heinrich Mehnert

B.App.Sc. Hons. 1 (Mathematics), Edith Cowan University (ECU), Western Australia M.App.Sc. (Mathematics and Planning), ECU

A thesis submitted to the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia In fulfilment of the requirements for the degree of Doctor of Philosophy (Ph.D.), November, 2003

Statement of Originality

The work presented in this thesis is, to the best of my knowledge and belief, original and my own work, except as acknowledged in the text. This material has not been submitted, either in whole or in part, for a degree at this or any other university.

Andrew J. H. Mehnert:

Keywords

- Chromatin
- Cervical cancer
- Cytology
- Papanicolaou test
- Automated screening
- Malignancy associated changes
- Statistical pattern recognition
- Texture features
- Mathematical morphology
- Complete lattice
- Self-dual filtering
- Segmentation
- Chromatin segmentation
- Seeded region growing
- Watershed transform
- Priority queue

- Adjacency graph
- Co-occurrence matrix
- Voronoi diagram
- Distance transform
- Hexagonal grid
- Skeleton by influence zones
- Minkowski functionals
- Curse of dimensionality
- Receiver operating characteristic curve
- Logistic regression
- Margination
- MACs-based classifier

Abstract

This thesis describes a set of image analysis tools developed for the purpose of quantifying the distribution of chromatin in (light) microscope images of cell nuclei. The distribution or pattern of chromatin is influenced by both external and internal variations of the cell environment, including variations associated with the cell cycle, neoplasia, apoptosis, and malignancy associated changes (MACs). The quantitative characterisation of this pattern makes possible the prediction of the biological state of a cell, or the detection of subtle changes in a population of cells. This has important application to automated cancer screening.

The majority of existing methods for quantifying chromatin distribution (texture) are based on the stochastic approach to defining texture. However, it is the premise of this thesis that the structural approach is more appropriate because pathologists use terms such as clumping, margination, granulation, condensation, and clearing to describe chromatin texture, and refer to the regions of condensed chromatin as granules, particles, and blobs. The key to the structural approach is the segmentation of the chromatin into its texture primitives. Unfortunately all of the chromatin segmentation algorithms published in the literature suffer from one or both of the following drawbacks: (i) a segmentation that is not consistent with a human's perception of blobs, particles, or granules; and (ii) the need to specify, a priori, one or more subjective operating parameters. The latter drawback limits the robustness of the algorithm to variations in illumination and staining quality.

The structural model developed in this thesis is based on several novel low-, medium-, and high-level image analysis tools. These tools include: a class of non-linear self-dual filters, called folding induced self-dual filters, for filtering impulse noise; an algorithm, based on seeded region growing, for robustly segmenting chromatin; an improved seeded region growing algorithm that is independent of the order of pixel processing; a fast priority queue implementation suitable for implementing the watershed transform (special case of seeded region growing); the adjacency graph attribute co-occurrence matrix (AGACM) method for quantifying blob and mosaic patterns in the plane; a simple and fast algorithm for computing the exact Euclidean distance transform for the purpose of deriving contextual features (measurements) and constructing geometric adjacency graphs for disjoint connected components; a theoretical result establishing an equivalence between the distance transform of a binary image and the grey-scale erosion of its characteristic function by an elliptic poweroid structuring element; and a host of chromatin features that can be related to qualitative descriptions of chromatin distribution used by pathologists.

In addition, this thesis demonstrates the application of this new structural model to automated cervical cancer screening. The results provide empirical evidence that it is possible to detect differences in the pattern of nuclear chromatin between samples of cells from a normal Papanicolaou-stained cervical smear and those from an abnormal smear. These differences are supportive of the existence of the MACs phenomenon. Moreover the results compare favourably with those reported in the literature for other stains developed specifically for automated cytometry. To the author's knowledge this is the first time, based on a sizable and uncontaminated data set, that MACs have been demonstrated in Papanicolaou stain. This is an important finding because the primary screening test for cervical cancer, the Papanicolaou test, is based on this stain.

Table of Contents

St	atem	ent of	Originality	iii	
K	eywo	\mathbf{rds}		\mathbf{v}	
\mathbf{A}	Abstract vi			vii	
Li	List of Figures xv				
Li	st of	Tables	3	xxii	
Notation xxv Abbreviations xxi				xxv	
				xxix	
P۱	Publications xx: Original Contributions xx:				
O					
A	cknov	wledge	ments	xxxv	
1	Intr	oducti	on	1	
	1.1	The P	apanicolaou test	3	
		1.1.1	The origin and nature of the Papanicolaou smear	3	
		1.1.2	Accuracy of the Pap test	4	
		1.1.3	Automated screening	5	

	1.2	Malignancy associated changes
	1.3	Statistical pattern recognition in computer vision
	1.4	Texture features
		1.4.1 Defining texture
		1.4.1.1 Structural approach
		1.4.1.2 Stochastic approach
		1.4.2 Texture feature extraction
	1.5	Problem definition
	1.6	Aim and objectives
	1.7	Scope of the research
	1.8	Structure of this thesis
2	The	Poretical Framework 29
4	2 1	Sets and hinary relations 30
	2.1	Feujvalonce relations
	2.2	Order relations
	2.0	2 3 1 Hasse diagrams 32
		2.3.1 Thasse diagrams
		2.3.2 The principle of duality with respect to order
		2.5.5 Quasi-ordering
	0.4	2.3.4 Isomorphism
	2.4	Lattices \dots
		2.4.1 Set theory definition of a lattice
		2.4.2 Algebra definition of a lattice
		2.4.3 Properties of lattices
		2.4.4 Modular and distributive lattices
		2.4.5 Complete lattices
		2.4.5.1 Complete Boolean lattice
		$2.4.5.2 \text{Power lattice} \dots \dots \dots \dots \dots \dots \dots 39$
	2.5	Operators on a complete lattice

	2.6	Mathe	matical m	orphology for complete lattices	43
		2.6.1	Adjuncti	ons	44
		2.6.2	Represen	tation theorem for increasing mappings	45
		2.6.3	Morphole	ogical filters	45
			2.6.3.1	Overfilters, underfilters, inf-overfilters, sup-underfilters, inf-filters, sup-filters, strong filters	46
	2.7	Mathe	matical m	orphology for binary images	47
		2.7.1	Represen ant mapp	tation theorem for increasing and translation invari-	49
		2.7.2	Three du	alities for binary dilations and erosions	49
	2.8	Mathe	matical m	orphology for grey-scale images	49
		2.8.1	Represen ant mapp	tation theorems for increasing and translation invari-	55
		2.8.2	Three du	alities for grey-scale dilations and erosions \ldots .	55
	2.9	Mathe	matical m	orphology for metric spaces	56
		2.9.1	Discrete	metric spaces	58
	2.10	Mathe	matical m	orphology for graphs	60
		2.10.1	Structuri	ng graphs	64
	2.11	Summ	ary		66
3	Fold	ling In	duced Se	elf-dual Filters	67
	3.1	Review	v of non-li	near self-dual operators and filters	70
		3.1.1	Self-dual	operators based on the activity ordering	72
			3.1.1.1	The centre operator	73
			3.1.1.2	Finite iterations of increasing centres and the middle filter	74
		3.1.2	Self-dual	toggle mappings	76
		3.1.3	Self-dual	morphological operators based on the switch operator	78
			3.1.3.1	Construction of morphological filters from switch operators	79

		3.1.4	Self-dual operators based on folded ordering $\ldots \ldots \ldots$	80
		3.1.5	Self-dual annular filters	81
	3.2	New a	pproach: Folding induced self-dual filters	83
		3.2.1	Fold-space and the folding operator	84
		3.2.2	Fold-space meta-operators	86
		3.2.3	Folding induced self-dual filters	88
		3.2.4	Constructing FISFs	88
		3.2.5	Implementation issues	92
	3.3	Discus	sion and summary	92
1	Chr	omotiv	Segmentation	05
4	4 1	Samaa	n Segmentation	9 0
	4.1	Segme	ntation methods	90
	4.2	The na	ature of chromatin	.06
		4.2.1	Fixation and staining	.06
		4.2.2	Absorbance, extinction, optical density	.07
		4.2.3	Chromatin structure as visualised by light microscopy 1	.08
	4.3	Review	v of previous approaches to chromatin segmentation $\ldots \ldots \ldots 1$.09
		4.3.1	Summary and conclusion	31
	4.4	New a	lgorithm for chromatin segmentation	33
		4.4.1	Rationale	33
		4.4.2	Description of the new algorithm	38
		4.4.3	Preferred embodiment	39
		4.4.4	Discussion	42
	4.5	The A	dams and Bischof seeded region growing algorithm 1	44
		4.5.1	Description of the algorithm	44
		4.5.2	Inherent order dependencies	45
		4.5.3	Implementation order dependencies	45
	4.6	Improv	ved seeded region growing algorithm 1	47
		4.6.1	Implementation of the ISRG	48

		4.6.2	A comment on boundary flagging	19
		4.6.3	Implementing a scan-order independent version of the water- shed transform	19
		4.6.4	Summary and conclusion	53
	4.7	A new	v fast priority queue for watershed segmentation	53
	4.8	Summ	ary	58
-	Б	,		24
5	Rep	oresent	ation and Description 16	51
	5.1	Adjace	ency graphs	53
	5.2	The V	oronoi diagram and related graphs	34
		5.2.1	The Delaunay graph	37
		5.2.2	Graphs related to the Delaunay triangulation	38
		5.2.3	β -skeletons	71
		5.2.4	A comment on structural stability	74
	5.3	The a	rea Voronoi diagram	74
	5.4	The d	istance transform	75
	5.5	Comp	uting the exact EDT on rectangular and hexagonal grids 18	33
		5.5.1	The hexagonal grid versus the square grid	34
		5.5.2	An equivalence between grey-scale erosion and distance trans-	
			formation	35
		5.5.3	The algorithm of Huang & Mitchell	37
		5.5.4	Adaptation to the hexagonal grid	39
			5.5.4.1 A comment on integer arithmetic)1
	5.6	Skelet	on by influence zones and perceptual graphs)1
	5.7	The A	djacency Graph Attribute Co-occurrence Matrix)3
		5.7.1	The grey-level co-occurrence matrix)5
		5.7.2	The adjacency graph attribute co-occurrence matrix 19)6
	5.8	Measu	urement of region attributes)8
		5.8.1	Minkowski functionals)0

		5.8.2	Estimat	es of the Minkowski functionals for $X \in \mathcal{P}\left(\mathbb{Z}^2\right)$ 202
		5.8.3	Grey-sca	ale images and dimensional measurements $\ldots \ldots \ldots 207$
		5.8.4	Other p	arameters/functionals
	5.9	Summ	ary	
6	Ар	olicatio	on to the	e Automated Screening of Cytology Slides 213
	6.1	Why ł	ouild a M	ACs-based screener?
	6.2	How t	o build a	MACs-based classifier?
	6.3	Overv	iew of sta	tistical pattern recognition
		6.3.1	The cur	se of dimensionality
		6.3.2	Dimensi	onality reduction
			6.3.2.1	Feature extraction
			6.3.2.2	Feature selection
		6.3.3	Choice of	of classifier
		6.3.4	Evaluati	ng classifier performance
	6.4	Case s	study: A	MACs-based classifier for Pap smear screening \ldots 234
		6.4.1	Cancer of	of the uterine cervix $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 234$
		6.4.2	Cells of	interest for MAC analysis
		6.4.3	Cytolog	y slides
		6.4.4	Image a	cquisition
		6.4.5	Image p	rocessing and analysis
			6.4.5.1	Artefact rejection
			6.4.5.2	Nucleus and chromatin features
			6.4.5.3	Nucleus statistics
		6.4.6	Culling	of slides unsuitable for MAC analysis
		6.4.7	Classific	ation using logistic regression
			6.4.7.1	Using a fitted logistic regression model as a classifier 254
			6.4.7.2	Comparison with Fisher's linear discriminant function 255
			6.4.7.3	Logistic regression analysis software

		6.4.8	Sample size considerations	256
		6.4.9	Experiment 1: Classification with a single nucleus-feature his- togram	256
			6.4.9.1 Interpretation of the results of Experiment 1	262
		6.4.10	Experiment 2: Feature selection	268
			6.4.10.1 Interpretation of the results of Experiment 2	272
	6.5	Summ	ary and discussion	273
7	Sun	nmary	and Conclusions	275
	7.1	Thesis	review	275
	7.2	Key co	ontributions and findings	280
	7.3	Implic	ations	281
	7.4	Limita	tions	282
	7.5	Oppor	tunities for further research	285
Bi	bliog	graphy		291
\mathbf{A}	DIn	nPAL 4	4.0	317
	A.1	Techni	cal details / history	319
	A.2	Lexica	l conventions	323
		A.2.1	Identifiers	323
		A.2.2	Constants	324
		A.2.3	Missing value constant	324
		A.2.4	Operators	324
		A.2.5	Meta-operator	324
		A.2.6	Separators	325
	A.3	Langu	age grammar	325
	A.4	Variab	les and layers	328
		A.4.1	Data types	328

	A.4.3 Creating and managing variables	330
	A.4.4 Creating .dpl Files	331
	A.4.5 Scalar variables	334
	A.4.6 String variables	334
	A.5 Sample program	334
в	The Watershed Transform	335
\mathbf{C}	FISF Implementation for MICROMORPH	339
D	DImPAL Programs	345
	D.1 Implementation of the new chromatin segmentation algorithm \ldots	345
	D.2 Additional code for constructing the generalised Delaunay graph	345
\mathbf{E}	Fast Priority Queue Implementation in C	347
\mathbf{F}	Minkowski Functionals in \mathbb{R}^n	351
\mathbf{G}	QML AutoCyte® Slide Diagnoses	353
н	Danielsson's G Shape Factor	361
Ι	R Program for Experiment 1	363
J	R Program for Experiment 2	367
K	R program for Empirical ROC Curve Analysis	371
\mathbf{L}	Slide Features	375
М	Slide Feature Box-and-Whisker Plots	379

List of Figures

1.1	The elements of an image analysis system	8
1.2	Textures that can be modelled using the structural approach	12
1.3	Textures that can be modelled using the stochastic approach	12
1.4	The interrelation between the various second-order statistics	14
1.5	Examples of symmetrical MRF neighbourhoods	15
1.6	Examples of MRF cliques.	16
1.7	Nuclei from several Papanicolaou-stained cervical cells as visualised	
	by light microscopy	21
1.8	Scene and nucleus segmentation	24
1.9	Flowchart summarising the aims and objectives of the thesis	28
2.1	Hasse diagrams	33
2.2	Closed balls in \mathbb{R}^2	58
2.3	Interpretations of \mathbb{Z}^2 in \mathbb{R}^2	59
2.4	Closed balls in \mathbb{Z}^2	60
2.5	Elementary graph dilation and erosion	63
2.6	Structuring graph defining the elementary neighbourhood N_1	65
3.1	Hasse diagrams showing natural and folded ordering	80
3.1 3.2	Hasse diagrams showing natural and folded ordering	80 82

3.4	FISF filtering when significant parts of the image are at either extreme of the grey-value range
4.1	Illustration of global thresholding
4.2	Papanicolaou-stained cervical cells
4.3	Illustration of the segmentation algorithm of Smeulders et al. (1978). 115
4.4	Illustration of the segmentation algorithm of Mever (1978)
4.5	Illustration of the ricefield transformation and its comparison with the watershed transform
4.6	Illustration of the segmentation algorithm of Komitowski & Zinser (1985)
4.7	Illustration of the segmentation algorithm of Kondo & Taniguchi (1986).124
4.8	Illustration of the segmentation algorithm of Young et al. (1986) 125
4.9	Illustration of the segmentation algorithm of Beil (1992)
4.10	Illustration of the segmentation algorithm of Wolf et al. (1995) 128
4.11	Representing a grey-scale image as a topographic relief
4.12	Flowchart showing the steps of the new chromatin segmentation al- gorithm
4.13	Demonstration of the new chromatin segmentation algorithm 141
4.14	Segmentation of dark particles using the preferred embodiment of the proposed chromatin segmentation algorithm
4.15	Detection of light particles
4.16	Order dependency of the SRG algorithm
4.17	ISRG algorithm applied to the test image in Figure 4.16(a) 147
4.18	Boundary flagging
4.19	Demonstration of the scan-order independent watershed transform. 152
4.20	Order independence of the ISRG implementation
4.21	Schematic of the new priority queue for the watershed transform 157
5.1	Ordinary Voronoi diagram in \mathbb{R}^2

5.2	Voronoi graph constructed from the Voronoi diagram shown in Fig-
	ure 5.1
5.3	Constructing the geometric-dual of a plane graph
5.4	The Delaunay graph derived from the Voronoi diagram of Figure 5.1. 169
5.5	Gabriel graph
5.6	Gabriel and relative neighbours
5.7	Relative neighbourhood graph
5.8	Minimum spanning tree
5.9	Area Voronoi diagram
5.10	Propagation masks used in the two-pass DT algorithm
5.11	Comparison of several discrete metrics
5.12	Larger propagation masks that can be used in the two-pass DT algo-
	rithm
5.13	Ambiguity of connectivity on the square grid
5.14	Examples of elliptic poweroids $g(x) = -(x^{T}\mathbf{A}x)^{\frac{\alpha}{2}}$ in \mathbb{R}^2
5.15	The domain of the circular paraboloid structuring element needed to
	calculate the Euclidean distance transform
5.16	Embedding a hexagonal grid image into a rectangular grid 190
5.17	New algorithm for computing the exact Euclidean distance transform on the beyagonal grid
5 18	Skeleton by influence zones (SKIZ)
5.19	Generalised Delaunay graph
5.20	Example of k-adjacency for $k = 3$
5.21	Estimating the connectivity number of a finite set X (large dots) on
0.21	the square lattice
5.22	Elementary test-line directions for the square lattice and the hexago- nal lattice
5.23	Estimating $D_{\alpha}(X)$ on the square lattice
5.24	Doubling the test-line directions for the square lattice and the hexag-
	onal lattice

6.1	Steps involved in building a statistical pattern recognition system 22 $$	20
6.2	An example of an ROC (receiver operating characteristic) curve 23	33
6.3	Epithelia of the cervix	35
6.4	Cells of the squamous epithelium	35
6.5	One of the AutoCyte slides used in the study	39
6.6	$AcCell^{\mathbb{M}}$ 2000 workstation	40
6.7	Number of nuclei-like objects per slide before and after artefact rejection.	11
6.8	Sample images captured by the cytometer: the first 64 images cap- tured from slide 0	12
6.9	Sample images captured by the cytometer: the first 64 images cap- tured from slide 1	13
6.10	Sample images after artefact rejection: the first 64 images retained from slide 0	16
6.11	Sample images after artefact rejection: the first 64 images retained from slide 1	17
6.12	The logistic function $f(x) = e^x/(1+e^x)$ restricted to the domain $[-10, 10]$	54
6.13	Obtaining the <i>cookie-cutting-distance</i> used to compute B8 25	58
6.14	Box-and-whisker plot of the feature M computed for each nucleus on slide $6. \ldots \ldots$	59
6.15	Split box-and-whisker plots for the features $F1$, $F2$, $F3$, $F4$, and $F5$. 26	52
6.16	The empirical ROC curve and associated classification summary for the logistic classifier built using the sample priors and equal costs of misclassification.	33
6.17	The empirical ROC curve and associated classification summary for the logistic classifier built using the sample priors and assuming that the cost of misclassifying an abnormal is 10 times worse than that of misclassifying a normal	54
6.18	The empirical ROC curve and associated classification summary for the logistic classifier built using population priors and equal costs of misclassification.	35

6.19	The empirical ROC curve and associated classification summary for	
	the logistic classifier built using population priors and assuming that	
	the cost of misclassifying an abnormal is 10 times worse than that of	
	misclassifying a normal	36
7.1	Hasse diagram showing the partial ordering on $\mathcal{T} \times \{\mp, -, +, \pm\}$ for	
	the case $T = \{0, 1, 2\}$	34
7.2	Compendium of images and graphs derived from the chromatin seg-	
	mentation algorithm	38

xxii

List of Tables

1.1	Ph.D. research contributing to the overall research initiative in the Cytometrics Project
ን 1	Complete lattices in image analysis
2.1	Complete lattices in image analysis
2.2	Comparison between linear and morphological image processing 51
4.1	Chromatin segmentation algorithms published in the literature 110
4.2	Classification of chromatin segmentation algorithms published in the
	literature
4.3	Data structures and labels used in the ISRG
5.1	Commonly used GLCM features
5.2	Examples of planar grey-scale graphs (adjacency graph + attribute)
	for which an adjacency graph attribute co-occurrence matrix can be
	defined
5.3	Minkowski functionals $W_k^{(n)}$ in \mathbb{R}^n
6.1	Feature selection methods
6.2	Commonly used classification methods
6.3	Commonly used error estimation methods
6.4	Qualitative interpretation of the AUC
6.5	Nucleus measurements
6.6	Blob measurements

6.7	Blob statistics computed for each nucleus
6.8	Comparison of the present study with similar studies published in the literature
6.9	The frequency with which features were selected in the 100 repetitions of stepwise model fitting
A.1	DImPAL's software components
A.2	DImPAL operators listed in decreasing order of precedence 325
A.3	DImPAL data types

Notation

\Rightarrow	Implies
\Leftrightarrow or iff	If and only if
\mathbb{N}	Set of natural numbers (positive integers)
\mathbb{Z}	Set of all integers
\mathbb{R}	Set of all real numbers
$\mathbb E$	Either \mathbb{Z} or \mathbb{R}
$\overline{\mathbb{Z}}$	Set of all integers including $+\infty$ and $-\infty$
\mathbb{R}	Set of all real numbers including $+\infty$ and $-\infty$
\mathbb{Z}^+	Same as \mathbb{N}
\mathbb{R}^+	Set of positive real numbers
\mathbb{Z}^n	<i>n</i> -dimensional discrete space
\mathbb{R}^{n}	n-dimensional Euclidean space
\mathbb{E}^n	Either \mathbb{Z}^n or \mathbb{R}^n
x,y	Points (and also position vectors) in \mathbb{E}^n ;
	e.g. $x^{T} = (x_1, x_2, x_3)$ in \mathbb{R}^3
p,q	Points in \mathbb{E}^n ; e.g. $p = (x_1, y_1)$ in \mathbb{R}^2
S	Arbitrary set with elements X, Y, \ldots
$\{\} \text{ or } \emptyset$	Empty set
$\mathcal{S} imes \mathcal{T}$	Cartesian product of two sets
\mathfrak{R}	Binary relation
\mathfrak{R}'	Converse relation of the relation \Re
$\mathcal{P}\left(\mathcal{S} ight)$	Power set of the set \mathcal{S} ; i.e. the set of all subsets
	of ${\mathcal S}$
$\mathcal{T}^{\mathcal{S}}$ or Fun $(\mathcal{S}, \mathcal{T})$	Set of all functions $f: \mathcal{S} \to \mathcal{T}$
\leq, \preceq	Partial order relations
\subseteq	The partial order relation "subset or equal to"

\lor and \land	Supremum and infimum, respectively, for the
	partial order relation \leq
Υ and λ	Supremum and infimum, respectively, for the
	partial order relation \preceq
\cup and \cap	The supremum (union) and the infimum (in-
	tersection), respectively, for the partial order
	relation \subseteq
(\mathcal{L},\leq)	A poset; i.e. a pair consisting of an arbitrary
	set and a partial order relation defined on it
U	Universal element of the poset (\mathcal{L}, \leq)
0	Null element of the poset (\mathcal{L}, \leq)
$\left(\mathcal{P}\left(\mathcal{S} ight) ,\subseteq ight)$	Complete Boolean lattice
$\left(\mathcal{T}^{\mathcal{S}},\leq\right)$ or $\left(\operatorname{Fun}\left(\mathcal{S},\mathcal{T}\right),\leq\right)$	Power lattice
$\mathcal{O}\left(\mathcal{L} ight)$	Set of all mappings $\psi : \mathcal{L} \to \mathcal{L}$
X	Arbitrary set with elements x, y, \ldots
X^{c}	Set complement of X
X^*	Negation (lattice complement) of X
ψ^*	Negative of the operator ψ ; i.e. $\psi^*(X) =$
	$[\psi\left(X^*\right)]^*$
δ	Dilation
ε	Erosion
$(arepsilon,\delta)$	Adjunction
ψ^n	Iteration (self composition), of ψ , <i>n</i> -times
$\psi \circ \eta$ or $\psi \eta$	Composition product meaning the application
	of η followed by the application of ψ
Ι	Identity mapping; i.e. $I(X) = X$
A_h	Translation of the set $A \in \mathcal{P}(\mathbb{E}^n)$ along the
	vector $h \in \mathbb{E}^n$
$A \oplus B$	Minkowski addition of sets $A, B \in \mathcal{P}(\mathbb{E}^n)$
$A \ominus B$	Minkowski subtraction of sets $A, B \in \mathcal{P}(\mathbb{E}^n)$
B	Symmetric set (reflection) of B ; i.e. $\dot{B} =$
	$\{-b b \in B\}$
$\delta_{B}\left(A ight)$	Dilation of the set A by the set (structuring
	element) B
$\varepsilon_B(A)$	Erosion of the set A by the set (structuring
	element) B

f_h	Translation of the function f along the vector
	<i>h</i> ; i.e. $f_h(x) = f(x - h)$
$f\oplus g$	Minkowski addition of functions
$f\ominus g$	Minkowski subtraction of functions
ğ	Reflection of g ; i.e. $\breve{g}(x) = g(-x)$
$\delta_{g}\left(f ight)$	Dilation of the function f by the function
	(structuring function) g
$\varepsilon_{g}\left(f ight)$	Erosion of the function f by the function
	(structuring function) g
$\delta_{B}\left(f ight)$	Dilation of the function f by the set (flat func-
	tion) B
$\varepsilon_{B}\left(f ight)$	Erosion of the function f by the set (flat func-
	tion) B
$d\left(x,y ight)$	Distance between x and y
(E,d)	Metric space; i.e. a pair consisting of a non-
	empty set E and a metric defined on it
d_p or L_p	Minkowski metric of power p
$B^{\circ}\left(x,r ight)$	The (topologically) open ball of centre x and
	radius r
$B\left(x,r ight)$	The (topologically) closed ball of centre x and
	radius r
δ^r	Metric dilation of size r
ε^r	Metric erosion of size r
G = (V, E)	Graph G with vertex set V and edge set E
$V\left(G ight)$	Set of vertices of the graph G
$E\left(G ight)$	Set of edges of the graph G
$N\left(v ight)$	Neighbourhood of a vertex v
$X \mid G$	Binary graph
$f \mid G$	Grey-scale graph
$\delta^r \left(X \mid G \right)$	Binary graph dilation of size r
$\varepsilon^r \left(X \mid G \right)$	Binary graph erosion of size r
$\delta^r \left(f \mid G \right)$	Grey-scale graph dilation of size r
$\varepsilon^r \left(f \mid G \right)$	Grey-scale graph erosion of size r
$A \setminus B$	Set difference; i.e. $A \cap B^{c}$
$d\left(x,X ight)$	Distance between a point x and a set X
$d_{\mathrm{H}}\left(X,Y ight)$	Hausdorff metric
D_X	Distance transform of X

xxvii

$O\left(n ight)$	Of order n complexity
x^{T}	Transpose of the vector x
$B\left(\lambda ight)$	Closed ball, centred at the origin, of radius λ
A	Matrix
$\bigoplus g_i$	$g_1\oplus g_2\oplus\ldots$
$\overset{\imath}{D}(p,q)$	Closed disk of diameter \overline{pq}
$W_k^{(n)}$	Minkowski functional of number k in \mathbb{R}^n
$A\left(X ight)$	Area of X
$U\left(X ight)$	Perimeter of X
$N^{(n)}\left(X\right)$	Euler-Poincarè constant (connectivity num-
	ber) of X in \mathbb{R}^n
$\mathcal{N}\left\{\star ight\}$	Number of configurations of type \star
$\lfloor a \rfloor$	The largest integer not greater than a
X	Column vector of predictor variables

Abbreviations

AGACM	Adjacency graph attribute co-occurrence matrix
AHTAC	Australian Health Technology Advisory Committee
AIC	An information criterion
AUC	Area under the ROC curve
BIC	Bayesian information criterion
CCR	Correct classification rate
CIN	Cervical intraepithelial neoplasia
CIS	Carcinoma in situ
CSSIP	Cooperative Research Centre for Sensor Signal and In-
	formation Processing
DImPAL	Digital image processing and analysis language
DNA	Deoxyribonucleic acid
DT	Distance transform
EDT	Euclidean distance transform
FDA	United States Food and Drug Administration
FIFO	First-in first-out queue
FISF	Folding induced self-dual filter
GLCM	Grey-level co-occurrence matrix
HSIL	High-grade squamous intraepithelial lesion
ISRG	Improved seeded region growing
IZ	Influence zone
LIFO	Last-in first-out queue
LSIL	Low-grade squamous intraepithelial lesion
MAC	Malignancy associated change
MRF	Markov random field
O.D.	Optical density

PDF	Probability density function
Ph.D.	Doctor of philosophy
PQ	Priority queue
QML	Queensland Medical Laboratory
RAG	Region adjacency graph
RE	Rare event
ROC	Receiver operating characteristic
SRG	Seeded region growing
WHO	World Health Organization
ZOI	Zone of influence

Publications

The research detailed in this thesis was conducted at the University of Queensland node of the Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP). At the start of candidacy the author signed a participation agreement with the CSSIP. The agreement includes several clauses relating to intellectual property arising during candidacy and the protection of commercially sensitive material. CSSIP has restricted open publication because of the commercial nature of some of the research described herein.

Patents

 Mehnert, A. J. H. & Jackway, P. T. (2002), 'Chromatin segmentation', *International Patent Application*. Number PCT/AU02/00969. Priority date 19 July 2001.

Refereed Journal Papers

- Mehnert, A. J. H. & Jackway, P. T. (1999b), 'On computing the exact Euclidean distance transform on rectangular and hexagonal grids', *Journal of Mathematical Imaging and Vision* 11, 223–230.
- Mehnert, A. J. H. & Jackway, P. T. (1997), 'An improved seeded region growing algorithm', *Pattern Recognition Letters* 18, 1065–1071.

Refereed Conference Papers

- Mehnert, A. J. H. & Jackway, P. T. (2000), Folding induced self-dual filters, in J. Goutsias, L. Vincent & D. S. Bloomberg, eds, 'Mathematical Morphology and its Applications to Image and Signal Processing', Vol. 18 of *Computational Imaging and Vision*, Kluwer Academic Publishers, Boston, pp. 99–108.
- Mehnert, A. J. H. & Jackway, P. T. (1999*a*), Inducing self-duality through greyscale folding, *in* 'Proceedings DICTA'99: The Fifth International/National Biennial Conference on Digital Image Computing, Techniques, and Applications', Curtin University, Perth, Western Australia, pp. 176–180.

Technical Reports

- Mehnert, A. J. H. & Bamford, P. C. (2002), [Cytometrics Project] Cell-image data sets, Technical Report TI013, Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), SPRI Building, Technology Park, Mawson Lakes, South Australia.
- Mehnert, A. J. H. (2002), An independent review and evaluation of the Cytometrics Project's classifiers for Papanicolaou stained cervical slides, Technical Report TI009, Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), SPRI Building, Technology Park, Mawson Lakes, South Australia.

Original Contributions

The major contributions of this thesis are:

- 1. A new and general method for constructing non-linear self-dual operators. The operators, called folding induced self-dual filters (FISFs), are constructed from arbitrary morphological (meta-) operators defined on an abstract space called *fold-space*. The *folded closing* of Evans et al. (1997) is a particular type of FISF. Importantly, other types of FISF can be designed that do not suffer the limitations of the folded closing in relation to the attenuation of salt-and-pepper impulse noise.
- 2. A new algorithm for segmenting chromatin in (light) microscope images of cell nuclei. The algorithm is based on seeded region growing and is, in its preferred embodiment, parameter-free. Moreover it yields a segmentation consistent with what a human would perceive to be chromatin particles.
- 3. A new seeded region growing algorithm, based on the original algorithm devised by Adams & Bischof (1994), that is independent of the order of pixel processing.
- 4. A new implementation of an ascending priority queue that permits the implementation of a fast version of the watershed transform (particular case of seeded region growing).
- 5. A new theoretical result establishing an equivalence between the distance transform of a binary image, where the underlying distance is based on a *positive definite quadratic* form, and the *erosion* of its characteristic function by an *elliptic poweroid* structuring element.

- 6. A new algorithm, based on the result in 5, for calculating the exact Euclidean distance transform of a binary image manifested on a hexagonal grid. The algorithm has several advantages over the only other known algorithm devised by Vincent (1991*a*): it is simpler, faster, and suited to hardware implementation.
- 7. A new generalisation of the grey-level co-occurrence matrix method to vertexweighted adjacency graphs. The new method, called the adjacency graph attribute co-occurrence matrix (AGACM) method, can be used to quantitatively characterise blob-like and mosaic patterns in the plane.
- 8. Corrections to formulae published in the literature and two new formulae for estimating the Minkowski functional $W_1^{(2)}$ (perimeter) for digital images manifested on square and hexagonal grids.
- 9. A structural model of chromatin texture from which it is possible to define features that can be directly related to the terms and adjectives, used by cytoprofessionals, to describe chromatin distribution.
- 10. New features—in particular *margination*—for the purpose of quantifying the distribution of nuclear chromatin.
- 11. Empirical evidence that it is possible to detect differences in the pattern of nuclear chromatin between samples of cells from a normal Papanicolaou-stained cervical smear and those from an abnormal smear using the features in 10. These differences are supportive of the existence of the malignancy associated changes (MACs) phenomenon. The classification results are based on 99 normal and 40 abnormal Pap-stained cervical slides collected as part of the (Australian) National Cervical Screening Program. Moreover the results compare favourably with those reported in the literature for other stains developed specifically for automated cytometry. To the author's knowledge this is the first time that MACs have been demonstrated in Papanicolaou stain based on a sizable and uncontaminated data set. This is an important finding because cervical screening programs world-wide use the Pap stain.

Acknowledgements

First and foremost I wish to thank Dr. Paul Jackway (my principal advisor during the first half of my candidacy) and Professor Dennis Longstaff (my principal advisor during the second half of my candidacy) for their guidance, support, and enthusiasm. I thank Professor Longstaff, in particular, for his patience and his faith in my ability to complete this thesis.

I thank the University of Queensland (UQ) and the Centre for Sensor Signal and Information Processing (CSSIP) for their financial support during my candidature: UQ Postgraduate Research Scholarship, UQ Postgraduate Completion Scholarship, and the CSSIP supplementary scholarship. Additionally, I acknowledge partial funding by the Australian Research Council for the research presented in Chapter 3.

I extend a big thank you to Professor Dennis Longstaff, Professor Vassilios Sarafis, Dr. Pascal Bamford, Dr. Andrew Bradley, and Julie Reeves for reading and commenting on various parts of this thesis.

I am indebted to Dr. Pascal Bamford for his tenacity in adapting the Cytometrics Project cytometer to work with Papanicolaou stain and for acquiring the cell-image data used in this thesis. I am also grateful to Pascal and his wife Melanie for their enduring friendship, and their support and encouragement during my candidacy.

Thanks also to all of the researchers in the Queensland node of the CSSIP for their comradeship, intellectual input, and the many intellectual and earthy discussions over cups of coffee. I extend a special thanks to Pat Bellett and his wife Paege for their friendship and encouragement.

The hardest thing about leaving Western Australia, to take up my candidacy at the University of Queensland, has been the separation from my family. I have missed them all very much. I thank them, and especially my parents, for their encouragement, love, and support. Last, but by no means least, I thank Julie, the love of my life, for believing in me and for giving me several good reasons to finish this thesis.

This thesis was prepared using L_YX^1 which is a WYSIWYM (what you see is what you mean) open source document processor. L_YX itself relies on ET_EX^2 to do the actual typesetting. The figures and diagrams in this thesis were produced using the packages: Xfig³, xv⁴, GIMP⁵, Gnuplot⁶, R⁷, and DImPAL (see Appendix A). All of the image analysis was performed using DImPAL. The R package was used to perform all of the statistical analyses. All of this software was run under the Linux operating system⁸.

¹ L_YX is free software and can be downloaded from http://www.lyx.org.

² \mathbb{IAT}_{EX} is a a program for typesetting documents. "The T_{EX} in \mathbb{IAT}_{EX} refers to Donald Knuth's T_{EX} typesetting system. The \mathbb{IAT}_{EX} program is a special version of T_{EX} that understands \mathbb{IAT}_{EX} commands" (Lamport, 1994, p. 5).

³ Freely available from http://www.xfig.org.

⁴ Written by John Bradley and freely available via ftp://ftp.cis.upenn.edu/pub/xv.

⁵ Freely available from http://www.gimp.org.

⁶ Freely available from http://www.gnuplot.info.

⁷ Freely available from http://www.r-project.org.

⁸ Linux (http://www.linux.org) is a free Unix-like operating system. It consists of the kernel developed by Linus Torvalds and software developed by the GNU Project (http://www.gnu. org).
l Chapter

Introduction

The known is finite, the unknown infinite; intellectually we stand on an islet in the midst of an illimitable ocean of inexplicability. Our business in every generation is to reclaim a little more land, to add something to the extent and the solidity of our possessions

T. H. Huxley, On the Reception of the Origin of Species, 1887

The research detailed in this thesis was conducted at the University of Queensland node of the Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP)¹. It constitutes part of a larger research initiative known as the Cytometrics Project. The project's name stems from *cyto*- which is a word element referring to cells and *metric* which pertains to measurement. The aim of the project is to develop computer algorithms to automate, or assist cytoprofessionals² with, the screening of slide specimens for the presence of biological abnormalities. Since its inception in 1993, the project has focused on cervical cytology³ and the *Papanicolaou (Pap) test* in particular. The Pap test is the primary screening test

¹ The CSSIP was established in 1992 under the Cooperative Research Centres program of the Australian Commonwealth Government. The parties in CSSIP are the following: The University of Adelaide, The University of South Australia, The Flinders University of South Australia, The University of Melbourne, The University of Queensland, Commonwealth of Australia (Defence Science and Technology Organization), Telstra Corporation Limited, Compaq Computers Australia Pty Limited, CEA Technologies Pty Limited, and RLM Systems Pty Limited.

² Cytoprofessionals are cytotechnologists and cytopathologists. "Cytotechnologists and cytopathologists are technical and medical laboratory specialists who possess the knowledge and skill required to make diagnostic interpretations of cellular specimens" (Greening, 1994, p. 328).

 $^{^3}$ Cytology is the scientific study of cells; particularly in relation to form, structure, and function.

for cervical cancer. In the United States alone, approximately 70 million Pap smears are screened annually (Greer, 1997, p. 248)⁴. Cibas (2003, p. 1) states that:

In the 1930s, before screening was introduced, cervical cancer was the most common cause of cancer deaths in women in the United States. Today it is not even one of the top ten.

Nevertheless, in the United States about 13000 cases of cervical cancer are diagnosed, and more than 4000 women die of the disease each year (ACS, 2002, p. 16)⁵. Anderson & Runowicz (2001, p. 753) state that:

Although most cases of invasive cervical cancer [in the United States] occur in the unscreened population, nearly one third of cases can be attributed to screening failure...: either unsatisfactory collection of the sample or misinterpretation of cervical cytology.

Whilst improved collection and smear preparation standards can reduce the number of errors due to inadequate (unrepresentative) smears, "using the conventional methodology of Pap smear examination, it may be difficult to reduce the errors attributable to faulty interpretations" (Grohs, 1994, p. xii). Human errors in Pap smear interpretation occur for a variety of reasons including (Grohs, 1994, p. xii): fatigue, habituation, inexperience and lack of knowledge, and the limitation of the subjective microscopic examination. A major objective of the Cytometrics Project is to reduce the error rate attributable to misinterpretation by supplementing, or even replacing, the subjective interpretation of smears with objective quantitative measurements.

The remainder of this chapter is organised as follows. Section 1.1 presents a brief overview of Pap smear cytology and the problems inherent in seeking to automate the screening of Pap smear slides. Section 1.2 discusses a phenomenon, known as malignancy associated changes (MACs), which current research world-wide suggests has the potential to overcome these problems. MACs have been demonstrated, in the quantitative cytology literature, to be subtle subvisual changes in otherwise normalappearing cells on cervical atypical smears. The features (measurements) reported

⁴ In Australia approximately 2.7 million Pap smears are screened annually (Pieris-Caldwell et al., 2002, p. xiii).

⁵ In Australia, statistics for 1987-1998 show that about 1000 cases of cervical cancer are diagnosed each year (Pieris-Caldwell et al., 2002, p. 57), and statistics for 1989-1999 show that about 300 women die each year from the disease (Pieris-Caldwell et al., 2002, p. 66).

to have the most discriminatory power are texture features. The feature-based approach to discrimination and classification is known as statistical pattern recognition. Section 1.3 presents a brief overview of the statistical pattern recognition approach used in the field of computer vision. Section 1.4 then presents an overview of the different approaches to quantifying texture (texture feature extraction) published in the literature. Section 1.5 outlines the research problem addressed in this thesis. Section 1.6 then presents the aim and objectives of the thesis and Section 1.7 outlines the scope of the research described in this thesis. Finally Section 1.8 describes the structure of this thesis.

1.1 The Papanicolaou test

This section briefly reviews the Papanicolaou (Pap) test. The review includes a description of the origin and nature of the Pap test, a discussion of the accuracy of the conventional Pap test, and a discussion of the state-of-the-art in automated Pap smear screening.

1.1.1 The origin and nature of the Papanicolaou smear

Dr. George Nicholas Papanicolaou is credited with the conception and development of the Pap smear or Pap test (Cibas, 2003, p. 2). In 1928 he reported "that cancer cells derived from the uterine cervix may be observed in human vaginal smears" (Koss, 1989, p. 737). However, this observation was initially poorly received (Kline, 1997, p. 205). Several years later Papanicolaou revived the method, in collaboration with Herbert Traut, resulting in the publication of a major paper (Papanicolaou & Traut, 1941) and a monograph (Papanicolaou & Traut, 1943). Several subsequent studies in the 1940s, on small groups of women, documented that vaginal or cervical smears could lead to the discovery of occult cancers of the uterine cervix and to precancerous changes (Koss, 1992, p. 3). This paved the way for the introduction of large-scale cervical cancer screening programs in the late 1950s (Grohs, 1994, p. xi). In the industrialised world (Greer, 1997, p. 246):

cytologic screening for cancer of the cervix and precancerous lesions and the subsequent treatments of these lesions have been effective in reducing the incidence and mortality of cervical cancer... In the highly screened population, the cervical cancer rate is 79% lower than the rate in the unscreened population. The Pap smear procedure or Pap test consists of the following steps: (i) collecting a sample of cells from in and around the cervix using a cotton swab, wooden spatula, or cervical brush; (ii) pressing the cells on a glass slide; (iii) applying a *fixative*⁶ to preserve the sample; (iv) staining⁷ the sample (to visualise the different cell components); and (v) visually inspecting the slide under a microscope for signs of abnormality (Hale, 1989; Patlak, 1996). The collection and fixing steps are usually performed by a medical practitioner who then sends the specimen to a cytology laboratory for staining and interpretation.

1.1.2 Accuracy of the Pap test

The conventional Pap test "is a uniquely labor-intensive complex process, the outcome of which depends entirely on human judgment and is not machine generated" (Koss, 1989, p. 737). A single smear can contain from 50 000 to 300 000 cells (Koss, 1989, p. 738). "An abnormal smear may contain only a few abnormal cells scattered through the thousands of normal cells" (Greer, 1997, p. 248). The careful screening of a Pap smear is, therefore, a time-consuming task, even for a very-well-trained cytotechnologist (Koss, 1989, p. 738). Potentially, fatigue or a lack of concentration can lead to a false interpretation. Unfortunately (DeMay, 1997, p. 229):

no cytology laboratory, no matter how well run, is completely free of diagnostic errors, the most important being false negatives [reports that fail to identify abnormal cells]. False negatives occur at a low, but well documented and probably irreducible rate of at least 5% to 10%, ie, at least 1 in every 10 to 20 positive cases will be missed in routine screening.

In a 1997 press statement the World Health Organization stated that in addition to the "inappropriate interpretation of results", the other principal cause of false negatives is "the poor quality of samples". Either abnormal cells don't make it onto the slide or they are obscured by overlapping cells, blood, mucus, and other debris (Linder, 1994, p. 26). In its press release the WHO states that the issue of

⁶ Fixation immobilises, kills, and preserves the cells. Moreover fixation "makes cells permeable to staining reagents and cross-links their macromolecules so that they are stabilized and locked in position" (Alberts et al., 2002, p. 554).

⁷ "Worldwide, the most commonly used cytological staining technique in gynecology is the Papanicolaou (Pap) stain, which was introduced as a routine staining method for cervical smears by George Papanicolaou in 1942. Many modifications of the original Pap stain have been described in the relevant literature, but the overall staining pattern is rather similar amongst the various methods" (Schulte & Wittekind, 1994, p. 201).

false negatives "can be overcome through improved sampling and the introduction of automated [screening] devices that can detect 30% to 50% more false negatives than humans".

1.1.3 Automated screening

"The need to automate the screening of the cervical Pap smear was recognized in the 1950s by Dr. Papanicolaou and co-workers" (Linder, 1994, p. 25). Indeed there has been a concerted effort since the 1950s to develop a viable automated cytometer. Numerous systems were developed including (Cibas, 2003; Husain, 1994): the Cytoanalyzer (1950s), TICAS (1968), Quantimet B (1960s), CERVIFIP (1970), CYBEST (1972), DIASCANNER (1976), FAZYTAN (1978), LEYTAS (1978), and BioPEPR (1981).

Early attempts at automation were unsuccessful for three principal reasons (Linder, 1994, p. 25):

1. The nature of the Pap smear.

"The conventional Pap smear includes a large and variable number of epithelial cells that are admixed with blood, inflammatory cells, and mucous. Cells and background material on the smear frequently overlap each other, making it difficult to identify diagnostically important cells" (Linder, 1997, p. 282).

- 2. Limited computing power.
- 3. An incomplete conceptual understanding of morphologic abnormalities.

It is perhaps not surprising, therefore, that it was only in September 1995 that the first automated system—the AutoPap 300 QC developed by NeoPath Inc.—was granted approval by the United States Food and Drug Administration⁸ (FDA) for commercial use; and then only for the purpose of screening slides already screened by humans (re-screening) (FDA, 1995). Another system—Papnet developed by Neuromedical Systems Inc.—was granted approval for the same purpose less than two months later (FDA, 1996). Cibas (2003, p. 5) gives the following summary of the present state of automated screening:

⁸ The U.S. Food and Drug Administration is an agency of the United States government's Department of Health and Human Services.

Although European investigators largely lost interest in cytology automation in the 1990s, researchers in the United States and Canada, supported primarily through venture capital, retained their enthusiasm, resulting in advances in research and development. Foremost in the field in recent years have been AutoCyte Inc. (formerly Roche Image Analysis Systems), Cytyc Corp., Neopath Inc., and Neuromedical Systems, Inc. An important three-way merger took place in 1999, when AutoCyte, after purchasing the intellectual property of Neuromedical Systems, merged with Neopath to form a new company called TriPath Imaging. In 1998, the FDA approved the AutoPap System (now called FocalPointTM)(TriPath Imaging, Burlington, NC) as a primary screener for cervicovaginal smears.

Whilst "advances in computer-processing power and software design have permitted cytology automation to occur..., a more significant factor has been the perception by both the cytology profession and the society at-large that cytology automation will improve the quality of cytology services" (Linder, 1997, p. 282). At the time of writing, all of the FDA approved Pap smear automated screeners are diagnostic cell screeners; i.e. they exhaustively examine all of the cells on a slide for the (rare) occurrence of diagnostic cells. They improve the quality of conventional cytology services in the sense that, in comparison to human screeners, they provide both objective classification and an improved sensitivity to the detection of abnormal cells e.g. in a prospective clinical trial FocalPoint[™] "detected significantly more abnormal slides...than conventional practice (86 vs 79%)" (Cibas, 2003, p. 6). Although these systems can reduce the number of false negatives attributed to "inappropriate interpretation of results", the problem remains that they cannot reduce the number of false-negatives associated with "poor quality samples". DeMay (1997, p. 230) states that "many, perhaps most, false negatives represent sampling errors, where a sample of abnormal cells from the patient fails to make it onto the glass slide". These machines, like human screeners, cannot label a smear as suspicious if there are no abnormal cells on the slide. However, a phenomenon known as *malignancy* associated changes (MACs) may offer a solution to this problem.

1.2 Malignancy associated changes

This section provides a brief overview of the MACs phenomenon. A recent and comprehensive review of the history and significance of the MACs phenomenon can be found in the Ph.D. thesis of Hallinan (2000).

The expression MACs was coined by Nieburgs, Zak, Allen, Reisman & Clardy (1959) to describe distinct visual changes in the nucleus of apparently normal cells "adjacent to or distant from malignant tumours" (Nieburgs & Goldberg, 1968, p. 35). The phenomenon itself was first reported more than forty years earlier by Gruner (1916). The work of Nieburgs et al. (1959) initiated a flurry of research throughout the 1960s seeking to verify the existence of MACs, and also to refine the list of its characteristic visual features or cues. Unfortunately by the close of the decade, the significance and definition of MACs was still unresolved. The first quantitative study of the phenomenon, using *computer image analysis* techniques, was undertaken by Klawe & Rowiński (1974) using the Quantimet B (see Section 4.3). Using a quantitative parameter consistent with the description of MACs by Nieburgs et al., they were able to show "a significant difference between the nuclei of buccal cells of healthy children and those from children with benign malignant tumours" (Palcic & MacAulay, 1994b, p. 157). MACs, in modern usage, refers to the following phenomenon (Nordin & Bengtsson, 1994, p. 44):

In the early 1980s, several researchers found evidence of *subvisual* [italics added] alterations in intermediate cells from cervical atypical smears: Cells which were considered normal when analyzed visually through a microscope differed slightly in appearance if taken from a smear containing cancer cells compared with cells from wholly normal smears. These alterations are too insignificant to be analyzed [measured] on a cell-by-cell basis; instead, populations of cells must be analyzed and the population parameters (means, variances) used to classify the smear.

The measurements or *features* that appear to have the most discriminatory power "are those describing DNA organization and its distribution in the cell nuclei" (Palcic & MacAulay, 1994b, p. 158). The most prominent of these features are *texture features* (Palcic, 1994, p. 41).

Formally, the feature-based approach to discrimination/classification is known as *statistical pattern recognition*. Statistical pattern recognition is the most widely used approach to pattern recognition in the field of *Computer Vision*.



Figure 1.1: The elements of an image analysis system (adapted from Gonzalez & Woods (1992, p. 573)).

1.3 Statistical pattern recognition in computer vision

This section provides a brief overview of statistical pattern recognition in computer vision. A more detailed treatment of statistical pattern recognition is given in Chapter 6.

Computer vision is a "branch of the field of artificial intelligence concerned with developing algorithms for analyzing the content of an image" (Castleman, 1996). The objective of computer image analysis is to endow a computer with the ability to discover, identify, and understand patterns in images that are relevant to the performance of an image-based task (Gonzalez & Woods, 1992). An example of such a task is the sorting of letters according to postcode in a postal sorting centre. Here the patterns are sequences of digits hand-written or typed on an envelope. A more relevant example is of course the recognition of textural changes in the nuclei of individual cells or populations of cells taken from Pap smears. The fundamental elements of a computer *image analysis system* are depicted in Figure 1.1. Broadly speaking, these elements fall into three categories (Gonzalez & Woods, 1992, p. 572): (1) *low-level processing:* image acquisition and preprocessing; (2) *intermediate-level processing:* image segmentation, and representation and description; and (3) *high-level processing:* pattern/object recognition and interpretation.

Low-level processing encompasses all those operations concerned with acquiring a digital image and processing it to correct for degradations such as noise and blurring. These operations require no *intelligence* on the part of the image analysis system

in the sense that no recognition or interpretation of the contents of the image is required.

Intermediate-level processing is concerned with locating and isolating objects of interest in an image (segmentation) and with their characterisation (representation and description). Robust autonomous segmentation invariably requires some level of in-built intelligence in order to be able to cope with anomalies such as gaps in extracted contours, or the presence of artefacts. Gonzalez & Woods (1992, p. 413) state that "the [segmentation] process determines the eventual success or failure of the analysis". The segmentation process yields a raw image in which either the regions are delineated by boundary pixels, or the pixels comprising each region are assigned a unique label. Depending on the nature of the images under study, this data may be further processed to obtain a simplified representation of the regions suitable for further processing. Such representations include: using a *chain code* to represent the boundary of each region, approximating regions by polygons, reducing the boundary of each region to a *signature* (one dimensional functional representation of the boundary), decomposing the boundary into segments, and reducing each region to its *skeleton*. Once a suitable representation has been established, descriptors can be derived for each of the regions. Such descriptors include: area and perimeter, and quantitative parameters that characterise properties such as shape, texture, and colour.

High-level processing involves the recognition and interpretation of the contents (segmented regions) of an image. The recognition step is classically referred to as *pattern recognition*. Pattern recognition by computer entails assigning patterns to classes without human intervention. The term *pattern* refers to the object or entity of interest; e.g. a Pap smear. The most widely used approach to pattern recognition in computer vision is *statistical pattern recognition* (an overview of statistical pattern recognition is given in Chapter 6). This approach involves: (i) computing numerical parameters, called *features*, for each pattern; and (ii) using these to assign patterns to classes. Features include quantitative measurements of object attributes, i.e. descriptors, and derived numerical parameters; e.g. combinations (usually linear) of descriptors.

1.4 Texture features

This section briefly reviews the different approaches to texture feature extraction published in the literature. Recent and comprehensive reviews of texture analysis methods can be found in the *Handbook of Pattern Recognition and Computer Vision* (Tüceryan & Jain, 1999) and the Ph.D. thesis of Smith (1998, Chapter 2).

1.4.1 Defining texture

Texture analysis has been the subject of serious study since the late sixties and early seventies (Levine, 1985, p. 423). It is remarkable, therefore, that there still does not exist a universally accepted general definition of texture (Smith, 1998, p. 1). Tüceryan & Jain (1999, p. 219) state that:

the intensity variations in an image which characterize texture are generally due to some underlying physical variation in the scene (such as pebbles on a beach or waves in water). Modelling this physical variation is very difficult, so texture is usually characterized by the two-dimensional variations in the intensities present in the image. This explains the fact that no precise, general definition of texture exists in the computer vision literature.

In practice (Tüceryan & Jain, 1999, p. 208):

the "definition" of texture is formulated by different people depending upon the particular application.

Smith (1998, p. 3) states that:

despite the lack of a universally agreed definition of texture, all researchers agree on two points. Firstly, within a texture there is significant variation in intensity levels between nearby pixels; that is, at the limit of resolution, there is non-homogeneity. Secondly, texture is a homogeneous property at some spatial scale larger than the resolution of the image.

These points of consensus imply that a texture possesses both shift (translation) invariance and local structure. Shift invariance means that "visual perception is basically independent of position in the image pattern" (Levine, 1985, p. 423). Local structure refers to the existence of texture primitives (basic patterns or shapes repeated over and over as in the case of a chessboard), or of more subtle dependencies

between the intensity of a pixel and the intensities of its near-neighbours. The following definition of texture, by Francos et al. (1993, p. 2665), encapsulates all of these ideas:

texture can be defined as a structure which is made up of a large ensemble of elements that resemble each other to a great extent, with some kind of "order" in their locations, so that there is no one element which attracts the viewer's eye in any special way. The human viewer gets an impression of uniformity when he looks at a "texture".

Unfortunately this definition does not facilitate a quantitative characterisation of texture. Rather, a precise algorithmic formulation of the definition of texture is needed. Two such computational approaches to the definition of texture have been the mainstay of the texture analysis literature: the *structural* approach and the *stochastic* approach (Smith, 1998, p. 5).

1.4.1.1 Structural approach

The structural approach assumes that a texture is composed of texture primitives and that these primitives are arranged according to certain placement rules (Tüceryan & Jain, 1999, p. 226). The "primitives may be of varying or deterministic shape, such as circles, hexagons, or even dot patterns" (Cross & Jain, 1983, p. 25). "The placement rule may be deterministic for periodic and very well structured textures, or stochastic for more random structures" (Francos et al., 1993, p. 2666). The structural approach is suited to describing textures such as a brick wall, tilings of the plane, and cellular structures such as tissue samples (Cross & Jain, 1983, p. 25). Examples of such textures are shown in Figure 1.2.

1.4.1.2 Stochastic approach

The stochastic approach assumes that a texture is a realisation of a (discrete) *stochastic process*⁹ governed by a set of parameters (Paget, 1999, p. 4). Usually the stochastic process is considered to be two-dimensional (2D); i.e. it is a *random field*. The stochastic approach is used to describe textures such as sand, grass, and water (see Figure 1.3). "The key feature of these images is that the primitives are very random in shape and cannot be easily described" using a structural model (Cross & Jain, 1983, p. 25).

⁹ A discrete stochastic process is an indexed sequence of discrete random variables.



Figure 1.2: Textures that can be modelled using the structural approach. (a) Steel mesh. (b) Cells in tissue.



Figure 1.3: Textures that can be modelled using the stochastic approach. (a) Grass. (b) Sand.

1.4.2 Texture feature extraction

Early surveys of texture analysis methods—such as Haralick (1979) and Levine (1985, Chapter 9)—categorise texture analysis methods as either *structural* or *statistical*. These categories correspond to the structural and stochastic definitions of texture respectively. In the structural methods (Francos et al., 1993, p. 2666):

the texture is characterized by a description of its primitives and their placement rules. Hence the purpose of the first stage of the analysis procedure is to define the local attributes which characterize the texture basic cell–the primitive. The primitives may be connected regions with some tonal property, which are described in addition by their geometric properties like shape, area, directionality, and by degree of homogeneity of these properties. The higher level of structural analysis methods is concerned with describing the placement rules and the spatial relations between primitive cells. The placement rule may be deterministic...or stochastic.

In the statistical approaches, the texture is characterised "by a collection of statistics of selected features" (Francos et al., 1993, p. 2666). In the simplest case, one can compute statistics—such as the mean, variance, skewness, and kurtosis—of the grey-level histogram. These statistics are called *first-order* statistics because they "depend only on individual pixel values and not on the interaction or co-occurrence of neighbouring pixel values" (Tüceryan & Jain, 1999, p. 211). Second- and higherorder statistics, however, do take into account spatial interactions that may exist between two or more pixels. The concept of first-, second-, and third-order statistics can be explained (Levine, 1985, p. 434):

in terms of "dropping" a point, a line, and a triangle, respectively, at random on an image pattern. In the first case we collect statistics on which pixel (monopole) gray level has been hit and represent the probability distribution in terms of a histogram. Second-order statistics are computed by randomly dropping a needle (dipole) of varying length and orientation and considering the gray levels at the two extremities. Similarly, the third-order distribution is obtained by examining the three vertices of triangles with arbitrary dimensions.



Figure 1.4: The interrelation between the various second-order statistics (reproduced from Tüceryan & Jain (1999, p. 222)).

Examples of second-order statistics include: difference statistics, co-occurrence matrix features (see Section 5.7.1), autocorrelation function features, spectral features, and autoregression model parameters. As Figure 1.4 shows, these statistics are not independent.

Second- and higher-order interactions can be modelled by Markov random field (MRF) models¹⁰. "These models assume that the intensity at each pixel in the image depends on the intensities of only the neighboring pixels" (Tüceryan & Jain, 1999, p. 227). More specifically, the intensity of each pixel p is modelled as a random variable X_p such that the probability that $X_p = x_p$ is conditional upon the values of those pixels neighbouring p. The Hammersley-Clifford theorem, also called the MRF-Gibbs equivalence theorem, establishes the form of the local conditional probability density function (LCPDF) necessary to define a valid MRF (Paget & Longstaff, 1998, p. 926). In particular, the theorem requires the neighbourhoods to "be symmetrical and self similar for homogeneous MRFs" (Paget, 1999, p. 39). Examples of such neighbourhoods are shown in Figure 1.5. The MRF-Gibbs equivalence theorem expresses the form of the LCPDF with reference to the local clique set associated with each neighbourhood (see Figure 1.6).

More recent surveys of texture analysis methods refine and expand the statistical/structural taxonomy:

1. Reed & du Buf (1993) categorise texture feature extraction methods as:

¹⁰ Picard (1991) has established a theoretical link between grey-level co-occurrence matrices and Markov random field models.



Figure 1.5: Examples of symmetrical MRF neighbourhoods (pixel *p* is shaded): (a) First-order (not statistical order) neighbourhood; (b) Second-order (not statistical order) neighbourhood; and (c) Eighth-order (not statistical order) neighbourhood.

• Feature-based

This category is essentially the classical statistical category excluding model-based methods such as autoregressive and MRF methods.

 $\bullet \ {\it Model-based}$

This category includes all those methods that "hypothesize underlying processes for textures" (Reed & du Buf, 1993, p. 359). Such methods include fractal models, autoregression models, and MRF models. "Since model parameters are used as texture features, model-based methods could be considered a subclass of feature-based methods" (Reed & du Buf, 1993, p. 359).

• Structural

This category is identically the classical *structural* category.

- 2. Tüceryan & Jain (1999) categorise texture feature extraction methods as:
 - Statistical

This category includes co-occurrence matrix features, autocorrelation features, power spectrum features, and difference statistics.

• Geometrical

This category includes methods based on a "definition of texture as being composed of 'texture elements' or primitives" (Tüceryan & Jain, 1999, p. 223). This category is essentially the classical *structural* category



Figure 1.6: Cliques: (a) Local clique set corresponding to a first-order neighbourhood; (b) Cliques types for a second-order neighbourhood (the first 3, from left to right, are the clique types for a first-order neighbourhood).

with the addition of Voronoi tessellation features. Given a finite set of distinct points in the Euclidean plane, the Voronoi tessellation or diagram (see Section 5.2) is a partitioning of the plane into regions, each of which contains the set of points closest to a particular point in the point set. The concept can be extended from points (pixels) to more complex primitives which must be "extracted from the gray level images using some low-level processing" (Tüceryan & Jain, 1990, p. 211).

• Model based

This category includes MRFs and *fractals*. Fractals model the "statistical quality of roughness and self-similarity at different scales" that are exhibited by many natural textures (Tüceryan & Jain, 1999, p. 229). A texture that can be modelled as a fractal can be characterised by a single parameter called the *fractal dimension*. Fractal methods are methods for estimating this dimension.

• Signal processing

This category primarily includes methods that "compute certain features from filtered images" (Tüceryan & Jain, 1999, p. 231). The category includes methods based on spatial domain filters, frequency domain filters, Gabor filters, and wavelets.

- 3. Smith (1998) categorises non-structural texture feature extraction methods as:
 - Parametric PDF (probability density function) methods

This category includes autoregressive methods, Gauss MRFs, and clique MRFs.

• Non-parametric PDF methods

This category includes grey-level co-occurrence methods, grey-level difference methods, and texture spectrum methods. With regard to the latter class of methods, Smith (1998, p. 27) writes that:

Texture Spectrum methods use PDF models which are sensitive to high-order interactions. Typically, Texture Spectrum methods use a histogram model in which the partitioning of the intensity space is sensitive to high-order interactions between pixels. This sensitivity to high-order interactions between pixels is made feasible by quantizing the intensity values to a small number of levels. • Harmonic methods

This category includes Fourier power spectrum and autocorrelation methods.

• Primitive methods

This category includes methods that "measure the density of a set of texture primitives, such as lines, edges and intensity extrema, in a texture" (Smith, 1998, p. 31). The category coincides with the *signal processing* category of Tüceryan & Jain (1999).

• Blob and mosaic methods

This category includes methods that "segment the image into small regions" (Smith, 1998, p. 35). Such a "method models the image as composed of one simple but variable primitive; the features measured model the variations of the primitive found in the image" (Smith, 1998, p. 36).

• Fractal methods

This category is a sub-category of the *model-based* categories of both Reed & du Buf (1993) and Tüceryan & Jain (1999).

• Line methods

This category includes methods that measure "features from one-dimensional (though possibly curved) subsets of the texture" (Smith, 1998, p. 38).

Smith (1998, p. 7) states that the emerging consensus in the literature is "that texture is a locally structured 2-D homogeneous random field". Recall from Section 1.4.1.2 that a random field is a two-dimensional stochastic process (each pixel is a random variable X_p). The adjective homogeneous is synonymous with stationary and refers to the property that $P(X_p = x_p | \text{the neighbours of } p)$ depends only on the neighbourhood configuration and not on the location of p. What this means is that the local statistics (with respect to the neighbourhood configuration)—mean, variance, etc.—are the same for each pixel in the texture.

1.5 Problem definition

As noted in Section 1.2, nuclear texture features appear to be the most useful in detecting MACs. These features characterise the distribution of *chromatin*. Chromatin is the substance in the cell nucleus that forms the chromosomes or DNA and can be readily stained. Rousselle et al. (1999, p. 63) note that "most of the existing

methods used to quantify nuclear texture are based on the statistical or probabilistic assessment of the grey levels distribution in the image". The underlying assumption is that chromatin texture is suitably modelled as a two-dimensional, locally structured, homogeneous random field. As Figure 1.7 illustrates, the validity of this assumption is questionable. Wolf et al. (1995, p. 25) state that there are a number of inherent difficulties in trying to describe chromatin structure as (stochastic) texture:

(1) nuclear patterns are mixtures of texture and structure (nucleoli [small rounded bodies]) in changing amounts; (2) the chromatin texture may vary within the nucleus; for example, there could be irregularly formed regions of dense chromatin; and (3) the nuclear texture is given on a small area, and boundary effects play an important role.

A major criticism of the features derived from the statistical/stochastic approach is that they "usually cannot be interpreted in terms of chromatin organization such [as] compaction and distribution" (Rousselle et al., 1999, p. 63). Hence they are difficult to relate to the terms and adjectives—such as heterogeneity, granularity, margination, condensation, compaction, clearing, clumping, clod-like, diffuse, blobs, particles, granules, and particles—used by cytologists to describe chromatin (a detailed examination of the appearance and structure of chromatin as visualised by light microscopy is presented in Chapter 4). All of these observations suggest that chromatin texture is more suitably modelled using a structural approach, and more specifically a *blob and mosaic* method. The key here is the segmentation of the chromatin into its primitives.

Several algorithms for chromatin segmentation have been published in the literature and are reviewed in Chapter 4. The algorithms can be categorised according to the underlying segmentation methodology used: global thresholding, top-hat transform, grey-scale thinning, local adaptive thresholding, and region growing/merging (an overview of segmentation methods is also provided in Chapter 4). The algorithms based on global thresholding make no use of spatial information and cannot ensure the segmentation of the chromatin into regions consistent with what a cytoprofessional might perceive to be blobs or particles. Moreover, global thresholding is notorious for its lack of robustness to noise and/or uneven and variable illumination. All of the algorithms, with the exception of those based on threshold decomposition (thresholding over all possible grey levels) and grey-scale thinning, have in common that one or more operational parameters must be specified a priori. Moreover these need to be tuned to the particular application. As a consequence none of these methods can be robust to changes in, or non-uniformity of, illumination and staining. This in turn affects the quality of the chromatin features computed from such segmentations. In summary, a better structural model of nuclear chromatin (as visualised by light microscopy) is needed.

A strong motivation for pursuing the structural approach to modelling chromatin texture is that it can potentially offer features with discriminatory power superior to the statistical/stochastic texture features. Hallinan (2000, p. 57) has pointed out that it is highly unlikely that the MACs phenomenon described by Nieburgs and his colleagues in 1959, hereinafter referred to as *classical MACs*, is the same phenomenon as that detected quantitatively using computers and texture features. Descriptions of classical MACs support a structural model:

- "irregular intranuclear chromatin clumping and nuclear background clearing" (Romsdahl et al., 1964, p. 1403);
- "coarse irregular chromatin clumps; irregularly prominent nuclear membrane; multiple regular areas of chromatin clearing with central pinpoint chromatin condensation; nuclear lobulation, folding and layering" (Chomet et al., 1966, p. 197).

A study group was formed in November 1966 at the 14th Annual Scientific Meeting of the American Society of Cytology to investigate the significance of the classical MAC phenomenon. The group convened several times in 1967 and 1968. The findings of the group were reported at the 16th Annual Meeting of the American Society of Cytology. In particular the group concluded that a MAC positive nucleus must possess the following eight characteristics (Meisels, 1969, p. 476):

- 1. Pale areas between chromatin bands
- 2. Areas are circular
- 3. Circular areas are of uniform size
- 4. Chromocenters are small and their attached chromatin bands surrounding circular areas are curved
- 5. Chromocenters and the curved chromatin bands have uniformity of diameters and staining quality



Figure 1.7: Nuclei from several Papanicolaou-stained cervical cells as visualised by light microscopy. The nuclear chromatin has the appearance of a patchwork or mosaic of dark and light regions.

- 6. The chromocenters and chromatin bands are not pyknotic
- 7. All or 5 of 6 circular areas are lightly stained
- 8. Four of the circular areas are present in a row and two appear together with two of the circular areas of the row in a quadrant formation of four circular areas of clusters of circular areas occupying an entire lobule.

1.6 Aim and objectives

In light of the comments and criticisms outlined in the previous section, the primary aim of the research described in this thesis was to develop a structural model of chromatin (as visualised by light microscopy) from which features can be defined that can be directly related to the terms and adjectives used by cytoprofessionals to describe chromatin distribution/texture. A secondary aim was to demonstrate that such features can be used to detect nuclear changes during neoplasia¹¹, and malignancy associated changes. To this end the research was undertaken with the following objectives:

1. To develop a class of non-linear self-dual filters for the purpose of attenuating impulse-type noise in digital images;

 ¹¹ A tumour or *neoplasm* is "a relentlessly growing mass of abnormal cells" (Alberts et al., 2002, p. 1314). Tumour growth is called neoplasia.

- 2. To develop a robust algorithm for segmenting the chromatin in a digitised image of a cell nucleus (as visualised by light microscopy);
- 3. To develop a representation and description of the segmented chromatin that characterises the spatial relationship between chromatin regions and additionally incorporates scalar feature parameters associated with these regions;
- 4. To demonstrate that features derived from this representation and description can be related to the descriptive terms used by cytoprofessionals to describe chromatin distribution; and
- 5. To demonstrate that features so-derived can be used to discriminate between normal and abnormal Papanicolaou-stained cervical cytology slides.

These objectives address low-, intermediate-, and high-level processing tasks associated with an image analysis system (see Figure 1.1).

1.7 Scope of the research

A major impediment to the progress of the research outlined in this thesis was the lack of cell-image data suitable for developing a chromatin segmentation algorithm and for training and testing a classifier. Given that, ideally, at least 500 measurable cells per slide are needed to accurately estimate the means and standard deviations of most cell features on a slide (Palcic & MacAulay, 1994*a*, p. 59), and that there needs to be at least ten times as many slides per class as the number of features (discussed in Section 6.3.1), it would take hundreds of hours of painstaking work by a cytoprofessional to compile the requisite database. The associated cost, and the commercial sensitivity of MACs-based research in general, precluded the possibility of obtaining a database of images from a third party. Consequently the Cytometrics Project purchased a cytometer and adapted it for use with the Papanicolaou stain (this is detailed in Chapter 6). The associated delays and uncertainties are reflected in the theoretical and algorithmic emphasis of Chapters 2-5 of this thesis.

The research described in this thesis is concerned with the preprocessing of cell nuclei images to remove noise, the segmentation of the chromatin within these images, the representation and description of the segmented chromatin, the extraction of features describing the distribution of the chromatin, and the classification of slides (populations of cells) using these features. Cell scene segmentation and nucleus segmentation (see Figure 1.8) are not the subject of this thesis. Details of these

Researcher	Area of research
Bradley (1996)	Machine learning / classification
Walker (1997)	Texture analysis
Smith (1998)	Texture analysis
Bamford (1999)	Scene and nucleus segmentation
Hallinan (2000)	Classification
Jones (2001)	Texture analysis

Table 1.1: Ph.D. research contributing to the overall research initiative in the Cytometrics Project.

steps can be found in Bamford & Jackway (2001), Bamford & Lovell (1999), and Bamford (1999).

As noted at the beginning of the chapter, the research outlined in this thesis constitutes part of a larger initiative to develop an automated image analysis system (cytometer) for screening Papanicolaou-stained cervical smears. The research of several Ph.D. students within CSSIP has contributed toward this goal (see Table 1.1).

1.8 Structure of this thesis

This chapter has:

- Described the rationale for undertaking the research described in this thesis and the environment in which this research was undertaken.
- Provided an overview of Pap smear cytology, including the motivation for automated Pap smear screening, and the difficulties associated with developing such a screener: in particular the difficulty identifying the rare occurrence of diagnostic cells on a slide because of the nature of the Pap smear, and sampling error. The literature suggests that the MACs phenomenon—subtle subvisual changes in otherwise normal-appearing cells on atypical smears—has the potential to solve these problems. MACs-based screening eliminates the need to exhaustively scan a slide looking for diagnostic cells. The phenomenon has been demonstrated in the literature on the basis of quantitative measurements (features) made on populations of cells. The literature suggests that the most discriminatory features are nuclear texture features.
- Provided an overview of statistical pattern recognition—a feature/measurementbased approach to classification—in computer vision.



Figure 1.8: Scene and nucleus segmentation. (a) The cytometer automatically captures scenes from the deposition area on the slide according to a predefined pattern. (b) In each scene the cytometer identifies the locations of nucleus-like objects. (c) For each location identified, the cytometer captures an in-focus image containing the nucleus-like object and then segments the nucleus-like object from this image.

- Provided an overview of the different approaches to texture feature extraction published in the literature. Two computational approaches to the definition of texture have been the mainstay of the texture analysis literature: the *structural* approach and the *stochastic* approach.
- Identified that the stochastic approach to quantifying chromatin texture has predominated in the literature and that the efficacy of this approach is questionable because: (i) chromatin texture exhibits structure—clumping, margination, granulation, condensation, clearing—which is not compatible with the definition of texture as a locally-structured two-dimensional homogeneous random field; and (ii) statistical/stochastic features are difficult to relate to the terms and adjectives used by cytoprofessionals to describe chromatin distribution.
- Defined the aim, objectives, and scope of this research.

The remainder of the thesis is organised as follows:

- Chapter 2 This chapter establishes the theoretical framework—fundamental concepts, definitions, and notation—used throughout the thesis. The chapter serves to acquaint the reader with sets and ordering (essential to the exposition in Chapter 3), complete lattices (variously used in this thesis to model binary and grey-scale images, the components of an abstract space of *folded functions*, and binary and grey-scale graphs), metric spaces, and mathematical morphology for complete lattices.
- **Chapter 3** This chapter deals with the topic of noise filtering, and in particular with attenuating impulse-type noise that can be induced by camera/framegrabber electronics when acquiring digital images. Noise filtering typically precedes image segmentation. The chapter presents a review of existing approaches to constructing non-linear self-dual filters based on morphological operators, and proffers a new theoretical approach to constructing non-linear self-dual filters from morphological operators defined on an abstract space called *fold space*. Self-dual filters, in general, are of interest in image analysis because they treat the dark and light areas of an image in an equivalent manner. This is useful when trying to remove salt-and-pepper noise or more generally when trying to filter images such as textures for which there is no distinction between foreground and background. Non-linear self-dual filters:

they do not induce ringing and blurring, they can be designed such that they do not reduce the high frequencies and dynamic range in the image, they can be designed such that they do not introduce new grey values into the image, and they can be designed to be independent of monotone changes in intensity.

- Chapter 4 This chapter deals with the problem of segmenting chromatin as visualised by light microscopy. The chapter presents an overview of grey-scale image segmentation methods published in the literature; an overview of the nature of chromatin and in particular its appearance and structure as revealed by light microscopy; a review of chromatin segmentation methods published in the literature; a new algorithm for the segmentation of chromatin based on seeded region growing; a new seeded region growing algorithm that retains the advantages of that proposed by Adams & Bischof (1994) but is independent of the order of pixel processing; and proffers a new ascending priority queue for use in implementing the watershed transform (a particular case of seeded region growing).
- Chapter 5 This chapter deals with the problem of representing and describing objects—e.g. chromatin particles—in the plane. In particular the chapter examines the notions of adjacency, distance, and measurement. The chapter presents an overview of (geometric) adjacency graphs; a review of the Voronoi diagram and the graphs related to it; a review of the area Voronoi diagram; a review of distance transform algorithms published in the literature; a new theoretical result concerning the distance transform of a binary image, where the underlying distance is based on a *positive definite quadratic form*, and the erosion of its characteristic function by an *elliptic poweroid* structuring element; a new algorithm for computing the exact Euclidean distance transform on the hexagonal grid; an overview of the skeleton by influence zones, the region adjacency graph, and the perceptual graph; a generalisation of the grey-level co-occurrence matrix method to vertex-weighted adjacency graphs; and an overview of the types of parameters (attributes) that can be measured for image objects.
- **Chapter 6** This chapter discusses the application of the material in the preceding chapters to the problem of screening cytology slides. In particular the chapter discusses the motivation for designing a MACs-based classifier for cytological screening; presents an overview of statistical pattern recognition, including important issues such as the *curse of dimensionality*, dimensionality reduction, classifier design, and classifier evaluation; and presents a case

study in which features derived from the representation and description of the segmented chromatin are used to build several MACs-based classifiers for screening Papanicolaou-stained cervical smears.

Chapter 7 This chapter reviews the thesis, summarises its key contributions and findings, and discusses the implications of these results. In addition, the chapter outlines the limitations of the research undertaken, and the opportunities for further research.

Chapters 3–6 address the aim and objectives outlined in Section 1.6. These are summarised in the flowchart shown in Figure 1.9. This flowchart is reproduced at the beginning of each of these middle chapters to remind the reader of the aim and objectives, and to identify which part or parts of the flowchart are addressed by the respective chapter.



Figure 1.9: Flowchart summarising the aims and objectives of the thesis and the corresponding chapters in which each is discussed and developed.

Chapter 2

Theoretical Framework

There cannot be a language more universal and more simple, more free from errors and obscurities, more worthy to express the invariable relations of natural things [than mathematics]

Jean Baptiste Joseph Fourier, Analytic Theory of Heat, 1822

The purpose of this chapter is to acquaint the reader with fundamental concepts and definitions that are used in subsequent chapters of this thesis. The first three sections of this chapter deal with sets and ordering. The concept of ordering in sets is central to the exposition on folding induced self-dual filters (FISFs) presented in the next chapter. Section 2.4 discusses lattices and in particular, complete lattices. The complete lattice, which is in essence an ordered set equipped with a supremum and infimum, is the algebraic framework that underlies all of the theoretical chapters in this thesis. In this thesis complete lattices are variously used to model binary and grey-scale images, components of an abstract space of folded functions (Chapter 3), and binary and grey-scale graphs (Chapter 5). Section 2.5 deals with operators on complete lattices (of which an FISF is an example). Section 2.6 discusses *mathematical morphology* for complete lattices. Mathematical morphology is a non-linear theory "concerned with the processing and analysis of images, using operators and functionals based on topological and geometrical concepts" (Heijmans, 1994a, preface). The complete lattice framework represents the most general algebraic framework in which mathematical morphology can be defined and studied. The adoption of this framework eliminates the need to replicate definitions and results for different objects spaces: binary images, grey-scale images, graphs, etc. Nevertheless, from an image analysis point of view the complete lattice is not in itself a rich

enough structure to describe the useful morphological operations. It is necessary to endow the complete lattice with certain geometrical and topological properties such as translation invariance and connectivity. In the case of binary and grey-scale images, for example, this permits the definition of mathematical morphology operators that "satisfy, besides the usual algebraic properties..., geometrical invariance properties (translation invariance) also" (Heijmans, 1994*a*, p. 14). For this reason Sections 2.7–2.10 give a brief treatment of mathematical morphology for binary images, grey-scale images, metric spaces, and graphs. The material in this chapter is drawn primarily from the following monographs: Serra (1988*d*), Heijmans (1994*a*), Birkhoff (1948), Donnellan (1968), and Szász (1963).

2.1 Sets and binary relations

A binary relation, defined on an arbitrary set, is a rule that prescribes a relationship between pairs of elements. Formally, a binary relation is defined as follows.

Definition 2.1.1 (binary relation). Let S be an arbitrary set and let \mathfrak{R} be a subset of $S \times S$. The set \mathfrak{R} is called a binary (or dyadic) *relation* on S. If the ordered pair $(X, Y) \in \mathfrak{R}$ then we write $X\mathfrak{R}Y$ and say that "the relation \mathfrak{R} holds between X and Y".

Example. Consider the set of natural numbers $\mathbb{N} = \{1, 2, 3, ...\}$ and the relation "is the immediate predecessor of". This relation is explicitly

$$\mathfrak{R} = \{(1,2), (2,3), (3,4), \ldots\}.$$

One can associate with each binary relation another relation called the converse relation.

Definition 2.1.2 (converse relation). If \mathfrak{R} is a binary relation defined on a set S then the *converse* relation is the relation \mathfrak{R}' such that $X\mathfrak{R}'Y$ if and only if $Y\mathfrak{R}X$.

Example. The converse of the relation \mathfrak{R} in the previous example is the relation "is the immediate successor of". This relation is explicitly $\mathfrak{R}' = \{(2, 1), (3, 2), (4, 3), \ldots\}$.

The following definition lists some important properties used to characterise binary relations.

Definition 2.1.3 (properties of binary relations). A binary relation \mathfrak{R} defined on a set \mathcal{S} is said to be

- 1. reflexive if $X \Re X$ for all $X \in S$;
- 2. *irreflexive* if $X \Re X$ for no $X \in S$;
- 3. transitive if $X\Re Y$ and $Y\Re Z \Rightarrow X\Re Z$ for all $X, Y, Z \in S$;
- 4. symmetric if $X\Re Y \Rightarrow Y\Re X$ for all $X, Y \in \mathcal{S}$; and
- 5. anti-symmetric if $X\Re Y$ and $Y\Re X \Rightarrow X = Y$ for all $X, Y \in S$.

The symbol "=" used in the definition of anti-symmetry denotes the *equality relation* which is the binary relation defined as follows.

Definition 2.1.4 (equality relation). Given a set S with elements X, Y, Z, ..., the binary relation $\mathfrak{R} = \{(X, X), (Y, Y), (Z, Z), ...\}$ is called the *equality relation*.

2.2 Equivalence relations

The equality relation is a special example of a type of binary relation known as an *equivalence relation*.

Definition 2.2.1 (equivalence relation). A binary relation \mathfrak{R} defined on a set \mathcal{S} is said to be an *equivalence relation* if it is reflexive, transitive, and symmetric.

Examples.

- 1. Let \mathcal{S} be the set of all triangles in the plane. The relation "is similar to" is an equivalence relation.
- 2. Consider the set of natural numbers \mathbb{N} . The relation $x\mathfrak{R}y \Leftrightarrow x = y$, for all $x, y \in \mathbb{N}$ is a special equivalence relation known as the equality relation. This relation is explicitly $\mathfrak{R} = \{(1, 1), (2, 2), (3, 3), \ldots\}$.

A feature of equivalence relations is that they induce a partition of the set on which they are defined into disjoint non-empty sets called equivalence classes. The relationship between equivalence relations and equivalence classes is characterised in the following theorem.

Theorem 2.2.2 (Donnellan (1968, p. 14)). Any equivalence relation \Re defined on a non-empty set S induces a partition of S into disjoint non-empty subsets, called equivalence classes, which contain all the elements of S. Conversely, any partition of S into disjoint subsets such that every member of S is contained in some subset and no member of S is in more than one subset, induces an equivalence relation on S.

2.3 Order relations

The equality relation is also an example of a type of binary relation known as a *partial order relation*.

Definition 2.3.1 (partial order relation). A binary relation \Re defined on a set S is said to be a *partial order relation* if it is reflexive, transitive, and anti-symmetric.

Remark. The equality relation is the only relation that is both an equivalence relation and a partial order relation (Szász, 1963, p. 14).

Example. The relation "is less than or equal to" defined on the set of natural numbers \mathbb{N} is a partial order relation. The ordered pair (3,7) is an element of this relation and we usually write that $3 \leq 7$.

An arbitrary set, together with a partial order relation defined on it, is known as a *partially ordered set* or *poset*.

Definition (partially ordered set). Let \mathfrak{R} be a partial order relation defined on a set \mathcal{S} . The pair $(\mathcal{S}, \mathfrak{R})$ is called a *partially ordered set* or *poset*.

Example. The pair (\mathbb{N}, \leq) is a poset.

A poset for which the partial order relation defines an ordering for all pairs of elements is called a *chain*.

Definition 2.3.2 (chain). Let (S, \mathfrak{R}) be a poset. If $X\mathfrak{R}Y$ or $Y\mathfrak{R}X$ or both for all $X, Y \in S$ then (S, \mathfrak{R}) is said to be *totally* or *linearly* ordered and the poset is called a *chain*.

Example. The poset (\mathbb{N}, \leq) is a chain. It is totally ordered because for every pair of distinct elements $x, y \in \mathbb{N}$ we can write that $x \leq y$ or $y \leq x$. The poset $(\mathcal{P}(\mathbb{R}^2), \subseteq)$, on the other hand, is not a chain—there exist many pairs of distinct elements (sets) $X, Y \in \mathcal{P}(\mathbb{R}^2)$ such that $X \nsubseteq Y$ and $Y \nsubseteq X$.

2.3.1 Hasse diagrams

If (S, \mathfrak{R}) is a poset and S is non-empty and finite then the partial order relation on S can be represented graphically as a *Hasse diagram*. Each element of S is drawn as a labelled dot in the plane. If X and Y are two distinct elements of S and $X\mathfrak{R}Y$ then the dot representing Y is drawn above the dot representing X. If, in addition, there does not exist another element $Z \in S$ such that $X\mathfrak{R}Z$ and $Z\mathfrak{R}Y$, then a line segment is drawn between the dots representing X and Y. Some examples of Hasse diagrams are shown in Figure 2.1.



Figure 2.1: Hasse diagrams. (a) The Hasse diagram for the chain (S, \leq) where $S = \{1, 2, 3, 4\}$. (b) The Hasse diagram for the poset $(\mathcal{P}(S), \subseteq)$ where $S = \{1, 2, 3\}$.

2.3.2 The principle of duality with respect to order

A feature of partial order relations is that their corresponding converse relations are also partial order relations.

Theorem 2.3.3 (Birkhoff (1948, p. 3)). The converse of any partial ordering is itself a partial ordering.

Remark. The implication of this theorem is that if $(\mathcal{S}, \mathfrak{R})$ is a poset then $(\mathcal{S}, \mathfrak{R}')$, where \mathfrak{R}' is the converse relation, is also a poset. Moreover, for every statement, definition, or property pertaining to $(\mathcal{S}, \mathfrak{R})$, a corresponding one is obtained for $(\mathcal{S}, \mathfrak{R}')$ by simply interchanging the role of \mathfrak{R} and \mathfrak{R}' . This is known as the *principle* of duality with respect to order. The poset $(\mathcal{S}, \mathfrak{R}')$ is said to be the dual of $(\mathcal{S}, \mathfrak{R})$ and vice versa.

2.3.3 Quasi-ordering

A binary relation that is reflexive and transitive is called a relation of quasi-ordering.

Definition 2.3.4 (quasi-ordering). A binary relation \mathfrak{R} defined on a set \mathcal{S} is said to be a relation of *quasi-ordering* if it is reflexive and transitive.

Remark. The difference between a relation of quasi-ordering and a partial order relation is that the latter is also anti-symmetric (we know a priori a relation of equality) (Donnellan, 1968, p. 264).

A relation of quasi-ordering can be used to define an equivalence relation as follows.

Theorem 2.3.5 (Schröder's theorem (Birkhoff, 1948, p. 4)). Let \mathfrak{R} be a relation of quasi-ordering defined on a set S. The relation \mathfrak{E} defined

 $X \mathfrak{E} Y$ if and only if $X \mathfrak{R} Y$ and $Y \mathfrak{R} X$

is an equivalence relation.

A relation of quasi-order together with an equivalence relation defined on the same set can be used to define a partial order relation on the equivalence classes as follows.

Theorem 2.3.6 (Donnellan (1968, p. 265)). Let \mathfrak{Q} be a relation of quasi-order defined on a set S (with elements x, y, z, \ldots), and let \mathfrak{E} be an equivalence relation defined on S that partitions it into classes $X, Y, Z \ldots$ of equivalent elements. Let Tbe the set $\{X, Y, Z, \ldots\}$. If we define a dyadic relation \leq on T as follows:

 $X \leq Y$ if and only if $x \mathfrak{Q} y$

for some $x \in X$ and some $y \in Y$, then (\mathcal{T}, \leq) is a poset.

2.3.4 Isomorphism

An *isomorphism* is an order preserving mapping between two posets, and a *dual isomorphism* is an order reversing mapping between two posets. Formally they are defined as follows.

Definition 2.3.7 (isomorphism and dual isomorphism). Let (S, \leq) and (\mathcal{T}, \leq) be two posets. Let the mapping $\Psi : S \to \mathcal{T}$ between the two posets be a bijection (it is one-to-one and onto, i.e. there is a one-to-one correspondence between the elements of S and the elements of \mathcal{T}). If both Ψ and its inverse Ψ^{-1} are order preserving, i.e.

 $X \leq Y$ if and only if $\Psi(X) \preceq \Psi(Y)$

for all $X, Y \in \mathcal{S}$ then Ψ is said to be an *isomorphism*. If both Ψ and its inverse Ψ^{-1} are order reversing, i.e.

$$X \leq Y$$
 if and only if $\Psi(Y) \preceq \Psi(X)$

for all $X, Y \in \mathcal{S}$ then Ψ is said to be a *dual isomorphism*.

Remarks.

- 1. If an isomorphism exists between two posets then the posets are said to be *isomorphic*.
- 2. An isomorphism between a poset and itself is called an *automorphism*.
- 3. A dual isomorphism between a poset and itself is called a *dual automorphism*.

Example. The familiar real number line is the poset (\mathbb{R}, \leq) . The negation operator, -, is a dual automorphism.

2.4 Lattices

A lattice may be looked at in two distinct ways: (i) as a poset (in the context of set theory), or (ii) as an algebra (in the context of algebra theory).

2.4.1 Set theory definition of a lattice

The set theory definition of a lattice first requires that the *infimum* and *supremum* of a poset be defined.

Definition 2.4.1 (infimum). Let (S, \mathfrak{R}) be a poset and let \mathcal{T} be a non-empty subset of S. A *lower* bound of \mathcal{T} is an $L \in S$ such that $L\mathfrak{R}X$ for every $X \in \mathcal{T}$. A greatest lower bound (infimum, inf, or \bigwedge) of \mathcal{T} is a lower bound L_0 such that $L\mathfrak{R}L_0$ for every other lower bound $L \in S$.

Definition 2.4.2 (supremum). Let (S, \mathfrak{R}) be a poset and let \mathcal{T} be a non-empty subset of S. An *upper* bound of \mathcal{T} is a $U \in S$ such that $X\mathfrak{R}U$ for every $X \in \mathcal{T}$. A *least upper bound* (supremum, sup, \bigvee) of \mathcal{T} is an upper bound U_0 such that $U_0\mathfrak{R}U$ for every other upper bound $U \in S$.

A lattice is a poset for which every pair of elements has a supremum and an infimum.

Definition 2.4.3 (lattice). Let (\mathcal{L}, \leq) be a poset. If every pair of elements $X, Y \in \mathcal{L}$ has an infimum (written $X \wedge Y$) and a supremum (written $X \vee Y$) then the poset is said to be a *lattice*.

When only the infimum or only the supremum exists for each pair of elements then the poset is called a semilattice. **Definition 2.4.4 (semilattice).** Let (\mathcal{L}, \leq) be a poset. If every pair of elements $X, Y \in \mathcal{L}$ has an infimum (written $X \wedge Y$) then the poset is said to be an *inf* semilattice. If every pair of elements $X, Y \in \mathcal{L}$ has a supremum (written $X \vee Y$) then the poset is said to be a sup semilattice.

Remark. A lattice is both an inf and sup semilattice.

2.4.2 Algebra definition of a lattice

"A set closed with respect to one or more specified finitary operations [i.e. n-ary operations where n is finite] is called an algebra" (Donnellan, 1968, p. 10). Hence a lattice can be defined as follows.

Definition 2.4.5 (Birkhoff (1948, p. 18)). A lattice \mathcal{L} is an algebra with two binary operations (symbolised by \vee and \wedge) satisfying for all $X, Y, Z \in \mathcal{L}$

- 1. *idempotence:* $X \wedge X = X$ and $X \vee X = X$;
- 2. commutativity: $X \wedge Y = Y \wedge X$ and $X \vee Y = Y \vee X$;
- 3. associativity: $X \land (Y \land Z) = (X \land Y) \land Z$ and $X \lor (Y \lor Z) = (X \lor Y) \lor Z$;
- 4. absorption: $X \land (X \lor Y) = X$ and $X \lor (X \land Y) = X$.

Remarks.

- 1. \mathcal{L} is an algebra because it is closed with respect to the operations \vee and \wedge ; i.e. $X \wedge Y \in \mathcal{L}$ and $X \vee Y \in \mathcal{L}$ for all $X, Y \in \mathcal{L}$.
- 2. The partial order relation associated with this lattice is given by $X \leq Y \Leftrightarrow X \wedge Y = X$, or equivalently $X \leq Y \Leftrightarrow X \vee Y = Y$ (Szász, 1963, p. 38).

The algebra definition of a semilattice is as follows.

Definition 2.4.6 (Szász (1963, p. 38)). A semilattice is an algebra with one binary operation that is *idempotent, commutative,* and *associative.*

Remark. A lattice is both a semilattice with respect to the operator \land and a semilattice with respect to the operator \lor .
2.4.3 Properties of lattices

Definition 2.4.7 (properties of lattices). In any lattice (\mathcal{L}, \leq) the following inequalities hold true for all $X, Y, Z \in \mathcal{L}$ (Szász, 1963, pp. 79–86):

- 1. modular inequality: $X \lor (Y \land Z) \le (X \lor Y) \land Z$ if $X \le Z$; and
- 2. distributive inequality: $X \lor (Y \land Z) \le (X \lor Y) \land (X \lor Z)$.

Remarks.

- 1. The modular inequality is self-dual.
- 2. Applying the principle of duality with respect to order, the dual of the distributive inequality must also hold true, namely:

$$X \land (Y \lor Z) \ge (X \land Y) \lor (X \land Z).$$

2.4.4 Modular and distributive lattices

A lattice is said to be *modular* if the modular inequality holds true when the (first) inequality (\leq) is replaced by an equality (=).

Definition 2.4.8 (modular lattice). A lattice (\mathcal{L}, \leq) is said to be *modular* if it satisfies

$$X \lor (Y \land Z) = (X \lor Y) \land Z \quad \text{if } X \le Z$$

for all $X, Y, Z \in \mathcal{L}$.

Remark. This identity is self-dual.

A lattice is said to be *distributive* if the distributive inequalities are in fact equalities.

Definition 2.4.9 (distributive lattice). A lattice (\mathcal{L}, \leq) is said to be *distributive* if

$$X \lor (Y \land Z) = (X \lor Y) \land (X \lor Z),$$

$$X \land (Y \lor Z) = (X \land Y) \lor (X \land Z)$$

for all $X, Y, Z \in \mathcal{L}$.

Remark. Every distributive lattice is modular but, in general, the converse is not true (Serra, 1988*d*, p. 124).

A feature of distributive lattices is that

if
$$X \wedge Z = Y \wedge Z$$
 and $X \vee Z = Y \vee Z$ then $X = Y$.

This property also holds true for modular lattices when X and Y are comparable; i.e. when either $X \leq Y$ or $Y \leq X$ (Serra, 1988*d*, p. 37).

2.4.5 Complete lattices

The complete lattice is the algebraic framework in which the theory of mathematical morphology is defined and studied.

Definition 2.4.10 (complete lattice). A lattice (\mathcal{L}, \leq) is said to be *complete* if every subset \mathcal{K} of \mathcal{L} has both a supremum (written $\bigvee \mathcal{K}$) and an infimum (written $\bigwedge \mathcal{K}$).

Remarks.

- 1. If \mathcal{K} has only a finite number of elements X_1, X_2, \ldots, X_n then we write $X_1 \wedge X_2 \wedge \ldots \wedge X_n$ or $\bigwedge_i X_i$ in place of $\bigwedge \mathcal{K}$ (the notation for the supremum is similarly expressed).
- 2. Implicit in this definition is existence of a greatest element U called the *uni-versal element* and a least element O called the *null element*.
- 3. "In a complete lattice every element is both an upper bound and a lower bound of the empty set" (Heijmans, 1994*a*, p. 20). Consequently $U = \bigvee \mathcal{L} = \bigwedge \emptyset$ and $O = \bigwedge \mathcal{L} = \bigvee \emptyset$.
- 4. Not every lattice is complete (Heijmans, 1994*a*, p. 20). Consider the interval (0, 1]. It is a lattice (and also a chain) for the usual partial ordering. However it is not a complete lattice because, for example, the set $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$ does not have a lower bound (this set is actually the infinite sequence $\{\frac{1}{n}\}_{1}^{\infty}$ for which $\lim_{n\to\infty}\frac{1}{n}=0$).

Example. The finite set $\{0, 1, \ldots m\}$ is a complete lattice for the usual partial ordering. The sets \mathbb{R} and \mathbb{Z} are not complete lattices because they each do not have a universal and null element. However, the extended sets $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ and $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$ are complete lattices.

The *complete Boolean lattice* and the *power lattice* are two important types of complete lattice that are used in image analysis.

2.4.5.1 Complete Boolean lattice

Definition 2.4.11 (complete Boolean lattice). A complete lattice (\mathcal{L}, \leq) is said to be a *complete Boolean lattice* if

- 1. it is distributive; and
- 2. for each $X \in \mathcal{L}$, there exists a unique $X^{c} \in \mathcal{L}$ (called the *complement* of X) such that $X \vee X^{c} = U$ and $X \wedge X^{c} = O$.

Example. Let S be an arbitrary set. The pair $(\mathcal{P}(S), \subseteq)$ is a complete Boolean lattice with null element \emptyset and universal element S. Set complementation is the complement operator.

Remark. In image analysis the complete Boolean lattice $(\mathcal{P}(\mathcal{S}), \subseteq)$ serves as a mathematical model for binary images (in which case \mathcal{S} is usually \mathbb{Z}^n or \mathbb{R}^n representing the space of pixel coordinates) and binary graphs (in which case \mathcal{S} is a finite set of vertices).

2.4.5.2 Power lattice

Definition 2.4.12 (power lattice). Let S be an arbitrary non-empty set and let (\mathcal{T}, \leq) be a complete lattice. Let \mathcal{T}^{S} denote the space of all mappings of the set S into the set \mathcal{T}^{-1} . The partial ordering on \mathcal{T} can be used to define the following partial ordering on \mathcal{T}^{S} :

$$f \leq g$$
 if $f(x) \leq g(x) \ \forall x \in \mathcal{S}$,

for $f, g \in \mathcal{T}^{\mathcal{S}}$. The pair $(\mathcal{T}^{\mathcal{S}}, \leq)$ is a complete lattice called a *power lattice*.

Remarks.

- 1. For simplicity, the same symbol \leq is used for both the lattice (\mathcal{T}, \leq) and $(\mathcal{T}^{\mathcal{S}}, \leq)$.
- 2. This lattice inherits many of the properties of the lattice (\mathcal{T}, \leq) . Importantly, if (\mathcal{T}, \leq) is distributive then $(\mathcal{T}^{\mathcal{S}}, \leq)$ is also distributive. Similar conclusions hold for other properties such as modularity (Heijmans, 1994*a*, p. 27).

¹ "A mapping of a set A into a set B is a subset of $A \times B$ in which each element of A occurs once and only once as the first component in the elements of the subset" (Ayres, 1965, p. 6). Such a mapping is better known as a *function*.

- 3. When $\mathcal{T} = \{0, 1\}$ the power lattice $(\mathcal{T}^{\mathcal{S}}, \leq)$ is isomorphic to the complete Boolean lattice $(\mathcal{P}(\mathcal{S}), \leq)$.
- 4. In image analysis the power lattice $(\mathcal{T}^{\mathcal{S}}, \leq)$ serves as a model for grey-scale images and grey-scale graphs. When dealing with grey-scale images \mathcal{S} is usually taken to be \mathbb{R}^n or \mathbb{Z}^n (representing pixel coordinates) and when dealing with grey-scale graphs \mathcal{S} is taken to be a finite set of vertices $V = \{v_1, v_2, \ldots, v_n\}$. The set \mathcal{T} defines a set of grey-levels that each element (e.g. pixel) of the domain space may map to. In image processing \mathcal{T} is typically one of the infinite sets $\mathbb{R} = \mathbb{R} \cup \{-\infty, +\infty\}$ or $\mathbb{Z} = \mathbb{Z} \cup \{-\infty, +\infty\}$, or the finite set $\{0, 1, \ldots, m\}$, each of which is a complete lattice for the usual partial order relation \leq .

A comparison between the lattices $(\mathcal{P}(\mathcal{S}), \subseteq)$ and $(\mathcal{T}^{\mathcal{S}}, \leq)$ is shown in Table 2.1.

2.5 Operators on a complete lattice

The following definition characterises some important properties of operators defined on a complete lattice.

Definition 2.5.1. Let (\mathcal{L}, \leq) be a complete lattice and let $\mathcal{O}(\mathcal{L})$ —or simply \mathcal{O} when the context is clear—be the set of all operators $\psi : \mathcal{L} \to \mathcal{L}$. An operator $\psi \in \mathcal{O}$ is said to be

- 1. increasing if $X \leq Y \Rightarrow \psi(X) \leq \psi(Y) \ \forall X, Y \in \mathcal{L};$
- 2. extensive if $X \leq \psi(X) \ \forall X \in \mathcal{L};$
- 3. anti-extensive if $\psi(X) \leq X \forall X \in \mathcal{L}$;
- 4. *idempotent* if $\psi(\psi(X)) = \psi(X) \ \forall X \in \mathcal{L}$;
- 5. a negation² if it is a dual automorphism and satisfies $\psi(\psi(X)) = X \forall X \in \mathcal{L};$
- 6. self-dual if $\psi(X) = [\psi(X^*)]^* \quad \forall X \in \mathcal{L}$, where * denotes a negation;
- 7. a *dilation* if it commutes with the supremum, i.e. $\psi(\bigvee_i X_i) = \bigvee_i \psi(X_i)$ for every family $\{X_i\}$ of elements in \mathcal{L} ;

² A complete lattice may or may not have a negation. The existence of this operator depends entirely upon the nature of \mathcal{L} . The set of all closed subsets of \mathbb{R}^2 together with the partial order relation \subseteq is an example of a complete lattice that does not possess a negation (Heijmans, 1994*a*, p. 13).

	Complete Boolean lattice $(\mathcal{P}(\mathcal{S}), \subseteq)$	Power lattice $(\mathcal{T}^{\mathcal{S}},\leq)$
Set	all subsets of ${\cal S}$	all functions $f: \mathcal{S} \to \mathcal{T}$
Elements	$A, B, \ldots \in \mathcal{P}\left(\mathcal{S}\right)$	$f, g, \ldots \in \mathcal{T}^{\mathcal{S}}$
Partial order relation	\subseteq	$f \leq g \text{ if} f(x) \leq g(x) \ \forall x \in \mathcal{S}$
Universal element	S	$u\left(x\right)=\bigvee\mathcal{T}$
Null element	Ø	$o(x) = \bigwedge \mathcal{T}$

 Table 2.1: Complete lattices in image analysis.

 \mathcal{S} is an arbitrary set (\mathcal{T}, \leq) is a complete lattice

- 8. an erosion if it commutes with the infimum, i.e. $\psi(\bigwedge_i X_i) = \bigwedge_i \psi(X_i)$ for every family $\{X_i\}$ of elements in \mathcal{L} ;
- 9. an opening if it is increasing, anti-extensive and idempotent;
- 10. a *closing* if it is increasing, extensive and idempotent.

Remarks.

- 1. Dilation and erosion are dual notions; i.e. they satisfy the principal of duality with respect to order.
- 2. Opening and closing are also dual notions. In the literature these operators are sometimes called algebraic opening and closing (e.g. Serra (1988d)) or generalised opening and closing (e.g. Haralick (1989)), respectively.

The partial ordering of the complete lattice of operands (\mathcal{L}, \leq) induces a partial ordering on $\mathcal{O}(\mathcal{L})$. Moreover $\mathcal{O}(\mathcal{L})$ is a complete lattice for this partial ordering.

Proposition 2.5.2. The partial ordering of the lattice (\mathcal{L}, \leq) induces the following partial ordering on the set of operators $\mathcal{O}(\mathcal{L})$:

$$\psi \leq \eta \text{ if } \psi(X) \leq \eta(X) \text{ for all } X \in \mathcal{L}$$

where $\psi, \eta \in \mathcal{O}(\mathcal{L})$. Moreover the pair $(\mathcal{O}(\mathcal{L}), \leq)$ is itself a complete lattice (Heijmans, 1994a, p. 45). The null element of this lattice is the operator $o \in \mathcal{O}$ that maps every element of \mathcal{L} onto the null element of (\mathcal{L}, \leq) ; i.e. o(X) = O for all $X \in \mathcal{L}$. The universal element of this lattice is the operator $u \in \mathcal{O}$ that maps every element of \mathcal{L} onto the universal element of (\mathcal{L}, \leq) , i.e. u(X) = U for all $X \in \mathcal{L}$.

Remarks.

- 1. $\mathcal{O}(\mathcal{L})$ is in fact the power lattice $\mathcal{L}^{\mathcal{L}}$.
- 2. To keep the notation simple the same symbols are used to represent the partial ordering, the supremum, and the infimum as are used for the complete lattice of operands (\mathcal{L}, \leq) . Hence, for example, if I denotes the identity mapping i.e. I(X) = X for all $X \in \mathcal{L}$ —then the extensive property of definition 2.5.1 can be written: $\psi \in \mathcal{O}$ is extensive if $I \leq \psi$ (here the symbol \leq refers to the partial order relation on the complete lattice $(\mathcal{O}(\mathcal{L}), \leq)$ rather than (\mathcal{L}, \leq)).

If the lattice of operands has a negation then the lattice of operators also has a negation.

Definition 2.5.3 (negative of an operator). If (\mathcal{L}, \leq) is a complete lattice with a negation, then the negative of an operator $\psi \in \mathcal{O}$ is defined

$$\psi^*\left(X\right) = \left[\psi\left(X^*\right)\right]^*.$$

Remark. In operator notation, the self-dual property of definition 2.5.1 can be written: $\psi \in \mathcal{O}$ is self-dual if $\psi = \psi^*$.

The following proposition lists several key properties concerning an operator and its negation.

Proposition 2.5.4 (Heijmans (1994*a*, **p. 46)).** If (\mathcal{L}, \leq) is a complete lattice with a negation and $\varphi, \psi, \psi_i \in \mathcal{O}(\mathcal{L})$ then

- 1. ψ is increasing if and only if ψ^* is increasing;
- 2. $\varphi \leq \psi$ if and only if $\psi^* \leq \varphi^*$;
- 3. $\left(\bigwedge_{i} \psi_{i}\right)^{*} = \bigvee_{i} \psi_{i}^{*}$ and $\left(\bigvee_{i} \psi_{i}\right)^{*} = \bigwedge_{i} \psi_{i}^{*};$

4.
$$(\varphi \psi)^* = \varphi^* \psi^*$$
.

2.6 Mathematical morphology for complete lattices

The monograph Image Analysis and Mathematical Morphology by Serra (1982) is generally considered to be the first systematic treatment of mathematical morphology as an approach to image analysis. The main theoretical foundations of the theory, however, were established years earlier in the monograph Random Sets and Integral Geometry by Matheron (1975). These monographs "discuss a number of mappings on subsets of the Euclidean plane (which serve as a model for continuous binary images), which have in common that they are based on set-theoretical operations (union, intersection, complementation) as well as translations" (Heijmans, 1994a, p. 15). Subsequently Matheron and Serra extended the theory of mathematical morphology to arbitrary complete lattices. This theoretical advancement is discussed in volume 2 of Image Analysis and Mathematical Morphology (Serra, 1988*d*). Their work has since been extended by other researchers, most notably Henk Heijmans and Christian Ronse. Heijman's monograph *Morphological Image Operators* (Heijmans, 1994*a*) presents a rigorous mathematical treatment of mathematical morphology on complete lattices that encompasses the pioneering work of Matheron and Serra as well as its more recent extensions. This section serves to present only a very brief overview of mathematical morphology; as much as is needed to support the remainder of this thesis. In what follows, (\mathcal{L}, \leq) is assumed to be a complete lattice with universal element U and null element O.

2.6.1 Adjunctions

"From a theoretical point of view, the adjunction is the most important notion in mathematical morphology" (Heijmans, 1994a, p. 12). It is a notion that is as fundamental to mathematical morphology as the concept of linearity is to traditional signal processing.

Definition 2.6.1 (adjunction). Let $\delta, \varepsilon \in \mathcal{O}(\mathcal{L})$. The pair (ε, δ) is called an *adjunction* on (\mathcal{L}, \leq) if

$$\delta\left(Y\right) \le X \Longleftrightarrow Y \le \varepsilon\left(X\right),$$

for all $X, Y \in \mathcal{L}$.

The following theorem establishes that δ is a dilation and that ε is an erosion.

Theorem 2.6.2 (Heijmans (1994*a***, p. 51)).** If (ε, δ) is an adjunction on (\mathcal{L}, \leq) then ε is an erosion and δ is a dilation.

From the definition of an adjunction it is easy to prove the following.

Proposition 2.6.3. If (ε, δ) is an adjunction on (\mathcal{L}, \leq) then³

- 1. the composition $\delta \varepsilon$ is an opening;
- 2. the composition $\varepsilon \delta$ is a closing;
- 3. the composition $\varepsilon \delta \varepsilon = \varepsilon$;
- 4. the composition $\delta \varepsilon \delta = \delta$.

³ In this thesis we follow the convention that the composition product $\delta \varepsilon$ means the application of ε followed by the application of δ .

Remark. "The study of openings and closings on complete lattices was initiated by Matheron and Serra in the early eighties" (Heijmans, 1994*a*, p. 215). In the monograph (Serra, 1988*d*) Matheron and Serra refer to adjunctional openings and closings as morphological openings and closings.

2.6.2 Representation theorem for increasing mappings

Dilations, erosions, openings, and closings are increasing mappings. The following theorem establishes a general representation for all increasing mappings.

Theorem 2.6.4 (representation of increasing mappings (Serra, 1988d, p. 20)). Every increasing operator $\psi \in \mathcal{O}(\mathcal{L})$ for which $\psi(O) = O$ can be written as a supremum of erosions in $\mathcal{O}(\mathcal{L})$.

Remark. Applying the principle of duality with respect to order it follows that every increasing operator $\psi \in \mathcal{O}(\mathcal{L})$ for which $\psi(U) = U$ can be written as an infimum of dilations in $\mathcal{O}(\mathcal{L})$.

2.6.3 Morphological filters

The study of morphological filters on lattices was pioneered by Matheron and Serra (Serra, 1988*d*). Morphological filters are defined to be increasing and idempotent operators on a complete lattice. Openings and closings are simple examples of morphological filters. Indeed "pedagogically speaking, openings and closings represent the basic material that is constantly used to generate all other [morphological] filters" (Serra, 1988*d*, p. 105).

Definition 2.6.5 (morphological filter (Serra, 1988*d*, p. 104)). An operator $\psi \in \mathcal{O}(\mathcal{L})$ that is both increasing and idempotent is called a *morphological filter*.

Remark. Prior to the generalisation of mathematical morphology to the complete lattice algebraic framework, the expression *morphological filter* was used by some authors to denote an image operator that is increasing and translation invariant; for example Giardina & Dougherty (1988). Translation invariance is a geometric property rather than an algebraic property. Translation invariant operators and morphological filters are discussed in Sections 2.7–2.10 in the context of binary images, grey-scale images, and graphs.

Unfortunately the class of morphological filters is not closed under composition, supremum, and infimum; i.e. if ψ and η are morphological filters then in general

 $\psi \circ \eta$, $\eta \circ \psi$, $\psi \lor \eta$, $\psi \land \eta$ are not morphological filters (Serra, 1988*d*, p. 105). This led Matheron to establish the following result.

Proposition 2.6.6 (composition of morphological filters (Serra, 1988*d*, p. 118)). If $\psi, \eta \in \mathcal{O}(\mathcal{L})$ are two morphological filters such that $\psi \leq \eta$ then

- 1. $\psi \leq \psi \eta \psi \leq \eta \psi \land \psi \eta \leq \psi \eta \lor \eta \psi \leq \eta \psi \eta \leq \eta;$
- 2. $\psi\eta$, $\eta\psi$, $\eta\psi\eta$, and $\psi\eta\psi$ are morphological filters.

This in turn led Matheron to introduce other classes of operators, such as underfilters and overfilters, as part of the necessary theoretical framework in which to comprehensively study morphological filters (Heijmans, 1994a, p. 410).

2.6.3.1 Overfilters, underfilters, inf-overfilters, sup-underfilters, inf-filters, sup-filters, strong filters

Definition 2.6.7 (Heijmans (1994*a***, p. 410)).** If (\mathcal{L}, \leq) is a complete lattice then an increasing operator $\psi \in \mathcal{O}(\mathcal{L})$ is called

- 1. an overfilter if $\psi \leq \psi \psi$; an underfilter if $\psi \psi \leq \psi$;
- 2. an inf-overfilter if $\psi(I \land \psi) = \psi$; a sup-underfilter if $\psi(I \lor \psi) = \psi$;
- 3. an *inf-filter* if it is a morphological filter that satisfies $\psi(I \land \psi) = \psi$; a *sup-filter* if it is a morphological filter that satisfies $\psi(I \lor \psi) = \psi$;
- 4. a strong filter if it is both a sup-filter and an inf-filter.

From this definition one can establish the following properties.

Proposition 2.6.8 (Heijmans (1994*a*, p. 410)).

- 1. Every extensive operator is an inf-overfilter. Every anti-extensive operator is a sup-underfilter.
- 2. If ψ is an overfilter, then ψ^n is an overfilter for every $n \ge 1$. If ψ is an underfilter, then ψ^n is an underfilter for every $n \ge 1$.

- 3. If ψ is an inf-overfilter, then ψ^n is an inf-overfilter for every $n \ge 1$. If ψ is a sup-underfilter, then ψ^n is a sup-underfilter for every $n \ge 1$.
- 4. If ψ is an inf-overfilter, then $I \lor \psi$ is an opening. If ψ is a sup-underfilter, then $I \lor \psi$ is a closing.
- 5. Every opening and every closing is a strong filter.

2.7 Mathematical morphology for binary images

From the point of view of mathematical morphology, a binary image is a set (of points or pixels) contained within some universal set \mathcal{S} . In other words the space of all binary images is the power set $\mathcal{P}(\mathcal{S})$; i.e. the set of all subsets of \mathcal{S} . Importantly, $(\mathcal{P}(\mathcal{S}), \subseteq)$ is a complete Boolean lattice. From the point of view of image analysis it is often desirable to imbue \mathcal{S} with additional structure: "it may be a group, a vector space, a metric space, a topological space, or a graph, just to mention a few structures relevant in the context of mathematical morphology" (Heijmans, 1994*a*, p. 72). Indeed several of these structures make an appearance in subsequent chapters of this thesis. In this section \mathcal{S} is taken to be either the Euclidean space \mathbb{R}^n or the discrete space \mathbb{Z}^n . The former is a vector space whilst the latter is a module⁴. Hereinafter the notation \mathbb{E}^n is used to denote either space. The remainder of this section presents a very brief overview of mathematical morphology for binary images.

In image analysis particular interest is paid to operators that are translation invariant. Translation invariance is defined as follows.

Definition 2.7.1 (translation invariance). An operator $\psi : \mathcal{P}(\mathbb{E}^n) \to \mathcal{P}(\mathbb{E}^n)$ is said to be *translation invariant* if

$$\psi\left(A_{h}\right) = \left[\psi\left(A\right)\right]_{h},$$

where $A \in \mathcal{P}(\mathbb{E}^n)$, $h \in \mathbb{E}^n$, and $A_h = \{a + h \mid a \in A\}$ is the translate of A along the vector h.

The only translation invariant dilations on $(\mathcal{P}(\mathbb{E}^n), \subseteq)$ are *Minkowski additions* and the only translation invariant erosions are *Minkowski subtractions*.

⁴ A module is conceptually very similar to a vector space. Indeed, most of the rules of vector manipulation hold.

Definition 2.7.2 (Minkowski addition and subtraction). Let $A, B \in \mathcal{P}(\mathbb{E}^n)$. The Minkowski addition \oplus and the Minkowski subtraction \oplus are defined as follows:

$$A \oplus B = \bigcup_{b \in B} A_b,$$
$$A \oplus B = \bigcap_{b \in B} A_{-b}.$$

Remarks.

1. Minkowski addition and Minkowski subtraction satisfy the following duality with respect to complementation:

$$(A \oplus B)^{c} = A^{c} \ominus \breve{B}$$
 and $(A \ominus B)^{c} = A^{c} \ominus \breve{B}$

where $\breve{B} = \{-b \mid b \in B\}.$

- Minkowski addition appeared for the first time in 1903 in the work of Minkowski (1903). Surprisingly, Minkowski did not define its dual: Minkowski subtraction. It is only much later in the work of Hadwiger (1950) that Minkowski subtraction is defined.
- 3. The definition of Minkowski subtraction given here is consistent with the definition originally given by Hadwiger (1950). However, Serra (1982) defines Minkowski subtraction as $A \ominus B = \bigcap_{b \in B} A_b$. Although this definition coincides with that used by Matheron (1975), it is not Hadwiger's original definition. This has led to much confusion in the literature (Heijmans, 1994*a*, p. 84).

Definition 2.7.3 (dilation and erosion). Let $A, B \in \mathcal{P}(\mathbb{E}^n)$. The operators $\delta_B, \varepsilon_B \in \mathcal{O}(\mathcal{P}(\mathbb{E}^n))$ defined

$$\delta_B(A) = A \oplus B,$$

$$\varepsilon_B(A) = A \ominus B,$$

are, respectively, a dilation and an erosion. The set B is called a structuring element.

The following result characterises all translation invariant adjunctions on $\mathcal{P}(\mathbb{E}^n)$.

Proposition 2.7.4 (Heijmans (1994a, p. 133)). For every $B \in \mathcal{P}(\mathbb{E}^n)$ the pair $(\varepsilon_B, \delta_B)$ is a translation invariant adjunction. Conversely, every translation invariant adjunction is of this form.

2.7.1 Representation theorem for increasing and translation invariant mappings

The following theorem is a stronger version of Theorem 2.6.4.

Theorem 2.7.5 (representation of increasing and translation-invariant operators (Heijmans, 1994*a*, p. 86)). If $\psi \in \mathcal{O}(\mathcal{P}(\mathbb{E}^n))$ is an increasing and translation invariant operator then

$$\psi(A) = \bigcup_{B \in \ker(\psi)} A \ominus B = \bigcap_{B \in \ker(\psi^*)} A \oplus \breve{B}$$

where ker $(\psi) = \{A \subseteq \mathbb{E}^n \mid 0 \in \psi(A)\}$ is called the kernel of the operator ψ .

Remark. This theorem was originally established by Matheron (1975) for increasing mappings on $\mathcal{P}(\mathbb{R}^n)$. Consequently this theorem is often called Matheron's representation theorem for increasing mappings.

2.7.2 Three dualities for binary dilations and erosions

There exist three types of duality between binary dilation and erosion:

1. adjunction:

$$\delta_B(X) \le Y \Longleftrightarrow X \le \varepsilon_B(Y);$$

2. duality with respect to order:

A dilation (respectively erosion) on the complete lattice $(\mathcal{P}(\mathbb{E}^n), \subseteq)$ is an erosion (respectively dilation) on the complete lattice $(\mathcal{P}(\mathbb{E}^n), \supseteq)$ and vice versa;

3. duality with respect to complementation:

$$(\delta_B(A))^c = \varepsilon_{\breve{B}}(A^c) \quad and \quad (\varepsilon_B(A))^c = \delta_{\breve{B}}(A^c).$$

2.8 Mathematical morphology for grey-scale images

In classical (linear) image processing grey-scale images are modelled as functions of the form $f : \mathbb{E}^n \to \mathcal{T}$. Adopting the notation of Heijmans (1994*a*), let Fun ($\mathbb{E}^n, \mathcal{T}$) denote the set of all such functions. The domain space represents image points or pixel coordinates and the codomain \mathcal{T} is either \mathbb{Z} or \mathbb{R} representing grey-levels. The sum f + g and the scalar multiple λf are defined:

$$(f+g)(x) = f(x) + g(x)$$
$$(\lambda f)(x) = \lambda f(x),$$

where $f, g \in \text{Fun}(\mathbb{E}^n, \mathcal{T})$ and $\lambda \in \mathcal{T}$. The set $\text{Fun}(\mathbb{R}^n, \mathbb{R})$ equipped with these operations is a vector space V^5 . Of particular interest in signal processing is a class of linear operators⁶ on V called *filters*. In this context the term *filter* is "commonly defined as any operator that is *linear*, continuous and invariant under translation" (Serra, 1988d, p. 102). Every linear filter can be expressed as the "convolution product $f * \varphi$ of a signal f by a (generalized) function φ " (Serra, 1986, p. 288).

From the point of view of mathematical morphology grey-scale images are likewise modelled as elements of the set Fun $(\mathbb{E}^n, \mathcal{T})$. However, the codomain \mathcal{T} is typically taken to be one of the infinite sets $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ or $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$, or the finite set $\{0, 1, \ldots, m\}$, each of which is a complete lattice for the usual partial order relation \leq . This ensures that $(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}), \leq)$ is a power lattice. When \mathcal{T} is the finite set $\{0, 1, \ldots, m\}$ then this lattice possesses the unique negation $f^*(x) = m - f(x)$, and when \mathcal{T} is one of the infinite sets $\overline{\mathbb{R}}$ or $\overline{\mathbb{Z}}$ the lattice possesses many negations (usually * is defined to be $f^*(x) = -f(x)$). The lattice is not, however, a complete Boolean lattice because, although it is distributive and possesses one or more negations, it does not possess a complement operator⁷. A comparison between linear and morphological image processing is shown in Table 2.2.

In classical signal processing translation invariance refers to spatial translation invariance. Indeed convolution is spatially translation invariant (Serra, 1988*d*, p. 184). Spatial translation invariance is defined as follows.

⁵ In contrast the set Fun $(\mathbb{Z}^n, \mathbb{Z})$ is not a vector space. It is a *module*. A module is conceptually very similar to a vector space. Indeed, most of the rules of vector manipulation hold. In a module, the coefficients are taken from a *ring* rather than a *field*. A ring is a more general algebraic object than a field. The set \mathbb{R} is both a ring and a field, whilst \mathbb{Z} is only a ring.

⁶ An operator $\psi: V \to V$ is said to be linear if (i) $\psi(f+g) = \psi(f) + \psi(g)$ for all signals $f, g \in V$ and (ii) $\psi(\lambda f) = \lambda \psi(f)$ for all $f \in V$ and all scalars λ .

⁷ Consider, for example, the case when \mathcal{T} is the set \mathbb{R} and $f^*(x) = -f(x)$. The lattice then has universal element $u(x) = +\infty$ and null element $o(x) = -\infty$ and in general $f(x) \lor f^*(x) \neq u(x)$ and $f(x) \land f^*(x) \neq o(x)$.

Table 2.2:	Comparison	between	linear	and	morpho	logical	image	processing.
	-				-	<u> </u>		· ·

	Linear image processing	Morphological image processing			
Image model	Vector space: set of vectors V , set of scalars K	Complete lattice: set of functions Fun $(\mathbb{E}^n, \mathcal{T})$, partial order relation \leq			
$Fundamental\ laws$	Addition, scalar multiplication	Supremum \bigvee , infimum \bigwedge			
Useful operations	Those that preserve the structure of the vector space and commute with the laws, i.e. convolutions $\phi(\sum_{i} a_{i}f_{i}) = \sum_{i} a_{i}\phi(f_{i})$	Those that preserve the structure of the complete lattice and commute with the laws, i.e. dilations $\phi(\bigvee_i f_i) = \bigvee_i \phi(f_i)$ and erosions $\phi(\bigwedge_i f_i) = \bigwedge_i \phi(f_i)$			

Definition 2.8.1 (spatial translation invariance). An operator ψ : Fun $(\mathbb{E}^n, \mathcal{T})$ \rightarrow Fun $(\mathbb{E}^n, \mathcal{T})$ is said to be *spatially translation invariant* if

$$\psi\left(f_{h}\right) = \left(\psi\left(f\right)\right)_{h}$$

for all $h \in \mathbb{E}^n$, where $f \in \text{Fun}(\mathbb{E}^n, \mathcal{T})$ and $f_h(x) = f(x-h)$ is the translate of f along the vector h.

Adopting the terminology of Heijmans (1994a), these operators are hereinafter called *H-operators* (i.e. horizontally translation invariant operators). In mathematical morphology, however, it is possible to define operators that are grey-scale translation invariant in addition to being spatially translation invariant. Grey-scale translation invariance is defined as follows.

Definition 2.8.2 (grey-scale translation invariance). An operator ψ : Fun(\mathbb{E}^n , \mathcal{T}) \rightarrow Fun(\mathbb{E}^n , \mathcal{T}) is said to be *grey-scale translation invariant* if

$$\psi(f+v)(x) = \psi(f(x)) + v$$

for all $v \in \mathcal{T}$, where $f \in \operatorname{Fun}(\mathbb{E}^n, \mathcal{T})$.

Once again, adopting the terminology of Heijmans (1994a), operators that are both spatially translation invariant and grey-scale translation invariant are hereinafter called *T*-operators (i.e. translation-invariant operators).

In image analysis it is common practice to change the relative scale of the grey values; e.g. to improve contrast or to suppress certain ranges of grey-values (Heijmans, 1994*a*, p. 104). Serra (1982) coined these transformations *anamorphoses*.

Definition 2.8.3 (anamorphosis (Heijmans, 1994*a*, p. 104)). An anamorphosis is a function $\nu : \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ which is continuous and increasing.

Remark. Examples of commonly applied anamorphoses are (Serra, 1982, p. 435): $f(x) \to af(x) + b$ for a, b > 0; $f(x) \to \log(f(x))$; $f(x) \to (f(x))^2$; and $f(x) \to \sqrt{f(x)}$.

Sternberg (1979) is responsible for extending Minkowski addition and subtraction to functions.

Definition 2.8.4 (Minkowski addition and subtraction). Let $f, g \in Fun(\mathbb{E}^n, \mathcal{T})$. The Minkowski addition \oplus and the Minkowski subtraction \ominus are defined as follows:

$$(f \oplus g)(x) = \bigvee_{h \in \mathbb{E}^n} \left\{ f(x-h) + g(h) \right\},$$
$$(f \oplus g)(x) = \bigwedge_{h \in \mathbb{E}^n} \left\{ f(x+h) - g(h) \right\}.$$

Remark. These operators are ill-defined for the finite set $\mathcal{T} = \{0, 1, \dots, m\}$ because this set is not closed under addition and subtraction; i.e. the subtraction or addition of two elements can generate a value that is outside of the set.

Analogous to the binary case, Minkowski addition is a dilation and Minkowski subtraction is an erosion.

Definition 2.8.5 (dilation and erosion). Let $f, g \in \text{Fun}(\mathbb{E}^n, \mathcal{T})$. The operators $\delta_g, \varepsilon_g \in \mathcal{O}(\text{Fun}(\mathbb{E}^n, \mathcal{T}))$ defined

$$\begin{aligned} \delta_g\left(f\right) &= f \oplus g, \\ \varepsilon_g\left(f\right) &= f \ominus g, \end{aligned}$$

are, respectively, a dilation and an erosion. The function g is called a *structuring* function.

Remark. If one adopts a truncation policy, i.e. truncating values below 0 and above m, then these operators can be applied to the case where $f \in \text{Fun}(\mathbb{E}^n, \{0, 1, \ldots m\})$ and $g \in \text{Fun}(\mathbb{E}^n, \mathbb{Z})$. Moreover δ_g is then a dilation and ε_g is an erosion on $\text{Fun}(\mathbb{E}^n, \{0, 1, \ldots m\})$ (Heijmans, 1994*a*, p. 385). However, the pair $(\varepsilon_g, \delta_g)$ is not in general an adjunction (see also the remarks following Proposition 2.8.6).

The following result characterises all H-adjunctions on $(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}), \leq)$.

Proposition 2.8.6 (Heijmans (1994a, p. 373)). If (e_h, d_h) is an adjunction on \mathcal{T} for every $h \in \mathbb{E}^n$ then the pair of operators (E, Δ) defined

$$\Delta(f)(x) = \bigvee_{h \in \mathbb{E}^n} d_h(f(x-h)),$$

$$E(f)(x) = \bigwedge_{h \in \mathbb{E}^n} e_h(f(x+h)),$$

is an H-adjunction on Fun $(\mathbb{E}^n, \mathcal{T})$. Moreover, every H-adjunction on Fun $(\mathbb{E}^n, \mathcal{T})$ is of this form.

Remarks.

1. When $\mathcal{T} = \overline{\mathbb{R}}$ or $\mathcal{T} = \overline{\mathbb{Z}}$ then the pair (e_h, d_h) defined

$$d_h(t) = t + g(h),$$

$$e_h(t) = t - g(h),$$

where $g \in \text{Fun}(\mathbb{E}^n, \mathcal{T})$, is an adjunction on \mathcal{T} . The operator Δ is then identically the dilation δ_q and the operator E is identically the erosion ε_q .

2. When $\mathcal{T} = \{0, 1, \dots, m\}$ then $d_h(t) = t + g(h)$ and $e_h(t) = t - g(h)$ are ill-defined. Moreover, if one adopts a truncation policy then, surprisingly, the pair (e_h, d_h) is not in general an adjunction (Heijmans, 1994*a*, p. 385). The situation is remedied if one redefines the addition and subtraction operators such that whenever $t \in \mathcal{T}$ takes on the value 0 this value cannot be changed by addition, and whenever t takes on the value m this value cannot be changed by a subtraction (see Heijmans (1994*a*, p. 386) for details).

When \mathcal{T} is either \mathbb{R} or \mathbb{Z} then Minkowski addition and subtraction are T-operators. The following result characterises all T-adjunctions on Fun $(\mathbb{E}^n, \mathcal{T})$.

Proposition 2.8.7 (Heijmans (1994a, p. 151)). For every $g \in \text{Fun}(\mathbb{E}^n, \mathcal{T})$, where \mathcal{T} is either \mathbb{R} or \mathbb{Z} , the pair $(\varepsilon_g, \delta_g)$ is a T-adjunction. Conversely, every T-adjunction is of this form.

When the structuring function only takes on the value 0 on its domain then it suffices to represent the function by a set (its domain). The expressions for δ_g and ε_g then simplify to the following.

Definition 2.8.8 (flat dilation and erosion). Let $f \in \operatorname{Fun}(\mathbb{E}^n, \mathcal{T})$ and $B \in \mathcal{P}(\mathbb{E}^n)$. The operators $\delta_B, \varepsilon_B \in \mathcal{O}(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}))$ defined

$$\delta_B(f) = f \oplus B$$

= $\bigvee_{h \in B} \{f(x-h)\},\$

$$\varepsilon_{B}(f) = f \ominus B$$
$$= \bigwedge_{h \in B} \{f(x+h)\},\$$

are, respectively, a (flat) dilation and a (flat) erosion. The set B is called a structuring element or flat structuring function.

2.8.1 Representation theorems for increasing and translation invariant mappings

The following two theorems are stronger versions of Theorem 2.6.4. The first pertains to the representation of increasing T-operators and the second to the representation of increasing H-operators.

Theorem 2.8.9 (representation of increasing T-operators

(Heijmans, 1994*a*, p. 109)). If $\psi \in \mathcal{O}(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}))$, where \mathcal{T} is either \mathbb{R} or \mathbb{Z} , is an increasing *T*-operator then

$$\psi\left(f\right) = \bigvee_{g \in \ker(\psi)} f \ominus g = \bigwedge_{g \in \ker(\psi^*)} f \oplus \breve{g},$$

where $\check{g}(x) = g(-x)$, and ker $(\psi) = \{g \in \operatorname{Fun}(\mathbb{E}^n, \mathcal{T}) \mid \psi(g)(0) \ge 0\}$ is called the kernel of the operator ψ .

Remark. This theorem is an extension of Matheron's representation theorem (Theorem 2.7.5) to grey-scale functions. Matheron's original theorem was extended to grey-scale functions independently by Maragos (1989) (with the requirement that fis upper semi-continuous), and by Giardina & Dougherty (1988). In both cases the proofs rely upon the *umbra transform* (Heijmans, 1994*a*, p. 376) and Matheron's original theorem. Crombez (1990) established an independent proof of this theorem without using the umbra transform or Matheron's representation theorem.

Theorem 2.8.10 (representation of increasing H-operators

(Heijmans, 1994*a*, p. 375)). Every increasing H-operator $\psi \in \mathcal{O}(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}))$ that satisfies $\psi(u) = u$, where $u(x) = \bigvee \mathcal{T}$, can be represented as a supremum of H-erosions.

Remark. Applying the principle of duality with respect to order it follows that every increasing H-operator $\psi \in \mathcal{O}(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}))$ that satisfies $\psi(o) = o$, where $o(x) = \bigwedge \mathcal{T}$, can be represented as an infimum of H-dilations.

2.8.2 Three dualities for grey-scale dilations and erosions

There exist three types of duality between grey-scale dilation and erosion (Heijmans, 1994a, p. 108):

1. adjunction:

 $\delta_g(h) \leq f \iff h \leq \varepsilon_g(f);$

2. duality with respect to order:

A dilation (respectively erosion) on the complete lattice (Fun $(\mathbb{E}^n, \mathcal{T}), \leq$) is an erosion (respectively dilation) on the complete lattice (Fun $(\mathbb{E}^n, \mathcal{T}), \geq$) and vice versa;

3. duality with respect to negation:

$$(\delta_g(f))^* = \varepsilon_{\breve{g}}(f^*)$$
 and $(\varepsilon_g(f))^* = \delta_{\breve{g}}(f^*).$

2.9 Mathematical morphology for metric spaces

The concept of a *metric space* is relevant to the discussion on graph morphology in the section to follow and also to the exposition in Chapter 5. A *metric* is essentially a generalisation of the concept of *distance*. In geometry distance is an essential notion (Preparata & Shamos, 1985, p. 4). In particular, the familiar Euclidean distance

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

is used to quantify the *nearness* of $x = (x_1, x_2, ..., x_n)$ to $y = (y_1, y_2, ..., y_n)$ in \mathbb{R}^n . This in turn permits the definition of continuity and convergence in \mathbb{R}^n (DePree & Swartz, 1988, 241). In real analysis the concept of a metric is used to extend the notions of continuity and convergence to the more abstract metric space.

Definition 2.9.1 (metric space). Let *E* be a non-empty set. The function *d* : $E \times E \to \mathbb{R}$ is called a *metric* if it satisfies the following properties for all $x, y, z \in E$:

- 1. $d(x,y) \ge 0$ and d(x,y) = 0 if and only if x = y (positive definite);
- 2. d(x,y) = d(y,x) (symmetry);
- 3. $d(x,y) \leq d(x,z) + d(z,y)$ (triangle inequality).

The pair (E, d) is called a *metric space*.

Examples. The following are metric spaces:

1. (\mathbb{R}, d) , where d = |x - y|.

2. $(C([a, b], \mathbb{R}), d)$, where $C([a, b], \mathbb{R})$ is the set of all real-valued continuous functions on [a, b] and

$$d(f,g) = \max_{t \in [a,b]} \left| f(t) - g(t) \right|.$$

3. (\mathbb{R}^n, d_p) , where d_p is the *Minkowski* metric of power p (Okabe et al., 1992, p. 185). The Minkowski metric, also called the L_p metric, is in fact a parameterised family of metrics defined as follows for $p = 1, 2..., \infty$:

$$d_p(x,y) = \begin{cases} \left(\sum_{i=1}^n |x_i - y_i|^p\right)^{1/p}, & \text{if } 1 \le p < \infty, \\ \max\{|x_i - y_i| \mid 1 \le i \le n\}, & \text{if } p = \infty. \end{cases}$$

When p = 1 the Minkowski metric is called the *Manhattan* or *city-block* or *taxi-cab* metric. When p = 2 it is the familiar Euclidean distance. When $p = \infty$ the Minkowski metric is called the *supremum* or *dominance* metric.

Definition 2.9.2 (open and closed balls). Let (E, d) be a metric space, $x \in E$, and $r \in \mathbb{R}^+$. The sets

$$B^{\circ}(x,r) = \{y \in E \mid d(x,y) < r\}$$

and

$$B(x,r) = \{y \in E \mid d(x,y) \le r\}$$

are called, respectively, the open ball and the closed ball of radius r and centre x.

Example. Consider the metric d_2 (i.e. the Euclidean metric) defined on the space \mathbb{R}^n . When n = 1, the open ball is an open interval, when n = 2 it is an open disk, and when n = 3 it is a sphere without its boundary. Other metrics defined on \mathbb{R}^n , however, lead to rather different balls as Figure 2.2 shows.

Definition 2.9.3 (metric dilation (Heijmans, 1994*a*, **p. 293)).** Let (E, d) be a metric space. The operator $\delta^r : \mathcal{P}(E) \to \mathcal{P}(E)$, for $r \ge 0$, defined

$$\delta^{r}\left(A\right) = \bigcup_{x \in A} B\left(x, r\right)$$

is a dilation called the *metric dilation* of size r.



Figure 2.2: Closed balls in \mathbb{R}^2 with respect to: (a) the Minkowski metric L_1 (the cityblock metric), (b) the Minkowski metric L_2 (Euclidean distance), and (c) the Minkowski metric L_{∞} (the dominance metric).

Remarks.

1. When E is \mathbb{Z}^n or \mathbb{R}^n then the metric dilation of size r > 0 can be written as

$$\delta^r = A \oplus \underbrace{B \oplus \dots \oplus B}_{r \text{ terms}}.$$

where B = B(0, 1); i.e. the closed unit ball centred at the origin.

2. The adjoint erosion of size r is defined

$$\varepsilon^{r}(A) = \{x \mid B(x, r) \subseteq A\}.$$

2.9.1 Discrete metric spaces

The notion of a discrete metric space arises in the context of digital images. A discrete metric space is defined as follows.

Definition 2.9.4 (discrete metric space (Heijmans, 1994*a***, p. 326)).** A metric space (E, d) is called *discrete* if for every $x \in E$ and for r > 0 sufficiently small, the closed ball B(x, r) contains only finitely many points.

The domain space of a digital image is \mathbb{Z}^n (the space of pixel coordinates). Unfortunately \mathbb{Z}^n is not isomorphic to \mathbb{R}^n . This means that there are several ways of interpreting (visualising) a digital image in \mathbb{R}^n . In the case of a two-dimensional digital image, the space \mathbb{Z}^2 is usually interpreted as either a square or a regular hexagonal lattice⁸ of points in the Euclidean plane \mathbb{R}^2 . In the former case the points,

⁸ The term lattice here has nothing to do with lattice theory.



Figure 2.3: Interpretations of \mathbb{Z}^2 in \mathbb{R}^2 : (a) points on a hexagonal grid; (b) points on a 4-connected square grid; and (c) points on an 8-connected square grid.

corresponding to the centres of pixels, are deemed to be the integer points in \mathbb{R}^2 . In the latter case, assuming that the vertical spacing is 1, the points are deemed to be the points au + bv where $u = \left(\frac{2}{\sqrt{3}}, 0\right)$, $v = \left(\frac{1}{\sqrt{3}}, 1\right)$ where $a, b \in \mathbb{Z}$. The choice of interpretation is usually "made at the moment of discretization" (Heijmans, 1994*a*, p. 325).

In the case of the hexagonal lattice, the six points surrounding a given point are equidistant from that point and are deemed to be connected to it. This connectivity can be explicitly shown by drawing an edge between each point and its six neighbours. This yields the hexagonal grid shown in Figure 2.3(a). In the case of the square lattice not all eight neighbours of a given point are equidistant from it and this gives rise to two possibilities: the 4-connected square grid and the 8-connected square grid as shown in Figure 2.3(b)-(c).

Given any two grid points x and y it is possible to define at least one *path* between them consisting of a sequence of grid points such that every pair of successive points is connected by an edge. The number of edges comprising the shortest path between them is denoted d(x, y). It is easy to show that d is a metric. It follows that the pair (\mathbb{Z}^2 , d) is therefore a (discrete) metric space. Figure 2.4 shows closed balls in \mathbb{Z}^2 for the discrete metric induced by the hexagonal grid, the 4-connected square grid, and the 8-connected square grid.

A set of pixels $X \subseteq \mathbb{Z}^2$ on a grid is said to be *connected* if for any two pixels $x, y \in X$ there exists at least one path connecting them such that all of the pixels on the path are elements of X. The notion of a connected set leads to the definition of a *connected component* or *flat zone*.



Figure 2.4: Closed balls in \mathbb{Z}^2 : (a) pixels arranged on the hexagonal grid (hexagonal metric); (b) pixels arranged on the 4-connected square grid (discrete city-block metric); and (c) pixels arranged on the 8-connected square grid (discrete chessboard metric).

Definition 2.9.5 (connected component). A connected component is a connected set whose pixels have the same value. For grey-scale images, connected components are also called *flat zones*.

2.10 Mathematical morphology for graphs

In image analysis graphs are often used to model the topological connectedness of and the geometric structure in the image under study (see Chapter 5). Formally, a graph is defined as follows (for a more detailed exposition on graph theory the reader is directed to the classic treatise *The Theory of Graphs and its Applications* by Berge (1962)).

Definition 2.10.1 (graph (Beineke & Wilson, 1997)). A graph G consists of a finite non-empty set V(G) of elements called *vertices* or *nodes*, and a finite set E(G) of distinct unordered pairs of distinct elements of V(G). The elements of E(G) are called *edges*. The set V(G) is called the *vertex set* and the set E(G) is called the *edge set*. It is convenient to abbreviate these two sets to V and E and to write that G = (V, E).

Remarks.

- 1. The elements of E are unordered pairs so that (v, w) is the same element as (w, v).
- 2. There are numerous variations on the basic definition of a graph. In particular:

- (a) on replacing *unordered* with *ordered* in the definition of a graph, one obtains the definition of a *directed graph* (this is a graph in which each edge has a direction; e.g. (v, w) is an arrowed edge from v to w);
- (b) by removing the restriction that the elements of E(G) must be distinct, one obtains the definition of a *multigraph* (this is a graph in which two or more edges may join pairs of vertices);
- (c) by removing the restriction that the pairs in E(G) consist of distinct elements, one obtains a graph with *loops* (a multigraph with loops is called a *general graph*).

A graph consistent with the above definition is also called a $simple \ graph^9$ because it has no self-loops and no multiple or directed edges.

If e = (v, w) is an edge of a graph then e is said to *join* the vertices v and w and these vertices are said to be *adjacent*. Furthermore the vertex w is said to be a *neighbour* of the vertex v and vice versa. In general the *neighbourhood* of a vertex is defined as follows.

Definition 2.10.2 (vertex neighbourhood (Beineke & Wilson, 1997)). Let G = (V, E) be an arbitrary graph. The *neighbourhood* N(v) of a vertex v is defined

$$N(v) = \{ w \in V \mid (v, w) \in E \}.$$

The notion of neighbours leads to the definition of paths and connectedness.

Definition 2.10.3 (path). Let G = (V, E) be an arbitrary graph. Let v and w be two distinct vertices. A path between v and w is defined to be a sequence of distinct vertices such that every pair of successive vertices are neighbours (i.e. are joined by an edge).

Definition 2.10.4 (connected graph). Let G = (V, E) be an arbitrary graph. If there exists a path between every pair of distinct vertices then G is said to be connected.

Geometrical realisations of graphs in the plane are fundamental to the exposition in Chapter 5. The following definition characterises such graphs.

⁹ The edge set of a simple graph can be viewed as an irreflexive and symmetric binary relation defined on the vertex set.

Definition 2.10.5 (plane and planar graphs (Wilson, 1985, p. 59)). A graph that can be drawn in the plane in such a way that no two edges (or rather, the curves representing them) intersect geometrically except at a vertex to which they are both incident is called a *plane graph*. A *planar* graph is a graph that is isomorphic to a plane graph.

In his pioneering work on graph morphology Vincent (1989) extended many morphological transformations—metric dilation and erosion, distance transforms (see Section 5.4), watershed transform (see Appendix B)—to simple graphs by modelling the graphs as elements of the complete Boolean lattice $(\mathcal{P}(V(G)), \subseteq)$ and introducing the following discrete metric on G.

Proposition 2.10.6. Let G = (V, E) be an arbitrary connected and simple graph. Let v and w be two distinct vertices and define d(v, w) to be the length (number of edges) of the shortest path between v and w. The pair (G, d) is a discrete metric space.

Remark. The distance d can be applied to a graph that is not connected by adopting the convention that the distance between two unconnected vertices is ∞ . However, d is then technically no longer a metric.

Hereinafter the notation X|G is used to denote a (binary) graph so that it is clear that $X \in \mathcal{P}(V(G))$. Graph dilation and graph erosion are defined to be metric dilation and metric erosion, respectively, as follows.

Definition 2.10.7 (binary graph dilation and erosion). Let G = (V, E) be a connected simple graph. The operator $\delta^r : \mathcal{P}(V(G)) \to \mathcal{P}(V(G))$ defined

$$\delta^{r}\left(X\left|G\right.\right) = \bigcup_{v \in X} N_{r}\left(v\right)$$

is called graph dilation of size r and the operator $\varepsilon^{r} : \mathcal{P}(V(G)) \to \mathcal{P}(V(G))$ defined

$$\varepsilon^r \left(X \left| G \right. \right) = \left\{ v \in X \left| N_r(v) \subseteq X \right. \right\}$$

is called graph erosion of size r, where $r \in \mathbb{Z}^+$, $N_r(v)$ is the closed ball $B(v, r) = \{w \in V(G) \mid d(v, w) \leq r\}$, and $X \in \mathcal{P}(V(G))$.

Remarks.

1.
$$N_1(v) = N(v) \cup v = B(v, 1).$$



Figure 2.5: Elementary graph dilation and erosion: (a) binary graph X | G (filled vertices); (b) $\delta(X | G)$; and (c) $\varepsilon(X | G)$.

- 2. In contrast to mathematical morphology for digital images, the number of vertices (points) contained in the neighbourhood (ball) $N_1(v)$ depends on the vertex v.
- 3. When r = 1 the dilation and erosion are denoted δ and ε respectively. Moreover they are called elementary dilation and erosion (see Figure 2.5).
- 4. $\delta^r = \underbrace{\delta \circ \delta \cdots \circ \delta}_{r \text{ times}}$ and $\varepsilon^r = \underbrace{\varepsilon \circ \varepsilon \cdots \circ \varepsilon}_{r \text{ times}}$.

Vincent (1989) extended these definitions to grey-scale graphs. A grey-scale graph is a simple connected graph whose vertices are assigned grey-values. Hence a grey-scale graph can be modelled as an element of the power lattice (Fun $(V(G), \mathcal{T}), \leq)$, where \mathcal{T} is typically one of the infinite sets $\mathbb{R} = \mathbb{R} \cup \{-\infty, +\infty\}$ or $\mathbb{Z} = \mathbb{Z} \cup \{-\infty, +\infty\}$, or the finite set $\{0, 1, \ldots, m\}$. Formally a grey-scale graph is defined as follows.

Definition 2.10.8 (grey-scale graph). Let G = (V, E) be a connected simple graph. The function $f \in Fun(V(G), \mathcal{T})$, also written f | G, is called a grey-scale graph.

Remark. When $\mathcal{T} = \{0, 1\}$ the power lattice $(\operatorname{Fun}(V(G), \mathcal{T}), \leq)$ is isomorphic to the complete Boolean lattice $(\mathcal{P}(V(G)), \subseteq)$.

The dilation and erosion operations for grey-scale graphs are as follows.

Definition 2.10.9. Let G = (V, E) be a connected simple graph. The operator $\delta^r \in \mathcal{O}(\operatorname{Fun}(V(G), \mathcal{T}))$ defined

$$\delta^{r}\left(f\left|G\right)\left(v\right) = \bigvee_{w \in N_{r}(v)} f\left(w\right)$$

is called *grey-scale graph dilation* of size r and the operator $\varepsilon^r \in \mathcal{O}(\operatorname{Fun}(V(G), \mathcal{T}))$ defined

$$\varepsilon^{r}\left(f\left|G\right.\right) = \bigwedge_{w \in N_{r}(v)} f\left(w\right)$$

is called grey-scale graph erosion of size r, where $f \in Fun(V(G), \mathcal{T})$.

2.10.1 Structuring graphs

Heijmans, Nacken, Toet & Vincent (1992) generalised Vincent's metric approach to defining graph morphology by introducing the notion of *structuring graphs* or *s-graphs*.

Definition 2.10.10 (s-graph (Heijmans et al., 1992, p. 28)). A structuring graph \mathcal{A} is a simple graph $G_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$ together with two non-empty sets $B_{\mathcal{A}}, R_{\mathcal{A}} \subseteq V_{\mathcal{A}}$ called the *buds* and the *roots* respectively.

Remark. The graph $G_{\mathcal{A}}$ need not be connected and the buds and roots may coincide.

The idea of an s-graph is analogous to that of a structuring element. The concept of a root corresponds to that of the origin (although an s-graph can have several roots). At each vertex v of the graph G under study an *embedding* of the s-graph is sought. An embedding is a mapping G_A into G such that one of the roots coincides with v and such that all of its vertices and edges coincide with vertices and edges in G. Those vertices of G that specifically coincide with the buds of the s-graph constitute part of the neighbourhood of v generated by the s-graph. The complete neighbourhood is determined by seeking all possible embeddings of the s-graph at v. Formally, the neighbourhood associated with an s-graph is defined as follows:

Definition 2.10.11 (neighbourhood function of an s-graph (Heijmans et al., 1992, p. 29)). Let G = (V, E) be a simple graph and let \mathcal{A} be an s-graph. The s-graph \mathcal{A} defines a neighbourhood function, denoted $N_{\mathcal{A}}$, on $\mathcal{P}(V(G))$ as follows:

$$N_{\mathcal{A}}(v | G) = \bigcup \{ \theta(B_{\mathcal{A}}) | \theta \text{ is an embedding of } \mathcal{A} \text{ into } G \text{ at } v \}$$



Figure 2.6: Structuring graph defining the elementary neighbourhood N_1 (arrow indicates the root, and both vertices are buds).

where $v \in V$.

Remarks.

- 1. Mathematically speaking, an embedding θ is a homomorphism of $G_{\mathcal{A}}$ into G (i.e. a mapping that preserves the vertices and edges of the graph).
- 2. The neighbourhood function N_1 associated with Vincent's elementary graph dilation and erosion is defined by the s-graph shown in Figure 2.6.

This neighbourhood function can then be used to define adjunctions on $(\mathcal{P}(V(G)), \subseteq)$ as follows.

Proposition 2.10.12 (Heijmans et al. (1992, p. 30)). Let G = (V, E) be a simple graph and let \mathcal{A} be an s-graph. The pair $(\varepsilon_{\mathcal{A}}, \delta_{\mathcal{A}})$ defined

$$\delta_{\mathcal{A}} (X | G) = \bigcup_{v \in X} N_{\mathcal{A}} (v | G),$$

$$\varepsilon_{\mathcal{A}} (X | G) = \{ v \in V | N_{\mathcal{A}} (v | G) \subseteq X \},$$

where $X \in \mathcal{P}(V(G))$, is an adjunction on the complete Boolean lattice $(\mathcal{P}(V(G)), \subseteq)$.

These definitions extend to grey-scale graphs also.

Proposition 2.10.13 (Heijmans & Vincent (1993, p. 191)). Let G = (V, E)be a simple graph and let \mathcal{A} be an s-graph. The pair $(\varepsilon_{\mathcal{A}}, \delta_{\mathcal{A}})$ defined

$$\delta_{\mathcal{A}}(f|G)(v) = \sup \left\{ f(w) \mid w \in \check{N}_{\mathcal{A}}(v|G) \right\},\$$

$$\varepsilon_{\mathcal{A}}(f|G)(v) = \inf \left\{ f(w) \mid w \in N_{\mathcal{A}}(v|G) \right\},\$$

where $\check{N}_{\mathcal{A}}(v|G) = \{w \in V(G) \mid v \in N_{\mathcal{A}}(w|G)\}$, is an adjunction on the power lattice (Fun $(V(G), \mathcal{T}), \leq$).

2.11 Summary

This chapter has established the fundamental definitions and concepts underlying the remainder of this thesis. This chapter has:

- Reviewed sets and binary relations, and ordering in sets: total, partial, and quasi-ordering.
- Defined the concept of a lattice from both a set theory point of view and as an algebra, and its properties; defined modular and distributive lattices; defined complete lattices, and in particular the complete Boolean lattice and the power lattice because of their importance in image analysis; and defined important properties and types of operators on a complete lattice.
- Reviewed the theory of mathematical morphology for complete lattices, including adjunctions, a representation theorem for all increasing mappings, and morphological filters.
- Briefly reviewed mathematical morphology for binary and grey-scale images, metric spaces, and graphs.

Chapter 3

Folding Induced Self-dual Filters

Copyrighted image removed

(c) 2004 The M.C. Escher Company - the Netherlands (www.mcescher.com). All rights reserved. Used by permission. $M.C.\ Escher's\ "Day\ and\ Night"$

This chapter deals with the topic of noise filtering, and in particular with attenuating impulse-type noise that can be induced by camera/frame-grabber electronics when acquiring digital images. Broadly speaking, *filters* are used in image and signal processing to extract the useful or essential part of a signal and to discard the extraneous components called *noise*. In the context of image analysis, noise filtering is a low-level processing task "requiring no intelligence on the part of the image analysis system" (Gonzalez & Woods, 1992, p. 572). It is a precursory step to intermediate-level process-



ing: segmentation, and image representation and description. In signal processing the term *filter* "is commonly defined as any operator that is *linear*, *continuous* and invariant under translation" (Serra, 1988*d*, p. 102). An operator ψ is said to be linear if $\psi(f+g) = \psi(f) + \psi(g)$ for all signals f and g. These transformations "preserve addition, and, beyond addition, the notion of group structure, thereby making reversibility an important feature" (Serra, 1988*d*, p. 10). An image degraded by motion blur, for example, can be modelled as a convolution (a linear operation) of the original image with a blurring function. If the blurring function can be derived analytically or can be estimated then *inverse filtering* (Gonzalez & Woods, 1992, p. 272) can be used to recover the original image from the degraded image (reversibility). When the phenomena are linear in nature, then the use of linear filters is justified; examples include the attenuation of specific frequencies in acoustic signals, the attenuation of additive noise, and correcting out-of-focus images. The most common types of linear filter are:

- *low-pass:* these filters attenuate the high frequencies, leaving low frequencies intact;
- *high-pass:* these filters attenuate the low frequencies, leaving high frequencies intact; and
- *band-pass:* these filters attenuate all but a selected band of frequencies.

For visual signals, (spatial) frequency is related to the perceived geometric detail. In a digital image, for example, high spatial frequency corresponds to a rapid change in brightness across image pixels and is synonymous with object edges and noise, whilst low spatial frequency corresponds to a gradual change in intensity. Heijmans & Ronse (1990, p. 247) state that:

workers in image processing have attempted to apply linear techniques to the analysis of images. One thought, for example, [was] that the global structure of an image would be derived from a low-pass filtering and the finer details from a high-pass filtering.

In reality, whilst some signals (such as acoustic signals) combine linearly by superposition, visual signals do not (Heijmans, 1994*a*, p. 6). The reason for this is that the world around us is composed of opaque objects. When one object is placed in front of another the light waves reflected from both objects do not sum (Serra, 1988*b*, p. 102). On the contrary, the object in front blocks the light emanating from the object behind it. This basic observation is the basis of a non-linear approach to image filtering using operators from mathematical morphology (Serra, 1982, 1988*d*). Morphological filters (Definition 2.6.5) are *increasing* (preserve order relationships between signals) rather than linear. In addition, morphological filters are *idempotent*. This means that repeated application of the filter has no additional effect; in the context of linear filtering this corresponds to the notion of an *ideal* filter. Idempotence is a desirable property because after applying an idempotent filter the practitioner is not faced with the dilemma of whether or not to apply the filter a second or indeed several times more.

Every linear filter can be expressed as the "convolution product $f * \varphi$ of a signal f by a (generalized) function φ " (Serra, 1986, p. 288). Convolution satisfies the property $f * \varphi = -(-f * \varphi)$ for all signals f. This property is known as *self-duality*. Self-dual filters are of interest because they treat both the light and dark areas of an image in an equivalent manner. They are well suited to situations where the aim is to "separate two components, one of which is sometimes lighter and sometimes darker than the other" (Serra, 1988*a*, p. 159). The elimination of *salt-and-pepper* noise is a prime example. Other examples include the filtering of images of natural scenes and textures for which there is no clear distinction between foreground and background. This is exactly the situation one encounters in relation to the filtering of digitised photomicrographs of Papanicolaou-stained cell nuclei; there is no obvious notion of foreground and background in relation to the chromatin texture in a cell nucleus.

Whilst all linear filters are self-dual, only some non-linear filters possess this property; the median filter is perhaps the best known example. Non-linear self-dual filters offer several advantages over their linear counterparts. Firstly, an increasing non-linear self-dual filter, such as the median filter, does not reduce the dynamic range and high frequencies in the image. This is in contrast to convolutions weighted moving averages—which are by their very nature smoothing operations (Serra, 1988*a*, p. 159). Secondly, it is possible to construct non-linear self-dual filters that are independent of monotone changes in intensity called *anamorphoses* (see Definition 2.8.3). Thirdly, non-linear self-dual filters can be constructed such that no new grey values are introduced in the image. Finally, it is possible to construct non-linear self-dual filters that are idempotent but do not induce the *ringing* degradation (Gonzalez & Woods, 1992, p. 205) that is characteristic of ideal linear filters.

Given that self-duality is not an intrinsic property of non-linear filters, how does one construct a self-dual non-linear filter? Remarkably, the generalisation of mathematical morphology to the complete lattice framework has been the catalyst for most of the research on this question: Serra (1988*a*, 1989); Serra & Vincent (1992); Heijmans (1994*b*, 1996); Evans et al. (1997); Ronse & Heijmans (1998); Heijmans & Ronse (1999). Evans et al. (1997) introduced a particularly novel approach motivated by the desire to attenuate salt-and-pepper noise using a single morphological closing. The approach involves the imposition of an alternative ordering, which they call *folded ordering*, on the image grey-values. Application of a morphological closing to this image and the subsequent re-imposition of the original ordering produces a self-dual filter. Evans et al. note, however, that although this approach makes the closing self-dual, folded ordering actually inhibits the filter's ability to attenuate salt-and-pepper noise. They subsequently abandon this ordering in favour of another that induces only *approximate* self-duality.

This chapter proffers a new theoretical approach to constructing non-linear selfdual filters from morphological meta-operators defined on an abstract space called *fold-space*. Fold-space represents a generalisation of the folded ordering proposed by Evans et al. (1997) (hereinafter referred to as the ESJ folded ordering). Selfdual operators that can be constructed from fold-space operators are called *folding induced self-dual filters* (FISFs). The ESJ folded closing turns out to be a particular type of FISF. Importantly, however, other types of FISF can be designed that do not suffer the limitations of the ESJ folded closing in relation to the attenuation of salt-and-pepper impulse noise.

The remainder of this chapter is organised as follows. In the next section the various lattice-theoretical approaches to constructing non-linear self-dual filters are briefly reviewed. Section 3.2 then introduces the new approach: folding induced self-dual filters. Finally Section 3.3 presents a summary of the chapter.

3.1 Review of non-linear self-dual operators and filters

The generalisation of the theory of mathematical morphology to the complete lattice algebraic framework in the 1980s, was the catalyst for much of the research into non-linear self-dual filtering. A summary of the principal literature is as follows:

• Serra (1988*a*) introduces the notion of *activity ordering* and based on this the notion of the *morphological centre*. The morphological centre is an operator that is constructed from an arbitrary family of operators. When the family

consists of a collection of operators and their duals, then the centre is selfdual. Moreover, Serra (1988*a*, p. 166) proposes a criterion which, if satisfied, guarantees that after a finite number of iterations of the morphological centre idempotence is achieved (for digital images).

- Serra (1989) introduces the concept of a *toggle mapping*. The simple thresholding operator is a trivial example of a toggle mapping. The morphological centre is a more sophisticated example. Although "toggle mappings are not necessarily self-dual...they can *always* admit a self-dual version" (Serra & Vincent, 1992, p. 95).
- Serra & Vincent (1992, p. 99) establish a theoretical link between the morphological centre, the median and weighted median filters.
- Heijmans (1994b) introduces the notion of a *switch* operator as a tool for constructing self-dual increasing operators on $\mathcal{P}(\mathbb{Z}^n)$. Moreover he proposes a means of modifying any increasing self-dual operator—using an approach based on Serra's morphological centre operator and the activity ordering—so that its iteration leads to idempotence (and hence a morphological filter).
- Heijmans (1996) refines and extends his work based on the switch operator. In particular he derives a general formula for translation invariant self-dual operators on *P*(Z²). Moreover he makes the observation that this formula extends to grey-scale images (this follows from the fact that every increasing operator on *P*(Eⁿ) can be extended to grey-scale functions on Eⁿ by thresholding (Heijmans, 1994*a*, p. 110)).
- Evans et al. (1997) introduce self-dual and approximate self-dual filters based on the notion of *alternative orderings*.
- Heijmans & Ronse (1999) introduce the concept of an annular operator on \$\mathcal{P}(E)\$ where \$E = \mathbb{Z}^n\$ or \$E = \mathbb{R}^n\$. This operator is defined in terms of two structuring elements. Heijmans & Ronse show that if the structuring elements satisfy some additional conditions then the annular operator can be made to be idempotent and/or self-dual. Moreover they show that binary annular operators extend to grey-scale images by thresholding. In a companion paper Ronse & Heijmans (1998) extend the theory of annular filters (idempotent annular operators) to lattices, and in particular complete lattices.

In the remainder of this section, each of the different approaches to constructing self-dual operators and filters are reviewed: self-dual operators based on the activity ordering, self-dual toggle mappings, self-dual operators based on the switch operator, self-dual operators based on the ESJ folded ordering, and self-dual annular operators.

3.1.1 Self-dual operators based on the activity ordering

The activity ordering is an invention of Serra (1988*a*). The activity ordering is an ordering that can be defined on the set of operators $\mathcal{O}(\mathcal{L})$ of a complete Boolean lattice (\mathcal{L}, \leq) . The ordering makes it possible to "compare the impact of two operators on an object $X \in \mathcal{L}$ " (Heijmans, 1994*a*, p. 63). It is defined as follows:

Definition 3.1.1 (activity ordering (Heijmans, 1994*a*, p. 64)). Let (\mathcal{L}, \leq) be a complete Boolean lattice and let $\psi, \eta \in \mathcal{O}(\mathcal{L})$. The operator ψ is said to be *more active* than η , written $\eta \leq \psi$, if

$$\begin{split} I \wedge \psi &\leq I \wedge \eta, \\ I \vee \eta &\leq I \vee \psi \end{split}$$

where I is the identity operator, i.e. I(X) = X for all $X \in \mathcal{L}$. The relation \preceq is called the *activity ordering*.

Remarks.

- 1. The activity ordering defines a partial ordering on $\mathcal{O}(\mathcal{L})$ (Heijmans, 1994*a*, p. 64).
- The definition above defines the activity ordering in terms of the identity operator. It is possible to replace the identity operator with any other operator to define different types of activity ordering (Heijmans, 1994*a*, p. 64), (Serra, 1988*a*, p. 168).
- 3. The symbols Υ and λ are used to denote the supremum and infimum, respectively, with respect to the activity ordering \preceq .

The definition of the activity ordering extends to the power lattice Fun (E, \mathcal{T}) , where E is an arbitrary non-empty set and \mathcal{T} is a complete chain (Heijmans, 1994*a*, p. 68).
Proposition 3.1.2. When (\mathcal{L}, \leq) is

- 1. a complete Boolean lattice then the activity ordering is a partial ordering and $(\mathcal{O}(\mathcal{L}), \preceq)$ is a complete lattice with the identity mapping as the null element and the complement operator as the universal element (Heijmans, 1994*a*, pp. 66).
- 2. the power lattice $\mathcal{L} = \operatorname{Fun}(E, \mathcal{T})$, where *E* is an arbitrary non-empty set and \mathcal{T} is a complete chain, then the activity ordering is a partial ordering (Heijmans, 1994*a*, p. 68) but $(\mathcal{O}(\mathcal{L}), \preceq)$ is only an inf semilattice (Serra, 1988*a*, p. 164).

3.1.1.1 The centre operator

The (morphological) centre operator is defined to be the infimum, with respect to the activity ordering, of an arbitrary family of operators defined on a complete Boolean lattice (\mathcal{L}, \leq) . It is a *centre* mapping in the sense that if $\psi_1, \psi_2 \in \mathcal{O}(\mathcal{L})$ are two increasing operators such that $\psi_1 \leq \psi_2$, then the centre operator with respect to ψ_1 and ψ_2 is an operator β that satisfies $\psi_1 \leq \beta \leq \psi_2$ (Heijmans, 1994*a*, p. 69).

Definition 3.1.3 (centre and anti-centre (Heijmans, 1994*a*, p. 67)). Let (\mathcal{L}, \leq) be a complete Boolean lattice and let $\{\psi_i\}$ be a family of arbitrary operators in $O(\mathcal{L})$. The infimum of this family of operators, with respect to the activity ordering, is called the centre of the operators and is given by

$$\lambda_i \psi_i = \left[I \land \left(\bigvee_i \psi_i \right) \right] \lor \left(\bigwedge_i \psi_i \right).$$

The supremum of this family of operators, with respect to the activity ordering, is called the *anti-centre* of the operators and is given by

$$\Upsilon_i \psi_i = \left[\nu \land \left(\bigvee_i \psi_i \right) \right] \lor \left(\bigwedge_i \psi_i \right),$$

where ν is the complement operator $\nu(X) = X^{c}$.

Remarks.

1. If all of the operators ψ_i are increasing then the centre is an increasing operator. However, the same is not true for the anti-centre (Heijmans, 1994*a*, p. 67). 2. The centre also extends to the power lattice $\mathcal{L} = \operatorname{Fun}(E, \mathcal{T})$. However, the anti-centre does not because in this case the lattice $(\mathcal{O}(\mathcal{L}), \preceq)$ is only an inf semilattice.

The following proposition prescribes the manner in which self-dual centre and anticentre operators can be constructed.

Proposition 3.1.4 (Heijmans (1994a, p. 67)). If $\{\psi_i\}$ is a family of operators such that for every ψ_i its negative ψ_i^* is also a member of the family, then both the centre and anti-centre are self-dual operators.

Example. Consider the lattice (Fun $(\mathbb{Z}^2, \overline{\mathbb{Z}}), \leq$) (and assume that the underlying grid is the square grid). Let *B* be a 3 × 3 square structuring element and let $\varphi_B = \varepsilon_B \delta_B$ (an adjunctional closing) and let $\gamma_B = \delta_B \varepsilon_B$ (the dual adjunctional opening). Consider the family { $\varphi_B \gamma_B \varphi_B, \gamma_B \varphi_B \gamma_B}$ }. From proposition 2.5.4 it follows that $(\varphi_B \gamma_B \varphi_B)^* = \varphi_B^* \gamma_B^* \varphi_B^* = \gamma_B \varphi_B \gamma_B$ and hence the family satisfies the above proposition. From proposition 2.6.6 it follows that $\gamma_B \varphi_B \gamma_B \leq \varphi_B \gamma_B \varphi_B$ because $\gamma_B \leq \varphi_B$. Hence the centre $\lambda \{\varphi_B \gamma_B \varphi_B, \gamma_B \varphi_B \gamma_B\} = (I \land \varphi_B \gamma_B \varphi_B) \lor \gamma_B \varphi_B \gamma_B$ is an increasing self-dual operator.

3.1.1.2 Finite iterations of increasing centres and the middle filter

The centre operator, like the median filter, is not idempotent. What this means is that after its application the practitioner is faced with the dilemma of whether or not to apply it one or several times more. Worse still, repeated application to a digital image may lead to oscillatory or periodic behaviour (Serra & Vincent, 1992, p. 99). In the case of the lattice Fun (E, \mathbb{R}) , a centre β , of a family of primitives $\{\psi_i\}$, does not oscillate under iteration if and only if (Serra & Vincent, 1992, p. 100):

$$f(x) \le (\beta f)(x) \le (\beta \beta f)(x) \le (\beta^3 f)(x) \le \cdots$$

or

$$\dots \leq \left(\beta^{3}f\right)(x) \leq \left(\beta\beta f\right)(x) \leq \left(\beta f\right)(x) \leq f(x)$$

for every $f \in \operatorname{Fun}(E, \mathbb{R})$, and for all $x \in E$. This dictates that the successive powers β^n of β be increasing with respect to the activity ordering, i.e. $\beta \leq \beta\beta \leq \beta^3 \leq \ldots \beta^{n-1} \leq \beta^n$ (Serra & Vincent, 1992, p. 100). This criterion is formalised in the following proposition. **Proposition 3.1.5 (Serra (1988a, p. 166)).** Let (\mathcal{L}, \leq) be a completely distributive lattice¹ and let $\{\psi_i\}$ be a family of increasing operators in $\mathcal{O}(\mathcal{L})$. Let $\eta = \bigvee_i \psi_i$ and $\xi = \bigwedge_i \psi_i$, and let β be the centre of the family $\{\psi_i\}$, i.e. $\beta = (I \land \eta) \lor \xi$. The sequence of successive powers β^n of β is increasing with respect to the activity ordering, i.e. $\beta \leq \beta\beta \leq \ldots \leq \beta^n$ and β^n , if for all integers $0 < n < \infty$

$$\xi_n \leq \eta_n,$$

where

$$\eta_n = \bigwedge_{k=0}^{n-1} \eta \beta^k = \eta \wedge \eta \beta \wedge \ldots \wedge \eta \beta^{n-1}$$

and

$$\xi_n = \bigvee_{k=0}^{n-1} \xi \beta^k = \xi \lor \xi \beta \lor \ldots \lor \xi \beta^{n-1}$$

Moreover β^n is then written

$$\beta^n = (I \wedge \eta^n) \vee \xi^n.$$

Remarks.

- 1. Although in theory iterations β^n are not necessarily bounded by a majorant with respect to the activity ordering, "in practice, as any numerical precision must be finite, there *always* exists an $n_0 < \infty$ such that $\beta^{n_0+1} = \beta^{n_0}$ " (Serra, 1988*a*, p. 168). This maximal iteration β^{n_0} is then both increasing and idempotent and therefore a morphological filter.
- 2. Examples of activity increasing sequences β^n can be found in Serra (1988*a*) and Heijmans (1994*a*).

Serra (1988*a*, p. 175) states that when the lattice (\mathcal{L}, \leq) is *finite*—such as the complete lattice of grey-scale digital images whose "spatial and numerical resolutions are finite and their extensions bounded" (Serra, 1988*b*, p. 111)—the idempotent limit of iterations of the centre of an inf-overfilter η and a sup-underfilter ξ , such that $\eta \leq \xi$, yields a unique strong filter called the *middle filter*. The middle filter is characterised by the following theorem.

¹ $\mathcal{P}(E)$ and Fun (E, \mathcal{T}) , where E is an arbitrary non-empty set and \mathcal{T} is a complete chain, are completely distributive lattices (Heijmans, 1994*a*, p. 26).

Theorem 3.1.6 (middle filter (Heijmans, 1994*a***, p. 428)).** Let (\mathcal{L}, \leq) be a modular complete lattice. If $\eta, \xi \in \mathcal{O}(\mathcal{L})$ are respectively an inf-overfilter and a sup-underfilter, such that $\eta \leq \xi$, then there exists a unique strong filter α , called the middle filter, that satisfies $\eta \leq \alpha \leq \xi$.

The relationship between the centre and the middle filter is characterised by the following proposition.

Proposition 3.1.7 (Serra (1988a, p. 174)). Let (\mathcal{L}, \leq) be a modular complete lattice. If $\eta, \xi \in \mathcal{O}(\mathcal{L})$ are respectively an inf-overfilter and a sup-underfilter, such that $\eta \leq \xi$, and β is the centre of η and ξ , then the function $n \to \beta^n$ is increasing with respect to the activity ordering. When (\mathcal{L}, \leq) is finite then the idempotent limit β^{n_0} of β^n is the middle filter α .

Remark. When the centre is self-dual then the middle filter is also self-dual.

3.1.2 Self-dual toggle mappings

Toggle mappings are the invention of Serra (1989). The notion encompasses several classes of mappings including the morphological centre. Toggle mappings are defined with reference to the power lattice (Fun $(E, \mathbb{R}), \leq$) where E is an arbitrary nonempty set. The idea of a toggle mapping is that one associates with an image $f \in \operatorname{Fun}(E, \mathbb{R})$

- 1. a family $\{\psi_i\}$ of mappings in $\mathcal{O}(\operatorname{Fun}(E, \mathbb{R}))$ called *primitives* and
- 2. a toggling criterion (decision rule) that determines at each point x, the best value from among the candidates $\psi_i(f)(x)$.

A simple example of a toggle mapping is thresholding (see Section 4.1). In this case there are two primitives—one that maps every image pixel to a foreground pixel, and the other that maps every image pixel to a background pixel—and the decision rule is nothing more than a comparison of the value of an image pixel with a constant (the threshold value). In this example, the two primitives are defined independently of f. However, this is not always the case; e.g. in the case of the morphological centre, "the primitives are themselves transformations acting on the initial image" (Serra & Vincent, 1992, p. 102). Formally, the toggle mapping is defined as follows. Definition 3.1.8 (toggle mapping (Serra & Vincent, 1992, p. 102)). Let $\{\psi_i\}$ be a family of mappings in $\mathcal{O}(\operatorname{Fun}(E,\overline{\mathbb{R}}))$ called *primitives*. Any mapping $\omega \in \mathcal{O}(\operatorname{Fun}(E,\overline{\mathbb{R}}))$ is called a *toggle mapping* of the primitives $\{\psi_i\}$ if

- 1. At each point x, $\omega(f)(x)$ is equal to one of the $\psi_i(f)(x)$, or f(x).
- 2. The decision rule that selects one of the $\psi_i(f)(x)$, say $\psi_{i_0}(f)(x)$, depends only on the various $\psi_i(f)(x)$, the value of f(x), and possibly one or more constants.
- 3. If at point x, at least one of the various ψ_i , say $\psi_{i_0}(f)(x)$, coincides with f(x) then

$$\omega(f)(x) = f(x) = \psi_{i_0}(f)(x).$$

Toggle mappings are not necessarily self-dual. However, it is possible to define a selfdual toggle mapping by extending the family of primitives to include their duals i.e. starting with the family $\{\psi_i\}$ define the family $\{\psi_i, \psi_i^*\}$ —and "symmetrizing the toggling criteria with respect to duality" (Serra & Vincent, 1992, p. 95).

Example. A simple algorithm for contrast enhancement of a digital image is as follows (Serra, 1982, p. 476). For a given pixel x_0 find the maximum and minimum values in a neighbourhood centred on it. Replace $f(x_0)$ in the output image with the value of the maximum or minimum depending on which value $f(x_0)$ is closer to. This mapping is explicitly the toggle mapping:

$$\omega(x) = \begin{cases} \delta_B(f)(x) & \text{if } \delta_B(f)(x) - f(x) < f(x) - \varepsilon_B(f)(x) \\ \varepsilon_B(f)(x) & \text{otherwise,} \end{cases}$$

where B is the elementary square or hexagon (depending on the underlying grid). Unfortunately this toggle mapping is not self-dual. A self-dual version can be defined as follows:

$$\omega'(x) = \begin{cases} \delta_B(f)(x) & \text{if } \delta_B(f)(x) - f(x) < f(x) - \varepsilon_B(f)(x), \\ f(x) & \text{if } \delta_B(f)(x) - f(x) = f(x) - \varepsilon_B(f)(x), \\ \varepsilon_B(f)(x) & \text{if } \delta_B(f)(x) - f(x) > f(x) - \varepsilon_B(f)(x). \end{cases}$$

By design toggle mappings generate jumps. It follows, therefore, that an idempotent toggle mapping offers a degree of control over this phenomenon; on successive application such a mapping cannot create any further jumps. The reader is directed to Theorem 5.5 in Serra & Vincent (1992) for a characterisation of a comprehensive class of idempotent toggle mappings. Serra & Vincent (1992) also provide examples of a variety of toggle mappings, including contrast mappings (Meyer & Serra, 1989), combined toggles, and amplifier toggle mappings.

3.1.3 Self-dual morphological operators based on the switch operator

The switch operator is an invention of Heijmans (1994b). The basic idea underlying the switch operator approach is, in the words of Heijmans (1996, p. 16),

"to construct self-dual operators ψ which are not necessarily idempotent, but which do satisfy the (weaker) constraint that they are *activityextensive*. The latter means that the sequence of iterates $\psi, \psi^2, \psi^3, \ldots$ is increasing with respect to the... activity ordering".

Formally, a switch operator is defined as follows.

Definition 3.1.9 (switch operator (Heijmans, 1994*a*, p. 468)). An antiextensive operator $\sigma : \mathcal{P}(E) \to \mathcal{P}(E)$ is called a *switch operator* if it satisfies

1.
$$\sigma(Y) \cap X \subseteq \sigma(X)$$
 if $X \subseteq Y \subseteq E$;

2.
$$\sigma(X \cup \{h\}) \cap \sigma(X^{c} \cup \{h\}) = \emptyset$$
, for $h \in E, X \subseteq E$

Remarks.

- 1. The first condition dictates that a point $h \in Y$ that is switched—i.e. its state changes from 1 to 0—by application of σ to Y, must also be switched by application of σ to a subset X of Y that also contains h.
- 2. The second condition dictates that "two complementary configurations centred at a point h cannot both force a switch of h" (Heijmans, 1994a, p. 469).

Proposition 3.1.10 (Heijmans (1994*a***, p. 469)).** To every increasing, selfdual operator ψ there corresponds a unique switch operator σ , and vice versa. This relationship is given by

$$\psi(X) = (X \setminus \sigma(X)) \cup \sigma(X^c);$$
$$\sigma(X) = X \cap \psi(X^c).$$

The expressions in Proposition 3.1.10 can be written as $\psi = \Psi(\sigma)$ and $\sigma = \Sigma(\psi)$, respectively, where Σ and Ψ are defined as follows.

Definition 3.1.11. Let the mappings $\Sigma, \Psi : \mathcal{O}(\mathcal{P}(E)) \to \mathcal{O}(\mathcal{P}(E))$ be defined by

$$\Sigma(\psi) = I \wedge \psi \nu$$
, where $\psi \in \mathcal{O}(\mathcal{P}(E))$,

$$\Psi\left(\sigma\right) = \left(I \wedge \nu\sigma\right) \lor \sigma\nu, \text{ where } \sigma \in \mathcal{O}\left(\mathcal{P}\left(E\right)\right),$$

where ν is the complement operator; i.e. $\nu(X) = X^{c}$ for all $X \in \mathcal{P}(E)$.

Remark. If σ is anti-extensive then $\Psi(\sigma)$ is the centre of $\nu\sigma$ and $\sigma\nu$ (Heijmans, 1996, p. 19).

Theorem 3.1.12 (Heijmans (1996, p. 25)). Every translation invariant switch operator $\sigma : \mathcal{P}(\mathbb{Z}^2) \to \mathcal{P}(\mathbb{Z}^2)$ is of the form $\sigma = \sigma_A$, where

$$\sigma_{\mathcal{A}}\left(X\right) = X \cap \bigcup_{A \in \mathcal{A}} X^{c} \ominus A$$

and \mathcal{A} is a collection of structuring elements satisfying $0 \notin A$ and $A \cap B \neq \emptyset$, for $A, B \in \mathcal{A}$. The corresponding self-dual operator $\psi_{\mathcal{A}} = \Psi(\sigma_{\mathcal{A}})$ is given by

$$\psi_{\mathcal{A}}(X) = \left(X \cap \bigcap_{A \in \mathcal{A}} X \oplus \check{A}\right) \cup \bigcup_{A \in \mathcal{A}} X \ominus A.$$

Remark. In other words, the above theorem states that $\psi_{\mathcal{A}}$ is the centre of the increasing operator $\bigvee_{A \in \mathcal{A}} \varepsilon_A$ and its negation $\bigwedge_{A \in \mathcal{A}} \delta_{\check{A}}$ (Heijmans, 1996, p. 25).

3.1.3.1 Construction of morphological filters from switch operators

Proposition 3.1.13 (Heijmans (1994a, p. 474)). Let $\sigma \in \mathcal{O}(\mathcal{P}(\mathbb{E}^n))$ be a switch operator and let ψ be the corresponding self-dual operator. Assume that ψ is continuous (this is true, e.g., when σ uses only finite structuring elements). Let $A_1, A_2, \ldots A_p$ be finite structuring elements which satisfy $\sigma(A_j) = \emptyset$. If $\gamma = \bigcup_{j=1}^{p} \delta_{A_j} \varepsilon_{A_j}$ and $\varphi = \gamma^*$ then the operator

$$\pi = (I \land \psi \varphi) \lor \psi \gamma$$

is self-dual and activity-extensive; i.e. $\pi \leq \pi\pi \leq \ldots \leq \pi^n$. Moreover the sequence π^n converges to a strong self-dual filter.



Figure 3.1: Hasse diagrams showing (a) natural ordering defined on the grey-level set $\{0, 1, \ldots, 255\}$; and (b) folded ordering, with the median as the fold point, defined on the set in (a).

Remarks.

- 1. The operator π is the centre of $\psi \varphi$ and $\psi \gamma$.
- 2. Heijmans (1996) proffers a more refined methodology based on the notion of *persistent* structuring elements.

3.1.4 Self-dual operators based on folded ordering

Evans et al. (1997) introduced the idea of imposing alternative orderings on the set of grey-levels for the purpose "of allowing a single idempotent morphological closing to filter both salt and pepper noise from an image" (Evans et al., 1997, p. 177). Of particular interest is an ordering they introduce, called *folded ordering*, that permits the construction of a self-dual filter from a single morphological closing. The ordering, defined on the finite grey-level set $\{0, 1, \ldots, m\}$, is simply a reversal of the natural ordering about some chosen grey-value called the *fold point*. Effectively, grey-values that are equidistant from the fold point are assigned the same rank (see Figure 3.1).

When the fold point is chosen to be the median of the set of grey-levels then the folded ordering is self-dual and can be used to construct a self-dual filter from a morphological closing in the following manner. The original grey-values are folded about the median (see Figure 3.2), a morphological closing is applied to the folded image, and then the folding is inverted (this requires the use of a template image that indicates the locations of the grey-values originally affected by the folded ordering). The process of folding about the median guarantees that regardless of whether the image or its negative are folded, the result is the same (the respective template images, however, will be different—each is the complement of the other). Consequently, a morphological closing—or indeed any operator—applied to the folded image yields a self-dual operation.

3.1.5 Self-dual annular filters

Serra (1988b, p. 107) introduced the annular opening $\gamma : \mathcal{P}(\mathbb{R}^n) \to \mathcal{P}(\mathbb{R}^n)$ defined

$$\gamma(X) = (X \oplus B) \cap X,$$

where B is a symmetric structuring element² that does not contain the origin. The annular opening is a translation-invariant algebraic opening (see Definition 2.5.1). In \mathbb{Z}^2 the behaviour of the operator is to remove isolated connected components (grains) in a set. The dual operator, the annular closing φ , is defined

$$\varphi(X) = (X \ominus B) \cup X.$$

In \mathbb{Z}^2 the behaviour of the operator is to "add to a set isolated points of the background X^c , in other words to remove isolated hole points" (Heijmans & Ronse, 1999, p. 1330). Heijmans & Ronse (1999) introduce the *annular operator* as a generalisation of the annular opening and closing. The behaviour of this operator is to remove "isolated points both in the foreground X and in the background X^{cr} " (Ronse & Heijmans, 1998, p. 49). The operator has the form

$$\psi(X) = (X \cap (X \oplus A)) \cup (X \ominus B),$$

where A and B are symmetric structuring elements. The operator is characterised in the following proposition.

² A structuring element B is said to be symmetric if $B = \breve{B}$ (see the remarks following Definition 2.7.2).



Figure 3.2: Illustration of folded ordering. (a),(b) Original 8-bit grey-scale image (sine-wave corrupted with salt-and-pepper noise) and its rendering as a surface. (c),(d) The folding of the original image about the median and its rendering as a surface. (e) Template image needed to invert folding.

Proposition 3.1.14 (Heijmans & Ronse (1999)). Let $A, B \in \mathcal{P}(\mathbb{Z}^2)$ be two symmetric structuring elements. The annular operator $\psi(X) = (X \cap (X \oplus A)) \cup (X \oplus B)$ is

- 1. a morphological filter if and only if $A \cap B \cap (A \oplus B) \neq \emptyset$;
- 2. a self-dual morphological filter if additionally A = B; and
- 3. a strong filter if $A \cap B \neq \emptyset$ and $A \cup B \subseteq A \oplus B$.

Heijmans & Ronse (1999, p. 1338) demonstrate that annular operators can be extended to grey-scale images using the fact that "every increasing operator on $\mathcal{P}(E)$ can be extended to grey-scale functions...on E by thresholding". In a companion paper Ronse & Heijmans (1998) extend the theory of annular filters to the more general setting of a modular lattice.

3.2 New approach: Folding induced self-dual filters

Evans et al. (1997) devised the *folded* closing, hereinafter called the ESJ folded closing, for the purpose of removing salt-and-pepper noise from an image. Unfortunately, they note that this operator does not allow image values less than the fold point to be replaced by values greater than the fold point and vice versa. Consequently it "can not completely remove pepper noise from a light area or salt noise from a darker region" (Evans et al., 1997, p. 179). The behaviour of the filter—and indeed any other constructed in the same manner—is constrained by the fact that it operates only on the image of folded grey-values and does not take into account the template image (the template image is used only to invert the folding – see Figure 3.2). In this section a new approach is presented based on the concept of the folding operator which generalises the notion of folded ordering in the case when the fold point is the median. The operator maps an image into an abstract space called *fold-space*. Effectively every image value is mapped to an ordered pair comprising a folded value and a template (indicator) value. Two operations, a meta-supremum and a meta-infimum are defined on this space such that they act as the usual supremum and infimum respectively on the folded image values, but at the same time operate on the indicator values. This makes possible, for example, the definition of meta-dilation and meta-erosion operators on fold-space. Again, these operators act as the usual dilation and erosion on the folded values but do some extra *housekeep*ing with respect to the indicator values. The notion of fold-space and the folding operator makes possible the definition of self-dual operators on the original image space called *folding induced self-dual filters* (FISFs). The ESJ folded closing represents one example from this class of operators. In this section it is shown that it is possible to design other types of FISF that overcome the limitations of the ESJ folded closing with respect to the filtering of salt-and-pepper impulse noise.

The core of the material in this section was presented at the fifth International Symposium on Mathematical Morphology and was published in volume 18 of the *Computational Imaging and Vision* series published by Kluwer Academic Publishers: Mehnert & Jackway (2000). The paper has since been cited in Heijmans & Keshet (2002) and Heijmans & Keshet (2001).

3.2.1 Fold-space and the folding operator

Folded ordering is a *distance ordering* (Barnett, 1976) defined on the finite greyvalue set $\{0, \ldots, m\}$. A distance ordering, sometimes used in statistical analysis, is an ordering of univariate data according to their absolute deviation or distance from some reference value.

Definition 3.2.1 (distance ordering). Let $S \in \mathcal{P}(\overline{\mathbb{R}})$ and let $\rho \in \overline{\mathbb{R}}$. The binary relation \mathcal{D}_{ρ} defined

$$a\mathcal{D}_{\rho}b$$
 if and only if $|a-\rho| \leq |b-\rho|$,

where $a, b \in S$, is called the distance ordering of S with respect to ρ .

Remarks.

- 1. When $\rho \leq \inf S$ then \mathcal{D}_{ρ} is equivalent to the partial order relation \leq .
- 2. When $\rho \geq \sup S$ then \mathcal{D}_{ρ} is equivalent to the partial order relation \geq .
- 3. When ρ is a value in the interval (inf S, sup S) then \mathcal{D}_{ρ} is only a relation of quasi-order; it is not anti-symmetric because $a\mathcal{D}_{\rho}b$ and $b\mathcal{D}_{\rho}a \not\Rightarrow a = b$ (because a and b may be two distinct values the same distance either side of ρ).

By Theorem 2.3.5 this ordering defines an equivalence relation on the set S:

 $a \equiv b$ if and only if $a \mathcal{D}_{\rho} b$ and $b \mathcal{D}_{\rho} a$,

where $a, b \in S$. In other words, a and b are equivalent if and only if they are equidistant from the reference value. By Theorem 2.3.6 this leads to the definition of a partial ordering on the set of equivalence classes. For example, in the case of ESJ folded ordering, two grey-values are deemed to be equivalent if they are equidistant from the fold point. This then defines the partial ordering (and in fact a total ordering) shown in Figure 3.1.

In order to formalise the idea of folded ordering, and to extend its definition to the power lattice (\mathcal{F}, \leq) , where $\mathcal{F} = \operatorname{Fun}(E, \mathcal{T})$, we now introduce the concept of *fold-space* and the *folding operator*. Hereinafter it is necessary to assume that the complete lattice (\mathcal{T}, \leq) , associated with the power lattice (\mathcal{F}, \leq) , is totally ordered; i.e. it is a complete chain (see Definition 2.3.2). Indeed this is the case when \mathcal{T} is one of the sets \mathbb{R}, \mathbb{Z} , or $\{0, 1, \ldots, m\}$ commonly used as the grey-level set for grey-scale images.

Let $\mathcal{H} = \operatorname{Fun}\left(E, \widetilde{\mathcal{T}}\right)$ be the set of all functions $\widetilde{f} : E \to \widetilde{\mathcal{T}}$ where $\widetilde{\mathcal{T}} = \mathcal{T} \times \{-1, 0, 1\}$. The set $\{-1, 0, 1\}$ is arbitrary in the sense that it can be any chain of three elements (they are indicator values only). Hereinafter \mathcal{H} is called *fold-space*. The folding operator is defined as follows.

Definition 3.2.2 (folding operator). Let $\sigma : \mathcal{F} \to \mathcal{H}$ be the *folding operator* defined point-wise as follows:

$$\sigma(f)(x) = \begin{cases} (f(x), 1), & \text{if } f(x) < f^*(x) \\ (f(x), 0), & \text{if } f(x) = f^*(x) \\ (f^*(x), -1), & \text{if } f(x) > f^*(x), \end{cases}$$

where * denotes a negation on \mathcal{F} .

Remark. When \mathcal{T} is the finite set $\{0, 1, \ldots, m\}$, the lattice (\mathcal{F}, \leq) possesses the unique negation $f^*(x) = m - f(x)$. When \mathcal{T} is one of the infinite sets \mathbb{R} or \mathbb{Z} , the lattice possesses multiple negations (Heijmans, 1994b, p. 31). In grey-scale morphology special attention is paid to the negation $f^*(x) = -f(x)$ wherein it is seen as the counterpart to set complementation on a boolean lattice.

The folding operator maps a function $f \in \mathcal{F}$ to a pair $\tilde{f} = (f_1, f_2) \in \mathcal{H}$ comprising the folded function $f_1 \in \mathcal{F}$ and an indicator function $f_2 : E \to \{-1, 0, 1\}$. The folding operator is a one-to-one mapping of \mathcal{F} into \mathcal{H} ; i.e. the image of every distinct element f of \mathcal{F} is a distinct element \tilde{f} of \mathcal{H} . The unfolding operator $\sigma^{-1} : \mathcal{H} \to \mathcal{F}$ is defined point-wise as follows:

$$\sigma^{-1}\left(\tilde{f}\right)(x) = \begin{cases} f_1(x), & \text{if } f_2(x) = 1\\ c, & \text{if } f_2(x) = 0\\ f_1^*(x), & \text{if } f_2(x) = -1, \end{cases}$$

where $\tilde{f} = (f_1, f_2), f_1 \in \mathcal{F}, f_2 : E \to \{-1, 0, 1\}$, and $c \in \mathcal{T}$ such that $c = c^*$. The constant c is called the *crease* and its existence and value are solely determined by the negation operator; e.g. when \mathcal{T} is the infinite set \mathbb{R} , then the negation $f^*(x) = -f(x) + k$ on the lattice (\mathcal{F}, \leq) prescribes the value of the crease to be k/2 (see also the comments in Section 3.2.5). The unfolding operator is the inverse of the folding operator when applied to the range of the folding operator. However, the set $\tilde{\mathcal{T}}$ associated with the space \mathcal{H} contains ordered pairs of the form (a, 0), where $a \neq c$, that are not images of any element of the set \mathcal{T} associated with the space \mathcal{F} . As defined, the unfolding operator maps all such pairs to the value c and is thus a many-to-one mapping for this set of pairs.

From the definitions of the folding and unfolding operators, it is easy to prove the following properties.

Proposition 3.2.3 (properties of the folding and unfolding operators).

1. If
$$\sigma(f) = (f_1, f_2)$$
 then $\sigma(f^*) = (f_1, -f_2)$ where $f \in \mathcal{F}$.
2. $\sigma^{-1}(\widetilde{f}) = [\sigma^{-1}((f_1, -f_2))]^*$ where $\widetilde{f} = (f_1, f_2) \in \mathcal{H}$.

3.2.2 Fold-space meta-operators

The folding operator maps a value $t \in \mathcal{T}$ into an ordered pair $(a_1, a_2) \in \widetilde{\mathcal{T}}$. The folded value a_1 is itself an element of \mathcal{T} . The indicator value a_2 is an element of $\{-1, 0, 1\}$. In keeping with the underlying idea of the ESJ folded ordering we can define the following equivalence relation on the set \widetilde{T} :

$$a \equiv b$$
 if and only if $a_1 = b_1$

where $a = (a_1, a_2)$ and $b = (b_1, b_2)$ are elements of the set \widetilde{T} . This relation is defined in terms of the equality relation defined on the first component of the elements of \widetilde{T} , i.e. the folded values. These folded values are elements of the complete chain

$$a \ll b$$
 if and only if $a_1 \leq b_1$

Unfortunately, however, the relation \ll is not itself a partial order relation. It is only a relation of quasi-ordering because although it is reflexive and transitive, it is not anti-symmetric; e.g. if $\mathcal{T} = \{0, 1, \ldots, m\}$ then it is clear that $(2, 0) \ll (2, 1)$ and $(2, 1) \ll (2, 0)$ but that $(2, 0) \neq (2, 1)$. Consequently, $(\tilde{\mathcal{T}}, \ll)$ is not a complete lattice³. This in turn means that \mathcal{H} is not a complete lattice. The definition of the equivalence relation above can equally be defined:

 $a \equiv b$ if and only if $a \ll b$ and $b \ll a$.

This relation induces a partition of $\widetilde{\mathcal{T}}$ into the subsets (equivalence classes) $X = \{(x, -1), (x, 0), (x, 1)\}, Y = \{(y, -1), (y, 0), (y, 1)\}, \ldots$ where $x, y, \ldots \in \mathcal{T}$ (Theorem 2.2.2). If we let \mathcal{S} be the set of equivalence classes X, Y, \ldots then the relation \trianglelefteq defined

 $X \trianglelefteq Y$ if and only if $a \ll b$,

for some $a \in X$ and $b \in Y$, is a partial order relation (by Theorem 2.3.6). It follows that the pair $(\mathcal{S}, \trianglelefteq)$ is a complete lattice. In fact \trianglelefteq defines a total ordering and so $(\mathcal{S}, \trianglelefteq)$ is a complete chain.

The problem with the ESJ folded closing is that, because it is defined on the complete chain of folded (grey) values, it is not possible to map (x, -1) to (x, 1) or vice versa. The solution is to define two meta-operators on (\tilde{T}, \ll) that act as a supremum and infimum on the folded values but at the same time operate on the indicator values (this is discussed further in Section 3.2.4). The meta-infimum and meta-supremum then make possible the definition of morphological meta-operators. Again, these operators are not morphological operators on fold-space, but rather on the space of folded values.

The range of operations permitted on the indicator values is dictated by the constraint that the resulting operation on \mathcal{F} must be self-dual. This motivates the definition of an FISF.

³ $\tilde{\mathcal{T}}$ is the product of two lattices and so it is possible to define a partial ordering called the *product ordering* (see (Heijmans, 1994*a*, p. 23)) such that $\tilde{\mathcal{T}}$ is a complete lattice. However, this ordering presupposes that the ordering of the indicator set $\{-1, 0, 1\}$ is important. For the purpose of generalising folded ordering this is not the case: it does not make sense to rank $(a_1, -1)$ above $(a_1, 1)$ or vice versa.

3.2.3 Folding induced self-dual filters

Definition 3.2.4 (folding induced self-dual filter). The product $\sigma^{-1}\Gamma\sigma \in \mathcal{O}(\mathcal{F})$, where $\Gamma \in \mathcal{O}(\mathcal{H})$, is called a *folding induced self-dual filter* (FISF) if for all $f \in \mathcal{F}$,

$$\sigma^{-1}\Gamma\sigma\left(f\right) = \left[\sigma^{-1}\Gamma\sigma\left(f^*\right)\right]^*.$$

Theorem 3.2.5. Let $\Gamma \in \mathcal{O}(\mathcal{H})$. The product $\sigma^{-1}\Gamma \sigma \in \mathcal{O}(\mathcal{F})$ is an FISF if $\Gamma\left(\widetilde{f}\right) = \left[\Gamma\left(\widetilde{f}^{\vee}\right)\right]^{\vee}$ for all $\widetilde{f} \in \mathcal{H}$ where $\widetilde{f}^{\vee} = (f_1, -f_2)$ when $\widetilde{f} = (f_1, f_2)$.

Proof. By definition $\sigma^{-1}\Gamma\sigma$ is an FISF if $\sigma^{-1}\Gamma\sigma(f) = [\sigma^{-1}\Gamma\sigma(f^*)]^*$ for all $f \in \mathcal{F}$. If we let $\tilde{f} = (f_1, f_2) = \sigma(f)$ then we can write the LHS as $\sigma^{-1}\Gamma((f_1, f_2))$. Using property 1 of Proposition 3.2.3 we can write the RHS as $[\sigma^{-1}\Gamma((f_1, -f_2))]^*$. If we let $\tilde{g} = (g_1, g_2) = \Gamma((f_1, f_2))$ and $\tilde{h} = (h_1, h_2) = \Gamma((f_1, -f_2))$ we can then write $\sigma^{-1}(\tilde{g}) = [\sigma^{-1}(\tilde{h})]^*$. Using property 2 of Proposition 3.2.3 this can be written as $\sigma^{-1}((g_1, g_2)) = \sigma^{-1}((h_1, -h_2))$. Hence if $\Gamma((f_1, f_2)) = (g_1, g_2)$ then $\Gamma((f_1, -f_2)) = (g_1, -g_2)$, i.e. $\Gamma(\tilde{f}) = [\Gamma(\tilde{f}^{\vee})]^{\vee}$.

Remark. This result in essence states that Γ must be an operator that is self-dual with respect to the second component of its argument; i.e. the indicator function.

3.2.4 Constructing FISFs

The ESJ folded closing can be written as the product $\sigma^{-1}\Gamma\sigma$ where $\Gamma\left(\tilde{f}\right)(x) = (\phi_B(f_1)(x), f_2(x))$, and $\phi_B \in \mathcal{O}(\mathcal{F})$ is the closing $\phi_B = \varepsilon_B \delta_B$ where $\delta_B, \varepsilon_B \in \mathcal{O}(\mathcal{F})$ are respectively a dilation and an erosion by a structuring element B (see Definition 2.8.8). This filter acts only on the first component of \tilde{f} . Hence according to Theorem 3.2.5 the ESJ folded closing is an FISF. We call FISFs constructed in this manner *type 1 FISFs*.

It is precisely because the ESJ folded closing does not take into account the second component of \tilde{f} that it performs poorly as a filter for salt-and-pepper noise. Values of the first component that are less than the fold point (crease) cannot be replaced by values greater than the fold point and vice versa. Consequently the ESJ folded closing "can not completely remove pepper noise from a light area or salt noise from a darker region" (Evans et al., 1997, p. 179). This then is the motivation for introducing the following definitions of the meta-supremum and meta-infimum operators respectively:

$$\dot{\bigvee} \{a, b\} = \begin{cases} a, & \text{if } b_1 < a_1 \\ b, & \text{if } a_1 < b_1 \\ (a_1, 0), & \text{if } a_1 = b_1 \text{ and } a_2 \neq b_2 \\ a, & \text{if } a_1 = b_1 \text{ and } a_2 = b_2, \end{cases}$$
$$\dot{\bigwedge} (a, b) = \begin{cases} a, & \text{if } a_1 < b_1 \\ b, & \text{if } b_1 < a_1 \\ (a_1, 0), & \text{if } a_1 = b_1 \text{ and } a_2 \neq b_2 \\ a, & \text{if } a_1 = b_1 \text{ and } a_2 \neq b_2 \\ a, & \text{if } a_1 = b_1 \text{ and } a_2 = b_2, \end{cases}$$

where $a, b \in \tilde{\mathcal{T}}$. These operators act as the usual supremum and infimum, respectively, on the first component; i.e., $\bigvee \{a, b\} = (a_1 \lor b_1, \cdot)$ and $\bigwedge \{a, b\} = (a_1 \land b_1, \cdot)$. However, the second component is determined such that the meta-supremum or meta-infimum of two distinct but equivalent elements of $\tilde{\mathcal{T}}$ is always the equivalent element with the second component equal to zero. The fold-space meta-closing $\Gamma = E_B \Delta_B$, where

$$\Delta_B\left(\widetilde{f}\right)(x) = \bigvee_{y \in B} \left\{ \widetilde{f}(x-y) \right\}, \text{ and } E_B\left(\widetilde{f}\right)(x) = \bigwedge_{y \in B} \left\{ \widetilde{f}(x-y) \right\},$$

satisfies Theorem 3.2.5 and can thus be used to construct a FISF. Like the ESJ folded closing this operator acts as a closing on the first component of \tilde{f} . For this reason we call Γ simply a *fold-space closing* even though it is not actually a closing on \mathcal{H} (\mathcal{H} is not even a complete lattice). We call FISFs constructed in this manner *type 2 FISFs*. Figure 3.3 demonstrates the effectiveness of this FISF in removing salt-and-pepper noise.

Unfortunately, as Figure 3.4 shows, both the ESJ folded closing and type 2 FISFs based on the fold-space closing perform poorly when large areas within the image are at either grey-value extreme. This then is the motivation for introducing the following alternate definitions of the meta-supremum and meta-infimum, respectively,

$$\overset{\cdot}{\bigvee} \{a, b, \ldots\} = \begin{cases}
(\alpha, \mu) & \text{if } \mu \text{ is unique} \\
(\alpha, 0) & \text{otherwise,} \\
\overset{\cdot}{\bigwedge} \{a, b, \ldots\} = \begin{cases}
(\beta, \nu) & \text{if } \nu \text{ is unique} \\
(\beta, 0) & \text{otherwise,}
\end{cases}$$



(a)

(b)



Figure 3.3: Demonstration of noise filtering of a natural scene. (a) Original 8-bit greyscale image. (b) Image corrupted with 60% salt-and-pepper noise (pixels randomly set to 0 or 255). (c) Result after applying a 5×5 median filter to the noisy image. (d) Result after applying a type 2 FISF, based on a fold-space closing with a 5×5 flat structuring element, to the noisy image.



Figure 3.4: Demonstration of noise filtering when significant parts of the image are at either extreme of the grey-value range. (a) Original 8-bit image with grey-value bands at 0, 128, and 255. (b) Original image corrupted with 50% salt-and-pepper noise (pixels randomly set to 0 or 255). (c) Result after applying an ESJ folded closing with a 5×5 flat structuring element to the noisy image (type 1 FISF). (d) Result after applying a 5×5 median filter to the noisy image. (e) Result after applying a type 2 FISF, based on a morphological closing with a 5×5 flat structuring element, to the noisy image. (f) Result after applying a type 3 FISF, based on a morphological closing with a 5×5 flat structuring element, to the noisy image.

where $\alpha = \bigvee \{a_1, b_1, \ldots\}, \mu = \text{mode} \{x \mid (\alpha, x) \in \{a, b, \ldots\}\}, \beta = \bigwedge \{a_1, b_1, \ldots\},$ and $\nu = \text{mode} \{y \mid (\beta, y) \in \{a, b, \ldots\}\}$. Again, these operators act as the usual supremum and infimum, respectively, on the first component. The operator $\Gamma = E_B \Delta_B$ defined in terms of these definitions satisfies Theorem 3.2.5. Again this operator acts as a closing on the first component of \tilde{f} and we call it a fold-space closing. We call FISFs constructed in this manner *type 3 FISFs*. The behaviour of this FISF is illustrated in Figure 3.4.

More generally the meta-supremum and meta-infimum permit the definition of foldspace morphological meta-operators. Again, these operators are not morphological operators on fold-space, but rather on the space of folded functions.

3.2.5 Implementation issues

The implementation of fold-space morphological (meta-)operators for digital images differs from the implementation of conventional grey-scale morphological operators only in that the supremum and infimum operations (either between two pixels or over a window of pixels) must propagate a template value in addition to a greyvalue. Given that a digital image is typically represented using a finite number of grey-values $\{0, 1, \ldots, m\}$ where m is a power of 2, the crease does not exist. However, the symmetric point with respect to the ordering does exist and so we can set c = m/2. Unfortunately c is not a representable grey-value. This presents a problem in the case of type 2 and type 3 FISFs when it comes to applying the unfolding operator. One possible solution is to replace any pixel values that would map to the crease with the grey-value of the preceding or succeeding representable grey-value (though the resulting filter is now only approximately self-dual). Two other possible solutions, which preserve self-duality, are: (1) to use only an odd number of grey-values, and (2) at the unfolding step, to replace any pixel values that would map to the crease with the corresponding pixel produced by applying a median filter (or indeed any other self-dual filter) to the original image.

A program to implement fold-space (meta-)dilation and (meta-)erosion (for type 2 FISFs) in MICROMORPH version 1.3⁴ is given in Appendix C. The implementation and example programs and images can be downloaded from the CSSIP anonymous ftp server: ftp://ftp.cssip.uq.edu.au/pub/other/fold.zip.

3.3 Discussion and summary

A discussion of several additional mathematical morphology approaches to constructing non-linear self-dual filters can be found in Soille (2003). These include methods based on *area* openings and closings, and morphological filters by *reconstruction*. Soille (2003, p. 263) also shows that the representation of self-dual, increasing and translation-invariant operators devised by Heijmans (1996) (see Theorem 3.1.12) can in fact be expressed as the anti-centre of a dual pair of thinning and thickening⁵. Soille (2003, p. 263) also comments on the recent inf-semilattice approach of Heijmans & Keshet (2001) stating that it is an "approach to the design

⁴ Software developed by the Centre for Mathematical Morphology, Paris School of Mines, and marketed by TRANSVALOR S.A., Paris France.

⁵ Thinning and thickening are defined in terms of the *hit-or-miss* transform which is itself defined in terms of the intersection of two erosions with suitably defined structuring elements (see Soille (2003, p. 140) for further details).

of self-dual morphological filters by defining a grey tone reference image so that a filter is applied if the current value of the input image is above that of the reference image and its dual otherwise". The approach is presented more recently in Heijmans & Keshet (2002). The paper discusses an alternative theoretical framework (Heijmans & Keshet, 2002, p. 55):

for morphological image processing that gives rise to image operators which are intrinsically self-dual... this alternative framework is entirely based upon the definition of a new self-dual partial ordering.

This partial ordering was introduced by Keshet (formerly Kresch) in Kresch (1998) and Keshet (2000). Heijmans & Keshet (2002) also discuss the folding induced selfdual filter (FISF) approach, presented in this chapter, and describe its relation to their approach. The reader is referred to Heijmans & Keshet (2002) for the details.

This chapter has:

- Discussed the rationale behind the desire to construct non-linear self-dual filters. In summary, such filters offer several advantages over their linear counterparts: they do not induce ringing and blurring, they can be designed such that they do not reduce the dynamic range and high frequencies in the image, they can be designed such that no new grey values are introduced in the image, and they can be designed such that they are independent of anamorphoses.
- Reviewed the principal lattice-theoretical approaches to constructing self-dual morphological operators and filters.
- Presented a new method for the construction of self-dual operators, called folding induced self-dual filters (FISFs), from arbitrary morphological (meta-) operators defined on an abstract space called fold-space; demonstrated that the folded closing proposed by Evans et al. (1997) is a particular type of FISF; demonstrated that other types of FISF can be designed that overcome the limitations of the folded closing with respect to the filtering of salt-and-pepper impulse noise; and discussed software implementation issues.

Chapter

Chromatin Segmentation



Nuclear chromatin is visualized by light microscopy as a mosaic of interchanging regions of low and high optical density (O.D.). The regions of high O.D. are well-defined as chromatin particles; features characterizing these particles enable the description of chromatin structure and the recognition of its changes during neoplasia

Dymitr Komitowski and Gerhard Zinser, 1985

The present chapter deals specifically with the problem of accurately and robustly segmenting the chromatin in digitised light microscopy images of cell nuclei. Conceptually, segmentation is the process of partitioning the domain of an image into subsets corresponding to the objects or features to be measured and/or classified. "In general, autonomous segmentation is one of the most difficult tasks in image processing. This step in the process determines the eventual success or failure of the analysis...For this reason, considerable care should be taken to improve the probability of rugged segmentation" (Gonzalez & Woods, 1992, p. 413).



The remainder of this chapter is organised as follows. Section 4.1 presents an overview of grey-scale image segmentation methods that have been published in the

literature. Section 4.2 presents an overview of the nature of chromatin and, in particular, the manner in which it is visualised using a light microscope, and its appearance and structure under the light microscope. Section 4.3 reviews and evaluates existing approaches to chromatin segmentation. The conclusion is that existing methods typically require the specification of one or more operational parameters—and are thus not robust to changes in, or non-uniformity of, illumination and staining—and/or do not produce a segmentation consistent with what a human observer would perceive as chromatin structure. Consequently in Section 4.4 a new method of chromatin segmentation is proposed that redresses these deficiencies. The new chromatin segmentation algorithm is based on the seeded region growing approach to segmentation. In Section 4.5 the seeded region growing algorithm proposed by Adams & Bischof (1994) is examined in detail wherein it is shown to be inherently dependent on the order of pixel processing. In Section 4.6 an improved seeded region growing algorithm is proffered that retains the advantages of the Adams & Bischof (1994) algorithm—fast execution, robust segmentation, and no tuning parameters—but is pixel order independent. In Section 4.7 a new fast ascending priority queue implementation is proposed that is suitable for implementing the marker-based watershed algorithms (particular cases of seeded region growing) of Meyer (1991). This permits the implementation of a fast watershed transform that is well suited for use in automated cytometry where near real-time processing is needed for an economically viable screening device. Finally, Section 4.8 presents a summary of the chapter.

The material in Section 4.6 has been published in *Pattern Recognition Letters:* Mehnert & Jackway (1997). The material in Section 4.4 is the subject of a pending International Patent Application: Mehnert & Jackway (2002).

4.1 Segmentation methods

The purpose of this section is to acquaint the reader with the various approaches to grey-scale image segmentation published in the literature. This will provide the necessary background for the review of chromatin segmentation methods to follow. For a more detailed overview of image segmentation methods the reader is referred to the surveys by Borisenko et al. (1987), Haralick & Shapiro (1985), and Fu & Mui (1981), and the reviews by Bamford (1999) and Pal & Pal (1993).

In his treatise *Image Analysis and Mathematical Morphology* Serra (1982, p. 456) states that:

the classical meaning in image analysis of segmenting a [grey-scale] picture, is to partition its support¹ into subsets inside which the picture has a homogeneous texture. We say that the picture has a homogeneous texture in a zone Z when it can be represented as a realization of a stationary random function with a range (of the covariance) that is small with respect to the dimensions of Z. However, this definition itself is not particularly operational since there is no suggestion as to how the zones Z might be detected. The question of segmentation is an exceedingly complex one. The problem becomes more precise (and thus accessible) when one knows a physical interpretation for the words "homogeneous textures". They can be cells of a certain type in a tissue, petrographic phases in a mineral, ridge lines in a relief, [chromatin particles in a cell nucleus] etc...

One of the most simple and popular techniques for image segmentation is thresholding (Pal & Pal, 1993, p. 1279). The technique is applicable when each of the zones is composed of a distinct range of grey-levels. Thresholding (also called greylevel thresholding) is the partitioning the density function (grey-level histogram) of the image such that each partition corresponds to a zone. In the simplest case the histogram is bimodal—it has two dominant peaks: one corresponds to the zone of interest (the foreground) and the other to the background—and a suitable threshold value lies somewhere in the valley between the two peaks (see Figure 4.1). Numerous parametric and non-parametric methods for automatically locating this value in the grey-level histogram have been published in the literature (Abutaleb, 1989). The threshold value determined using any of these methods is called a *global threshold* because its value is determined solely from the grey-level histogram (Gonzalez & Woods, 1992, p. 444). A drawback of global threshold methods is that they may "fail to detect thresholds if these are not properly reflected as valleys in the histogram" (Pal & Pal, 1993, p. 1280). In this case a local threshold method may be more appropriate. Local thresholding methods determine the value of the threshold using additional information derived from some local property of the image pixels; e.g. the mean value in a neighbourhood surrounding each pixel (Gonzalez & Woods, 1992, p. 444), or the modulus of the gradient (Serra, 1982, p. 457).

In principle many of the methods of threshold selection developed for bi-level thresholding can be extended to the case where the grey-level histogram is multi-modal. However, multi-level thresholding "is generally less reliable than its single-threshold

 $^{^{-1}}$ The support of a function (image) is the domain in which the function takes defined values.



Figure 4.1: Illustration of global thresholding. (a) Original image of the nucleus of a Papanicolaou-stained cervical cell. (b) Grey-level histogram of the original image. (c) Thresholding with respect to the value 135 (value selected after a visual inspection of the histogram). (d) Thresholding with respect to the value 144 (value chosen automatically such that the entropy of the resulting foreground and background histograms is maximised (Abutaleb, 1989, p. 23)). (e) Binary opening $\delta_B \epsilon_B$ where B is a disk of radius 5 pixels (to remove small artefacts).

counterpart" (Gonzalez & Woods, 1992, p. 444). The reason for this is that as the number of different zones increases, it becomes increasingly more difficult to distinguish the peaks in the grey-level histogram (Rosenfeld & Kak, 1981, p. 66).

For many images, that at first glance would seem to be amenable to global thresholding methods, it is not possible to find a partitioning of the grey-level histogram that yields a satisfactory segmentation. This is true in particular for images that contain shadows, or noise, or that have been captured under non-uniform illumination, or that contain very small and sparse objects so that the image is almost entirely background and the objects produce imperceptible peaks in the histogram. In this situation more elaborate segmentation techniques need to be employed. One approach, which is an extension of global thresholding, is to use an *adaptive threshold* method². Typically such methods partition the image into small non-overlapping blocks and a threshold is determined for each block independently (Pratt, 1991; Pal & Pal, 1993). When a block contains both foreground and background then the histogram will be bimodal and a threshold can be readily determined. However, "if a block contains objects only, or background only, it will not have a bimodal histogram...; but a threshold can still be assigned to it by interpolation from the... thresholds that were found for nearby bimodal blocks" (Rosenfeld & Kak, 1981, p. 70).

When the zones in an image are connected regions of pixels that have only a small grey-level variation then a high grey-scale variation between two adjacent pixels is likely to indicate that the pixels belong to two different zones. It should then be possible to segment these zones using some neighbourhood properties (Soille, 2003, p. 268). However, when the zones to be segmented are each highly textured then it may be the case that the grey-level variation within a zone is greater than that occurring at the boundary of two zones. In this case "local texture measurements can be performed so as to obtain similar values for pixels belonging to similar textures and therefore high variations between two neighbour pixels belonging to two different regions [zones]" (Soille, 2003, p. 268). Reviews of texture analysis and texture segmentation methods, as well as an original method based on zero crossings information, can be found in the Ph.D. thesis of Smith (1998). A treatment of texture analysis and segmentation, based on mathematical morphology, can be found in Soille (2003).

Literally hundreds of segmentation techniques have been published in the literature (Pal & Pal, 1993, p. 1278) and "this number continually increases each year" (Zhang,

² Some inconsistency in terminology exists in the literature. Some authors use *local thresholding* to mean *adaptive thresholding* (Pal & Pal, 1993, p. 1279). Adaptive thresholding is also called *dynamic thresholding*.

1996, p. 1335). The reason that there are so many image segmentation techniques is that no general theory of image segmentation has yet been developed and as a consequence no universal method of segmentation has emerged (Pratt, 1991, p. 597). The practitioner is thus faced with the situation where no single algorithm is applicable to all images and where, for a given class of images, not all algorithms are equally suitable (Pal & Pal, 1993, p. 1278). To further complicate matters, no universal quantitative metric of segmentation performance has yet been devised (Pratt, 1991, p. 597). Often, the performance of a newly developed algorithm is subjectively compared with that of a handful of existing algorithms using only a few test images (Zhang, 1996, p. 1335).

Although no universal theory of image segmentation exists, several taxonomies of segmentation methods have been published in the literature. Recent taxonomies include: Zhu & Yuille (1996), Pal & Pal (1993), and Gonzalez & Woods (1992, Chapter 7). Gonzalez & Woods (1992) broadly classify segmentation algorithms into two categories:

1. Discontinuity methods

These methods identify abrupt changes in the grey-levels within an image and use these as the basis for determining a partition of the image. Such abrupt changes (discontinuities) are characteristic of edges, points, and lines. The most common approach to detecting discontinuities is to centre a square or rectangular mask³ (e.g. 3×3) over each pixel in turn and to compute the sum of products of the mask coefficients and the corresponding image pixel greylevels (Gonzalez & Woods, 1992, p. 414). Mask coefficients are chosen such that the sum of products (response) is zero when all of the image pixels covered by the mask have the same grey-level value. Ideally, the resulting response should be high for pixels that lie on the boundary between zones in the image. Well known masks include the Roberts, Prewitt, and Sobel operators (gradient estimators) and the Laplacian (second-order derivative estimate). In practice, however, the boundaries may be incomplete or contain breaks, and spurious edge pixels are likely to be present because of factors such as noise and nonuniform illumination. Edge linking and other boundary detection methods are then typically used to assemble meaningful boundaries from the detected edge pixels. Other discontinuity methods include the *Hough transform* which

 $^{^3}$ The shape of the mask is dictated by the underlying grid. Thus hexagonal masks are used on the hexagonal grid.

can be used to link pixels that lie on a curve of specified shape (Gonzalez & Woods, 1992, p. 432).

2. Similarity methods

These methods seek to partition an image into regions (zones) of similar greylevel. These methods include thresholding, region growing, and region splitting and merging. Region growing methods begin with a set of seed regions—single pixels or connected components (see Definition 2.9.5)—and grow these into larger aggregates by appending neighbouring pixels that have similar properties. The watershed transform—a segmentation method originating from mathematical morphology—is a special case of seeded region growing (see Appendix B). Splitting and merging methods begin with an arbitrary partitioning of the image and then merge and/or split these regions in an iterative fashion in an attempt to create regions of pixels with similar properties.

Pal & Pal (1993) classify (grey-scale) image segmentation methods into five⁴ categories:

1. Grey-level thresholding

2. Iterative pixel classification

These methods include relaxation, MRF (Markov random field), and neural network methods. Relaxation methods iteratively add neighbouring pixels to regions (classes) on the basis of compatibility. The compatibility is assessed using either a probabilistic or fuzzy set theory approach. MRF methods use Markov random field or Gibbs random field models to model the spatial interactions in the digital image. Neural network methods are based on training a neural network (NN)—e.g. a feed-forward NN (multi-layer perceptron) or a Hopfield-type NN—to segment an image; e.g. the histograms of a sample of images can be used as the input to the NN and the desired threshold value corresponding to each used as the output.

3. Surface based segmentation

Algorithms developed for range image segmentation. These algorithms assume that the image data can be "interpreted as noisy samples from a piece-wise smooth surface function" (Pal & Pal, 1993, p. 1282)

 $^{^4}$ Pal & Pal (1993) actually include an additional category for colour images.

4. Edge detection

The mask-based methods in the *discontinuity methods* category of Gonzalez & Woods (1992) exemplify the methods in this category.

5. Methods based on fuzzy set theory

These methods include fuzzy thresholding, fuzzy clustering, and fuzzy edge detection.

Zhu & Yuille (1996) classify image segmentation methods into four categories:

1. Local filtering (edge detectors)

These methods correspond to the mask-based methods in the *discontinuity methods* category of Gonzalez & Woods (1992). A problem with local filtering methods is that they cannot guarantee closed unbroken contours (hence the need for edge-linking methods).

2. Snakes and balloons

These methods belong to the class of *deformable contour models* (also called *active contour models*). A deformable contour is a planar curve (usually closed) and an associated energy function. The initial position of the contour must be specified; e.g. it might be initialised manually such that it roughly surrounds the object of interest. The aim is to minimise the energy of the contour. If the energy function is appropriately defined, and the contour appropriately initialised, then a minimum energy is achieved when the contour exactly encloses the object (region) of interest. The *snake* is an active contour model introduced by Kass et al. (1987). In the words of Jain et al. (1998, p. 111):

a snake is modeled as being able to deform elastically but any deformation increases its internal energy causing a 'restitution force' which tries to bring it back to its original shape. But at the same time, the snake is immersed in a potential energy field (created by the image) which causes a force acting on the snake. These two forces [energies] balance each other and the contour actively adjusts its position and shape until it reaches a local minimum of the energy:

$$\mathcal{E}_{\text{snake}} = \int_{0}^{1} \left\{ \mathcal{E}_{\text{int}} \left(v \left(s \right) \right) + \mathcal{E}_{\text{image}} \left(v \left(s \right) \right) + \mathcal{E}_{\text{con}} \left(v \left(s \right) \right) \right\} ds,$$

where s is the parameterization of the contour, v(s) is a point on the contour.

The internal energy term imposes smoothness. It is usually defined in terms of the first and second-order derivatives of the contour. The image energy term attracts the contour to the desired object. For example, this might be defined in terms of the magnitude of the image gradient so that the contour is attracted to the salient edges in the image. The external constraint energy term permits an external constraint to be placed on the contour. For example, Kass et al. (1987) use this term to add, interactively using a graphical user interface, spring and repulsive forces to selected points on the contour. This can be used to move the snake out of one local energy minimum into another. Indeed a snake is sensitive to its initial position and to image noise, and can become trapped in a local minimum (Jain et al., 1998, p. 112). The balloon model of Cohen (1991) introduces an additional energy/force term to the snake that can push the contour in or out along its normal. This helps the contour to "trespass spurious isolated weak image edges, and counters its tendency to shrink" (Jain et al., 1998, p. 112). As a consequence the balloon is less susceptible to noise and to initial position.

3. Region growing and merging

The methods in this category are the region growing and region splitting and merging methods in the *similarity methods* category of Gonzalez & Woods (1992).

4. Global optimisation

These methods determine a segmentation of the image domain into regions corresponding to distinct objects by minimising an energy function designed in conjunction with Bayes's theorem (Kervrann, 2001, p. 163). The input image f is modelled as "being a degraded version of an ideal image which is assumed to be piecewise smooth" (Zhu & Yuille, 1996, p. 886). The underlying probability model consists of two parts: a *prior model* and a *data model* (Mumford, 1994, p. 135) The prior model is a model of possible image segmentations. The model should capture prior knowledge about the boundaries and regions and any other scene structures that may be relevant. If S denotes the description of a particular segmentation, then the prior model is specified by the probability distribution p(S) giving the probabilities for all possible S. The data model is a model of all of the images that are consistent with the prior model. It is specified by p(f, S) for all possible images f and all possible S. Together they define (Freund, 1984, p. 131) the conditional probabilities p(f|S) of any image f given the segmentation S:

$$p(f|S) = \frac{p(f,S)}{p(S)}.$$

From Bayes's theorem (Freund, 1984, p. 140) it follows that:

$$p(S|f) = \frac{p(S) p(f, S)}{p(f)}$$

 $\propto p(S) p(f, S)$

which is the probability of obtaining the segmentation S given the image f (called the *posterior probability* of S). The most likely segmentation is the one that maximises this probability. This is equivalent to minimising the energy functional E(S) defined (Mumford, 1994, p. 136):

$$E(\mathcal{S}) = -\log (p(\mathcal{S}|f))$$

= $-\log (p(\mathcal{S})) - \log (p(f,\mathcal{S}))$
= $E_{p}(\mathcal{S}) + E_{d}(f,\mathcal{S}).$

The first term, $E_{\rm p}$, is a measure of how reasonable each segmentation model is. Lower values indicate that the segmentation is more common, and higher values less common. The second term, $E_{\rm d}$, is a fidelity term that describes the interaction between the observed data and the data model (Kervrann, 2001, p. 163). Morel & Solimini (1995, p. xii) argue that "most segmentation algorithms try to minimize, by several very different procedures, one and the same *Segmentation energy*" and that this energy is that associated with the Mumford-Shah model. Morel & Solimini summarise the model as follows (Ω denotes the domain of the image):

The Mumford-Shah model defines the segmentation problem as a joint smoothing/edge detection problem: given an image g(x), one seeks simultaneously a "piecewise smoothed image" u(x) with a set K of abrupt discontinuities, the "edges" of g. Then the "best" segmentation of a given image is obtained by minimizing the functional

$$E(u,K) = \int_{\Omega \setminus K} \left(|\nabla u(x)|^2 + (u-g)^2 \right) dx + \text{length}(K).$$

The first term imposes that u is smooth outside the edges, the second that the piecewise smooth image u(x) indeed approximates g(x), and the third that the discontinuity set K has minimal length (and therefore in particular is as smooth as possible).

The first and third terms together constitute $E_{\rm p}$ and the second term is $E_{\rm d}$. Minimisation of the functional is usually very difficult requiring computationally expensive algorithms such as simulated annealing (Zhu & Yuille, 1996, p. 886).

A relatively new approach to image segmentation, called the *level-set* approach, is not addressed in any of the taxonomies above. The method is described by Medioni et al. (2000, p. 25) as follows:

The main idea is to describe a curve Γ (or a surface) as the zero level set of a function of higher dimension Φ ...Then, instead of evolving the curve Γ , we consider the evolution of the function Φ and extract the zero levels to see the results. This method permits to easily handle topology changes, and has proved to be a powerful tool.

The reader is referred to Sethian (1999) and Osher & Fedkiw (2003) for an introduction to level-set methods and their application in computer vision. The level-set formulation of the classical snake is described in Osher & Fedkiw (2003, Chapter 12). The formulation depends on the image gradient to stop the evolution. Another active contour model formulation described in Osher & Fedkiw (2003, Chapter 12) uses a stopping criterion based on the Mumford-Shah segmentation technique.

Bamford (1999, p. 73) points out that, although (these) classifications of segmentation methods provide "useful summaries, and convenient methods for labelling, new and existing techniques...[they] have limited practical use in matching algorithms to applications". Moreover, in lieu of any universal theory of image segmentation, and therefore of any guiding principles for solving a particular image segmentation problem, it remains the case that (Bamford, 1999, p. 198):

the development of a solution to a specific image segmentation problem is often a fairly arbitrary process, depending upon the person developing it and their background.

4.2 The nature of chromatin

This section presents an overview of what chromatin is, the manner in which it is visualised using a light microscope, and its appearance and structure under the light microscope. This provides the necessary background for the review of previous approaches to chromatin segmentation presented in the next section.

Every human—and more generally mammalian—cell has three main components: cell membrane, cytoplasm, and nucleus (Koss, 1992, p. 15). The nucleus and cytoplasm can be seen in Figure 4.2. The cell membrane encloses the cytoplasm and "acts as a selective barrier that enables the cell to concentrate nutrients gathered from its environment and retain the products it synthesizes for its own use, while excreting its waste products" (Alberts, Johnson, Lewis, Raff, Roberts & Walter, 2002, p. 11). The cytoplasm contains all of the other cell organelles and is the site where most of the cell's intermediary metabolism occurs (Alberts et al., 2002, p. 660). "Within the cytoplasm, enclosed in its own membrane or envelope, there is a smaller, approximately spherical dense structure-the nucleus" (Koss, 1992, p. 15). The nucleus of a cell occupies about 10% of the total cell volume and contains nearly all of its DNA (deoxyribonucleic acid) (Alberts et al., 2002, p. 197). DNA consists of large molecules with a three-dimensional structure of a double helix (Alberts et al., 2002, p. 193). It governs "the genetic and functional aspects of cell activity" (Koss, 1992, p. 15). In particular it carries the genes—"the information that specifies all the proteins that make up an organism" (Alberts et al., 2002, p. 198). The complete DNA sequence⁵, or *genome*, is divided between a set of thread-like structures called chromosomes. "Each chromosome consists of a single, enormously long linear DNA molecule associated with proteins that fold and pack the fine DNA thread into a more compact structure. The complex of DNA and protein is called *chromatin*" (Alberts et al., 2002, p. 198). The word *chromatin* stems from the Greek *chroma* meaning *coloured*. Chromatin is so-named because of its ability to take on stain.

4.2.1 Fixation and staining

Cells or tissues are nearly invisible when viewed under a conventional light microscope (Schulte & Wittekind, 1994, p. 200). Consequently they are usually *fixed* and *stained* prior to microscopic examination. The fixation process principally serves to preserve the biologic material and to prepare it for the uptake of dye (Giroud, 1994,

 $^{^5}$ The nucleus of a human cell is about $6\mu{\rm m}$ in diameter and yet it "contains approximately 2 meters of DNA if stretched end-to-end" (Alberts et al., 2002, p. 198).



Figure 4.2: Papanicolaou-stained cervical cells.

p. 186). The staining process serves "to enhance the contrast of cells and tissues versus the unstained background" (Schulte & Wittekind, 1994, p. 200). Although other contrast enhancement techniques—such as polarisation microscopy and interference contrast microscopy—exist, "these methods are not frequently used in routine cytology and do not have the importance of conventional staining in diagnostic practice" (Schulte & Wittekind, 1994, p. 200).

4.2.2 Absorbance, extinction, optical density

Staining is a physicochemical process that adheres colour to cells and tissues (Schulte & Wittekind, 1994, p. 200). As light passes through the stained material its intensity (power per unit area) is reduced. A measure of this reduction is given by the parameter *absorbance* (A) which is defined:

$$A = -\log\left(\frac{I}{I_0}\right),\,$$

where I_0 is the intensity of the incident light and I is the intensity of the transmitted light. Absorbance is also called *extinction* (E) or *optical density* (O.D.) in the older literature (James & Tanke, 1991, p. 144). The *Beer-Lambert* law relates the absorbance of a stain, i.e. an aqueous or alcoholic solution of powdered dye, to its concentration (James & Tanke, 1991, p. 144):

$$A = k.c.l,\tag{4.1}$$

where k is a constant called the *extinction coefficient* (which is a characteristic of the dye), c is the concentration of the solubilised dye, and l is the length of the path that the light travels as it passes through the stain. The Beer-Lambert law shows that concentration is not linearly related to intensity, but rather to absorbance. For *stoichiometric* stains—stains for which the amount of stain uptake in the nucleus is proportional to the amount of DNA—it is preferable to work with a digital image for which the grey-values represent optical density rather than intensity.

4.2.3 Chromatin structure as visualised by light microscopy

Chromatin is visualised by light microscopy as a mosaic of interchanging dark and light regions (Komitowski & Zinser, 1985, p. 178). "Light-microscope studies in the 1930s distinguished between two types of chromatin in the interphase nuclei... of cells: a highly condensed form, called **heterochromatin**, and all the rest, which is less condensed, called **euchromatin**" (Alberts et al., 2002, p. 222). Giroud (1994, p. 191) states that:

three kinds of information can be extracted from chromatin patterns as they appear after fixation and staining. The first, defined as *condensation*, is the degree of chromatin coiling, considering that chromatin architecture ranges from condensed, typically representing the genetically inactive heterochromatin, to decondensed, representing the transcriptionally active euchromatin. The second, defined as *distribution*, is the amount of chromatin in the various degrees of chromatin condensation. The third, defined as *organization*, concerns the topographic arrangement of chromatin at the various levels of chromatin condensation.

Koss (1992, p. 46) states that:

in well-fixed and stained cells, within the homogeneous background of the nucleus (sometimes referred to as nuclear "sap"), one can observe a fine network of thin, threadlike linear condensations, known as the *linin network*. Located at various points in the network are small, dark granules of odd shapes, the *chromocenters*.

These granules of condensed chromatin are also described in the literature as *particles* (Sandritter et al., 1974; Danielsen et al., 1989; Komitowski & Zinser, 1985), and *blobs* (Smeulders et al., 1978).
Doudkine et al. (1995, p. 286) state that:

in cytopathology chromatin is typically referred to as "coarsely-clumped" or "finely-clumped", as having a "salt-and-pepper" or "smooth" appearance, or as having a "cart-wheel-like" or "clockface" distribution.

Beil (1992, p. 129) states that:

chromatin structure is caused by various cellular processes, such as chromatin condensation, configuration of nucleoli.... Pathologists use adjectives like granular, clod-like, diffuse, etc. for the characterization of chromatin "textures".

4.3 Review of previous approaches to chromatin segmentation

This section reviews chromatin segmentation algorithms that have been published in the literature. The key approaches are identified, described, and critiqued. Related algorithms are also discussed.

The invention of the Taxonomic Intra-cellular Analytic System (TICAS) by Wied et al. (1968) spawned the first quantitative studies—e.g. Bartels et al. (1968) and Bartels et al. (1969)—of cell texture by means of digital image analysis⁶. However, Rowiński et al. (1972) appear to have been the first to attempt to quantify nuclear texture using a structural approach; i.e. based on the segmentation of chromatin into texture primitives. Remarkably, as Table 4.1 shows, only a handful of researchers have since attempted this mode of analysis. This section reviews the different algorithms that have been devised for the segmentation of chromatin. In the discussion that follows, a variety of stains are mentioned. Some of these stain only the nucleus, whilst others—most notably the Papanicolaou stain—stain both the nucleus and the cytoplasm. From the point of view of chromatin segmentation, the type of stain used is irrelevant. However, from the point of view of nucleus segmentation (not reviewed here) the type of stain used is relevant: it is more difficult to segment nuclei from images in which both nuclei and cytoplasm are stained.

⁶ An earlier system, CYDAC (Cytophotometric Data Conversion), was able to discriminate between five different types of leukocytes (white blood cells) on the basis of cytomorphologic features. However, the TICAS was able to "distinguish between cells without appreciable morphological differences, on the basis of differences in the... absorption pattern" (Bartels et al., 1968, p. 205).

Algorithm	Brief description	Related algorithms
Rowiński et	Multiple grey-level thresholds to generate several binary im-	Klawe & Rowiński (1974)
al. (1972)	ages.	
Sprenger et	Single grey-level threshold to detect condensed chromatin.	de Campos Vidal et al.
al. (1973)		(1973), Sandritter et al.
		(1974), Al et al. (1978),
		Danielsen et al. (1989)
Smeulders et	Region growing/merging from local maxima (with respect to	Smeulders et al. (1979)
al. (1978)	optical density).	
Meyer (1978)	Top-hat transform used to detect chromatin particles and	Meyer (1980), Giménez-Mas
	interchromatinic channels.	et al. (1995)
Rodenacker	Ricefield transformation: Independent grey-scale thinning	Rodenacker et al. (1987),
et al. (1983)	and thickening. Thinning (resp. thickening) yields a skele-	Rodenacker (1992)
	ton that divides the image into a mosaic of connected com-	
	ponents; one for each minimum (resp. maximum).	
Komitowski	Local adaptive thresholding to detect regions of high optical	
& Zinser	density.	
(1985)		
Kondo &	Partitioning the image into regions each containing a single	
Taniguchi	local maximum (with respect to optical density) followed	
(1986)	by local adaptive thresholding to segment a particle of high	
	optical density within each region.	
Young et al.	Partitioning the grey-level histogram into 3 parts and using	Madachy & Fu (1988),
(1986)	this to to label each nucleus pixel as low, medium, or high	Doudkine et al. (1995),
	optical density.	Palcic et al. (2000)
Tanaka et al.	Unspecified algorithm in the CYBEST Model 4; most likely	Tanaka et al. $(1987b)$
(1987a)	grey-level thresholding.	
Beil (1992)	Region growing and merging.	
Wolf et al.	Watershed of the gradient followed by region merging.	Beil et al. (1995)
(1995)		
Albregtsen et	Region growing.	
al. (1995)		
Walker &	Thresholding over all possible grey-levels to generate a stack	Jones (2001)
Jackway	of binary images.	
(1996)		
Jackway	The original image is decomposed into a set of multiscale	
(1996)	images. For each multiscale image, the watershed of the	
	gradient is computed using the regional minima (for negative	
	scales) or maxima (for positive scales) of the multiscale image	
	as markers.	

 Table 4.1: Chromatin segmentation algorithms published in the literature.

Segmentation algorithm of Rowiński, Pieńkowski & Abramczuk

Rowiński et al. (1972) used the Quantimet B Image Analysis Computer (Metals Research⁷, Cambridge, England) to quantitatively characterise the morphology of the chromatin of Feulgen-stained lymphocytes⁸. "In this instrument the microscopic image is projected onto the screen of a television camera, [and] the detected signal passes into the computer" (Rowiński et al., 1972, p. 76). The algorithm devised by Rowiński et al. to segment the chromatin involves:

- 1. aligning the image of a single nucleus in the measuring frame of the Quantimet; and
- 2. globally thresholding the digitised optical density image for several different threshold values.

The first threshold value is the one that detects all of the chromatin (i.e. the entire nucleus). The second and subsequent thresholds are defined to be the value of the preceding threshold plus a fixed constant. The threshold value for which no chromatin is detected is used as the reference optical density level. The resulting segmentation effectively consists of a collection of binary images corresponding to intervals of optical density. The drawbacks of this algorithm are that:

- 1. it is based on global thresholding and is thus sensitive to noise and to nonuniformity of illumination and/or staining;
- 2. a step value must be specified; and
- 3. the result is a stack of binary images rather than a single partitioning of the grey-scale image.

Klawe & Rowiński (1974) applied the same segmentation method (also using the Quantimet B) to the nuclei of cells from buccal (oral) smears.

 $^{^7}$ Metals Research Limited later merged with Cambridge Instruments which in turn was incorporated into Leica.

 $^{^8}$ Lymphocytes are "mononuclear cells that are the predominant cells in immune organs" (Knox et al., 1994, p. 1037)

The segmentation algorithm of Sprenger, Moore, Naujoks, Schlüter & Sandritter

Sprenger et al. (1973) used the EPERC (equiprobable extinction range contours) computer program, written by Bartels et al. (1968), to quantitatively characterise the chromatin pattern in Feulgen-stained cells from cervical smears. For each digitised image of a nucleus, a fixed threshold value was used to isolate the nucleus from the background. The algorithm devised by Sprenger et al. to segment the chromatin involves:

- 1. choosing a single threshold value (with respect to optical density) that represents the cut-off between weakly-stained (non-condensed) and densely-stained (condensed) chromatin; and
- 2. globally thresholding the grey-level image to produce a binary image of the condensed chromatin.

Sprenger et al. choose the threshold value to be that value for which a predetermined proportion of the nucleus area (namely, 80%) is labelled as non-condensed. The main drawbacks of the algorithm are that:

- 1. it is that it is based on global thresholding and is thus sensitive to noise and to non-uniformity of illumination and/or staining; and
- 2. it artificially imposes a fixed value on the proportion of the nucleus that should be labelled as non-condensed.

De Campos Vidal et al. (1973) used the method of Sprenger et al., and two variations on the method, to quantitatively characterise the differences in nuclear structure in liver cells as visualised by the Feulgen, Giemsa, toluidine blue, and gallocyanin staining methods. The two variant methods differ from that of Sprenger et al. only in terms of the method used to select the threshold value. In the first case a fixed threshold is used. In the second, a "squares ratio" procedure (see de Campos Vidal et al. (1973) for details) is used.

The algorithm of Sandritter et al. (1974) is another variation on that of Sprenger et al. The algorithm was devised to segment condensed chromatin in digitised images of the nuclei of Feulgen-stained breast epithelium and glandular cells. The first steps of the algorithm are exactly the same as that of Sprenger et al., namely that a

fixed threshold-value is chosen and the image is thresholded to obtain the condensed chromatin. However, in addition, the original image is thresholded a second time, using a "somewhat lower fixed value" (with respect to optical density), and all those pixels that lie adjacent to a pixel detected by the first threshold are also labelled as condensed chromatin.

The segmentation algorithm of Al et al. (1978), implemented on the Leyden Television Analysis System (LEYTAS)⁹, is yet another variation on that of Sprenger et al. The algorithm was devised to segment the chromatin structure in digitised images of the nuclei of cervical cells stained according to the acriflavine-Feulgen-SITS method. The LEYTAS permits the processing of multiple cell nuclei in a digitised field of view. An initial threshold is used to detect all of the pixels in the nuclei and none in the background. A second threshold is applied to the original image to detect areas of high optical density (this step is equivalent to the algorithm of Sprenger et al.). Subsequent steps in the algorithm use binary opening $\delta_B \epsilon_B$ (see Definition 2.5.1, Proposition 2.6.8, and Definition 2.7.3), binary dilation, and set differencing to produce three binary images: one containing nuclei with hyperchromatic chromatin (large optically dense areas of chromatin), one containing nuclei of coarse chromatin (i.e. particles of optically dense chromatin), and one containing nuclei with hypochromatic chromatin (weakly-stained chromatin).

The segmentation algorithm of Danielsen et al. (1989) is the same as that of Sprenger et al. except that the threshold value is chosen interactively. Danielsen et al. used the method to segment the heterochromatin (densely-stained) particles in digitised electron micrographs (electron microscopy) of ultra-thin tissue sections prepared from biopsies from mice livers. The boundary of each nucleus was defined interactively.

Segmentation algorithm of Smeulders, Cornelisse, Vossepoel & Ploem

The segmentation algorithm devised by Smeulders et al. (1978) partitions a nucleus image into a number of sub-regions that visually correspond to chromatin aggregates (blobs). The image data used by Smeulders et al. are digitised photographic negatives of nuclear absorption images of cervical cell nuclei stained with acriflavine-Feulgen-stilbene. These images were processed, with software developed by van der

⁹ Developed in Leiden by Professor Ploem and his colleagues in collaboration with the Leitz company (Husain, 1994, p. 9).

Ploeg et al. (1977), to isolate the nucleus and eliminate the background in preparation for chromatin segmentation. Smeulders et al. (1978, p. 221) give the following description of their chromatin segmentation algorithm:

The method used starts by searching for a local greylevel maximum. From here those points on the slopes are added to the blob, which present continuously decreasing greylevels. The size of the blob is restricted by a minimum greylevel and a minimal greylevel gradient. The procedure is repeated through the entire image. Finally blobs superimposed on the slope of a second blob are merged by a nonlinear algorithm.

From this description it is clear that the algorithm consists of a region growing step, initiated from a set of seed regions, followed by a region merging step. The seed regions are the local maxima of the input image. These correspond to the local minima in the original photomicrograph. Smeulders et al. do not give any details concerning the region merging step. An example of the segmentation produced by the algorithm is shown in Figure 4.3. The algorithm has two major drawbacks:

- 1. The seed regions are not grown in parallel. Rather, the input image is scanned (presumably in raster order) until a local maximum is found and this region is then grown in an iterative fashion. At each iteration all of those pixels that border the growing region are examined. Those pixels that are of lesser or equal grey-value are added to the region. The growth continues until either there are no more candidate neighbouring pixels, or the candidate pixels have grey-values that fall below a predefined minimum grey-level, or the corresponding modulus of the gradient evaluated for each candidate pixel falls below a predefined value. This procedure is repeated for each local maximum. As a consequence the regions are not permitted to grow in competition. This is likely to lead to different results depending on the order in which the input image is scanned.
- 2. Two parameters must be specified a priori: a minimum grey-level value and a minimum grey-level gradient.

The segmentation algorithm described by Smeulders et al. (1979) appears to be essentially the same as that of Smeulders et al. (1978). Indeed the algorithm operates on the same type of image data and the same nucleus isolation preprocessing step is required. In the words of Smeulders et al. (1979, p. 200), the algorithm



Figure 4.3: Illustration of the segmentation algorithm Smeulders et al. (1978). Reproduced from Smeulders et al. (1978) with the permission of the publisher. (a) Nucleus from a cervical cell stained with acriflavine-Feulgen-stilbene. (b) Line printer representation of the segmented regions (only 16 blobs are shown). The aspect ratio distortion is an artefact of the line printer representation.

involves the segmentation of the regions with relative high optical densities in the image starting from local O.D. maxima (starting points) which are found by scanning the digitized image from top left to bottom right. The segmentation procedure is restricted to a fixed percentage of the nuclear area by an O.D. threshold computed for each nucleus individually from the intranuclear histogram. Only picture points with O.D. values above this threshold are included in the segmentation procedure. The individual segments are found by a dilatation from the starting points to the adjacent points with continuously decreasing O.D. values. Dilatation stops when the threshold O.D. value is reached or when the slope in O.D. is no longer declining¹⁰.

Presumably overlapping regions are merged as a post-processing step.

Segmentation algorithm of Meyer

The segmentation algorithm devised by Meyer (1978) segments a nucleus image into either chromatin particles (dark regions) or interchromatinic channels (light regions). Meyer illustrates the algorithm on a digitised image of the nucleus of a Feulgen stained cervical cell. The algorithm is well-known today as the top-hat $transform^{11}$ and finds more general use as a segmentation tool (see, for example, Gonzalez & Woods (1992) and Soille (2003)). The top-hat transform is defined to be the arithmetic difference $f - \psi(f)$ where $f \in \operatorname{Fun}(\mathbb{E}^n, \mathcal{T})$ and $\psi \in \mathcal{O}(\operatorname{Fun}(\mathbb{E}^n, \mathcal{T}))$ is an opening (see Definition 2.5.1). This difference is non-negative on the domain of f because the opening operation is anti-extensive. Meyer (1978) uses the opening $\psi = \delta_B \epsilon_B$ (see Definition 2.8.8) where B is a disk (or rather its digital equivalent) of sufficient size. If the grey-scale nucleus image is viewed as a topographic landscape (the light pixels corresponding to high areas and the dark pixels to low areas), this opening removes positive peaks that are thinner than the diameter of the disk. Consequently the arithmetic difference $f - \psi(f)$ yields an image containing only these peaks. Negative peaks can be obtained by applying the transform to the negative of the image, or equivalently by computing the arithmetic difference $\psi^*(f) - f$ where ψ^* is the dual closing. To obtain a binary mask of the chromatin particles (respectively interchromatinic channels), the top-hat (respectively inverted top-hat) image

¹⁰ Dilatation is the term used by Matheron (1975) to denote the Minkowski sum $A \oplus \check{B}$. When B is symmetric about the origin, i.e. $B = \check{B}$, then this operator is identical to the more modern definition of binary dilation. As used by Smeulders et al. (1979) dilatation means iterative dilation by the unit ball B.

¹¹ The term *top-hat* does not appear in Meyer (1978). It appears later in Meyer (1979).

must be thresholded and intersected with the nucleus mask. Meyer's algorithm is illustrated in Figure 4.4. Meyer (1978) gives no details concerning the segmentation of the nucleus. However he discusses the problem in several subsequent papers including: Meyer (1979), Meyer (1980), and Meyer (1986).

The major drawback of the Meyer algorithm is that the size of the disk (width of the top-hat) and the threshold value (height of the top-hat) must be tuned to the application. Indeed Meyer (1980, p. 165) states that it is necessary to "heuristically search for the dimension which gives the best visual segmentation... [and] once these parameters have been found, they remain the same for the whole study".

The segmentation algorithm of Giménez-Mas et al. (1995) is a trivial variation on that of Meyer (1978). It involves the computation of several independent top-hat transforms. The height (threshold) of each top-hat is kept the same, but the width (size of the disk structuring element) is made to vary. In the words of Giménez-Mas et al. (1995, p. 41), the steps involved are:

(1) opening, consecutively using a structuring element size of one, two, four, six and eight pixels; (2) subtracting each of these images from its original grey level image; and (3) densitometrically thresholding the resulting images, from 10 to 255. These operations... [result] in a sequence of binary images.

Segmentation algorithm of Rodenacker, Gais, Jütting & Burger

The segmentation algorithm devised by Rodenacker et al. (1983), which they call the *ricefield transformation*, is based on grey-scale *thinning* as defined by Serra (1982, p. 450). The thinning of a two-dimensional grey-scale image $f \in \text{Fun}(\mathbb{Z}^2, \mathbb{R})$ by a pair of disjoint flat structuring elements $\mathbf{B} = (B_1, B_2)$ is defined in terms of grey-scale erosion and dilation (see Definition 2.8.8) as follows:

THIN
$$(f, \mathbf{B})(x, y) = \begin{cases} (f \oplus B_1)(x, y) & \text{if } (f \oplus B_1)(x, y) < f(x, y) \le (f \ominus \breve{B}_2)(x, y) \\ f(x, y) & \text{otherwise.} \end{cases}$$

The ricefield transformation is specifically based on the sequential homotopic thinning algorithm shown in Algorithm 1. The input image is successively thinned using a sequence of eight pairs of structuring elements. For each \mathbf{B}_i shown in Algorithm 1,



Figure 4.4: Illustration of the segmentation algorithm of Meyer (1978). (a) Nucleus of a Papanicolaou-stained cervical cell with a disk structuring element superimposed at the top left. (b) Binary mask of the nucleus. (c) Grey-scale closing $\epsilon_B \delta_B$ with the disk structuring element *B* shown in (a). (d) Inverted top-hat—closing minus the original image—within the nucleus mask. (e) Thresholding of the top-hat image.

• denotes a pixel belonging to the first structuring element of the ordered pair, \circ denotes a pixel belonging to the second structuring element of the ordered pair, and the origin is at the centre¹². Sequential thinning is repeated until idempotence. Application to the original image yields the *lower ricefield* (LR) segmentation. Application to the negative of the original image yields the *upper ricefield* (UR) segmentation¹³. An example of the ricefield transformation is shown in Figure 4.5. Divide lines (grey-scale skeleton) in both the LR and UR delineate flat connected components. In the case of the LR each connected component corresponds to a local minimum of the original image. In the case of the UR each connected component corresponds to a local minimum of a local maximum of the original image. Rodenacker et al. (1983) demonstrate the algorithm on a digitised microscope image of an epithelial cell (both nucleus and cytoplasm). To obtain the portions of the LR and UR segmentations that correspond to chromatin it is necessary to restrict their domains to that of the nucleus. Rodenacker et al. (1983) do not discuss the issue of nucleus segmentation.

The drawbacks of the ricefield transformation algorithm are that:

- The UR and LR segmentations do not correlate visually with what an observer might perceive to be chromatin blobs, particles, or clumps. Instead they define zones (connected components) around the image extrema. "Neighborhood relations between such regions can be derived...[allowing] subsequent mathematical morphological processing on an elevated level of abstraction" (Rodenacker, 1992, p. 35).
- 2. The algorithm is inefficient (computationally expensive and thus slow) because the entire set of image pixels must be scanned at each thinning step.
- 3. The algorithm is sensitive to noise (Rodenacker, 1992, p. 44).

Remark. A comparison of Figures 4.5(b) and (c) with Figures 4.5(d) and (e) respectively shows that the grey-scale skeletons generated by homotopic thinning are very similar to the watershed lines produced by the watershed transform (see Appendix B). Indeed it has been proved that the watersheds correspond to the closed arcs of the grey-scale skeleton produced by homotopic thinning (Vincent & Soille, 1991, p. 586).

¹² The structuring elements used here are those appearing in Rodenacker (1992). They differ slightly from those appearing in Rodenacker et al. (1983).

¹³ Equivalently the UR can be obtained from sequential *thickening*, rather than thinning, of the original image. See Rodenacker (1992) for further details.

Algorithm 1 Pseudocode for the sequential thinning algorithm used in the ricefield transformation.

1: Let

$\mathbf{B}_1 =$	•		。 。, 。	$\mathbf{B}_2 =$	•	• 0	0,	$\mathbf{B}_3 =$	• • •	• 0	• • , •	$\mathbf{B}_4 =$	0	• •	• • ,
$\mathbf{B}_5 =$	0 0 0	•	• • , •	$\mathbf{B}_6 =$	0	0	• ,	$\mathbf{B}_7 =$	0 •	0	∘ ・,	$\mathbf{B}_8 =$	•	0	0

2: Let g be a grey-scale image of the same dimensions as f but taking the value 0 everywhere

3: while $f \neq g$ do 4: g = ffor i = 1 to 8 do 5: $f = \text{THIN}(f, \mathbf{B}_i)$ 6: 7: end for 8: end while



(a)



Figure 4.5: Illustration of the ricefield transformation and its comparison with the watershed transform. (a) Nucleus of a Papanicolaou-stained cervical cell. (b) Lower ricefield. (c) Upper ricefield. (d) Watershed transform of (a) – compare with (b). (e) Watershed transform of the photographic negative of (a) – compare with (c).

Segmentation algorithm of Komitowski & Zinser

The segmentation algorithm devised by Komitowski & Zinser (1985) partitions a nucleus image into regions of high O.D. which the authors call chromatin particles. The images used by Komitowski & Zinser (1985) are digitised photomicrographs of cell nuclei in tissue from rat livers and from the human colon, prepared using three different techniques: aceto-carmine staining of squash preparations, Feulgen staining of squashed tissue fragments fixed in Carnoy's solution, and Feulgen staining of paraffin-embedded sections. As a preprocessing step, the nucleus is isolated¹⁴ in the input image. Komitowski & Zinser (1985, p. 179) state that their chromatin segmentation algorithm is based on a "localization algorithm with a locally adaptive threshold". Further, they state that their method "identifies local maxima of the O.D. within the nuclear images and calculates disjunct, maximally large regions surrounding them" (Komitowski & Zinser, 1985, p. 179). No further detail is given. An example of the segmentation produced by the algorithm is shown in Figure 4.6. The drawbacks of this algorithm are those associated with local adaptive thresholding including:

- 1. the need to prescribe the manner in which the image is initially partitioned into blocks;
- 2. the need to prescribe the manner in which a threshold is determined for each block; and
- 3. sensitivity to noise.

Segmentation algorithm of Kondo & Taniguchi

The segmentation algorithm devised by Kondo & Taniguchi $(1986)^{15}$ partitions a cell nucleus image into regions which the authors call chromatin granules (densely stained DNA proteins). The images used by Kondo & Taniguchi are digitised images of cell nuclei from Pap smears. The algorithm operates only on those pixels that constitute the nucleus. This presupposes that the nucleus has been isolated in the input image, although Kondo & Taniguchi do not give any details of how this is done. The chromatin segmentation algorithm itself comprises three steps: (i) the

¹⁴ Specific details concerning this step can be found in Zinser & Komitowski (1983).

¹⁵ This is a translation of a paper originally published in Japanese in September 1985. Refer to Kondo & Taniguchi (1986) for details.



Figure 4.6: Illustration of the segmentation algorithm of Komitowski & Zinser (1985). Reproduced from Komitowski & Zinser (1985) with the permission of *Analytical and Quantitative Cytology and Histology*. The image marked 'A' is a digitised image of a cell nucleus from a rat liver (paraffin section and Feulgen staining). The image has been preprocessed to isolate the nucleus. The image marked 'B' shows the segmented particles as a binary image superimposed on the image 'A'.

local maxima (with respect to optical density) are identified in the image (these correspond to local minima of intensity); (ii) the input image is partitioned into subimages (regions), each containing a single maximum; and (iii) a chromatin granule (densely stained blob of chromatin) is segmented from each region in turn using local adaptive thresholding. Kondo & Taniguchi propose three different methods for the partitioning step:

1. Partitioning using a Voronoi neighbourhood

The Voronoi neighbourhood of a local maximum comprises all those pixels that are closer to it than to any other local maximum. The union of these neighbourhoods constitutes a complete tessellation of the nucleus. The concept of Voronoi neighbourhoods is treated in depth in Chapter 5.

A drawback of this method is that it does not use the topography of the input image to determine a region around each minimum. Consequently it is possible that the region determined around a minimum cuts through one or more adjacent chromatin particles.

2. Area expansion by difference direction

This method relies on the property that a chromatin granule is densely stained at the centre. The method involves an expansion (region growing) of each local maximum in the direction of lower density. "This expansion is applied only to the pixels which do not have more than one path from more than one maximum... In other words, if there is an ambiguity in the expansion by density difference, this procedure is not applied; the procedure is applied only to the pixels with certainty" (Kondo & Taniguchi, 1986, p. 13). This method is very similar to the region growing step of the algorithm of Smeulders et al. (1978).

A drawback of the density difference method is that the growth is not prescribed by topographic distance (i.e. if the image is viewed as a landscape then the growth is not prescribed by the topography of the landscape).

3. Region partitioning by directed tree

This method is based on a method devised by Narendra & Goldberg (1980) for segmenting natural images (LANDSAT) into relatively uniform regions.

A drawback of the directed tree method is that it is necessary to specify, a priori, a sensitivity parameter to control growth.

The local adaptive thresholding method of segmenting a granule from each region has several potential drawbacks including sensitivity to noise and the need to prescribe the manner in which the threshold value is determined for each sub-image. An example of the segmentation produced by the algorithm is shown in Figure 4.7.

Segmentation algorithm of Young, Verbeek & Mayall

The segmentation algorithm devised by Young et al. (1986) partitions a nucleus image into regions of low, medium, and high optical density. The algorithm presupposes that the nucleus has been isolated in the input image, although Young et al. (1986) do not give any details of how this is done. Young et al. illustrate their method on digitised light microscope images obtained from foam cells in human nipple aspirate fluid, and rat urothelial cells. The method of staining is unspecified. The chromatin segmentation algorithm itself involves nothing more than partitioning the grey-level histogram of the nucleus image into three parts (i.e. choosing two grey values), and then using this division to label each nucleus pixel. The result is a segmentation comprising regions of low, medium, and high optical density. The manner in which the partition points (threshold values) are determined must be specified a priori. Young et al. determine these thresholds based on a fixed percentage of the mean grey-value in the nucleus: one is defined to be this percentage below the mean and the other this percentage above the mean. An example of the



Figure 4.7: Illustration of the segmentation algorithm of Kondo & Taniguchi (1986). Reproduced from Kondo & Taniguchi (1986) with the permission of the publisher. The image quality in the original published article is poor. Top row (from left to right): tessellations produced by the Voronoi neighbourhood approach, the expansion by difference direction approach, and the partitioning by directed tree approach respectively. Bottom row (from left to right): local adaptive thresholding applied to the corresponding images in the top row to yield chromatin granules.

segmentation produced by the algorithm is shown in Figure 4.8. The algorithm has two major drawbacks:

- 1. It is based on global thresholding and is thus sensitive to noise and to nonuniformity of illumination and/or staining; and
- 2. It requires the specification of two threshold values. The manner in which these are chosen must be specified a priori. Moreover they must be tuned to the particular application.

The segmentation algorithm of Madachy & Fu (1988) is identically the algorithm of Young et al. (1986) (although the connection is not made by the authors). Madachy & Fu apply the algorithm to digitised light microscopy images of cells from cervical biopsy tissue. The staining method is not specified.

Doudkine et al. (1995) describe a variation of the algorithm of Young et al. (1986) namely in terms of the manner in which the two thresholds are determined—that is then used to compute a class of texture features the authors call *discrete texture features*. These features are included in the United States Patent of Palcic et al. (2000).



(c)

Figure 4.8: Illustration of the segmentation algorithm of Young et al. (1986). (a) Nucleus of a Papanicolaou-stained cervical cell. (b) Histogram of the grey-values. (c) Segmentation using thresholds based on 80% of the mean and 120% of the mean.

Segmentation algorithm in the CYBEST model 4

CYBEST "is an elaborate series of high-resolution, automated cytology screening systems for uterine cancer detection" (Husain, 1994, p. 9). The most recent model, model 4, is described in Tanaka et al. (1987*a*) and Tanaka et al. (1987*b*). CYBEST takes as input Papanicolaou-stained smears. Details of the nucleus and cytoplasm segmentation methods are given in Tanaka et al. (1987*a*). The methods used date back to the late 1970s and are based on grey-level thresholding. Unlike its predecessors, CYBEST model 4 additionally segments the chromatin into (dark) granules. No specific details of the algorithm used to segment the chromatin are given. However, a diagram that appears in both papers suggests that the algorithm is based on thresholding.

Segmentation algorithm of Beil

The segmentation algorithm devised by Beil (1992) is similar to the algorithm of Young et al. (1986) in that it segments a nucleus image into three types of region. However, rather than segmenting purely on the basis of grey-level, Beil's algorithm seeks to partition the image into three topographic features: mountain, valley, and slope. The images used by Beil are not specified. In the words of Beil (1992, p. 131), the algorithm

first... detects all edges between two neighbouring points P_i which are higher than a constant value. These points P_i represent origins for a region growing. The region growing procedure uses local operators for the assignment of the image points to regions. After harmonizing inconsistencies we obtain an image containing three texton classes.

Like the algorithm of Smeulders et al. (1978), this algorithm consists of a seeded region growing step followed by a merging step. An example of the segmentation produced by Beil's algorithm is shown in Figure 4.9. The major drawback of the algorithm is that the seed points are selected by thresholding. This is likely to be sensitive to noise and to non-uniformity of illumination and/or staining. In addition the process of "harmonizing inconsistencies", though not specified, is likely to involve the comparison of attributes of abutting regions. The decision to merge two regions is likely to be based on a threshold which must be specified a priori.



Figure 4.9: Illustration of the segmentation algorithm of Beil (1992). Reproduced from Beil (1992) with the permission of *Image Analysis & Stereology (formerly Acta Stereologica)*. From left to right: original image, result after an edge oriented segmentation, gradient line representation of the original image.

Segmentation algorithm of Wolf, Beil & Guski

The segmentation algorithm of Wolf et al. (1995) partitions a nucleus image into homogeneous regions (textons) of chromatin¹⁶. The images used by Wolf et al. are of cervical cells, from tissue sections obtained by colposcopic biopsies, stained by the Feulgen method. The algorithm is based on the watershed algorithm (see Appendix B) and region merging. The first step involves determining the watershed of the gradient of the nucleus image. This is done using a modification of the classic watershed algorithm of Vincent & Soille (1991). The result is an over-segmentation; i.e. too many regions are delineated and as a consequence the result does not correspond very well to the chromatin patches in the original image. The second step involves selectively merging the regions segmented in the first step. Specifically, this step involves fitting a plane to each segmented region using standard least-squares techniques—the "plane is defined by ax+by+c = z, with (x, y) the pixel position and z its gray level" (Wolf et al., 1995, p. 2)—and then iteratively merging neighbouring regions based on merging criteria related to the standard deviation of grey-levels in regions. The decision to merge two regions is based on the evaluation of a single parameter which is then compared to a predefined threshold. An example of the segmentation produced by the algorithm is shown in Figure 4.10. The major drawback of the algorithm is that a threshold value must be specified a priori. This is likely to be sensitive to noise and to non-uniformity of illumination and/or staining.

 $^{^{16}}$ The algorithm is also described in a subsequent paper by Beil et al. (1995).



Figure 4.10: Illustration of the segmentation algorithm of Wolf et al. (1995). Reproduced from Wolf et al. (1995) with the permission of *Analytical and Quantitative Cytology* and *Histology*. From left to right: original image, watershed of the gradient, and result after region merging.

Segmentation algorithm of Albregtsen, Schulerud & Yang

The segmentation algorithm of Albregtsen et al. (1995) does not partition the nucleus image into disjoint regions, but rather into overlapping regions. At each pixel position, it segments "a region of consistent connected neighbouring pixels..., forming a local textel of pixels belonging to the same gray level population" (Albregtsen et al., 1995, p. 496). Features are computed for the region and then the region is discarded. The image data used by Albregtsen et al. are digitised transmission electron *microscope* images, at a primary magnification of 2500, of Feulgen stained mouse liver cells. As a preprocessing step each image is 3×3 median filtered to remove noise and the nucleus is isolated by manual tracing. The segmentation algorithm itself is based on a modification of the K-Nearest Connected Neighbours (KNCN) filter concept of Lønnestad (1988). The algorithm is applied to each pixel in turn to generate a connected component of relatively homogeneous grey-levels. Although Albregtsen et al. only describe their algorithm in words, it can be expressed as pseudocode as shown in Algorithm 2. The constants K_{\min} , K_{\max} , and σ_{\min} must be specified a priori. The constant K_{\min} prescribes the minimum size that the region can have. The constant $K_{\rm max}$ prescribes an upper limit on the size that the region can grow. The constant σ_{\min} prescribes a lower limit on σ which is needed to ensure that the region will grow in strictly homogeneous areas. The algorithm has two major drawbacks:

- 1. The constants K_{\min} , K_{\max} , and σ_{\min} must be specified a priori and must be tuned to the particular application.
- 2. The algorithm is not scan order independent: the order in which unselected neighbours are examined can influence the size and shape of the resulting

Alg	orithm 2 Pseudocode for the algorithm of Albregtsen et al. (1995).
1: \$	select the pixel at (x, y)
2: 1	for $i = 1$ to K_{\min} do
3:	examine all of the unselected neighbours of the selected $pixel(s)$
4:	select the pixel that is closest in grey-value to the pixel at (x, y)
5: 0	end for
6: 0	compute the mean μ and the standard deviation σ of the K_{\min} selected pixels
7: i	${\rm if}\sigma<\sigma_{\rm min}{\rm then}$
8:	$\sigma = \sigma_{\min}$
9: 0	end if
10: 7	i = 1
11: 1	repeat
12:	i = i + 1
13:	examine all of the unselected neighbours of the selected $pixel(s)$
14:	if the pixel that is closest in grey-value to μ , has a grey value that is within 2σ of
	$\mu ~ {f then}$
15:	select the pixel and update μ and σ
16:	$ {\bf if} \sigma < \sigma_{\min} {\bf then} $
17:	$\sigma=\sigma_{ m min}$
18:	end if
19:	end if
20: 1	until $i = K_{\text{max}}$ or no pixel was selected

region.

A 1

Segmentation algorithm of Walker & Jackway

Walker & Jackway (1996) used the statistical geometrical features (SGF) method of Chen et al. (1995) to quantitatively characterise the chromatin in the nuclei of Papanicolaou-stained cervical cells from cervical slides prepared using the Thin-Prep[™] technique. The SGF method involves the computation of statistics "of geometrical attributes of connected regions in a sequence of binary images obtained from a texture image" (Chen et al., 1995, p. 537). Specifically, the first step of the SGF method involves thresholding the original image for each discrete grey-level to generate a stack of binary images. The main drawback of this approach is that the result is a stack of binary images rather than a single partitioning of the grey-scale image.

Jones & Jackway (2000) introduced a novel texture representation which they call granolds (the term is derived from granulometry of thresholds). Jones (2001, Chapter 6) uses the technique to quantitatively characterise the chromatin in nuclei of Papanicolaou-stained cervical cells from cervical slides prepared using the Thin-PrepTM technique (the results are discussed in Section 6.4.9.1). Like the SGF method, the first step involves thresholding the original image for each discrete grey-level to generate a stack of binary images.

Segmentation algorithm of Jackway

The segmentation algorithm of Jackway (1996) yields a number of segmentations of the nucleus image across different scales. Jackway illustrates the method on an image of a Papanicolaou-stained cell from a cervical smear. The multiscale images are produced by the *multiscale-morphological-dilation-erosion* defined for $f : \mathbb{R}^2 \to \mathbb{R}$ as

$$(f \circledast g_{\sigma})(x, y) = \begin{cases} (f \oplus g_{\sigma})(x, y), & \text{if } \sigma > 0\\ f(x, y), & \text{if } \sigma = 0\\ (f \oplus g_{\sigma})(x, y) & \text{if } \sigma < 0 \end{cases}$$

where $g_{\sigma} : \mathbb{R}^2 \to \mathbb{R}$ is defined

$$g_{\sigma}(x,y) = |\sigma| g\left(|\sigma|^{-1} x, |\sigma|^{-1} y \right)$$
 for all $\sigma \neq 0$

and where g(x, y) is a non-positive, anticonvex, even function for all $(x, y) \in \mathbb{R}$ with g(0, 0) = 0; e.g. $g(x, y) = -(x^2 + y^2)$. A slightly modified¹⁷, but equivalent, description of the algorithm is given in Algorithm 3 (the notions of regional minima and maxima are explained in Definition 4.4.1 and Definition 4.4.2 respectively).

The Jackway (1996) algorithm has two major drawbacks:

- 1. A set of scales must be specified a priori; and
- 2. The segmentation at each scale does not correlate visually with what an observer might perceive to be chromatin blobs, particles, or clumps.

¹⁷ The original algorithm specifies the use of homotopy modification of the gradient such that only certain minima are retained. This modified gradient is then segmented using the classical watershed transform (see Appendix B). However, the marker-based version of the watershed algorithm permits the imposition of minima and the computation of the watersheds to be combined in a single algorithm (Soille, 2003, p. 281)

Algorithm 3	Pseuc	locode	for [·]	the	Jackwa	y (1	l996) a	algoritl	nm.
-------------	-------	--------	------------------	-----	--------	------	---------	----------	-----

1: select a set of scales $\{\sigma_k\}$ of interest; e.g. $\{-1.6, -0.9, -0.4, -0.1, 0^-, 0^+, 0.1, 0.4, 0.9, 1.6\}$

2: for each σ_k do

- 3: compute $f \circledast g_{\sigma_k}$, where f is the input image
- 4: locate the regional minima $\{N_i\}$ (for $\sigma_k \leq 0$) or the regional maxima $\{M_i\}$ (for $\sigma_k \geq 0$) of $f \circledast g_{\sigma_k}$
- 5: compute the magnitude of the gradient $|\nabla (f \circledast g_{\sigma_k})|$
- 6: compute the watershed of the gradient using $\{N_i\} \cup \{M_i\}$ as markers
- 7: end for

4.3.1 Summary and conclusion

As shown in Table 4.2, the algorithms reviewed can be categorised according to the underlying segmentation method(s) used: global thresholding, top-hat transform, grey-scale thinning, local adaptive thresholding, and region growing/merging. The global thresholding algorithms range from the single threshold approach of Sprenger et al. (1973) through to the complete threshold decomposition approach of Walker & Jackway (1996). Local adaptive thresholding is used outright by Komitowski & Zinser (1985), and as a secondary step in the algorithm of Kondo & Taniguchi (1986). The top-hat transform approaches include the single top-hat of Meyer (1978), and the multiple top-hat approach of Giménez-Mas et al. (1995). The only approach based on grey-scale thinning is that of Rodenacker et al. (1983). Both Kondo & Taniguchi (1986) and Smeulders et al. (1978) use region growing to determine regions-of-interest around local maxima (with respect to optical density). Beil (1992) uses region growing to grow seed pixels into regions belonging to three texton classes. Wolf et al. (1995) use region growing—specifically the watershed transform of the gradient image—to segment the image into candidate regions for region merging. Albregtsen et al. (1995) uses region growing to grow a local textel of pixels from a given pixel. Jackway (1996) uses region growing—specifically the watershed of a homotopically modified gradient—multiple times to segment a set of multi-scale images generated from a multi-scale decomposition of the original image.

A criticism of the threshold decomposition, grey-scale thinning, and multi-scale decomposition algorithms is that they do not segment the chromatin into regions that correspond visually with what a human observer would perceive to be blobs, particles, or clumps. In the case of threshold decomposition, the result is a stack of binary images each of which contains connected components (see Definition 2.9.5). In the case of grey-scale thinning the result is a grey-scale image of flat connected components delineated by divide lines. In the case of the multi-scale decomposition algorithm the result comprises several sets of watershed regions. Moreover, "the watershed arcs move spatially with varying scale and are not a subset of those at zero scale" (Jackway, 1996, p. 913).

A criticism of the global thresholding algorithms is that they utilise only the greylevel histogram and thus make no use of spatial information. Worse still, as Figure 4.8 illustrates, it is not even obvious how the grey-level histogram should be partitioned; i.e. what the threshold value(s) should be. Moreover, under conditions of uneven and variable illumination and/or staining, and the presence of noise, thresholding is generally unsatisfactory.

The local adaptive thresholding algorithms are an improvement over the global thresholding methods in that local pixel properties are utilised. In the case of the algorithm by Kondo & Taniguchi (1986), for example, the nucleus image is first partitioned into subimages, and global thresholding is applied to each in turn. The threshold value for each subimage is derived from the minimum and maximum grey-values in the grey-level histogram for the subimage. However, the problem remains that the manner in which the threshold value is determined must be prescribed. Indeed, Kondo & Taniguchi (1986, p. 18) state that thresholding "must be performed experimentally for various subimages to determine different optimum thresholds".

The algorithm of Meyer (1978) is an improvement over the global threshold methods in that both geometric and grey-scale information are utilised. However, the algorithm (top-hat transform) requires the specification of both a grey-level threshold and the size of the structuring element.

The region growing algorithms seek to grow seed pixels/regions until the entire image has been partitioned. Smeulders et al. (1978) and Kondo & Taniguchi (1986) initiate the region growing from local maxima (with respect to optical density). Beil (1992) initiates region growing from a set of seed points. Albregtsen et al. (1995) initiates region growing, independently, from each image pixel. Wolf et al. (1995) use the watershed transform to grow regions about the regional minima in the gradient image. Jackway (1996) uses the watershed transform to grow regions about the regional minima of homotopically modified gradient images. In each case, however, the algorithm has one or more drawbacks (already outlined) that affect the quality/robustness of the segmentation.

The grey-scale thinning algorithm of Rodenacker et al. (1983) is similar to the region growing algorithms in that the resulting regions correspond to "topological properties like path connectedness and local extrema" (Rodenacker, 1992, p. 35). However, as outlined above, the algorithm does not yield regions that correlate visually with what an observer might perceive to be chromatin blobs, particles, or clumps. Of all the approaches to chromatin segmentation, thresholding appears to be the most popular. Indeed global thresholding is basis for the discrete texture features documented in the United States Patent of Palcic et al. (2000). These features are implemented in the Cyto-SavantTM image cytometer¹⁸. This system has been used in several studies into malignancy associated changes (MACs); e.g. Mairinger et al. (1999), Hanselaar et al. (1998), Ikeda et al. (1998), Anderson et al. (1997), Poulin et al. (1995), and Guillaud et al. (1995). The popularity of thresholding is likely due to its simplicity. However, in view of the criticisms above, as well as the existence of more sophisticated region growing algorithms, it is surprising that thresholding predominates.

A characteristic that all, but the threshold decomposition algorithms and the greyscale thinning algorithm, have in common is that one or more operational parameters must be specified: e.g. threshold values, region merging criteria, a set of scales. Moreover, these parameters need to be tuned to the particular application. As a consequence none of these methods are robust to changes in, or non-uniformity of, illumination and staining. The quality of the resulting segmentations produced is therefore questionable. This in turn affects the quality of any features subsequently computed from these segmentations. In the next section a new algorithm is presented for chromatin segmentation. In its preferred embodiment the algorithm is parameter-free.

4.4 New algorithm for chromatin segmentation

This section presents a new algorithm for chromatin segmentation devised by the author. The method is the subject of an International Patent Application (Mehnert & Jackway, 2002) filed by Fisher Adams Kelly¹⁹ on 19 July 2002 on the behalf of the CSSIP. The author is the principal inventor. The application claims priority from an Australian Provisional Application (Mehnert & Jackway, 2001) filed on 19 July 2001.

4.4.1 Rationale

Jain et al. (1998, p. 110) state that:

¹⁸ The original system was marketed by Oncometrics Imaging Inc. Oncometrics was acquired by AccuMed International Inc. in 1998 and the system was renamed the AcCell-SAVANT. AccuMed in turn was acquired by Ampersand Medical Corporation in February 2001.

¹⁹ Fisher Adams Kelly Patent & Trademark Attorneys, Brisbane, Australia.

 Table 4.2: Classification of chromatin segmentation algorithms published in the literature.

Global	Top-hat	Grey-scale	Local	Region grow-
${\it thresholding}$	transform	thinning	adaptive	ing/merging
			${f thresholding}$	
Rowiński et al.	Meyer (1978),	Rodenacker et	Komitowski &	Smeulders et al.
(1972), Klawe &	Meyer (1980),	al. (1983),	Zinser (1985)	(1978),
Rowiński (1974)	Giménez-Mas et	Rodenacker		Smeulders et al.
	al. (1995)	(1992)		(1979)
Sprenger et al.			Kondo & Tar	niguchi (1986)
(1973), de Cam-				
pos Vidal et al.				
(1973),				
Sandritter et al.				
(1974), Al et al.				
(1978),				
Danielsen et al.				
(1989)				
Young et al.				Beil (1992)
(1986),				
Madachy & Fu				
(1988),				
Doudkine et al.				
(1995), Palcic et				
al. (2000)				
Walker &				Wolf et al.
Jackway (1996),				(1995), Beil et
Jones (2001)				al. (1995)
				Albregtsen et al.
				(1995)
				Jackway (1996)

a model-free or structure-free image interpretation approach is doomed by the underconstrained nature of the problem. Imperfect image data can be augmented with extrinsic information such as geometrical models of the objects that are likely to be present in the scene in order to facilitate object recognition.

The discussion in Section 4.2.3 suggests that chromatin structure, as visualised by light microscopy, can be described in terms of:

- 1. regions of high optical density that are blob-like, particle-like, or granule-like in appearance;
- 2. regions of low optical density or clearing (which can also be considered as blob-like);
- 3. homogeneous background sap; and
- 4. the topographic arrangement of the blobs.

Although Koss (1992) states that under favourable conditions the linin network can also be seen, in the experience of the author this is rarely true for the images captured using the Cytometrics Project cytometer²⁰. For this reason the proposed algorithm is designed only to segment the dark and/or light blobs in a grey-level image of a cell nucleus. These grey-levels may represent either intensity or optical density. If the grey-scale image is viewed as a topographic relief, as depicted in Figure 4.11, then the light particles correspond to mountains and the dark particles correspond to valleys²¹. The dark particles are associated with the minima and the light particles are associated with the maxima. The algorithms of Smeulders et al. (1978), Komitowski & Zinser (1985), and Kondo & Taniguchi (1986) determine a single dark particle for each local maximum (with respect to optical density). In the proposed algorithm a dark particle is determined for each regional minimum (with respect to grey-level). The difference between a regional minimum and a local minimum is that a regional minimum is a plateau of pixels that is surrounded by pixels of higher grey-level, whilst a local minimum is a single pixel that is surrounded

²⁰ The cytometer uses a CCD camera (with square pixels of side length 14μ m) coupled to a light microscope (fitted with a $40\times$ objective lens with a numerical aperture of 0.75) to acquire 8bit intensity images of Papanicolaou-stained cells. After digitisation, a typical nucleus image comprises around 2000 pixels. See Section 6.4.4 for more details.

²¹ This topographical description coincides with the "mountain, valley, and slope" characterisation of chromatin structure proffered by Beil (1992).

by pixels of greater or equal grey-level. Formal definitions of local and regional extrema are as follows.

Definition 4.4.1 (local and regional minima). Let $f \in Fun(\mathbb{Z}^n, \mathcal{T})$ be a greyscale image (see Section 2.8), then:

1. a pixel x is called a *local minimum* of f if

$$\forall y \in \delta^1\left(\{x\}\right), \ f\left(y\right) \ge f\left(x\right);$$

2. a pixel x is called a *strict local minimum* of f if

 $\forall y\in \delta^{1}\left(\left\{ x\right\} \right) \backslash\left\{ x\right\} ,\ f\left(y\right)>f\left(x\right);\text{ and }$

3. a set of pixels M is called a *regional minimum* of f if

$$\begin{cases} \forall x \in M, & f(x) = t, \\ \forall y \in \delta^1(M) \setminus M, & f(y) > t. \end{cases}$$

Remarks.

- 1. A strict local minimum is a regional minimum that consists of a single pixel.
- 2. δ^1 is a metric dilation of size 1 (see Definition 2.9.3).

Definition 4.4.2 (local and regional maxima). Let $f \in Fun(\mathbb{Z}^n, \mathcal{T})$ be a greyscale image (see Section 2.8), then:

1. A pixel x is called a *local maximum* of f if

$$\forall y \in \delta^1(\{x\}), \ f(y) \le f(x).$$

2. A pixel x is called a *strict local maximum* of f if

$$\forall y \in \delta^1\left(\{x\}\right) \setminus \{x\}, \ f(y) < f(x).$$

3. A set of pixels M is called a *regional maximum* of f if

$$\left\{ \begin{array}{ll} \forall x \in M, & f\left(x\right) = t, \\ \forall y \in \delta^{1}\left(M\right) \backslash M, & f\left(y\right) < t. \end{array} \right.$$



Figure 4.11: Representing a grey-scale image as a topographic relief. (a) Grey-scale image. (b) Topographic surface generated by considering each grey-value to be a height.

Remark. A strict local maximum is a regional maximum that consists of a single pixel.

The proposed algorithm is similar to that devised by Kondo & Taniguchi (1986) in the sense that it involves the steps of locating minima in the grey-scale image (which, if the grey-values represent intensity, correspond to maxima in the optical density image), partitioning the image into regions each containing a single minimum, and then segmenting a single blob in each region. However, the new algorithm differs in several respects:

- 1. It locates regional minima with respect to grey-level, rather than local maxima with respect to optical density;
- 2. In its preferred embodiment, it grows these minima using the watershed transform (see Appendix B). If the input image is viewed as a topographic landscape then the boundaries of the regions determined by the watershed transform (the watersheds) are guaranteed to lie between the minima as determined by the topography of the surface; and
- 3. In its preferred embodiment, it segments a single blob in each region using the watershed transform rather than local adaptive thresholding.

4.4.2 Description of the new algorithm

A flowchart of the steps that make up the new algorithm is shown in Figure 4.12. The input image is denoted by f on the flowchart. It is a grey-scale image²² in which the chromatin is visualised as a patchwork of light and dark regions. The first, optional, step is preprocessing. Depending on the quality of f, this step may include filtering to attenuate noise, deconvolution to correct for lack-of-focus, and up-sampling to facilitate line rendering in subsequent steps. The preprocessed image is denoted f' on the flowchart. The next step is to locate the regional minima in the preprocessed image f'. The output from this step is a binary image Mcontaining connected components, each of which marks the location of a regional minimum. Several methods for detecting regional minima exist in the literature; e.g. the grey-scale reconstruction method of Vincent (1993, p. 184). The next, optional, step is to filter the image M according to a priori specified contrast criteria. This involves computing a contrast valuation, with respect to the preprocessed image f', for each regional minimum, and then discarding those minima that do not satisfy the contrast criteria. The filtered image is denoted M' on the flowchart. Two particularly useful contrast measures are *dynamics* devised by Grimaud (1992) and symmetrical dynamics devised by Vachier & Vincent (1995). The next step is to compute a zone of influence (ZOI) around each of the connected components in M'. The image containing the ZOIs is denoted by Z on the flowchart. Depending on the method of implementation, Z may be either a binary image of lines delineating the ZOIs or it may be a grey-scale image in which each ZOI has its own unique numerical label. Several different methods can be used to compute Z:

- 1. Application of the watershed transform to the preprocessed image f' using the connected components of M' as markers.
- 2. Application of a scan-order-independent seeded region growing algorithm such as that of Mehnert & Jackway (1997) or Beare & Talbot (1999)—to the preprocessed image f' using the connected components of M' as seeds.
- 3. Computation of the influence zone (IZ) around each connected component of M' using an a priori specified metric. The IZ of a connected component is the set of all pixels that are closer to it than to any other connected component (Soille, 2003, p. 170). See also Definition 5.6.1.

²² The grey-values may be optical density values or intensity values. The algorithm segments dark (with respect to grey-level) blobs. To segment light blobs the algorithm can be applied to the photographic negative (see Section 4.4.4) of the grey-scale image.

The next step is to segment a single blob (chromatin particle) within each ZOI in Z. This is done using a region growing procedure. Two possibilities are:

- 1. Application of the watershed transform to the modulus of the gradient of the preprocessed image f' using both the connected components of M' and the boundary lines of Z as markers (these boundary lines may or may not include the boundary of the entire nucleus itself). Numerous methods for computing the modulus of the gradient exist in the literature; e.g. Rivest et al. (1993).
- 2. Application of a scan-order-independent seeded region growing algorithm to the preprocessed image f' using both the connected components of M' and the boundary lines of Z as seeds (these boundary lines may or may not include the boundary of the entire nucleus itself).

Segmentations produced by two variants of the algorithm, including the preferred embodiment described next, are shown in Figure 4.13.

4.4.3 Preferred embodiment

The preferred embodiment of the new algorithm is as follows:

- The input image f is preprocessed. This involves the application of a 3 × 3 median filter (Gonzalez & Woods, 1992, p. 191) followed by up-sampling by factor 3. The method of up-sampling by factor 3 involves nothing more than replacing each pixel with a 3 × 3 block of pixels of the same grey-value.
- 2. The regional minima are identified in the preprocessed image f'.
- 3. The dynamics of these minima are computed but are not used to filter out unwanted minima, but rather are retained for subsequent use as chromatin features. Consequently M' = M.
- 4. The watershed transform is applied to the preprocessed image f' using the connected components of M' as markers. The divide lines define Z.
- 5. The magnitude of the gradient of f' is estimated using Beucher's gradient (Soille, 2003, p. 85):

$$\left|\nabla f'\right| \approx \delta_B\left(f'\right) - \varepsilon_B\left(f'\right)$$

where B is the unit ball for the grid on which f' is manifest.



Figure 4.12: Flowchart showing the steps of the new chromatin segmentation algorithm.



Figure 4.13: Demonstration of the new chromatin segmentation algorithm. (a) Isolated nucleus image (Papanicolaou stain) after the application of a 3×3 median filter and upsampling by factor 3. (b) Regional minima (superimposed in white). (c) Watershed transform of (a) using the regional minima as markers. (d) Beucher's gradient of (a). (e) Watershed transform of (d) using both the regional minima in (b) and the watersheds of (c) as markers. (f) Alternative: improved seeded region growing algorithm of Mehnert & Jackway (1997) applied to (a) using both the regional minima in (b) and the watersheds of (c) as seeds.



Figure 4.14: Segmentation of dark particles using the preferred embodiment of the proposed chromatin segmentation algorithm.

6. The watershed transform is applied to the morphological gradient of the image f' using the connected components of M' and the divide lines of Z as markers.

An implementation of this algorithm, in DImPAL (see Appendix A), is shown in Appendix D. A sample of segmentations produced by the algorithm is shown in Figure 4.14.

4.4.4 Discussion

If the input image is replaced with its photographic negative²³, the dark particles segmented by the proposed algorithm will correspond to the light particles in the original image as shown in Figure 4.15. Consequently the algorithm can be used to segment a nucleus image into dark particles, light particles, and background. For

²³ For an 8-bit grey-scale image, with grey-values ranging from 0 to 255, the photographic negative is obtained by subtracting each pixel value from 255.



Figure 4.15: Detection of light particles. (a) Original image (after noise removal and up-sampling). (b) Photographic negative (after noise removal and up-sampling). (c) Segmentation of the photographic negative (superimposed on the photographic positive).

an intensity image these correspond to condensed chromatin, areas of chromatin clearing, and nuclear sap respectively.

The proposed algorithm can, in principle, be generalised to higher dimensional images, and to multispectral and other multi-valued images. This is possible because the seeded region growing algorithms can be extended to such images. A threedimensional version of the watershed transform also exists: Cotsaces & Pitas (1998).

The preferred embodiment of the proposed segmentation algorithm is based on the marker-based watershed transform of Meyer (1991). The watershed transform is robust to slight optical changes. Indeed Wolf et al. (1995, p. 31) experimentally compared co-occurrence texture features with texture features derived from their chromatin segmentation algorithm (which is based on the watershed transform of the gradient) and found that the watershed-based features had "the best robustness against slight optical changes".

The proposed segmentation algorithm is, in it preferred embodiment, a parameterfree method of segmenting chromatin particles. Moreover, in comparison to existing methods, it is the author's opinion that the proposed algorithm produces a discernibly better segmentation of nuclear chromatin; i.e. for any given nucleus image the method yields a segmentation of chromatin particles that corresponds well with what a human observer might intuitively perceive to be blobs or particles²⁴.

Seeded region growing (which includes the marker-based watershed transform) is the cornerstone of the proposed chromatin segmentation algorithm. In the next

²⁴ To verify this claim it would be necessary to undertake a study involving the blind testing of several cytologists or similar experts.

section the classical seeded region growing algorithm of Adams & Bischof (1994) is examined in detail. It is shown that it is inherently dependent on the order of pixel processing. This means, for example, that raster order processing and anti-raster order processing do not, in general, lead to the same segmentation. An improved algorithm is proffered in Section 4.6.

4.5 The Adams and Bischof seeded region growing algorithm

Adams & Bischof (1994) proposed a novel region growing algorithm called seeded region growing (SRG). The algorithm is fast, robust, and parameter free. It takes a grey-level image and a set of seeds—individual pixels or connected components—as inputs. The seeds play the same role as the markers used in watershed segmentation (see Appendix B). They mark each of the objects (regions) to be segmented. The SRG algorithm operates on the premise that the pixels within a region are similar. The algorithm grows the seed regions in an iterative fashion. At each iteration all those pixels that border the growing regions are examined. The pixel that is most similar to a region that it borders is appended to that region. Unfortunately the SRG algorithm is inherently dependent on the order of processing of the image pixels. One implication of this is that raster order processing and anti-raster order processing do not, in general, lead to the same segmentation. This order dependency is particularly evident when the regions are small and of very similar grey value. Order dependency is clearly an undesirable property, especially when the images to be segmented have no obvious orientation (which is the case for nuclear chromatin).

4.5.1 Description of the algorithm

The seeded region growing approach to image segmentation is to partition an image g into regions with respect to a set of n seed regions. Each seed region is a connected component comprising one or more pixels and is represented by a set A_i , where i = 1, 2, ..., n. Let T be the set of all unallocated pixels that border at least one of the A_i , i.e.

$$T = \left\{ x \notin \bigcup_{i=1}^{n} A_i \mid N(x) \cap \bigcup_{i=1}^{n} A_i \neq \emptyset \right\},\tag{4.2}$$

where N(x) represents the set of immediate neighbours—6 for the hexagonal grid and either 4 or 8 for the square grid—of the pixel x (see Figure 2.3). A single step
of the algorithm involves examining the neighbours of each $x \in T$ in turn. If N(x) intersects a region A_j then a measure, $\delta(x)$, of the difference (similarity) between x and the intersected region is calculated²⁵. In the simplest case $\delta(x)$ is defined:

$$\delta(x) = \left| g(x) - \max_{y \in A_j} \{ g(y) \} \right|$$

where g(x) is the grey value of the pixel x. If N(x) intersects more than one region then A_j is taken to be that region for which $\delta(x)$ is a minimum (alternatively, the pixel x can be flagged as a boundary pixel for display purposes). In this way a δ value is determined for each $x \in T$. Finally, the pixel $z \in T$ that satisfies

$$\delta(z) = \min_{x \in T} \{\delta(x)\} \tag{4.3}$$

is appended to the region corresponding to $\delta(z)$. The new state of the regions $\{A_i\}$ then constitute the input to the next iteration. This process continues until all of the image pixels have been assimilated.

4.5.2 Inherent order dependencies

The SRG algorithm has two inherent pixel order dependencies. The first manifests itself whenever, during an iteration, several $x \in T$ determine the same, minimum, δ value. Equation 4.3 then offers several possible choices for z. The particular zchosen influences the running mean of the region that it is assigned to. This in turn influences the δ values calculated for the $x \in T$ in the next iteration, and ultimately affects the final segmentation. This problem is illustrated in Figure 4.16. The second order dependency manifests itself whenever the chosen z has the same δ value for several regions that it borders. Once again resolution of the deadlock ultimately influences the final segmentation (this would be the situation if the centre pixel in Figure 4.16(d) was the first of the five '3's, with a δ value of 2, to be processed).

4.5.3 Implementation order dependencies

In implementing the SRG algorithm Adams and Bischof utilise a data structure called the sequentially sorted list (SSL). In their implementation the SSL is a linked list of pixel addresses, ordered with respect to δ . A pixel can be arbitrarily inserted into the list in the position prescribed by its δ value. However, only the pixel

 $^{^{25}}$ The symbol δ used in this section does not refer to dilation.



Figure 4.16: Order dependency of the SRG algorithm. (a) Grey value test image with four seeds marked (the initial $\{A_i\}$). (b) Each $x \in T$ is shown with its δ value as a superscript numeral. (c) Result after 9 iterations. (d) Result after 13 iterations. (e) Final result assuming that the '3's in (d) are scanned in raster order. (f) Final result assuming that the '3's in (d) are scanned in anti-raster order.

with the smallest δ value can be removed from the SSL. Effectively, the SSL stores the points of the set T ordered according to δ . Adams & Bischof note that their implementation does not update previous entries in the SSL to reflect new differences from a region whose mean has been updated. They state that "this leads to negligible difference in the results, but greatly enhanced speed" (Adams & Bischof, 1994, p. 643). As a consequence, in addition to the pixel order dependencies induced by the SRG algorithm, the SRG implementation is subject to two other pixel order dependencies. The first order dependency manifests itself during the initial process of adding the neighbours of the seed regions to the SSL. In particular, if a pixel borders two or more seed regions it is given a δ value based on its similarity to that seed region which happens to be first in terms of the order of processing of the image pixels. Once inserted into the SSL the pixel position is never updated. The second order dependency manifests itself whenever the neighbours of a newly labelled pixel are added to the SSL. The order in which the neighbours are scanned can affect the δ value assigned to each and hence their ordering within the SSL.



Figure 4.17: ISRG algorithm applied to the test image in Figure 4.16(a). The unshaded squares are unresolved ties. Note that the '3's in Figure 4.16(d) have been shared equally between the top-left and bottom-right regions.

4.6 Improved seeded region growing algorithm

This section proffers an improved seeded region growing algorithm developed by the author. The algorithm was published in *Pattern Recognition Letters*: Mehnert & Jackway (1997). The paper has subsequently been cited in Soille (2003), Hao et al. (2001), Pitkänen (2001), and Tomori et al. (1999).

The first order dependency in the SRG algorithm is eliminated if all of the pixels $x \in T$ that have the same minimum δ value are processed in parallel. This means that no pixel can be labelled, and therefore no region means can be updated, until all other pixels at that priority have been examined. Thus for the situation depicted in Figure 4.16(d) the '3's (with a δ value of 2) must be assigned labels independently of one another. Only once all the labels have been determined are the region means updated. If a pixel cannot be labelled because it is equally similar to two or more adjacent regions (same δ values)—the situation that gives rise to the second pixel order dependency of the SRG algorithm—then it is marked as *tied* and takes no further part in the region growing process. After all the pixels have been labelled, the tied pixels are independently re-examined to see whether or not the ties can be resolved. Any remaining ties can of course be resolved by imposing additional assignment criteria if required: e.g. assigning the tied pixel to the largest region, and failing this, assigning it to the region with the larger mean, and so on. However, this constitutes a post-processing step and is not part of the improved seeded region growing algorithm. The behaviour of the ISRG algorithm is illustrated in Figure 4.17. Parallel processing ensures that all pixels of the same priority are processed on an equal basis. Consequently the '3's in Figure 4.16(d) are shared equally between the top-left region and the bottom-right region.

4.6.1 Implementation of the ISRG

The proposed implementation of the ISRG algorithm utilises an ascending priority queue $(PQ)^{26}$, and several LIFO (last-in, first-out) queues. In contrast to the more familiar LIFO queue, the elements of an ascending priority queue are ordered from smallest to largest. A new element can be inserted arbitrarily but only the smallest element can be removed (Tenenbaum & Augenstein, 1986, p. 181). In the proposed implementation the elements of the PQ are LIFO queues. Each LIFO queue contains pixels at a specific priority; i.e. with the same δ value. When a new pixel is inserted into the PQ it is added to the LIFO queue corresponding to the pixel's δ value. Instead of removing individual pixels from the PQ, the entire LIFO queue corresponding to the smallest δ (highest priority) is removed. This permits the processing of all the pixels at the highest priority at the same time. As each pixel is removed from the highest priority LIFO queue its label is determined and inserted into a LIFO queue of labels (LQ), and the pixel is inserted into a LIFO holding queue (HQ). A pixel's label is determined by examining its neighbours. If those neighbours that possess a region label all have the same label then the pixel is deemed also to have this label. If, however, the pixel is surrounded by neighbours with several different region labels then the pixel is deemed to have the label of the neighbour that determines the minimum δ value. In the event of a tie the pixel is marked as tied and inserted into the priority queue with an infinite (in reality just a very large) δ value. This guarantees that a second attempt at resolving ties is made after all other pixels have been labelled. Any pixels that are still tied remain unclassified (the ties can be resolved as a post-processing step if required). Once all of the pixels in the highest priority LIFO queue have been processed, they are labelled. This is done by successively removing a label and a pixel from the LQ and HQ respectively, and assigning the label to the pixel. Thus all the pixels at the highest priority are labelled independently. Ideally previous entries in the PQ should be updated whenever the region means are updated. As a compromise between not updating any of the of the previous entries (as is the case with the SRG implementation) and updating all of the entries (computationally expensive) the proposed implementation does the following. As each pixel is labelled, any of its neighbours that are either unlabelled or marked as being in the PQ are inserted—once only into a LIFO neighbours holding queue (NHQ). After labelling has completed each pixel is successively removed from the NHQ and inserted into the PQ. Thus a single pixel can be inserted into the PQ more than once. However, in the event that a pixel

²⁶ Breen & Monro (1994) discuss and evaluate different data structures for implementing a priority queue.



Figure 4.18: Where two or more regions abut an arbitrary decision has to be made as to which region's border pixel should be flagged as a boundary pixel.

removed from the highest priority LIFO queue—removed from the PQ—already has a region label then it is not relabelled.

The pseudocode for the proposed implementation is shown in Algorithm 4.

4.6.2 A comment on boundary flagging

The ISRG algorithm produces an image in which each individual region has a unique numeric label. If desired, for display purposes, it is a relatively straight forward task to flag a single pixel wide boundary between regions. This can be done either within the algorithm or as a post-processing step. Unresolved ties are of course natural candidates for boundary flagging. Where two or more regions abut, however, an arbitrary decision has to be made as to which region's border pixel should be flagged as a boundary pixel (see Figure 4.18). This introduces an order dependency. The implication of this is that there may be slight discrepancies in the borders produced by a raster order processing as opposed to an anti-raster order processing.

4.6.3 Implementing a scan-order independent version of the watershed transform

The classic marker-based versions of the watershed transform (Meyer, 1991; Beucher & Meyer, 1993) can be seen as a particular case of seeded region growing: the seeds are the markers and the difference measure δ is simply

$$\delta\left(x\right) = g\left(x\right).$$

Like the SRG algorithm the classic (marker-based) watershed algorithms are scanorder dependent (Dobrin et al., 1994, p. 215). However, the ISRG algorithm can be used to implement a scan-order independent version. This involves changing the definition for DELTA to $DELTA = \text{grey_value}(pixel)$ and removing "or has the label $IN_PRIORITY_QUEUE$ " from the *if* statement on line 39 of Algorithm 4. Moreover it is no longer necessary to maintain $REGION_MEAN[]$. The behaviour of this scan-order independent watershed algorithm is illustrated in Figure 4.19

Algorithm 4 Improved seeded region growing. {see Table 4.3 for a description of the data structures and labels} 1: assign a unique label to each seed region and initialise *REGION_MEAN*[] 2: add the pixels neighbouring the seed regions to the NHQ and label them as IN_QUEUE 3: while the PQ is not empty or the NHQ is not empty do while the NHQ is not empty do 4: remove pixel from the NHQ5:6: examine all of its neighbours to find the minimum DELTAinsert *pixel* into the PQ, with a priority equal to the minimum DELTA, and 7: label it as IN_PRIORITY_QUEUE end while 8: if the PQ is not empty then 9: 10: remove FQ from the PQwhile the FQ is not empty do 11: remove *pixel* from the FQ12:if *pixel* has the label *IN_PRIORITY_QUEUE* or the label *TIED* then 13:examine all of its neighbours that have a region label 14:15:if they all have the same label then add this label to the LQ16:else 17:18: examine all of its neighbours to find the minimum DELTA if there is no tie then 19:add the corresponding region label to the LQ20:21: else add TIED to the LQ22: if *pixel* is not already labelled *TIED* then 23:insert it into the PQ with a priority of ∞ 24:25:end if end if 26:27:end if add *pixel* to the HQ and label it as IN_QUEUE 28:29:end if end while 30: 31: end if while the HQ is not empty do 32: remove *label* from the LQ33: remove pixel from the HQ34: 35: assign *label* to *pixel* if the *label* is not *TIED* then 36: update REGION_MEAN[label] 37: examine all of the pixel's neighbours 38: if a neighbour is unlabelled or has the label IN_PRIORITY_QUEUE then 39:add it to the NHQ and label it as IN_QUEUE 40: 41: end if end if 42: end while 43:

44: end while

 Table 4.3: Data structures and labels used in the ISRG.

PQ	
	Ascending <i>priority queue</i> : list of LIFO queues each of which con- tains pixels of a particular priority <i>DELTA</i> .
DELTA	
	Difference (similarity) measure between a pixel and a region that it neighbours; e.g.
	$DELTA = \text{grey_value}(pixel) - REGION_MEAN[j] ,$
	where $REGION_MEAN[j]$ is an array used to hold the running mean grey value for the <i>j</i> -th region.
FQ	
	First (highest priority) queue in the PQ .
NHQ	
	<i>Neighbours holding queue</i> : used to hold pixels that neighbour one or more regions.
HQ	
	Holding queue: used to accumulate pixels removed from the NHQ .
LQ	
	Labels queue: used to hold region labels corresponding to the pixels in the HQ .
TIED	
	Label assigned to a tied pixel.
IN_QUEUE	
	Label assigned to a pixel that is in the HQ or NHQ .
IN_PRIORITY_QUEUE	
	Label assigned to a pixel that is in the PQ .



Figure 4.19: Demonstration of the scan-order independent watershed transform. The first row shows four rotations of a test image. The test image contains four regions of constant grey-level: 50, 100, 200, 250. The three regions in the test image with the lowest intensity are the regional minima. The second row shows the result after applying the scan-order independent watershed transform (ties are shown in black) using the regional minima as markers. The third and fourth rows show the result after applying the two versions of Meyer's algorithm (implementation is based on the pseudocode in Dobrin et al. (1994)) using the regional minima as markers.

4.6.4 Summary and conclusion

The proposed improved seeded region growing algorithm offers the same benefits as the algorithm proposed by Adams & Bischof (1994) but with the added advantage of pixel order independence. The algorithm was initially developed in response to an early attempt by the author to accurately segment the chromatin in images of cell nuclei. The limited pixel resolution of these images—e.g. Figure. 4.20(a) is 88H \times 85V pixels—and the similarity of adjacent chromatin clumps meant that the SRG algorithm produced markedly different segmentations for raster order and anti-raster order processing (see Figure 4.20(c),(d)). In contrast, the proposed ISRG algorithm produces a consistent segmentation because it is not dependent on the order of pixel processing (see Figure 4.20(e),(f)). The added complexity of the ISRG algorithm, as compared to the SRG algorithm, is reflected in increased execution time when implemented on a computer. As part of this research the SRG algorithm and the ISRG algorithm were implemented in C on a DEC3000 workstation by the author. To counter floating point imprecision a threshold difference was defined below which two δ values are deemed to be the same. The priority queue was implemented using a binary tree. For a 256×256 8-bit test image and four small seeds, the execution time for the SRG implementation was 3 seconds whilst that for the ISRG implementation was 15 seconds. Further optimisation of the ISRG implementation is possible. For example, according to Breen & Monro (1994), the speed performance of the priority queue—and hence the ISRG implementation—is improved if a SplayQ data structure is used.

4.7 A new fast priority queue for watershed segmentation

This section presents a new ascending priority queue implementation developed by the author that is suitable for implementing the marker-based watershed algorithms of Meyer (1991). The advantage of the implementation is that it is based on several static memory arrays and does not involve multiple dynamic memory allocations and deallocations. This means that the implementation is fast.

Like the ISRG algorithm, the classic marker-based watershed algorithms of Meyer (1991) (see Appendix B) are based on a priority queue of queues. In the case of the ISRG algorithm, the elements of the priority queue are LIFO queues, whilst in the case of the watershed algorithms the elements are FIFO (first-in first-out) queues.



(a)

(b)

(c)



Figure 4.20: Order independence of the ISRG implementation. (a) High resolution micrograph of the nucleus of a cell. (b) Seed regions (local extrema). (c) SRG implementation applied to (a) in raster order. (d) SRG implementation applied to (a) in anti-raster order. A careful comparison with (c) reveals significant differences. (e) ISRG implementation (boundary/tied pixels shown in white) applied to (a) in raster order. (f) ISRG implementation applied to (a) in anti-raster order. Minor discrepancies between (e) and (f) are an artefact of boundary flagging.

LIFO and FIFO queues are dynamic data structures and as a consequence the priority queue of queues is itself a dynamic data structure. The implementation of such data structures in software involves dynamic memory allocation and deallocation. For example, when a new element is added to a LIFO queue, a block of memory is allocated to store the element and the memory address (pointer) of the preceding block of memory (holding the preceding element of the queue). The overhead associated with dynamic memory management (usually handled by the underlying operating system) incurs a time penalty. For the ISRG and watershed algorithms, which involve numerous queue operations, this affects execution speed.

The priority queue used in the marker-based watershed algorithms differs from that used in the ISRG in two respects:

- 1. the maximum number of elements, i.e. the number of priority levels, in the priority queue is known a priori (equal to the number of distinct grey-levels in the input image); and
- 2. the maximum length of each queue (element) within the priority queue can be determined a priori (from the grey-level histogram).

Consequently, for the marker-based watershed algorithms it is possible to implement a priority queue of queues using a statically allocated block of memory, i.e. the memory is allocated once at the start of processing and deallocated at the end of processing thus eliminating the overhead of multiple memory allocations/deallocations. To process an *n*-bit grey-scale image with N pixels, static memory to hold the following four arrays is needed:

1. heap[]

An array of length N. Each element of the array can store a pixel address (pointer). The array is used to hold the elements of the FIFO queues.

2. head[]

An array of length 2^n . Each element is a number representing an offset into the heap[] array. The first element records the position of the first element (head) of the FIFO queue of priority 0. The second element records the position of the head of the FIFO queue of priority 1 and so on.

3. tail[]

An array of length 2^n . Each element is a number representing an offset into the

Algorithm 5 Pseudocode for initialising the static memory for the priority queue.

Require: heap[], head[], tail[], frequency[] to be indexed from 0**Require:**<math>frequency[] to have been initialised 1: $head[0] \leftarrow frequency[0] - 1$ 2: $rear[0] \leftarrow head[0]$ 3: **for** i in $\{1, 2, ..., 2^n - 1\}$ **do**

4: $head[i] \leftarrow head[i-1] + frequency[i]$ 5: $rear[i] \leftarrow head[i]$ 6: end for

7: $highest_existing_priority \leftarrow 0$

heap[] array. The first element records the position of the last element (tail) of the FIFO queue of priority 0. The second element records the position of the tail of the FIFO queue of priority 1 and so on.

4. frequency[]

An array of length 2^n . The first element is a count of the number of pixels with grey-level 0. The second element is a count of the number of pixels with grey-level 1 and so on.

The pseudocode for initialising this memory for use as a priority queue is shown in Algorithm 5. A schematic showing the state of these arrays after initialisation is shown in Figure 4.21.

As elements are added to the priority queue, the individual FIFO queues are filled right-to-left rather than left-to-right. The reason for this is that, by definition of the watershed algorithms, when a FIFO queue corresponding to the highest priority has been emptied it is suppressed. If, in the future, a pixel of higher priority is added to the priority queue then its address must be added to the highest priority FIFO queue remaining in the priority queue. Thus the size of this highest priority FIFO queue may exceed the size initially allocated for it. However, since the space allocated to the previously suppressed queues is no longer in use, this highest priority FIFO queue can safely grow into the unused space. Pseudocode for adding a pixel to and removing a pixel from the *i*-th FIFO queue is given in Algorithm 6 and Algorithm 7 respectively. Pseudocode for adding a pixel to and removing a pixel from the *a* and Algorithm 9 respectively. An implementation in the C programming language is given in Appendix E.



Figure 4.21: Schematic of the new priority queue for the watershed transform.

Algorithm 6 Pseudocode for adding $pixel_address$ to the *i*-th FIFO queue. $heap[tail[i]] \leftarrow pixel_address$ $tail[i] \leftarrow tail[i] - 1$

Algorithm 7 Pseudocode for removing the next *pixel_address* from the *i*-th FIFO queue.

 $\begin{array}{ll} \mathbf{if} & head[i] = tail[i] \ \mathbf{then} \\ & pixel_address \leftarrow NULL \\ \mathbf{else} \\ & pixel_address \leftarrow heap \left[head[i] \right] \\ & head[i] \leftarrow head[i] - 1 \\ \mathbf{end} \ \mathbf{if} \end{array}$

Algorithm 8 Pseudocode for adding *pixel_address* to the priority queue.

 $i \leftarrow \max\left(pixel_priority, highest_existing_priority\right)$

add *pixel_address* to the *i*-th FIFO queue

Algorithm 9 Pseudocode for removing the next *pixel_address* from the priority queue.

 $i \leftarrow highest_existing_priority$ remove the next $pixel_address$ from the *i*-th FIFO queue while $pixel_address = NULL$ and $highest_existing_priority < 2^n$ do $highest_existing_priority \leftarrow highest_existing_priority + 1$ $i \leftarrow highest_existing_priority$ remove the next $pixel_address$ from the *i*-th FIFO queue end while

4.8 Summary

This Chapter has:

- Described the various approaches to grey-scale image segmentation published in the literature including: discontinuity-based methods, similarity-based methods, local filtering, snakes and balloons, region growing and merging, global optimisation, and level set methods.
- Described chromatin, the manner in which it is visualised using a light microscope, and its appearance and structure under the light microscope. The conclusion is that nuclear chromatin, as visualised by light microscopy, can be modelled as sets of light and dark blobs arranged in a grey background sap. Topographically speaking the blobs correspond to hills and depressions.
- Critically reviewed previous approaches to chromatin segmentation. The conclusion is that existing methods typically require the specification of one or more operational parameters—and are thus not robust to changes in, or non-uniformity of, illumination and staining—, and/or do not produce a segmentation consistent with a human observer's perception of chromatin particles.
- Presented a new algorithm for chromatin segmentation based on seeded region growing. The algorithm is, in its preferred embodiment, parameter free. Moreover the algorithm yields a segmentation consistent with what a human would perceive to be chromatin particles and areas of clearing. In its preferred embodiment, the algorithm is based on the watershed transform (a special case of seeded region growing).
- Critically reviewed the seeded region growing algorithm of Adams & Bischof (1994). The conclusion is that the algorithm is inherently dependent on the order of pixel processing.

- Presented a new improved seeded region growing algorithm that retains the advantages of the Adams & Bischof (1994) algorithm but is independent of the order of pixel processing.
- Presented a new implementation of an ascending priority queue for use in implementing the watershed transform (a special case of seeded region growing). This permits the implementation of a fast watershed transform suitable for use in automated cytometry where near real-time processing is needed for an economically viable screening device.

The next chapter deals with the representation and description of segmented chromatin particles. This permits the quantitative characterisation of the attributes of these particles as well as their topographical arrangement within the nucleus.

Chapter

Representation and Description

Copyrighted image removed

(c) 2004 Papunya Tula Artists Pty. Ltd. - Australia. All rights reserved. Used by permission.

Kenny Williams Tjampitjinpa's "Kiwirrkura Community Site"

The motivation for this chapter is the need to quantitatively characterise the attributes and arrangement of the dark and/or light chromatin particles produced by the chromatin segmentation algorithm presented in the last chapter. To this end the chapter proffers a method for characterising both blob-like and mosaic patterns (texture) in the plane. The method, called the *adjacency graph attribute cooccurrence matrix* (AGACM) method, combines both structural and statistical/stochastic aspects of texture. The image under study is first reduced to a geometric *adjacency graph*



with vertices corresponding to individual regions (objects)—e.g. cells, aggregated chromatin, watershed regions, flat zones, individual pixels—and edges correspond-

ing to an adjacency relationship between regions. When the regions are the individual pixels of the image then the underlying grid—usually the hexagonal grid, 4-connected square grid, or the 8-connected square grid—serves as the adjacency graph. When the regions constitute a complete partitioning of the image (see for example Figure 4.20) then the region adjacency graph (RAG) serves as the adjacency graph. When the regions are disjoint connected components (see Definition 2.9.5) then the family of neighbourhood graphs—relative neighbourhood, Delaunay, and Gabriel—stemming from the Voronoi diagram are suitable candidates for the adjacency graph. Next, region attributes—average grey-level, dynamic, area, perimeter, etc.—are assigned to each vertex of the adjacency graph. The resulting vertexweighted adjacency graph constitutes a representation and description of the image under study. This representation and description in turn facilitates the quantitative description of the underlying pattern. For example it is possible to compute statistics—mean, variance, etc.—of the number of neighbours each region has, of the number of neighbours of attribute i each region has, of the shortest distance (graph or grid) from a region to another region with attribute i and so on. It is also possible to quantitatively characterise co-occurrence; e.g. the number of times a region of attribute i is adjacent to a region of attribute j. Further, these co-occurrences can be summarised in a co-occurrence matrix and summary measures (co-occurrence matrix features) derived from it. In fact, a co-occurrence matrix (AGACM) can be defined for each attribute and for k-adjacency; e.g. 2-adjacent regions whose vertices are joined by a path containing two edges.

The remainder of this chapter is organised as follows. In the next section the notion of a (geometric) adjacency graph is defined and several such graphs used in image analysis are described. Section 5.2 reviews the ordinary Voronoi diagram and the graphs related to the planar Voronoi diagram. These graphs characterise the adjacency relations existing between sets of points in the plane. Section 5.3 reviews an extension of the planar Voronoi diagram, called the area Voronoi diagram, for which the generators are areas (regions) rather than points. The Euclidean distance transform (EDT) is the key to defining the area Voronoi diagram for digital images. Section 5.4 reviews the distance transform and distance transform algorithms. Section 5.5 establishes a new theoretical result concerning the distance transform of a binary image, where the underlying distance is based on a *positive definite quadratic* form, and the erosion of its characteristic function by an *elliptic poweroid* structuring element. Moreover it is shown that the well-know EDT algorithm of Huang & Mitchell (1994) is a special case of this result. In addition a new algorithm is presented for computing the EDT on hexagonal grids. The algorithm is faster and less complex than the only other known algorithm devised by Vincent (1989). Section 5.6 discusses the extension of the area Voronoi diagram and related graphs to connected components of a binary digital image. Section 5.7 reviews the notions of generalised co-occurrence, the grey-level co-occurrence matrix method of Haralick et al. (1973), and formally defines the AGACM. Section 5.8 presents an overview of the types of parameters (attributes) that can be measured for image objects. In particular, the *Minkowski functionals* are reviewed because they constitute the basis of any valid measurement. Estimators for two-dimensional binary and greyscale images are presented for both the square and hexagonal grids. These include corrections to the literature as well as new estimators. The issue of *dimensionality* of measurements on grey-scale images is also discussed. Dimensional measurements are robust to changes to image magnification at the time of capture and to changes in contrast/brightness. Finally, Section 5.9 presents a summary of the chapter.

The material presented in Section 5.5 has been published in the *Journal of Mathematical Imaging and Vision:* Mehnert & Jackway (1999b). The paper has subsequently been cited in Talbot & Appleton (2002), Staunton (2001), and Rosenfeld (2000).

5.1 Adjacency graphs

This section briefly introduces the notion of a (geometric) adjacency graph and describes several such graphs used in image analysis.

Recall (see Section 2.10) that the vertices of a connected simple graph G = (V, E) are said to be adjacent if they are joined by an edge. The set of edges E defines an irreflexive and symmetric binary relation on the set of vertices V. Hereinafter such a relation is referred to as an *adjacency relation*, and a connected simple graph is called an *adjacency graph*.

Adjacency graphs are used in image analysis to model both the *topological connect-edness* of and the geometric structure in the image under study. The grid associated with a two-dimensional digital image—usually the 4- or 8-connected square grid, or the 6-connected hexagonal grid (see Section 2.9.1)—describes the connectivity between the pixels. For an image of finite size, the grid is in fact an adjacency graph. The vertices of the graph are the grid points and the edges are identically the grid edges. The edges define the adjacency relation "is connected to" on the set of vertices (representing pixels). If the image has been completely partitioned into regions then it is possible to construct another type of adjacency graph called

the region adjacency graph (RAG) (van der Heijden, 1994, p. 270). The RAG is constructed by associating a vertex with each region and joining pairs of vertices if the corresponding regions are direct neighbours (i.e. they abut). The edges define the adjacency relation "is a direct neighbour of" on the set of vertices (representing regions). If the image has been partitioned into disjoint regions (i.e. the regions do not abut) then the usual notion of *neighbour* or *adjacency* is not so clear cut. This motivates the definition of alternative adjacency relations to describe neighbourliness; e.g. "is the nearest neighbour of". Many such relations have been defined in the literature for point sets. However, Kirkpatrick & Radke (1985, p. 223) remark that:

if we insist that the internal structure of a point set be described by a connected graph, then it is natural to turn to the (Euclidean) *minimum* spanning tree (MST)... as a minimal descriptor.

The MST is one of several graphs—including the *Delaunay graph* (DG), *Gabriel graph* (GG), and the *relative neighbourhood graph* (RNG)—related to the *Voronoi diagram*. As noted by Vincent (1989) these graphs are of interest because:

- the adjacency relation is not defined in terms of any parameters as is the case, for example, for k-nearest neighbour graphs;
- 2. they are all connected and $planar^1$ (for point sets in the plane);
- 3. they are unique (except for the MST); and
- 4. they satisfy $MST\subseteq RNG\subseteq GG\subseteq DG$ (Preparata & Shamos, 1985, p. 263).

5.2 The Voronoi diagram and related graphs

This section briefly introduces the Voronoi diagram and several graphs related to it. The definitions in this section are adapted from Okabe et al. (1992) and Preparata & Shamos (1985).

The Voronoi diagram is an established tool of computational geometry². "In a sense, a Voronoi diagram records everything one would ever want to know about proximity

¹ See Definition 2.10.5.

² Broadly speaking computational geometry "is the study of algorithms for solving geometric problems on a computer". The field was formally christened by Michael Shamos in 1978 in his Ph.D. thesis (cited in O'Rourke (1993, preface)).

to a set of points (or more general objects)" (O'Rourke, 1993, p. 168). The concept of the Voronoi diagram is simple: "given a finite set of distinct, isolated points in a continuous space, we associate all locations in that space with the closest member of the point set" (Okabe et al., 1992, p. 1). The result is a partitioning of the space into regions; one region for each point (see Figure 5.1). Formally, the ordinary Voronoi diagram is defined as follows.

Definition 5.2.1 (ordinary Voronoi diagram). Let $P = \{p_1, \ldots, p_n\}$ be a set of points in \mathbb{R}^m , where $2 \le n < \infty$, and let *d* denote the Euclidean metric. The region defined by

$$V(p_i) = \{q \in \mathbb{R}^m \mid d(q, p_i) < d(q, p_j) \; \forall i \neq j\}$$

is called the (ordinary) Voronoi polyhedron associated with the point p_i and the set

$$\mathcal{V} = \{V(p_1), V(p_2), \dots, V(p_n)\}$$

is called the *ordinary Voronoi diagram* generated by P.

Remark. From the definition of the Voronoi polyhedron $V(p_i)$ it is clear that it is an open set because it does not contain its boundary. Alternatively, one can replace the < symbol with the \leq symbol and define the Voronoi polyhedron to be the closed set:

$$\overline{V(p_i)} = \{q \in \mathbb{R}^m \,|\, d(q, p_i) \le d(q, p_j) \,\,\forall i \neq j\}\,.$$

Both definitions are acceptable (Okabe et al., 1992, p. 67).

The Voronoi diagram in \mathbb{R}^2 is called the *planar Voronoi diagram*. When the generating set P consists of at least three points and these points are not collinear, then it is possible to construct a planar graph, called the Voronoi graph, from the planar Voronoi diagram by:

- 1. associating a vertex with each intersection of Voronoi edges;
- 2. making an arbitrary cut along each of the infinite edges;
- 3. introducing a dummy vertex beyond the extent of the truncated diagram; and
- 4. connecting the end point of each cut edge to the dummy vertex with a line segment.

An example of a Voronoi graph is shown in Figure 5.2.



Figure 5.1: Ordinary Voronoi diagram in \mathbb{R}^2 . The shaded region is a Voronoi polygon.



Figure 5.2: Voronoi graph constructed from the Voronoi diagram shown in Figure 5.1.

5.2.1 The Delaunay graph

The Delaunay graph is the (geometric-) dual of the Voronoi graph (Okabe et al., 1992, p. 101). Formally, the geometric dual of a plane graph is defined as follows (Wilson, 1985, p. 72).

Definition 5.2.2 ((geometric-) dual graph). Given a plane graph G = (V, E), its dual graph G^* is constructed by:

- 1. choosing a single point in each face³ of G—these constitute the vertices v_i^* of G^* ; and
- 2. corresponding to each edge e of G, a line is drawn which crosses e (but no other edge) and joins the vertices v_i^* which lie in the faces adjoining e—these lines constitute the edges of G^* .

Remark. If G is both plane and connected then G^* is plane and connected (Wilson, 1985, p. 73).

Figure 5.3 shows the construction of the dual of a graph.

 $^{^{3}}$ A plane graph divides the plane into a number of regions called faces.



Figure 5.3: Constructing the geometric-dual of a plane graph.

The dual graph of the Voronoi diagram is isomorphic to the *straight-line dual* of the Voronoi diagram. The straight-line dual is obtained by associating a vertex with each point p_i and joining two vertices if their Voronoi polygons share an edge (Preparata & Shamos, 1985, p. 208). The Delaunay graph is thus a simple connected planar graph (Preparata & Shamos, 1985, p. 211). If no four points of the generating set P are cocircular (non-cocircularity assumption) then the Delaunay graph (defined as the straight-line dual of the Voronoi diagram) is a triangulation⁴ called the Delaunay triangulation (Okabe et al., 1992, p. 89). Otherwise it is called a Delaunay pretriangulation because it will contain one or more regions with four or more sides. A Delaunay graph is shown in Figure 5.4.

5.2.2 Graphs related to the Delaunay triangulation

In the remainder of this section, it is assumed that the generating set P satisfies the non-cocircularity assumption so that the Delaunay graph is a triangulation. Several graphs are related to the Delaunay triangulation including the Gabriel graph, relative neighbourhood graph, and the minimum spanning tree.

⁴ Given a set of points in the plane, if pairs of points are joined by non-intersecting straight line segments such that every region internal to the convex hull is a triangle, the result is called a triangulation (Preparata & Shamos, 1985, p. 189).



Figure 5.4: The Delaunay graph derived from the Voronoi diagram of Figure 5.1. The graph is shown superimposed on the Voronoi diagram (broken lines).



Figure 5.5: Gabriel graph.

The Gabriel graph⁵ (Gabriel & Sokal, 1969) is defined by associating a vertex with each point in P and joining pairs of points $p_i, p_j \in P$ by an edge if and only if the closed disk with diameter $\overline{p_i p_j}$, denoted $D(p_i, p_j)$, contains no other point $p_k \in P$ in its interior (see Figure 5.6(a)). A simple algorithm for constructing this graph is to delete each edge from the Delaunay graph which does not intersect its dual Voronoi edge (Preparata & Shamos, 1985, p. 263). A Gabriel graph is shown in Figure 5.5. The edge set of the Gabriel graph is a subset of the edge set of the Delaunay triangulation.

The relative neighbourhood graph (Toussaint, 1980) is defined such that a pair of points $p_i, p_j \in P$ are joined by an edge if and only if

$$d(p_i, p_j) \le \min_{k(\neq i, j)} \max \left\{ d(p_i, p_k), d(p_j, p_k) \right\}.$$

This is equivalent to saying that a pair of points $p_i, p_j \in P$ are joined by an edge if the lune formed by the intersection of two closed disks, one centred at p_i and one

⁵ Gabriel & Sokal (1969, p. 267) describe the underlying adjacency relation as *contiguity*.



Figure 5.6: Gabriel and relative neighbours. (a) Two points are Gabriel neighbours if no other point is contained in the interior (open shaded region) of their "disk of influence". (b) Two points are relative neighbours if no other point is contained in the interior (open shaded region) of their "lune of influence".

centred at p_j , each of radius equal to the length of $\overline{p_i p_j}$, contains no other point $p_k \in P$ in its interior (see Figure 5.6(b)). A relative neighbourhood graph is shown in Figure 5.7. The edge set of the relative neighbourhood graph is a subset of that of the Gabriel graph.

The minimum spanning tree is defined to be the connected graph with the fewest number of edges that can be defined on P such that the sum of the lengths of the edges is the minimum possible. A minimum spanning tree is shown in Figure 5.8. The edge set of the minimum spanning tree is a subset of the edge set of the relative neighbourhood graph.

5.2.3 β -skeletons

A pair of points $p_i, p_j \in P$ are considered to be Gabriel neighbours if the interior of their disk of influence contains no other points $p_k \in P$. Similarly, they are considered to be relative neighbours if their lune of influence contains no other points $p_k \in P$. Kirkpatrick & Radke (1985) have proposed a family of lune-based neighbourhoods of influence $N(p_i, p_j, \beta)$, indexed by the parameter $\beta \in \mathbb{R}^+$, that give rise to a family of graphs they call β -skeletons. This family includes the GG ($\beta = 1$), RNG ($\beta = 2$), and the MST ($\beta \to \infty$).



Figure 5.7: Relative neighbourhood graph.



Figure 5.8: Minimum spanning tree.

5.2.4 A comment on structural stability

Tüceryan & Chorzempa (1991) used Monte Carlo methods to study the structural stability of the Delaunay triangulation, Gabriel graph, relative neighbourhood graph, and minimum spanning tree under random positional noise. The Delaunay triangulation was shown to be the least sensitive to such noisy conditions.

5.3 The area Voronoi diagram

"Since the early 1970s the ordinary Voronoi diagram has been extended or generalized in many directions, and those generalized Voronoi diagrams facilitate many practical applications in various fields" (Okabe et al., 1992, p. 123). One possible generalisation is to extend the generator from a point to an area. To facilitate this it is necessary to introduce the concept of the distance between a point and an area (set).

Definition 5.3.1. Let (E, d) be a metric space (see Definition 2.9.1). The distance between a point $x \in E$ and a non-empty set $X \subseteq E$ is defined

$$d(x, X) = \inf_{y \in X} d(x, y).$$

Remark. The distance d(x, X) is not a metric because it does not satisfy the properties given in Definition 2.9.1. However, it plays an important role in the definition of the *Hausdorff* metric (Heijmans, 1994*a*, p. 228). In the setting (\mathbb{R}^n, d), where *d* is the Euclidean distance, the Hausdorff metric is defined

$$d_{\mathrm{H}}(X,Y) = \sup_{z \in E} |d(z,X) - d(z,Y)|$$

where X and Y are non-empty $compact^6$ subsets of \mathbb{R}^n . It is not a metric on the space \mathbb{R}^n but rather, the set of all non-empty compact subsets of \mathbb{R}^n . When the two sets X and Y each consist of a single point the Hausdorff metric coincides with the Euclidean distance (Serra, 1988*d*, p. 73).

⁶ A subset of \mathbb{R}^n is compact if and only if it is topologically closed (i.e. it contains its boundary) and *bounded*. A subset X of \mathbb{R}^n is said to be bounded if there exists a real number L > 0 such that $d(x, y) \leq L$ for all $x, y \in X$ (Heijmans, 1994*a*, p. 228). A closed ball with finite radius is an example of a compact set.



Figure 5.9: Area Voronoi diagram.

Definition 5.3.2 (area Voronoi diagram). Let $A = \{A_1, A_2, \ldots, A_n\}$ where $1 \le n < \infty$ and $A_i \in \mathcal{P}(\mathbb{R}^2)$. Assume that the A_i are non-empty *connected*⁷ closed sets and that they are disjoint, i.e. $A_i \cap A_j = \emptyset$ for all $i \ne j$. The set

$$V(A_i) = \{ p \mid d(p, A_i) < d(p, A_j) \text{ for all } j \neq i \}$$

is called the *area Voronoi region* associated with A_i and the set

$$\mathcal{V}(A) = \{V(A_1), V(A_2), \dots, V(A_n)\}\$$

is called the *area Voronoi diagram* generated by A.

An area Voronoi diagram is shown in Figure 5.9

In the image processing community, the distance d(x, X) is better known in the guise of the *distance transform*. Importantly, the distance transform provides the key to implementing the area Voronoi diagram in the digital image setting.

5.4 The distance transform

This section presents a brief overview of the distance transform and distance transform algorithms. For a comprehensive review of distance transform algorithms, the

⁷ A set C is said to be disconnected if there exist a pair of topologically open sets A and B such that $(A \cap C) \cap (B \cap C) = \emptyset$. A set is said to be connected if it is not disconnected (Okabe et al., 1992, p. 20).

reader is referred to the Ph.D. thesis of Cuisenaire (1999), and more recently to the papers by Maurer et al. (2003) and Lotufo & Zampirolli (2001).

The distance transform (DT) is an important tool in image processing. Its uses include:

- obtaining dilations and erosions of binary images by arbitrarily sized disks (balls) (Heijmans, 1994a);
- 2. determining skeletons, skeletons by influence zones, conditional bisectors, and ultimate erosions (Serra, 1982);
- 3. constructing shape factors (Danielsson, 1978); and
- determining neighbourhood graphs for the purpose of studying the clustering and spatial repartition of image objects (Vincent, 1989; Heijmans, Nacken, Toet & Vincent, 1992).

Formally, the DT is defined as follows.

Definition 5.4.1 (distance transform). Let (E, d) be a metric space and let $X \subseteq E$. The *distance transform* (DT) is a function $D_X : E \to \mathbb{R}$ that assigns to each point $x \in E$ its shortest distance to the set X^c :

$$D_X(x) = d(x, X^c).$$

Remark. In the literature the DT is sometimes defined to be the function $x \mapsto d(x, X)$. This function is, according to the definition above, the DT of the complement of X.

The naive implementation of the DT involves computing, for each foreground pixel in turn, its distance to every background pixel. Execution time is thus proportional to the product of the number of background pixels and the number of foreground pixels; i.e. for an image containing n pixels the naive implementation is of $O(n^2)$ time complexity. Rosenfeld & Pfaltz (1966) proposed a two-pass algorithm—consisting of a forward pass (from left to right and from top to bottom) followed by a backward pass (from right to left and from bottom to top)—for computing the DT based on either the city-block or chessboard metrics. The algorithm has O(n) complexity. The underlying idea of the algorithm is that distances can be propagated locally from one foreground pixel to its neighbouring foreground pixels. Borgefors (1984) improved and generalised this algorithm to chamfer metrics (discussed below) and



Figure 5.10: Propagation masks used in the two-pass sequential DT algorithm. Top row: masks used in the forward pass on the square and hexagonal grids respectively. Bottom row: masks used used in the backward pass on the square and hexagonal grids.

extended the algorithm to the hexagonal (also called *honeycomb*) metric⁸ on the hexagonal grid. The distance propagation masks used in the forward and backward passes for the square and hexagonal grids are shown in Figure 5.10. In the case of the city-block metric a = 1 and $b = +\infty$. In the case of the chessboard metric a = 1 and b = 1. As Figure 5.11(a)-(d) shows, these discrete metrics are crude approximations to the discrete Euclidean metric.

In the case of the square grid a better approximation to the Euclidean metric is obtained by weighting the distances in the propagation masks to reflect their true Euclidean distance; e.g. setting a = 1 and $b = \sqrt{2}$. This gives rise to the class of *chamfer-a-b* metrics⁹. Borgefors (1986) proposed the chamfer-3-4 metric (integer weights) as a good approximation to (three times) the Euclidean distance. Borgefors (1986) also considered larger propagation masks concluding that the chamfer-5-7-11 metric (integer weights) offers a better approximation than the chamfer-3-4 metric. Moreover, Borgefors (1986) concluded that it is not worth considering larger propagation masks, e.g. 7×7 , with integer weights because the added computational complexity does not warrant the slight improvement in approximation¹⁰. The chamfer-*a-b-c* masks are shown in Figure 5.12.

⁸ This metric was first discussed in the literature by Rosenfeld & Pfaltz (1968), who proposed a parallel algorithm for its implementation, and subsequently by Luczak & Rosenfeld (1976). Borgefors & Sanniti di Baja (1988) coined the term *honeycomb*.

⁹ Theoretical results concerning chamfer metrics, as well as several efficient algorithms for implementing mathematical morphology operations using ball structuring elements defined by the chamfer metrics, can be found in Nacken (1996).

¹⁰ Butt & Maragos (1998) summarise several subsequent approaches to obtaining optimal chamfer masks, and propose their own novel approach motivated by multi-scale considerations.



Figure 5.11: Comparison of several discrete metrics. The binary image consists of all foreground pixels with the exception of a single background pixel located 100 pixels in from each edge. Quantisation has been applied to the grey-levels so that the closed balls associated with each metric can be visualised. In addition, each DT has been normalised so that the horizontal distance from the pixel is the same for each. (a) Euclidean distance. (b) City-block distance (4-connected square grid). (c) Chessboard distance (8-connected square grid). (d) Honeycomb distance (6-connected hexagonal grid). (e) Chamfer 3-4 distance (8-connected square grid). (f) Chamfer 5-7-11 distance (8-connected square grid). (g) Chamfer 3-5 distance (hexagonal grid). All of the discrete metrics (b)–(g) are anisotropic.



Figure 5.12: Larger propagation masks that can be used in the two-pass DT algorithm. Top row: masks used in the forward pass on the square and hexagonal grids respectively. Bottom row: masks used used in the backward pass on the square and hexagonal grids respectively.

In the case of the hexagonal grid, the distances in the elementary propagation masks already reflect their true Euclidean distance. However, Borgefors (1988, 1989) proposed weighting distances in larger masks to obtain better approximation to the Euclidean distance thus generalising chamfer metrics to the hexagonal grid. The hexagonal chamfer-a-b masks are shown in Figure 5.12. Borgefors (1988) proposed the hexagonal chamfer-3-5 metric (integer weights) as a good approximation to (three times) Euclidean distance (see Figure 5.11(g)).

DTs based on the city-block, chessboard, honeycomb, and chamfer metrics have enjoyed popular usage in the image analysis community because they are easily and efficiently implemented; e.g. using a two-pass sequential algorithm, or the FIFO queue algorithm of Vincent (1991*b*). Unfortunately, as Figure 5.11 shows, none of these metrics are isotropic. Exact Euclidean distance is required for isotropy.

Danielsson (1980) proposed a four-pass sequential algorithm for computing a quasi-Euclidean DT on the square grid. The algorithm is based on vector propagation. Each pixel is initially assigned a two-component label: (0,0) for foreground pixels and $(+\infty, +\infty)$ for background pixels. The first component records the relative *x*-coordinate of the nearest foreground pixel, whilst the second component records the relative *y*-coordinate of the nearest foreground pixel. During the four-pass scan, these components, rather than distances, are propagated. The algorithm yields almost perfect Euclidean distance. However, non-systematic errors occur for certain configurations of foreground pixels (see Danielsson (1980) and Vincent (1991*a*) for details). The problem is that the algorithm is based on the assumption that the distance (vector) assigned to any given pixel can be deduced from the vectors assigned to its neighbouring pixels. Whilst this holds true for the Euclidean metric in the continuous plane, and for the discrete metrics above, it does not in general hold true for the Euclidean metric on the discrete grid (Cuisenaire, 1999, p. 16). The problem is that when the Voronoi region of an area in the continuous plane is restricted to a discrete grid, the result is not necessarily a single connected component (Vincent, 1991*a*, p. 521).

Yamada (1984) proposed the first exact EDT algorithm on the square grid. The algorithm is based on the vector propagation idea of Danielsson (1980) (with some improvements) but is a parallel rather than sequential algorithm. For a parallel DT algorithm, in each pass through the image all of the pixels can be processed independently and in parallel rather than sequentially. However, in contrast to the sequential DT algorithms, a parallel DT algorithm can only propagate distance labels at a distance prescribed by the size of the neighbourhood mask. Consequently, parallel DT algorithms are generally much slower than sequential DT algorithms on a general purpose computer. In contrast to the DT algorithms already discussed, the time complexity of parallel DT algorithms is not fixed. Instead the complexity is proportional to the product of the number of pixels and the largest distance in the image. Hence parallel DT algorithms are of $O\left(n^{\frac{3}{2}}\right)$ complexity (Cuisenaire & Macq, 1999; Maurer et al., 2003).

Vincent (1991*a*) proposed the first sequential algorithm for computing the exact EDT. The algorithm can be implemented on both square and hexagonal grids. Hence it is also the first exact EDT algorithm for the hexagonal grid. The algorithm is based on the idea of ordered propagation, as opposed to raster/anti-raster propagation. The algorithm propagates chains—data structures encoding the boundaries of the object pixels—in the image using a set of rewriting rules. Like the algorithms of Danielsson (1980) and Yamada (1984), vector information rather than distance is propagated. The algorithm is more efficient than that of Yamada (1984) because it does not require multiple complete scans of the image pixels. Its execution time is highly dependent on the image content and thus its time complexity lies somewhere between O(n) and $O(n^{\frac{3}{2}})$. Indeed, Vincent (1991*a*, p. 524) concludes, experimentally, that his algorithm runs in approximately half the time of the quasi-EDT of Danielsson (1980).
Shih & Mitchell (1992) devised an exact EDT algorithm based on grey-scale erosion by an inverted conical structuring element¹¹. The algorithm is discussed in detail in the next section. In summary the algorithm is based on a complex dilation/supremum decomposition of the (infinitely) large conical structuring element into smaller 3×3 structuring elements. The resulting algorithm requires the computation of m(m+1)/2 erosions by 3×3 structuring elements, where m is the largest distance in the image (m is of the order $n^{\frac{1}{2}}$). Each erosion involves a pass through all n pixels, either in parallel or sequentially, and thus the resulting algorithm is of $O(n^2)$ complexity.

Huang & Mitchell (1994) adopted the approach of Shih & Mitchell (1992) but instead devised an exact algorithm for computing squared Euclidean distance based on erosion by an inverted elliptic paraboloid. This structuring element has a simple dilation decomposition into smaller 3×3 structuring elements. The resulting algorithm requires the computation of m erosions, where m is the largest distance in the image. The algorithm is thus of $O\left(n^{\frac{3}{2}}\right)$ complexity. The algorithm is discussed in detail in the next section.

Breu et al. (1995) devised the first O(n) algorithm for computing the exact EDT on the square grid. The algorithm is based on the idea that the EDT can be obtained as a by-product of the generation of the Voronoi diagram in the Euclidean plane. Breu et al. show that the Voronoi diagram of the centres (grid points) of the foreground pixels can be constructed in linear time because of the regular arrangement of the grid points. Their algorithm directly constructs the intersection of the Voronoi diagram of the foreground pixels with each row of the image. This is accomplished in two row-wise passes through the image. At the end of the two passes, each pixel has been assigned the identity (location) of the closest foreground pixel. The algorithm then computes the EDT from this image.

Mehnert & Jackway (1999b) proved an equivalence between the distance transform of a binary image, where the underlying distance is based on a *positive definite quadratic* form, and the erosion of its characteristic function by an *elliptic poweroid* structuring element (this is presented in the next section). The algorithms devised by Shih & Mitchell (1992) and Huang & Mitchell (1994) are particular cases of this result. In addition Mehnert & Jackway (1999b) proposed a method for computing the exact EDT on the hexagonal grid based on first embedding the image in a rectangular grid and then applying the algorithm of Huang & Mitchell (1994) with

¹¹ The idea was first proposed by Sternberg (1986), although in its dual form; i.e. as grey-scale dilation by a sufficiently large conical structuring element.

appropriate aspect ratio correction. These results are discussed in detail in the next section.

Cuisenaire & Macq (1999) proposed the first, apparently, O(n) algorithm for computing the exact EDT based on ordered propagation. They only verified this time complexity experimentally. The algorithm comprises two steps. In the first step an approximate EDT is produced using an ordered propagation scheme (the result is similar to that produced by the algorithm of Danielsson (1980)). In the second step the exact EDT is obtained by considering a sequence of larger neighbourhoods restricted to the boundaries of the Voronoi regions. The algorithm does not readily extend to rectangular and hexagonal grids.

Lotufo & Zampirolli (2001) proposed a faster implementation of the algorithm of Huang & Mitchell (1994) based on the further decomposition of each 3×3 structuring element into four one-dimensional structuring elements: two 1×2 structuring elements and two 2×1 structuring elements. In addition Lotufo & Zampirolli (2001) implement the directional erosions using fast propagation algorithms based on FIFO queues. They claim the algorithm is faster than the linear time algorithm proposed by Cuisenaire & Macq (1999).

Maurer et al. (2003) recently proposed an O(n) algorithm for computing the exact EDT (the algorithm actually produces squared distances). The algorithm, like that of Breu et al. (1995), is based on the idea that the EDT can be computed as a by-product of the generation of the Voronoi diagram. However, in contrast to the algorithm of Breu et al. (1995), squared Euclidean distance is computed directly rather than from the "closest foreground pixel" image. In addition, this algorithm is applicable to both square and rectangular pixels.

Although several exact EDT algorithms have now been devised for the square grid including the parallel algorithms of Yamada (1984); Chen & Chuang (1994); Embrechts & Roose (1996) and the sequential algorithms of Vincent (1991*a*); Paglieroni (1992); Mullikin (1992); Ragnemalm (1992); Saito & Toriwaki (1994); Breu et al. (1995); Hirata (1996); Kimmel et al. (1996); Eggers (1998); Guan & Ma (1998); Cuisenaire (1997); Cuisenaire & Macq (1999); Lotufo & Zampirolli (2001); Maurer et al. (2003)—, the algorithms of Vincent (1991*a*) and Mehnert & Jackway (1999*b*) remain the only ones reported in the literature that operate on the hexagonal grid. The algorithm of Mehnert & Jackway (1999*b*) has two major advantages over that of Vincent (1991*a*): (i) simplicity and (ii) suitability for hardware implementation using a pipeline architecture (because it is based on the the algorithm of Huang & Mitchell (1994)). The algorithm is based on the idea of first embedding the hexagonal grid image in a rectangular grid and then applying the algorithm of Huang & Mitchell (1994) with the appropriate aspect ratio correction. Using the fast implementation of Huang and Mitchell's algorithm devised by Lotufo & Zampirolli (2001) leads to a highly efficient serial implementation. An O(n) exact EDT should also be possible by replacing Huang and Mitchell's algorithm with the anisotropic version of the recent algorithm of Maurer et al. (2003). The material in Mehnert & Jackway (1999b) is presented in the next section.

5.5 Computing the exact Euclidean distance transform on rectangular and hexagonal grids

In this section a new result is proved establishing an equivalence between the distance transform of a binary image, where the underlying distance is based on a positive definite quadratic form, and the erosion of its characteristic function by an elliptic poweroid¹² structuring element. The well-known algorithms devised by Shih & Mitchell (1992) and Huang & Mitchell (1994), for calculating the exact Euclidean distance transform (EDT) of a binary digital image manifested on a square grid, are particular cases of this result. The former algorithm uses erosion by a circular cone to calculate the EDT whilst the latter uses erosion by an elliptic paraboloid (which allows for pixel aspect ratio correction) to calculate the square of the EDT. The algorithm of Huang & Mitchell (1994) is arguably the better of the two because: (i) the structuring element can be decomposed into a sequence of dilations by 3×3 structuring elements (a similar decomposition is not possible for the circular cone) thus reducing the complexity of the erosion, and (ii) the algorithm only requires integer arithmetic (it produces squared distance). The algorithm is amenable to both hardware implementation using a pipeline architecture and efficient implementation on serial machines. Unfortunately the algorithm does not directly transpose to, nor has a corresponding analogue on, the hexagonal grid (the same is also true for the algorithm of Shih & Mitchell (1992)). In this section, however, it is shown that if the hexagonal grid image is embedded in a rectangular grid then the algorithm of

¹² The expression (negative) elliptic poweroid was coined by Jackway (1995) to describe the parametrised family of functions $g_{\sigma} : \mathbb{R}^n \to \mathbb{R}$ defined $g_{\sigma}(x) = -|\sigma| \left(\sqrt{x^{\mathsf{T}} \mathbf{A} x} / |\sigma| \right)^{\alpha}$ where $\alpha > 0, \sigma \neq 0$, and \mathbf{A} is a symmetric positive definite matrix. The expression is used herein to mean the wider family for which \mathbf{A} need only be positive definite (i.e. not necessarily symmetric).

Huang & Mitchell (1994) can be applied, with a spect ratio correction, to obtain the exact EDT^{13} on the hexagonal grid.

The material in this section was published in the *Journal of Mathematical Imaging* and Vision: Mehnert & Jackway (1999b). The paper has since been cited in Talbot & Appleton (2002), Staunton (2001), and Rosenfeld (2000).

5.5.1 The hexagonal grid versus the square grid

The image processing community has tended to favour the use of the square grid over the hexagonal grid because of the predominance of bit-mapped raster digitising and display devices. The hexagonal grid is, however, superior to the square grid for several reasons (Serra & Laÿ, 1985):

- 1. improved compression of Euclidean information;
- 2. better approximation of Euclidean isotropy;
- 3. the elementary neighbourhood comprises fewer elements (seven as opposed to nine for the 8-connected square grid); and
- 4. for homotopy problems there is no ambiguity of connectivity (see Figure 5.13)

Borgefors (1989, p. 97) states that:

The hexagonal grid is not very common. However, for some applications it is preferable to the square grid....Natural scenes, which mostly consist of rounded shapes, are better represented on the hexagonal grid than on the square one, especially in low resolution. In the hexagonal grid all six neighbors to a pixel are equally connected and have equal distance to the central pixel. Thus the annoying 4-neighbor/8-neighbor problem in the square grid does not occur.

In lieu of a means of acquiring a digital image on a hexagonal grid it is possible to convert square grid images to the hexagonal grid. Serra & Laÿ (1985), for example, have proposed a simple conversion algorithm that produces nearly identical results for convolution, and is also satisfactory for increasing or homotopic transformations.

¹³ The algorithm generates squared distance. Therefore, if desired, to obtain actual distance it is necessary to take the square root of each squared (integer) distance. This can of course be done using a lookup table.



Figure 5.13: Ambiguity of connectivity on the square grid. (a) For pixels arranged in a hexagonal pattern both foreground (grey pixels) and background have the same connectivity. (b) For pixels arranged in a square pattern, the foreground and background cannot have the same connectivity. If both the foreground and background are assumed to be 8-connected in the example, then paradoxically the hole in the centre of the "diamond" (grey pixels) must be connected to the background surrounding the diamond. To resolve this ambiguity it is usual to assign opposite types of connectivity to the foreground (objects(s) of interest) and background pixels: e.g. 8-connectivity for the foreground and 4-connectivity for the background (Rosenfeld & Kak, 1981, p. 207).

5.5.2 An equivalence between grey-scale erosion and distance transformation

Let X be an arbitrary subset of \mathbb{R}^n (not necessarily topologically open or closed). Let $B(\lambda)$ be a ball in \mathbb{R}^n , centred at the origin, of radius $\lambda \geq 0$. When d(x, y) is the Euclidean distance then the set of points $X_{\lambda} = \{x \mid D_X(x) > \lambda\}$ is identically the binary erosion of X by the ball $B(\lambda)$, i.e. $X_{\lambda} = X \ominus B(\lambda)$ (Serra, 1988*d*; Preteux & Merlet, 1991). In \mathbb{R}^2 it is easy to see that if these balls (disks) are stacked atop one another, for increasing λ , the result is a cone (see the isolines of Figure 5.18(b)). Indeed, Shih & Mitchell (1992) realised that if the set $X \subseteq \mathbb{Z}^2$ is represented as a function f(x) that has the value $+\infty$ for $x \in X$ and 0 otherwise, then the EDT of X is identically the grey-scale erosion of f by the cone $g(x) = -\sqrt{x^T x}$, i.e. $D_X(x) = (f \ominus g)(x)$. More recently, Huang & Mitchell (1994) found that if f is instead eroded by a circular paraboloid $h(x) = -x^T x$ then the result is identically the square of the EDT of X. Both of these results are in fact particular cases of a more general relationship that exists between DTs based on positive definite quadratic forms (Johnson & Wichern, 1988) and erosion by elliptic poweroid structuring elements (see Figure 5.14). The relationship and its proof follow. **Proposition 5.5.1.** Let $X \in \mathcal{P}(\mathbb{R}^n)$ and $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be its characteristic function¹⁴ defined

$$f(x) = \begin{cases} \infty & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases}$$

Let $g: \mathbb{R}^n \to \overline{\mathbb{R}}$ be the elliptic poweroid defined

$$g\left(x\right) = -\left(x^{\mathsf{T}}\mathbf{A}x\right)^{\frac{\alpha}{2}}$$

where **A** is an $n \times n$ positive definite matrix and $\alpha \in \mathbb{R}^+$. If

$$d(x,y) = \sqrt{(x-y)^{\mathsf{T}} \mathbf{A} (x-y)}$$

then

$$(f \ominus g)(x) = [D_X(x)]^{\alpha}.$$

Proof. Consider the LHS:

$$(f \ominus g) (x) = \bigwedge_{y \in \mathbb{R}^{n}} \{ f(y) - g(x - y) \}$$

= $\bigwedge \{ \{ f(y) - g(x - y) \mid y \in X \}, \{ f(y) - g(x - y) \mid y \in X^{c} \} \}$
= $\bigwedge \{ \{ \infty \mid y \in X \}, \{ [(x - y)^{\mathsf{T}} \mathbf{A} (x - y)]^{\frac{\alpha}{2}} \mid y \in X^{c} \} \}$
= $\bigwedge_{y \in X^{c}} [(x - y)^{\mathsf{T}} \mathbf{A} (x - y)]^{\frac{\alpha}{2}}.$

Consider the RHS:

$$\begin{bmatrix} D_X(x) \end{bmatrix}^{\alpha} = \begin{bmatrix} d(x, X^c) \end{bmatrix}^{\alpha} \\ = \begin{bmatrix} \bigwedge_{y \in X^c} d(x, y) \end{bmatrix}^{\alpha} \\ = \begin{bmatrix} \bigwedge_{y \in X^c} \sqrt{(x - y)^{\mathsf{T}} \mathbf{A} (x - y)} \end{bmatrix}^{\alpha} \\ = \bigwedge_{y \in X^c} \begin{bmatrix} (x - y)^{\mathsf{T}} \mathbf{A} (x - y) \end{bmatrix}^{\frac{\alpha}{2}}.$$

Hence $(f \ominus g)(x) = [D_X(x)]^{\alpha}$.

¹⁴ This definition of the characteristic function differs slightly from that used in real analysis. In real analysis the characteristic function takes on the value 1, rather than ∞ , for $x \in X$ (DePree & Swartz, 1988, p. 60).



Figure 5.14: Examples of elliptic poweroids $g(x) = -(x^{\mathsf{T}}\mathbf{A}x)^{\frac{\alpha}{2}}$ in \mathbb{R}^2 . (a) Inverted circular cone: $\alpha = 1$ and $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. (b) Inverted elliptic paraboloid: $\alpha = 2$ and $\mathbf{A} = \begin{bmatrix} 1 & 0.25 \\ 0.25 & 3 \end{bmatrix}$.

Remarks.

- 1. The proposition still holds true when \mathbb{R}^n is replaced by \mathbb{Z}^n .
- 2. When **A** is the identity matrix, n = 2, and $\alpha = 1$ then g is an inverted circular cone (see Figure 5.14), $d(x, y) = \sqrt{(x_1 y_1)^2 + (x_2 y_2)^2}$ (Euclidean distance), D_X is the EDT, and $(f \ominus g)(x) = D_X(x)$. This is the basis of the EDT algorithm of Shih & Mitchell (1992).

3. When $\mathbf{A} = \begin{bmatrix} m^2 & 0 \\ 0 & n^2 \end{bmatrix}$, where $m, n \in \mathbb{R}$, n = 2, and $\alpha = 2$ then g is an inverted elliptic paraboloid, $d(x, y) = \sqrt{[m(x_1 - y_1)]^2 + [n(x_2 - y_2)]^2}$, and

inverted emptic paraboloid, $a(x,y) = \sqrt{[m(x_1 - y_1)]} + [n(x_2 - y_2)]$, and $(f \ominus g)(x) = [D_X(x)]^2$. If m = n = 1 then g is an inverted circular paraboloid and D_X is the EDT of X. This is the basis of the EDT algorithm of Huang & Mitchell (1994).

5.5.3 The algorithm of Huang & Mitchell

The EDT algorithms proposed by both Shih & Mitchell (1992) and Huang & Mitchell (1994) employ structuring element decomposition (Shih & Mitchell, 1991) to reduce

Algorithm 10 The EDT algorithm of Huang & Mitchell (1994) (rectangular grid).

- 1: Convert the input binary image to a grey-scale image by mapping each object pixel (usually binary 1) to positive infinity (just a very large value) and each background pixel (usually binary 0) to zero.
- 2: Set i = 1 (iteration counter).
- 3: Erode by the structuring element (origin at centre):

$$g_i = \begin{bmatrix} -2(m^2 + n^2)i + m^2 + n^2 & -2n^2i + n^2 & -2(m^2 + n^2)i + m^2 + n^2 \\ -2m^2i + m^2 & 0 & -2m^2i + m^2 \\ -2(m^2 + n^2)i + m^2 + n^2 & -2n^2i + n^2 & -2(m^2 + n^2)i + m^2 + n^2 \end{bmatrix}$$

where m is the horizontal grid spacing and n is the vertical grid spacing (for square pixels m = n = 1).

- 4: If the image has changed then increment i and go to step 3.
- 5: Replace each pixel with its square root (optional).

the complexity of the erosion by the circular cone or paraboloid, respectively, to a set of erosions by 3×3 structuring elements. Unfortunately in the case of the circular cone, a complex supremum and dilation decomposition is needed (Shih & Mitchell, 1992). However, in the case of the circular paraboloid, and more generally the elliptic paraboloid, a simple dilation decomposition is possible. This decomposition is the basis of the algorithm by Huang & Mitchell (1994). The algorithm is shown in Algorithm 10.

The algorithm by Huang & Mitchell (1994) works because on the rectangular grid the elliptic paraboloid $g(x) = -x^{\mathsf{T}} \mathbf{A} x$, for **A** defined in item 3 of the last remarks, has the dilation decomposition:

$$g = \bigoplus_i g_i.$$

Moreover to generate $[D_X(\mathbf{x})]^2$ it is necessary only to erode by an elliptic paraboloid of sufficient size (hence the algorithm iterates until no change). For example, consider the binary image X depicted in Figure 5.15. If m = n = 1 then it is clear that the domain of the circular paraboloid structuring element needed to calculate the EDT of X is a disk equal in size to the largest disk contained wholly within X.

Huang & Mitchell (1994) prove that the pixels (distance values) that change in a given iteration must be one of the eight neighbours of the pixels that changed in the previous iteration. This leads to a highly efficient implementation of their algorithm on a serial machine. The implementation is akin to the propagation algorithm devised by Vincent (1991*a*).



Figure 5.15: The domain of the circular paraboloid structuring element needed to calculate the Euclidean distance transform of the set X is a disk equal in size to the largest disk contained wholly within X.

5.5.4 Adaptation to the hexagonal grid

The corner stone of the algorithm by Huang & Mitchell (1994) is Pythagoras's theorem for right-angled triangles. This explains the formulation of each 3×3 structuring element g_i —e.g. $a_{11} = a_{12} + a_{21}$ —and the reason the algorithm produces squared Euclidean distance. The square of the Euclidean distance between two points $p = (x_1, y_1)$ and $q = (x_2, y_2)$ is given by

$$d^{2}(p,q) = (x_{1} - x_{2})^{2} + (y_{1} - y_{2})^{2}.$$

For both the square and hexagonal grids the difference $\Delta y = y_1 - y_2$ is always an integer multiple of the vertical spacing of the respective grid. On the square grid the difference $\Delta x = x_1 - x_2$ is equal to an integer multiple of the horizontal spacing (which is the same as the vertical spacing) of the square grid. Consequently it is possible to propagate Euclidean distance on the square grid using the elementary $3 \times$ 3 square neighbourhood as Huang & Mitchell (1994) have shown. On the hexagonal grid, however, Δx is equal to an integer multiple of half the horizontal spacing. As a result the elementary hexagonal neighbourhood does not have sufficient point density to be able to propagate Euclidean distance. The solution is to increase the horizontal point density of the hexagonal grid by inserting a new point between every pair of points in a row as shown in Figure 5.16 to form a rectangular grid. Each new point (pixel) is labelled as foreground (binary 1). This scheme ensures

Algorithm 11 Calculation of the exact EDT on the hexagonal grid.

- 1: Convert the input hexagonal grid image to a rectangular grid image by adding a foreground pixel (binary 1) between each pair of pixels in a row. The rectangular grid image will therefore have as many rows as the hexagonal grid image but twice as many columns.
- 2: Apply the algorithm of Huang & Mitchell (1994) described in Section 5.5.3 with $m = b/\sqrt{3}$ and n = b.
- 3: Convert the rectangular grid distance map to a hexagonal grid distance map by omitting each distance value corresponding to a pixel added in step 1.



Figure 5.16: Embedding a hexagonal grid image into a rectangular grid. (a) Hexagonal grid image. (b) Rectangular grid image formed from (a) by adding a foreground pixel between each pair of points in a row. Note that the transformation guarantees that the only background points are those corresponding to the original hexagonal grid.

that the only background points on the rectangular grid are those corresponding to the hexagonal grid. The algorithm of Huang & Mitchell (1994) can now be applied with m = a/2 and n = b where a and b are the horizontal and vertical spacing, respectively, of the hexagonal grid¹⁵. The EDT on the hexagonal grid is obtained by simply discarding the distance values generated for the pixels inserted into the hexagonal grid to form the rectangular grid. The proposed algorithm is shown in Algorithm 11. The algorithm is illustrated in Figure 5.17.

¹⁵ For the regular hexagonal grid $a = 2b/\sqrt{3}$.



Figure 5.17: New algorithm for computing the exact Euclidean distance transform (EDT) on the hexagonal grid. (a) Binary image manifested on the hexagonal grid. (b) EDT of (a) generated by the new algorithm.

5.5.4.1 A comment on integer arithmetic

If $b = \sqrt{3}$ in step 2 then $m^2 = 1$ and $n^2 = 3$ and therefore the successive erosions by g_i require only integer arithmetic. Hence given a binary image manifested on a hexagonal grid with vertical spacing b' (which implies a horizontal spacing of $a' = 2b'/\sqrt{3}$) the actual distance map can be recovered simply by multiplying entries by $b'/\sqrt{3}$.

5.6 Skeleton by influence zones and perceptual graphs

This section discusses the generalisation of the area Voronoi diagram, the Delaunay graph, and the Gabriel graph to connected components $A_i \in \mathcal{P}(\mathbb{Z}^2)$ of a binary image.

The definition of the area Voronoi diagram, Definition 5.3.2, readily extends to connected components $A_i \in \mathcal{P}(\mathbb{Z}^2)$. In image processing the Voronoi region associated

with a connected component A_i is called an *influence zone* (IZ). An IZ is defined as follows.

Definition 5.6.1 (influence zone). Let (\mathbb{Z}^2, d) be a metric space. Let $X \in \mathcal{P}(\mathbb{Z}^2)$ be a binary image containing $n \ge 1$ disjoint connected components A_1, A_2, \ldots, A_n ; i.e. $X = \bigcup_{i=1}^n A_i$ and $A_i \cap A_j = \emptyset$ for all $i \ne j$. The set

$$IZ(A_i) = \{p \mid d(p, A_i) < d(p, A_j) \text{ for all } j \neq i\}$$

is called the *influence zone* associated with the connected component A_i .

A simple algorithm for computing the set of influence zones is to compute the watershed transform (see Appendix B) of the distance transform of X^c using the the connected components as markers (these correspond to the regional minima (see Definition 4.4.1) of the distance transform). Each catchment basin defines an influence zone. The collective boundaries of the influence zones (the watersheds) is called the *skeleton by influence zones* (SKIZ) of X. The algorithm is illustrated in Figure 5.18. The RAG constructed on the set of influence zones generalises the idea of the Delaunay graph (see Figure 5.19).

The definition of the Gabriel graph can also be extended to connected components of a binary image (Vincent, 1989, p. 374). In image processing this graph is called the *perceptual graph* (Meyer, 1982). It is defined as follows.

Definition 5.6.2 (perceptual graph). Let (\mathbb{Z}^2, d) be a metric space. Let $X \in \mathcal{P}(\mathbb{Z}^2)$ be a binary image containing $n \geq 1$ disjoint connected components A_1, A_2, \ldots, A_n ; i.e. $X = \bigcup_{i=1}^n A_i$ and $A_i \cap A_j = \emptyset$ for all $i \neq j$. The perceptual graph is constructed by associating a vertex with each connected component, v_1 with A_1, v_2 with A_2 etc., and joining a pair of vertices, (v_i, v_j) by an edge if and only if there exist two points $p \in A_i$ and $q \in A_j$ such that

$$\forall r \in X \setminus \{p,q\}, \ r \notin D(p,q),$$

where D(p,q) is the closed disk with diameter \overline{pq} .

Meyer (1988; 1989) proposed a general method for constructing the perceptual graph from the distance transform of X^c that is valid for any type of digital grid and for all types of distance transform. The method is based on the detection of *saddle*



Figure 5.18: Skeleton by influence zones (SKIZ). (a) The EDT of the background between the (digitised) points shown in Figure 5.1. (b) Isolines of the EDT (compare this with the painting at the beginning of the chapter). (c) SKIZ: watersheds of the EDT using the regional minima as markers.

zones in the distance transform and then following *downstream* paths from each to a regional minimum. The reader is referred to Meyer (1989) for further details.

Meyer (1989) in fact devised a general methodology and several algorithms for constructing various types of skeleton and related structures—including the SKIZ and the perceptual graph—from binary and grey-scale digital images manifested on any type of grid and for all types of distance transforms.

5.7 The Adjacency Graph Attribute Co-occurrence Matrix

This section introduces the *adjacency graph attribute co-occurrence matrix* (AGACM). Features derived from this matrix can be used to quantitatively characterise both



Figure 5.19: Generalised Delaunay graph. (a) SKIZ of the segmented chromatin particles shown in Figure 4.13(e). (b) Region adjacency graph (RAG). Note: the influence zones of the particles were clipped to the nuclear mask before determining the RAG (see Appendix D for the DImPAL program written to produce this graph).

blob-like and mosaic patterns (texture) in the plane. The AGACM method combines both structural and statistical/stochastic aspects of texture. The AGACM is a generalisation of the well-known grey-level co-occurrence matrix (GLCM) devised by Haralick et al. (1973). The GLCM is a matrix of joint probability estimates. The element in row i and column j is an estimate of the probability that a pair of pixels, satisfying a particular distance or adjacency relation (co-occurrence), have grey-value i - 1 and grey-value j - 1 respectively. Similarly, the AGACM is a matrix of joint probability estimates. However, the AGACM is defined in terms of generalised co-occurrence which is described by Haralick (1979, p. 802) as follows:

To define the concept of generalized cooccurrence, it is necessary to first decompose an image into its primitives. Let Q be the set of all primitives on the image. Then we need to measure primitive properties such as mean gray tone, variance of gray tones, region, size, shape, etc. Let T be the set of primitive properties and f be a function assigning to each primitive in Q a property of T. Finally, we need to specify a spatial relation between primitives such as distance or adjacency. Let $S \subseteq Q \times Q$ be the binary relation pairing all primitives which satisfy the spatial relation. The generalized cooccurrence matrix P is defined by:

$$P(t_1, t_2) = \frac{\#\{(q_1, q_2) \in S \mid f(q_1) = t_1 \text{ and } f(q_2) = t_2\}}{\#S}$$

 $P(t_1, t_2)$ is just the relative frequency with which two primitives occur with specified spatial relationship in the image, one primitive having property t_1 and the other primitive having property t_2 .

Specifically, the AGACM is a co-occurrence matrix defined on an adjacency graph. The basic idea was first proposed by Toriwaki & Yokoi (1988). They extended the definitions of Delaunay, Gabriel, and relative neighbours to disjoint connected components of a two-dimensional digital image (they considered only the square grid and the chessboard and city-block metrics). They proposed using these to define generalised co-occurrence matrices. Furthermore they surmised that such matrices could be used to quantitatively characterise the texture in a two-dimensional grey-scale image. However, they did not address the manner in which texture primitives might be extracted, stating only that it is necessary to "derive a suitable neighbor graph from an input picture".

5.7.1 The grey-level co-occurrence matrix

Only a very brief overview of the GLCM is given here. For a comprehensive review of the GLCM method and of co-occurrence-based texture methods in general, the reader is referred to the Ph.D. thesis of Walker (1997).

A very simple way to quantitatively characterise the texture of a grey-scale image is to calculate moments of its grey-level histogram. An obvious limitation of this approach is that it fails to take into account any spatial, i.e. second-order, interactions between pixels comprising the texture. "One way to bring this type of information into the texture-analysis process is to consider not only the distribution of intensities, but also the positions of pixels with equal or nearly equal intensity values" (Gonzalez & Woods, 1992, p. 508). This is the basis of the well known GLCM method. The GLCM is a square matrix \mathbf{C} constructed such that element c_{ij} is the relative frequency with which a pixel of grey-level i-1 occurs at a fixed distance and direction from a pixel of grey-level j-1. In other words, given a fixed distance d and a fixed angle θ , element c_{ij} is an estimate of the joint probability $p(i, j | d, \theta)$. Usually no distinction is made between opposite directions so that $p(i, j | d, \theta) = p(i, j | d, \theta + \pi)$ and $p(i, j | d, \theta) = p(j, i | d, \theta)$, and hence **C** is symmetric; i.e. $c_{ij} = c_{ji}$. The GLCM is thus an estimate of a discrete probability density function (distribution). Classically, scalar parameters associated with this distribution are used to quantitatively characterise the texture of the image in the region over which the GLCM was computed. Some of the more commonly used

Uniformity or energy	$\sum_{i}\sum_{j}c_{ij}^{2}$	
Entropy	$-\sum_{i}\sum_{j}c_{ij}\log c_{ij}$	
Homogeneity	$\sum_{i} \sum_{j} \frac{1}{1 + (i-j)^2} c_{ij}$	
Inertia	$\sum_{i} \sum_{j} \left(i - j\right)^2 c_{ij}$	
Correlation	$-\sum_{i}\sum_{j}\frac{(i-\mu)(j-\mu)}{\sigma^2}c_{ij}$	
Shade	$\sum_{i} \sum_{j} \left(i + j - 2\mu\right)^3 c_{ij}$	
Prominence	$\sum_{i} \sum_{j} \left(i+j-2\mu\right)^4 c_{ij}$	
Variance	$\sum_{i} \sum_{j} \left(i - \mu\right)^2 c_{ij}$	
where $\mu = \sum_{i} i \sum_{j} c_{ij} = \sum_{j} j \sum_{i} c_{ij}$ and		
$\sigma^{2} = \sum_{i} (i - \mu)^{2} \sum_{j} c_{ij} = \sum_{j} (j - \mu)^{2} \sum_{i} c_{ij}.$		

Table 5.1: Commonly used GLCM features (Walker, 1997, p. 47).

features are listed in Table 5.1. Given that the size of the GLCM is dependent on the range of grey-levels present in the image, in practice the grey-levels are usually requantised to yield a more manageable matrix.

5.7.2 The adjacency graph attribute co-occurrence matrix

To compute a GLCM both a fixed distance and angle (i.e. a displacement vector) must be specified. The choice of angle is of course constrained by the underlying grid; e.g. on the (8-connected) square grid θ is typically taken to be one of 0°, 45°, 90° or 135°. The choice of distance, too, presupposes the existence of a discrete metric. Recall (see Section 2.9.1) that the underlying grid induces a metric: city-block for the 4-connected grid, chessboard for the 8-connected grid, and honeycomb for the hexagonal grid. Recall also that for a digital image of finite size, the grid is in fact an adjacency graph. An image can thus be represented as a grey-scale graph; i.e. an adjacency graph with real or integer vertex values. In addition, the

regularity of the underlying grid makes it possible to assign a direction label—one of 4 directions for the 4-connected square grid, 8 directions for the 8-connected square grid, and 6 directions for the 6-connected hexagonal grid—to each of the edges of this graph.

Given that the GLCM is effectively nothing more than a co-occurrence matrix defined on a grey-scale graph, it follows that it is possible to define such a matrix for other types of grey-scale graph. In the case of a two-dimensional digital image two obvious possibilities arise. Firstly, if the image has been segmented into a complete mosaic of regions—e.g. labelled *flat zones*, or watershed regions—then it is possible to construct a grey-scale graph from the RAG by simply assigning a region attribute—e.g. area, volume, mean grey-value—to each vertex of the RAG. Secondly, if the image has been segmented into disjoint regions (e.g. condensed chromatin particles in the nucleus of a cell) then it is possible to construct a grey-scale graph from a neighbourhood graph—e.g. Delaunay graph, perceptual graph—by once again assigning a region attribute to each vertex of the graph. More generally, for any vertex-weighted adjacency graph (grey-scale graph) it is possible to construct a co-occurrence matrix. This leads to the following definition of the *adjacency graph attribute co-occurrence matrix* (AGACM).

Definition 5.7.1. Let $k \in \mathbb{N}$. Let f | G be a grey-scale graph (see Definition 2.10.8) with grey value set \mathcal{T} . Let $\mathcal{T}' = \{0, \ldots, m\}$ represent the quantisation of \mathcal{T} into m+1 levels (any finite set of m+1 elements is isomorphic to this set) and let $\hat{f} | G$ denote the corresponding quantised grey-scale graph. Let \mathbf{A}_k be the $m \times m$ matrix defined such that entry a_{ij} is equal to the number of times (frequency) that a pair of vertices $v, w \in V(G)$, such that $\hat{f}(v) = i - 1$ and $\hat{f}(w) = j - 1$, are k-adjacent; i.e. d(v, w) = k (see Proposition 2.10.6). Let \mathbf{C}_k be the matrix formed by dividing every element of \mathbf{A}_k by the total number of k-adjacent pairs. The matrix \mathbf{C}_k is called the *adjacency graph attribute co-occurrence matrix* (AGACM).

Remarks.

- 1. An AGACM can be defined for any adjacency graph including neighbourhood graphs associated with regions in three or more dimensions.
- 2. Once an AGACM has been constructed it can be treated as though it were a GLCM and hence the usual GLCM features can be computed from it.
- 3. To find the k-adjacent neighbours of a given vertex v using only graph dilation (see Definition 2.10.7):

Objects/regions	Adjacency graph	Attributes
pixels	grid; e.g. 4—or 8-connected square grid, 6-connected hexagonal grid	grey value
mosaic of regions; e.g. flat zones, watershed regions	region adjacency graph	grey-value statistics, Minkowski functionals, number of neighbours, statistics of the distance to neighbours
disjoint connected components; e.g. chromatin particles	geometric adjacency graph; e.g. Delaunay, perceptual	grey-value statistics, Minkowski functionals, number of neighbours, statistics of the distance to neighbours

Table 5.2: Examples of planar grey-scale graphs (adjacency graph + attribute) for which an adjacency graph attribute co-occurrence matrix can be defined.

- (a) construct the binary graph $X \mid G$ by setting $f \mid G$ equal to 1 at vertex v and to 0 elsewhere; and
- (b) compute the set difference $\delta^k(X|G) \setminus \delta^{k-1}(X|G)$.

This is illustrated in Figure 5.20

Table 5.2 lists several planar grey-scale graphs from which an AGACM can be constructed.

5.8 Measurement of region attributes

The quantitative measurement of image objects is a two-step process: "geometrical transformations and then measurements" (Serra, 1986, p. 292). The geometric transformations serve to partition the domain of the image, \mathbb{E}^n , into disjoint sets $R_1, R_2, \ldots, R_m \subset \mathbb{E}^n$ locating the objects/regions to be measured. This is classically referred to as segmentation (see Section 4.1). Measurement is then the process of evaluating one or more numerical parameters associated with these regions. Such parameters quantitatively describe various region attributes (properties). They include (Bengtsson & Nordin, 1994; van der Heijden, 1994):





Figure 5.20: Example of k-adjacency for k = 3. (a) Binary graph X | G with the vertex of interest set to Boolean 1. (b) $\delta^2(X | G)$. (c) $\delta^3(X | G)$ (d) The 3-adjacent neighbours of the vertex of interest: $\delta^3(X | G) \setminus \delta^2(X | G)$.

- 1. *Morphometric* parameters for describing the geometry—shape, size, boundary, position, and orientation—of a region. These parameters can be computed from the binary mask (silhouette) of a region.
- 2. *Radiometric/densitometric* parameters for describing the irradiance (intensity) or optical density of a region. These are computed from the grey-level histogram of the region.
- 3. *Texture* parameters for describing the spatial variation of grey-levels within a region.
- 4. *Relational/contextual* parameters for describing a region's relation to other regions; e.g. the number of neighbours it has, and statistics of the distances to neighbouring regions.

Compendia of such parameters (features) for two-dimensional grey-scale images can be found in Bengtsson & Nordin (1994) and Palcic et al. (2000). Although these compendia describe features for characterising the nucleus and/or cytoplasm of cells, they can be applied more generally to other image objects including chromatin particles. However, given the large number of features that have been devised—e.g. Palcic et al. (2000) list more than 70 different features—there must exist a high degree of statistical dependence between sets of features. With regard to morphometric parameters and radiometric parameters, however, integral geometry provides a powerful result concerning measurements that satisfy a few basic properties. Specifically, any such measurement can be written in terms of a linear combination of just a few of them called the *Minkowski functionals*.

5.8.1 Minkowski functionals

"The mind imagines concepts such as surface area or width of a body, only by more or less implicit reference to convex figures" (Serra, 1982, p. 93). Hadwiger (1957) proved that any numerical parameter that can be defined for a compact¹⁶ convex¹⁷ set (called an *ovoid*), and satisfying a few strong invariance properties (discussed below), can be expressed in terms of a linear combination of just a few of them called the *Minkowski functionals* (also called the *quermass integrals*). In \mathbb{R}^n there exist n + 1 Minkowski functionals. They are defined by a recurrence relation on

 $^{^{16}}$ See Footnote 6.

 $^{^{17}}$ A set $X \subset \mathbb{R}^n$ is said to be convex if the line segment joining any two points of X lies wholly within X.

sub-dimensions of the space (see Appendix F). The k-th functional is denoted $W_k^{(n)}$. Table 5.3 lists the functionals for the spaces \mathbb{R}^0 , \mathbb{R} , \mathbb{R}^2 , and \mathbb{R}^3 . The functional $N^{(i)}(X)$ is 1 if $X \neq \emptyset$ and 0 if $X = \emptyset$ (Serra, 1982, p. 103). The Minkowski functionals satisfy the following properties:

• isometry invariance:

$$W_{k}^{(n)}\left(\tau\left(X\right)\right) = W_{k}^{(n)}\left(X\right)$$
 where $\tau: \mathbb{R}^{n} \to \mathbb{R}^{n}$ is an isometry¹⁸;

• increasingness:

$$X \subseteq Y \; \Rightarrow \; W_k^{(n)}\left(X\right) \le W_k^{(n)}\left(Y\right);$$

- C-additivity: $W_k^{(n)}(X) + W_k^{(n)}(Y) = W_k^{(n)}(X \cup Y) + W_k^{(n)}(X \cap Y);$
- homogeneity: $W_k^{(n)}(\lambda X) = \lambda^{n-k} W_k^{(n)}(X), \quad \lambda > 0; \text{ and}$
- continuity:

$$d_{\mathrm{H}}(X,Y) \to 0 \Rightarrow W_{k}^{(n)}(X) \to W_{k}^{(n)}(Y).$$

where $d_{\rm H}$ is the Hausdorff metric (see the remark following Definition 5.3.1) and $X, Y \in P(\mathbb{R}^n)$ are ovoids. The Minkowski functionals in fact generalise to more arbitrary shapes. In particular they generalise to sets formed from the finite union of ovoids. The class of all such sets is called the *convex ring*. In the case of the convex ring, the functional $N^{(i)}(X)$ is the Euler-Poincarè constant (connectivity number); e.g. $N^{(1)}(X)$ is the number of line segments in $X \subset \mathbb{R}$. The convex ring serves as an archetype for binary images consisting of random collections of particles. Importantly, Hadwiger's characterisation result above also carries over to the convex ring; i.e. that all measurements that possess the above properties can be written as a linear combination of the Minkowski functionals.

¹⁸ An isometry of \mathbb{R}^n is a mapping of \mathbb{R}^n onto itself that preserves distances (DePree & Swartz, 1988, p. 268). Translations, rotations, and reflections (in lines) are examples of isometries in \mathbb{R}^2 (Allenby, 1991, p. 233).

k = 0k = 2k = 3k = 1 $\frac{4}{3}\pi N^{(3)}$ $\frac{1}{3}S$ n = 3V $\frac{1}{3}M$ Dimension of $\frac{1}{2}U$ $\pi N^{(2)}$ n=2Α the space \mathbb{R}^n L $2N^{(1)}$ n = 1 $N^{(0)}$ n = 0

Table 5.3: Minkowski functionals $W_k^{(n)}$ in \mathbb{R}^n (Serra, 1982, p.104).

N denotes connectivity number, L length, A area,

Minkowski functional of number k

V volume, U perimeter,

S surface area, and M norm.

5.8.2 Estimates of the Minkowski functionals for $X \in \mathcal{P}\left(\mathbb{Z}^2\right)$

Estimation of the Minkowski functionals of $X \in \mathcal{P}(\mathbb{Z}^2)$ amounts to estimating A(X), U(X), and $N^{(2)}(X)$. The estimate of $N^{(2)}$ derives directly from Euler's formula for a disconnected plane graph (Wilson, 1985, p. 66):

$$v - e + f = k + 1$$

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 graph divides the plane into, including one infinite region), and k is the number of connected components (connected graphs). In the case of a finite set X manifested on the hexagonal grid, the grid itself is a planar graph (the grid points are the vertices and the grid edges are the graph edges). In the case of the square grid, if X is deemed to be 4-connected then again the grid is a planar graph. However, if X is deemed to be 8-connected then the grid is not a planar graph and Euler's formula is not applicable. This is remedied if the the diagonal edges associated with instances of $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ are omitted (see Figure 5.21). It is then easy to show—e.g. Mehnert (1994, Chapter 4)—that the connectivity number



Figure 5.21: Estimating the connectivity number of a finite set X (large dots) on the square lattice. (a) The planar graph resulting if 4-connectivity is adopted. (b) The non-planar graph resulting if 8-connectivity is adopted. (c) The modification of (b) to be planar so that Euler's formula can be used.

of a set X manifested on a 4-connected square grid, an 8-connected square grid, and the hexagonal grid, respectively, can be written (Serra, 1982, p. 201):

$$N_4(X) = \mathcal{N} \left\{ \begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right\} - \mathcal{N} \left\{ \begin{array}{cc} 1 & 1 \\ 1 & 0 \end{array} \right\}$$
(5.1)

$$N_8(X) = \mathcal{N} \left\{ \begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right\} - \mathcal{N} \left\{ \begin{array}{cc} \cdot & 1 \\ 1 & 0 \end{array} \right\}$$
(5.2)

$$N_{\rm H}(X) = \mathcal{N} \left\{ \begin{array}{cc} 0 & 0 \\ & 1 \end{array} \right\} - \mathcal{N} \left\{ \begin{array}{cc} 0 \\ 1 & 1 \end{array} \right\}, \tag{5.3}$$

where $\mathcal{N} \{\bigstar\}$ means the number of occurrences of \bigstar .

To estimate A(X) it is necessary only to count the number of foreground pixels (value '1') in the set and to multiply this by the area of a single pixel: a^2 on the square grid with spacing a, and $\frac{2}{\sqrt{3}}a^2$ on the hexagonal grid with vertical spacing a. The area estimate for the square grid and the area estimate for the hexagonal grid, respectively, can be written:

$$A_{\rm S}\left(X\right) = \mathcal{N}\left\{1\right\}a^2\tag{5.4}$$

and

$$A_{\rm H}(X) = \mathcal{N}\{1\} \frac{2}{\sqrt{3}}a^2.$$
 (5.5)

Serra (1982, p. 105) estimates U(X) using a digital interpretation of Crofton's formula for \mathbb{R}^2 . This formula relates the perimeter of X to its sections viz.

$$\frac{1}{\pi}U(X) = \frac{1}{\pi} \int_0^{\pi} d\alpha \int_{-\infty}^{+\infty} N^{(1)} \left[X \cap \Lambda\left(y,\alpha\right)\right] dy$$
(5.6)

where $\Lambda(y, \alpha)$ is a test line with direction α and passing through the point $y \in \mathbb{R}^2$, and the ordinate y is on an axis normal to Λ . Crofton's formula can be written (Serra & Laÿ, 1985, p. 11):

$$U(X) = \pi \left[\frac{1}{\pi} \int_0^{\pi} D_\alpha(X) \, d\alpha\right],\,$$

where $D_{\alpha}(X) = \int_{-\infty}^{+\infty} N^{(1)} [X \cap \Lambda(y, \alpha)] dy$. The part of the formula within the square brackets is an average over all possible directions α . In \mathbb{Z}^2 the number of directions α is limited by the underlying grid. Figure 5.22 shows four possible test directions for the square lattice and three possible test directions for the hexagonal lattice based on the elementary (first-order) neighbourhood. The differential dyis approximated by Δy . On the square lattice this leads to the approximations: $D_0^*(X) = a\mathcal{N}\left\{\begin{array}{cc} 0 & 1 \end{array}\right\}, \ D_{\frac{\pi}{2}}^*(X) = a\mathcal{N}\left\{\begin{array}{cc} 1 \\ 0 \end{array}\right\}, \ D_{\frac{\pi}{4}}^*(X) = \frac{a}{\sqrt{2}}\mathcal{N}\left\{\begin{array}{cc} 0 & \cdot \\ \cdot & 1 \end{array}\right\}, \text{ and}$ $D_{\frac{3\pi}{4}}^{*}(X) = \frac{a}{\sqrt{2}} \mathcal{N} \left\{ \begin{array}{c} \cdot & 1\\ 0 & \cdot \end{array} \right\}$ (see Figure 5.23), and to the following unbiased estimate

of U proposed by Serra & Laÿ (1985, p. 12) ¹⁹:

$$U_{\rm S}^*(X) = \frac{\pi a}{4} \left[\left(\mathcal{N} \left\{ \begin{array}{cc} 0 & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 1 \\ 0 \end{array} \right\} \right) + \frac{1}{\sqrt{2}} \left(\mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ \cdot & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} \cdot & 1 \\ 0 & \cdot \end{array} \right\} \right) \right]. \quad (5.7)$$

Similarly, for the hexagonal lattice with vertical spacing a, this yields the following estimate of U^{20} :

$$U_{\rm H}^*(X) = \frac{\pi a}{3} \left[\mathcal{N} \left\{ \begin{array}{cc} 0 & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 1 \\ 0 & \cdot \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ 1 & 1 \end{array} \right\} \right].$$
(5.8)

¹⁹ The formula proffered by Serra (1982, p. 228, H.7.) is positively biased. It appears that the

constant $\frac{\pi}{3} \approx 1$ rather than $\frac{\pi}{4}$ has been used. ²⁰ The formula proffered by Serra & Laÿ (1985, p. 12) is incorrect. It incorrectly specifies the constant $\frac{\pi}{3}$ where $b = \frac{2a}{\sqrt{3}}$.



Figure 5.22: Elementary test-line directions for the square lattice and the hexagonal lattice.



Figure 5.23: Estimating $D_{\alpha}(X)$ on the square lattice. The test line $\Lambda\left(y, \frac{\pi}{4}\right)$ intersects the set X ten times. If the lattice spacing is a then $\Delta y = \frac{a}{\sqrt{2}}$ and an estimate of $D_{\frac{\pi}{4}}(X)$ is $D_{\frac{\pi}{4}}^{*}(X) = \frac{a}{\sqrt{2}} \mathcal{N} \left\{ \begin{array}{c} \cdot & 1\\ 0 & \cdot \end{array} \right\}$.

Serra (1982, p. 222) suggests a possible refinement to these estimates based on a doubling of the number of test directions. To do this it is necessary to consider a second-order neighbourhood as shown in Figure 5.24. This leads to the following estimates of U (not published in the literature) for the square and hexagonal grids respectively:

$$U_{\rm S}^{**}(X) = \frac{\pi a}{8} \left[\left(\mathcal{N} \left\{ \begin{array}{cc} 0 & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 1 \\ 0 \end{array} \right\} \right) + \\ & \frac{1}{\sqrt{2}} \left(\mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ \cdot & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} \cdot & 1 \\ 0 & \cdot \end{array} \right\} \right) + \\ & \frac{1}{\sqrt{5}} \left(\mathcal{N} \left\{ \begin{array}{cc} \cdot & 1 \\ \cdot & \cdot \\ 0 & \cdot \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} \cdot & \cdot & 1 \\ 0 & \cdot & \cdot \end{array} \right\} + \\ & \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot & \cdot \\ \cdot & \cdot & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ \cdot & \cdot \\ \cdot & 1 \end{array} \right\} \right) \right]$$
(5.9)

and

$$U_{\rm H}^{**}(X) = \frac{\pi a}{6} \left[\left(\mathcal{N} \left\{ \begin{array}{cc} 0 & 1 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 1 \\ 0 & \cdot \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ 1 & \cdot \end{array} \right\} \right) + \frac{1}{\sqrt{3}} \left(\mathcal{N} \left\{ \begin{array}{cc} 1 \\ \cdot \\ 0 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} \cdot \\ 0 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ 0 \end{array} \right\} + \mathcal{N} \left\{ \begin{array}{cc} 0 & \cdot \\ \cdot \\ 0 \end{array} \right\} \right) \right] \right] . \quad (5.10)$$

Serra (1982, p. 222) notes that the quality of estimation afforded by doubling the number of directions is not necessarily any better because of the bias introduced by considering a larger elementary neighbourhood.

5.8.3 Grey-scale images and dimensional measurements

Recall that a two-dimensional grey-scale image f, with non-negative grey-values, can be represented as a topographic relief if each grey-level is considered to be a height (see Figure 4.11). The points lying on and below the surface of the relief constitute a three-dimensional set called the *subgraph* of f. Formally, the subgraph of a two-dimensional grey-scale image $f \in \operatorname{Fun}(\mathbb{R}^2, \mathbb{R}^+_0)$, where $\mathbb{R}^+_0 = \mathbb{R}^+ \cup \{0\}$, is defined:

$$SG(f) \triangleq \left\{ (x,t) \in \mathbb{R}^2 \times \mathbb{R}_0^+ \middle| 0 \le t \le f(x) \right\}.$$



Figure 5.24: Doubling the test-line directions for the square lattice and the hexagonal lattice.

Equivalently, the SG (f) can be seen as a stacking of all of the cross-sections of f. The cross-section of f at level t is defined:

$$X_f(t) \triangleq \left\{ x \in \mathbb{R}^2 \, | \, f(x) \ge t \, \right\}.$$

Given that SG $(f) \subset \mathbb{R}^3$ it is possible to compute the four Minkowski functionals V, S, M, and $N^{(3)}$. If the grey-level axis represents a third spatial dimension—this situation occurs, for example, when studying a geographical relief (Serra, 1988c, p. 314)—then the space in which SG (f) resides is physically homogeneous (all the dimensions have the same physical meaning). In this case the Minkowski functionals all have physical meaning. If, as is usually the case, the grey-scale dimension represents the response of a sensor (light intensity, electrical intensity, etc.) then the space in which SG (f) resides is not physically homogeneous and not all of the Minkowski functionals nor all image measurements have physical significance. This problem has tended to be "overlooked by the computer vision community" (Rivest & Soille, 1995, p. 751). To be physically significant Rivest et al. (1992) and Soille et al. (1992) argue that a measurement should possess an additional property called *dimensionality.* A measurement on a grey-scale image is said to be dimensional if it can be related to the same measurement applied to this image after a scaling of the image plane and an independent scaling of the grey-scale axis (Soille et al., 1992, p. 127). If Λ_1 represents a scaling (magnification) of the image plane or domain of definition by a factor λ_1 , and Λ_2 represents a linear scaling of the grey-scale values with gain λ_2 and offset o, i.e. $(\Lambda_2 f)(x) = \lambda_2 f(x) + o$, then a measurement $W : \mathbb{R}^n \to \mathbb{R}$ is said to be dimensional if (Soille & Rivest, 1996, p. 218):

$$W\left(\Lambda_{1}\Lambda_{2}f\right) = \lambda_{1}^{k_{1}}\lambda_{2}^{k_{2}}W\left(f\right) + \lambda_{1}^{k_{1}}k_{o}W\left(\phi\right),$$

where $k_1, k_2 \in \mathbb{R}, k_o \in \{0, 1\}$ and ϕ is a two-dimensional grey-scale image, with the same domain as f, but only taking on the value o, i.e. $\phi(x) = o$. Dimensional measurements are robust to changes to image magnification at the time of capture and to changes in contrast/brightness. "In practice, however, this is only true for a given range of parameter modifications because both spatial and amplitude quantisations occur when sensing a continuous object" (Soille, 2003, p. 51). The volume Vand the connectivity number $N^{(3)}$ of SG (f) are dimensional measurements. When the domain of f resides in \mathbb{Z}^2 the estimate of the volume V is simply the sum of the grey-scale values. As for the connectivity number, Serra (1988c, p. 314) proposes that $N^{(3)}(f)$ be computed by summing the connectivity number $N^{(2)}$ over all cross-sections:

$$N^{(3)}(f) = \int_0^\infty N^{(2)}(X_t(f)) \, dt.$$

Furthermore he proffers formulae for the hexagonal and square grids that permit the calculation of $N^{(3)}(f)$ without having to threshold the image.

Rivest & Soille (1995, p. 752) note that an often used method for computing the surface area of a three-dimensional object, based on approximating it by planar triangles, is not dimensional. A dimensional method for computing the surface area is to compute the volume of the morphological gradient²¹ (also called Beucher's gradient) of f (Rivest & Soille, 1995, p. 752):

$$S(f) = V\left(\delta_B(f) - \varepsilon_B(f)\right) \tag{5.11}$$

where B is the structuring element representing the unit disk.

The histogram of grey-levels of an image derives from the notion of V(f) (Rivest et al., 1992, p. 140) and hence statistics of the histogram are dimensional.

5.8.4 Other parameters/functionals

Serra (1982, p. 110) notes that the C-additivity property should be thought of as the link needed to extend the Minkowski functionals to the convex ring, and "not a sine qua non condition for experiments". The P2A shape factor defined

$$P2A(X) = \frac{\left[U(X)\right]^2}{4\pi A(X)},$$

for example, is a widely used quantitative measure of shape (normalised to be 1 for a circle) and yet it is not C-additive²². Other such functionals include (Rivest et al., 1992, p. 138): convexity number, roughness, and fractal dimension²³.

²¹ The morphological gradient is itself dimensional (Soille et al., 1992, p. 133).

 $^{^{22}}$ The ratio or product of two Minkowski functionals is not C-additive.

²³ Soille & Rivest (1996) review several alternative algorithms for computing the fractal dimension and identify those that are dimensional.

5.9 Summary

This Chapter has

- Presented an overview of (geometric) adjacency graphs—region adjacency graph and graphs stemming from the Voronoi diagram—used in image processing.
- Reviewed the ordinary Voronoi diagram and graphs related to the planar Voronoi diagram: Delaunay graph, Gabriel graph, relative neighbourhood graph, minimum spanning tree, and β-skeletons.
- Reviewed a generalisation of the ordinary Voronoi diagram called the area Voronoi diagram. It was noted that the (Euclidean) distance transform is the key to extending the area Voronoi diagram to connected components of a binary image.
- Reviewed distance transform algorithms published in the literature.
- Established that the erosion of the characteristic function of a binary image $X \in \mathcal{P}(\mathbb{R}^n)$ by the elliptic poweroid structuring element $g(x) = -(x^{\mathsf{T}}\mathbf{A}x)^{\frac{\alpha}{2}}$, where \mathbf{A} is $n \times n$ positive definite and $\alpha \in \mathbb{R}^+$, is equivalent to finding the distance transform of X, based on the distance $d(x, y) = \sqrt{(x-y)^{\mathsf{T}}\mathbf{A}(x-y)}$, raised to the power α . Moreover it was shown that the EDT algorithm of Huang & Mitchell (1994), for binary images manifested on the square grid, corresponds to the case when n = 2 and $\alpha = 2$.
- Presented a new method for computing the EDT on the hexagonal grid. The method is a procedural extension to the EDT algorithm of Huang & Mitchell (1994). The extension involves embedding the input binary image, manifested on a hexagonal grid, in a rectangular grid. Huang and Mitchell's algorithm can then be applied, with the appropriate aspect ratio, to obtain the distance transform on the rectangular grid. The EDT on the hexagonal grid is then obtained by discarding those distance values that do not correspond to a point on the original hexagonal grid. The algorithm is both amenable to hardware implementation using a pipeline architecture and efficient implementation on serial machines. Moreover the algorithm requires only integer arithmetic.
- Presented an overview of the skeleton by influence zones (which is the area Voronoi diagram for connected components in a binary image), the RAG constructed on the influence zones (which generalises the Delaunay graph), and the perceptual graph (which generalises the Gabriel graph).

- Presented a generalisation of the well-known grey-level co-occurrence matrix method (GLCM) to vertex-weighted adjacency graphs (grey-scale graphs). The image under study is first reduced to an adjacency graph with vertices corresponding to individual regions (objects) and edges corresponding to an adjacency relationship between regions. Next, region attributes—average grey-level, area, perimeter, etc. —are assigned to each vertex of the adjacency graph. Finally, for each attribute a co-occurrence matrix—called an adjacency graph attribute co-occurrence matrix (AGACM)—is defined. Co-occurrence matrix features can then be used to quantitatively describe the arrangement of the regions. Of particular interest is that the AGACM method can be used to characterise blob-like and mosaic patterns in the plane; e.g. chromatin particles.
- Presented an overview of the types of parameters (attributes) that can be measured for image objects. The convex ring—the class of sets in Rⁿ whose elements are finite unions of compact convex sets—provides a realistic Euclidean model for digital images. The Minkowski functionals form the basis of any valid measurement that can be made on compact convex sets. Estimators for the Minkowski functionals for two-dimensional binary and grey-scale images were presented for both the square and hexagonal grids. Corrections to the literature, as well as new estimators for the perimeter were proffered.

Chapter 6

Application to the Automated Screening of Cytology Slides

There are two schools: one, of pattern recognition in which many measurements, 20-100 per cell structure, are analyzed statistically to find which are the best for recognizing a pattern... A completely different approach [is to]... extract features by image transformation, and if the feature we extract is satisfying to the eye and to the mind, then we make the measurement

Fernand Meyer, 1980

The chromatin segmentation, representation and description methods presented in the preceding chapters make it possible to extract features that quantitatively describe nuclear chromatin distribution (pattern) as visualised by light microscopy. Possible applications of these features include (Mehnert & Jackway, 2002): artefact rejection, the detection of malignancy associated changes (MACs), and the detection of nuclear changes during neoplasia. Recent research by Rousselle et al. (1999) suggests that chromatin segmentation features might also be used to assess the chromatin patterns



in living cells during the cell cycle. This would make it possible to measure the alterations in the evolving chromatin patterns that result from pathological or environmental influences.

This chapter considers one possible application: the automated screening of cytology slides. The aim is not to mimic the method of screening used in conventional manual screening—i.e. to exhaustively search for biologically abnormal cells on a slide—but rather to distinguish slides containing wholly normal cells from slides containing abnormal cells using statistical pattern recognition and chromatin segmentation features. Specifically, this chapter considers the automated screening of Papanicolaou-stained cervical smears (for the reasons outlined in Chapter 1). Several studies reported in the literature have demonstrated that MACs can be detected in cervical smears that have been stained with a stoichiometric stain (see Section 6.4.9.1). For a stoichiometric stain, such as Thionin-Feulgen, the amount of stain uptake in the nucleus is proportional to the amount of DNA present. Consequently it is possible to measure chromatin variation by measuring the grey-tone (optical density) variation within the nucleus. Unfortunately, the Papanicolaou stain is not stoichiometric (Schulte & Wittekind, 1994, p. 202). Nevertheless the Papanicolaou stain "is still the stain of choice for visual screening and control of...slides" (Schulte & Wittekind, 1994, p. 208). Indeed, worldwide it is the most commonly used cytological staining technique in gynaecology (Schulte & Wittekind, 1994, p. 201). Consequently there exists a significant advantage in being able to automatically screen conventional Pap-stained cervical smears.

The remainder of this chapter is organised as follows. Section 6.1 examines the rationale behind a MACs-based screener (classifier). Section 6.2 examines how such a classifier can be designed and motivates the statistical pattern recognition approach. Section 6.3 presents an overview of statistical pattern recognition including important issues such as the *curse of dimensionality*, dimensionality reduction, choice of classifier, and evaluation of classifier performance. Section 6.4 presents a case study involving the application of statistical pattern recognition and chromatin segmentation features to the problem of Pap smear screening. In particular the results of two experiments, on 40 abnormal and 99 normal Pap smear slides collected as part of the (Australian) National Cervical Screening Program, are reported. The first experiment evaluates the performance of a classifier trained and evaluated using features purposefully designed to measure *chromatin margination*. The second experiment determines a subset of chromatin features that have the most discriminatory power for the Pap smear classification problem.

6.1 Why build a MACs-based screener?

When cytotechnologists screen a specimen (slide) for biological abnormalities they exhaustively review all of the cells on the slide searching for perhaps only a very few clearly abnormal cells (called *diagnostic cells*). If none are found the slide is classified as *negative* (normal), otherwise it is classified as *positive* (abnormal). This rare event (RE) approach to screening gives rise to several scenarios in which a specimen may be incorrectly classified as negative (a false negative):

- 1. Diagnostic cells may be overlooked or misinterpreted by the cytotechnologist (e.g. because of fatigue or inexperience);
- 2. Diagnostic cells may be obscured by other material such as blood, dust, and mucous; and
- 3. No diagnostic cells may have made it to the slide from the patient in the first place (sampling error).

"Many, perhaps most, false negatives represent sampling errors, where a sample of abnormal cells from the patient fails to make it onto the glass slide" (DeMay, 1997, p. 230). With regard to automated screening, the RE approach presents a formidable challenge for state-of-the-art technology and image analysis techniques (Bengtsson & Nordin, 1994, p. 41). The hardware (robotics, camera, framegrabber, etc.) must be able to scan a specimen with sufficient speed and produce images of sufficient quality to permit the evaluation of the specimen within a reasonable amount of time; e.g. "it is generally suggested, on the basis of current economic considerations, that a [primary screening] system should take no longer than three to four minutes to process a slide" (Bartels & Vooijs, 1999, p. 9). The software must be able to accurately locate, focus, segment, and measure every cell on the specimen. Complicating factors include the type of stain (e.g. Papanicolaou, Thionin-Feulgen) and preparation (e.g. smear, liquid-based preparation), the presence of artefacts—such as dust, blood, and mucous—, and coverage and registration problems associated with scanning the slide to produce a mosaic of field-of-view images. The literature suggests that an alternative screening approach, based on a phenomenon known as malignancy associated changes (MACs), can be used to build an automated screener that does not suffer from the drawbacks associated with the RE approach. MACs, in modern usage, refers to subtle subvisual changes—predominantly textural—in otherwise normal-appearing cells on cervical atypical smears. "These alterations are too insignificant to be analyzed on a cell-by-cell basis; instead, populations of cells must be analyzed and the population parameters (means, variances) used to classify the smear" (Nordin & Bengtsson, 1994, p. 44). Using image cytometry MACs have been "found in different tissues, including the epithelia of the oral cavity, the gastrointestinal tract, the bronchial system, the hematopoietic system, and the cervix uteri" (Kasper, Haroske, Geissler, Meyer & Kunze, 1997, p. 482).

In summary the MACs approach offers the potential for reliably classifying a cytological specimen without the need for scrutinising all of the cells on the slide—as must be done by a human screener or RE-based automated screener—or for detecting any diagnostic cells that may be present. However, there is a caveat: "with current techniques, not all cancer patients display MACs, and some apparently healthy individuals do" (Hallinan, 2000, p. 58). Unfortunately the reason for this is not yet understood because "the biologic nature of MACs... is not yet fully understood" (Kasper et al., 1997, p. 483). The implication of this is that there may exist a fundamental limit to the performance than can be expected from a MACs-based classifier.

6.2 How to build a MACs-based classifier?

Conceptually, to build a classifier to discriminate between normal (not MAC-affected) slides and abnormal (MAC-affected) slides involves testing, for a given slide, "the hypothesis that the cell sample found on the slide was drawn from the gigantic population of all normal cells" (Nordin & Bengtsson, 1994, p. 48). There are two fundamentally different approaches that can be taken: "Either each cell is assigned an atypia index and the distribution of this index is analyzed, or the distributions—perhaps even the values—of the various [computed] cell features are analyzed" (Nordin & Bengtsson, 1994, p. 48). The Cytometrics Project has adopted the latter approach because (Nordin & Bengtsson, 1994, p. 48):

a single atypia index for all cell types is probably too simple a strategy... The various types of cells found on a typical slide simply differ too much, even if they are all normal.

In practice the aforementioned hypothesis must be tested on the basis of only a subsample of the cells found on the slide. There are several reasons for this including:
- 1. The fact that no automated image analysis system (cytometer) can hope to accurately locate, segment, and measure each and every cell on the slide. This is especially true for Papanicolaou-stained slides; and
- 2. For commercial viability a single slide must be processed within a fixed amount of time—e.g. 4 minutes—and cost/hardware constraints may preclude the exhaustive scanning of the slide.

As noted in Chapter 1 the feature-based approach to classification is called *statistical* pattern recognition. The term pattern refers to the object or entity of interest; e.g. a cytology slide. In this approach (Jain, Duin & Mao, 2000, p. 6):

each pattern is represented in terms of d features or measurements and is viewed as a point in a d-dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated.

In relation to designing a MACs-based classifier for cytology slides the pattern classes are, in the simplest case, *normal* and *abnormal*¹. The efficacy of such a classifier depends on the discriminatory power of the MACs phenomenon and on the feature set used to quantify it.

6.3 Overview of statistical pattern recognition

For a recent and comprehensive review of the field of statistical pattern recognition the reader is referred to Jain et al. (2000). The classic texts on the subject include Duda & Hart (1973), Devijver & Kittler (1982), and Fukunaga (1990). Duda & Hart have recently released a revised and much updated second edition, Duda, Hart & Stork (2001), of their original monograph. Other monographs on the subject include McLachlan (1992) which is specifically a treatment of multivariate statistical methods, and Ripley (1996) which includes treatment of neural networks within a statistical framework.

¹ If the MACs phenomenon, or the feature set used to quantify it, is sufficiently discriminatory then it may be possible to design a MACs-based classifier with several pattern classes; e.g. normal, mild dysplasia, moderate dysplasia, and severe dysplasia and carcinoma in situ.

From the point of view of statistical pattern recognition, the problem of designing a classifier is one of estimating density functions in high-dimensional space and partitioning this space into the regions of interest (Fukunaga, 1990, p. 3). These probability distributions (class densities) determine the decision boundaries that separate the classes in feature space. If these distributions are completely specified, i.e. the class densities and the prior probabilities are known, then the theoretically best classifier is the *Bayes classifier* (also called *Bayes rule*) because it minimises the probability of classification error (Fukunaga, 1990, p. 3). In practice, however, these are usually not known and must be estimated (learnt) from a set of example patterns (feature vectors) called the training set. The techniques used to estimate these densities can be broadly classified into *supervised* and *unsupervised* learning. In the former case the samples in the training set are labelled, i.e. the class membership of each observation is known. In the latter they are not and must be "learned along with the structure of each class" (Jain et al., 2000, p. 9). In both cases, the specific strategy used to estimate the class densities depends on whether their particular distributional forms are known (or can be assumed); e.g. multivariate normal. If they are then the estimation problem reduces to one of estimating the parameters of the distributional forms from the training data (*parametric estimation*). If they are not then an unstructured approach to estimation must be used (non-parametric *estimation*). Ripley (1996, p. 4) points out that:

the traditional methods of statistics and pattern recognition are either *parametric* based on a family of models with a small number of parameters, or *non-parametric* in which the models used are totally flexible. One of the impacts of neural network methods on pattern recognition has been to [offer]...something in between, families of models with large but not unlimited flexibility given by a large number of parameters.

In the case of supervised learning a common strategy for parametric estimation "is to replace the unknown parameters in the density functions by their estimated values, resulting in the so-called Bayes plug-in classifier" (Jain et al., 2000, p. 9). The most commonly used such classifiers are *linear* and *quadratic* classifiers. In multivariate statistical analysis such classifiers fall under the heading of *discriminant analysis*. Discriminant analysis techniques are closely related to multivariate linear regression models and generalised linear models (encompassing *logistic regression* discussed in Section 6.4.7). With regard to non-parametric estimation in supervised learning, two basic strategies exist: either the density functions are estimated from the training data (e.g., Parzen window approach), or the decision boundary is constructed directly from the training data (e.g., k-nearest neighbour rule) (Jain et al., 2000, p. 9).

In contrast to supervised learning, in the case of unsupervised learning the number of classes are usually not a priori known and must be estimated. In addition, "unsupervised methods are generally designed for *visualization*, either to show views of the data which indicate groups, or to show affinities between the examples by displaying similar examples close together" (Ripley, 1996, p. 287). If the particular distributional forms of the class-conditional densities are known or can be assumed then *mixture decomposition* (Jain et al., 2000, p. 30) can be used to estimate their parameters from the unlabelled data. This approach is based on the idea that "each pattern has been produced by one of a set of alternative (probabilistically modeled) sources" (Jain et al., 2000, p. 30). Generally, however, the structure of the class-distributional densities is unknown and other *cluster analysis* methods must be used. Broadly speaking, cluster analysis methods identify clustering or grouping in training patterns without making assumptions about the number of classes or the structure of each class. Classes are identified on the basis of similarity or dissimilarity measured in terms of distances (Johnson & Wichern, 1988, p. 543). In multivariate statistics, statistical distance $d(x,y) = \sqrt{(x-y)^{\mathsf{T}} \mathbf{A}(x-y)}$ (see also Proposition 5.5.1) is used to measure the distance between two multivariate observations (patterns) x and y. Ordinarily $\mathbf{A} = \mathbf{S}^{-1}$ where **S** is the sample covariance matrix. However, without prior knowledge of the distinct groups (classes) the covariance matrix cannot be calculated (Johnson & Wichern, 1988, p. 545). Consequently **A** is often set to be the identity matrix yielding the Euclidean distance. Other distance functions can be defined, although it is advisable to use distance functions satisfying the properties of a metric (see Section 2.9) (Johnson & Wichern, 1988, p. 545). An elementary and very popular clustering method is the k-means clustering algorithm (Duda et al., 2001, p. 526). In its simplest form the algorithm has three steps (Johnson & Wichern, 1988, p. 566):

- 1. Randomly partitioning the original data into k initial clusters (classes) and computing the centroid (mean) for each;
- 2. Assigning each pattern to the cluster that has the nearest centroid (mean) and immediately updating the centroid of the cluster receiving the pattern and of the cluster losing the pattern; and
- 3. Repeating step 2 until no further assignments can be made.



Figure 6.1: Steps involved in building a statistical pattern recognition system (adapted from Jain et al. (2000, p. 8)).

The steps involved in building a statistical classifier (pattern recognition system) are shown in Figure 6.1. Two modes of operation are illustrated: training (learning) and testing (classification). The preprocessing step encapsulates all of the operations needed to establish a representation/description of the pattern (object) of interest that is suitable for subsequent feature extraction. In the case of a cytology slide (pattern), for example, this step might include the segmentation of all of the cell nuclei in a set of digitised field-of-view images from a light microscope, converting the grey-values to optical density values (normalisation), the computation of nucleus features such as circularity, perimeter, area, volume, and texture features, and the subsequent computation of slide summary statistics (e.g. mean and standard deviation) for each nucleus feature. The feature extraction step involves the computation of new features from this representation/description "based on transformations or combinations of the original feature set" (Jain et al., 2000, p. 12). In the cytology slide example such features might include the principal components (Johnson & Wichern, 1988, p. 340) of the slide statistics. In the training mode, feature extraction is combined with feature selection to determine the best subset of features for discriminating between the different pattern classes. In the case of supervised learning, where the true class label of each pattern is known a priori, the feedback loop permits the feature extraction/selection strategies to be optimised. In the classification mode, a new unseen pattern is classified on the basis of feature values computed for only the best subset of features (as determined in the training mode).

6.3.1 The curse of dimensionality

To build a statistical classifier it is necessary to obtain sample data with which to train and test the classifier. The sample must contain observations representative of each class. The size of the sample dictates the number of features that can sensibly be used to build a classifier to discriminate between the classes. For a given, fixed, sample size, increasing the number of features has two conflicting effects (Smith, 1998, p. 103):

- 1. The additional features provide more information and therefore should improve classification performance; and
- 2. The feature space increases in dimensionality and as a consequence the observations in each class become less representative of that class (because more observations are required to form a representative sample in high-dimensional space than in low-dimensional space). As a result, classification accuracy on the training set increases but the performance on unseen data decreases.

This phenomenon is called the *curse of dimensionality* (also called the *peaking phenomenon*). It (Smith, 1998, p. 103):

has been described theoretically and observed empirically. For fixed sample size and increasing dimensionality [i.e. increasing number of features], the usual pattern is for classification accuracy to increase to a peak, and thence to decrease to poor accuracy. It is most significant for statistically dependent features, though it can occur in independent features. The phenomenon occurs for all forms of features, including continuous features, quantized features, and non-numeric discrete features. The phenomenon occurs in both parametric and non-parametric classification techniques.

As a general rule of thumb (Jain et al., 2000, p. 11):

it is generally accepted that using at least ten times as many training samples [patterns] per class as the number of features... is a good practice to follow in classifier design. The more complex the classifier, the larger should the ratio of sample size to dimensionality be to avoid the curse of dimensionality.

6.3.2 Dimensionality reduction

There are two main reasons for wanting to reduce the dimensionality of feature space; i.e. the number of features to be used for classification. The first is to avoid the curse of dimensionality. The second is to reduce the measurement and computational cost. Two commonly used approaches to feature set reduction are feature extraction and feature selection. Devijver & Kittler (1982, p. 192) point out that "ideally, the problem of feature selection and extraction on the one hand and the classifier design [model selection] on the other hand should not be considered independently". Indeed Ripley (1996, p. 327) notes that feature selection and extraction methods "are being supplanted by model selection methods".

6.3.2.1 Feature extraction

The aim of feature extraction is to reduce the dimensionality of the original feature space by combining (either in a linear or non-linear fashion) the initial set of features (measurements). Multivariate statistical analysis is a source of several linear feature extraction methods including *principal components analysis* (PCA), *factor analysis* (FA), and *linear discriminant analysis* (LDA). PCA is an unsupervised feature extraction method; i.e. class labels, if they exist, are not taken into account. FA and LDA, on the other hand, are supervised feature extraction methods. Other linear feature extraction methods include *projection pursuit* and *independent component analysis* (Jain et al., 2000, p. 12). Non-linear feature extraction methods include (Jain et al., 2000, p. 13):

- 1. The *Kernel PCA* method The input data are mapped into a new feature space using a prescribed non-linear mapping such as a polynomial of degree *p*. Standard PCA is then performed;
- 2. *Multidimensional scaling* The original feature space is projected into a twoor three-dimensional space in such a way that the distance matrix of the new feature space is as close as possible to that of the original feature space; and
- 3. Methods based on neural networks and self-organising maps.

6.3.2.2 Feature selection

The aim of feature selection methods is to find individual features that "are likely to have good discriminatory power" (Ripley, 1996, p. 327). More specifically the

aim is to find "the minimally sized feature set that is necessary and sufficient" for the classification problem at hand (Dash & Liu, 1997, p. 132). For a feature set containing d features the number of candidate subsets is $2^d - 1$ (excluding the empty set). An exhaustive search through all possible subsets of features is typically impractical because of the exponential increase in the number of subsets that must be evaluated as the number of features increases. The curse of dimensionality imposes an upper limit $m, m \leq d$, on the number of features that can be used. Even so the number of subsets to be searched is combinatorial:

$$\left(\begin{array}{c}d\\m\end{array}\right) + \left(\begin{array}{c}d\\m-1\end{array}\right) + \ldots + \left(\begin{array}{c}d\\1\end{array}\right)$$

Consequently, a variety of other strategies have been devised based on heuristic or random search methods (Dash & Liu, 1997, p. 132). Table 6.1 lists well-known feature selection methods found in the literature. All of these procedures have three steps in common (Dash & Liu, 1997, p. 132):

- 1. a generation procedure to generate the next candidate subset;
- 2. a *criterion function* to evaluate the discriminatory power of the subset under consideration; and
- 3. a stopping criterion to decide when to stop.

An obvious choice for the criterion function is the classification error rate. However, as a computational short cut, instead of fitting each new model to the data and evaluating its classification error rate, simpler criterion functions are used. These functions² seek to measure the class separability afforded by a set of features. The stopping criterion can be based on either the generation procedure or the criterion function. In the former case, the criteria include (Dash & Liu, 1997, p. 133):

- 1. stopping when the desired number of features have been selected; and
- 2. stopping after a fixed number of iterations.

In the latter case stopping criteria include (Dash & Liu, 1997, p. 133):

 $^{^2}$ Dash & Liu (1997) divide criterion functions into five different categories: distance measures, information measures, dependence measures, consistency measures, and classifier error rate measures.

Method	Property	Comments
Exhaustive Search	Evaluate all $\begin{pmatrix} d \\ m \end{pmatrix}$ possible subsets of size m .	Guaranteed to find the optimal subset; not feasible for even moderately large values of m and d .
Branch-and-Bound Search	Uses the well-known branch-and-bound search method; only a fraction of all possible feature subsets need to be enumerated to find the optimal subset.	Guaranteed to find the optimal subset provided the criterion function satisfies the mono- tonicity property; the worst- case complexity of this algo- rithm is exponential.
Best Individual Features	Evaluate all the d fea- tures individually; se- lect the best m indi- vidual features.	Computationally simple; not likely to lead to an optimal subset.
Sequential Forward Selection (SFS)	Select the best sin- gle feature and then add one feature at a time which in com- bination with the se- lected features maxi- mizes or minimizes the criterion function as the case may be.	Once a feature is retained, it cannot be discarded; computa- tionally attractive since to se- lect a subset of size 2, it exam- ines only $(d-1)$ possible sub- sets.
Sequential Backward Selection (SBS)	Start with all of the d features and successively delete one feature at a time.	Once a feature is deleted, it cannot be brought back into the optimal subset; requires more computation than se- quential forward selection.
"Plus <i>l</i> -take away <i>r</i> " Selection	First enlarge the fea- ture subset by l fea- tures using forward se- lection and then delete r features using back- ward selection.	Avoids the problem of fea- ture subset "nesting" encoun- tered in SFS and SBS meth- ods; need to select values of l and r $(l > r)$.
Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS)	A generalization of "plus l -take away r " method; the values of l and r are determined automatically and up- dated dynamically.	Provides close to optimal solu- tion at an affordable computa- tion cost.

Table 6.1: Feature selection methods (adapted from Jain et al. (2000, p. 16)).

- 1. stopping when the addition (or deletion) of a feature does not produce a better subset; and
- 2. stopping when the optimal value of the criterion function has been found.

In the traditional statistical approaches to feature (model) selection the *log-likelihood* evaluated at the *maximum likelihood* (ML) estimate is the most commonly used measure of fit (criterion function) (Ripley, 1996, p. 60). The *log-likelihood* function is the log of the *likelihood function*³. The likelihood function (Hosmer & Lemeshow, 2000, p. 8):

expresses the probability of the observed data as a function of the unknown parameters. The *maximum likelihood estimators* of these parameters are chosen to be those that maximize this function. Thus, the resulting estimators are those which agree most closely with the observed data.

This method of estimation coincides with the more familiar *least squares* method of estimation used in linear regression (under the additional (tentative) assumption that the errors of the fitted model have normal distribution) (Johnson & Wichern, 1988, p. 284).

"It is often more convenient to work with the *deviance*, minus twice the log-likelihood shifted to be zero for the 'perfect' model" (Ripley, 1996, p. 60). The deviance is defined (Hosmer & Lemeshow, 2000, p. 13):

$$D = -2 \ln \left[\frac{(\text{likelihood of the fitted model})}{(\text{likelihood of the saturated model})} \right]$$

"A saturated model is one that contains as many parameters as there are data points" (Hosmer & Lemeshow, 2000, p. 12). The traditional statistical approaches to feature selection fall into two camps (Ripley, 1996, p. 60):

1. Iterative feature selection.

There are three basic approaches: *forward*, *backward*, and *stepwise*. Forward selection starts with no features and progressively adds features one at a time. Backward selection (elimination) starts with all of the features and progressively removes one at a time. Stepwise selection starts with a set of features

 $^{^3}$ Mathematically it is usually easier to work with the log of the likelihood function (Duda et al., 2001, p. 86).

(not necessarily all of the features) and progressively either adds or drops a feature. "At any step in the procedure the most important variable [feature], in statistical terms, is the one that produces the greatest change in the log-likelihood relative to a model not containing the variable" (Hosmer & Lemeshow, 2000, p. 116). The statistical significance of the change is assessed using a likelihood ratio test, or equivalently a difference in deviances, for some a priori specified level of significance (Ripley, 1996, p. 60). The selection process stops when no more variables can be added or deleted, as the case may be.

2. Penalising fit.

These procedures are based on the idea of penalising the measure of fit according to the size of the model because "normally we would expect the largest models to fit best" (Ripley, 1996, p. 60). The most common such approach is "based on the idea that the deviance will be smaller on the training set than on a test set of comparable size, since we actually chose the parameters to minimize the deviance on the training set" (Ripley, 1996, p. 61). To take this into account, the deviance should be penalised. This is the basis of the AIC^4 (an information criterion). It is defined to be the deviance plus twice the number of parameters in the fitted model. A characteristic of the AIC is that it tends to choose models of larger and larger size as the size, n, of the training data set is increased (Ripley, 1996, p. 61). An alternative criterion, called the BIC (Bayesian information criterion), corrects for this tendency by applying a penalty of $\log n$ (rather than 2) times the number of parameters in the fitted model (Ripley, 1996, p. 65). The definition of AIC is motivated by information theory and stems from the Kullback-Liebler distance between distributions, whilst the definition of BIC is motivated by Bayesian theory and stems from integrated likelihood (Li & Nyholt, 2001). Whilst in principle it is necessary to search all models to find the best, in practice, for computational reasons, "we may have to confine the search to only some of the models in the family: this could even be done by a stepwise search" (Ripley, 1996, p. 60).

6.3.3 Choice of classifier

Duda et al. (2001, p. 454) state that:

⁴ "AIC was named by Akaike (1974) as 'An Information Criterion', although it seems commonly believed that the A stands for Akaike" (Ripley, 1996, p. 34).

If the goal is to obtain good generalization performance, there are no context-independent or usage-independent reasons to favor one learning or classification method over another. If one algorithm seems to outperform another in a particular situation, it is a consequence of its fit to the particular pattern recognition problem, not the superiority of the algorithm.

"In practice, the choice of a classifier is a difficult problem and it is often based on which classifier(s) happen to be available, or best known, to the user" (Jain et al., 2000, p. 17). This is often dictated by what is available in commercial and public domain software such as SAS⁵, MINITAB⁶, S-PLUS⁷, R⁸, and SNNS⁹. An important consideration in the choice of classifier should be whether or not the model fits the data; i.e. whether the model assumptions are met. For example, linear discriminant analysis is based on the assumption that the covariance matrix for each class is the same. Another consideration is interpretability. A *decision tree* is attractive because it partitions feature space in a hierarchical fashion thus making it possible to "interpret the decision rule in terms of individual features" (Jain et al., 2000, p. 19). There is a caveat however. Whilst classification trees are often easy to interpret they are not amongst the highest performers (Ripley, 1996, p. 10). Table 6.2 summarises commonly used classifiers.

6.3.4 Evaluating classifier performance

The classification error is the ideal measure of the performance of a classifier. In practice this must be estimated from the available data. The simplest approach is to partition the available data into two sets: a training set and a test set. The classifier is trained using the training set and tested using the test set. The estimate of the classification error (error rate) is then given by the proportion of misclassified test set samples. Jain et al. (2000, p. 25) note that:

⁵ SAS® is commercial software developed by the SAS Institute Inc., headquartered in Cary, North Carolina (http://www.sas.com).

⁶ MINITAB® is commercial software developed by Minitab Inc. (http://www.minitab.com).

⁷ S-PLUS® is commercial software developed by Insightful Corporation, headquartered in Seattle, Washington (http://www.insightful.com).

⁸ R is available as Free Software under the terms of the Free Software Foundation's (http: //www.gnu.org) GNU General Public License in source code form. The R homepage is located at http://www.R-project.org.

⁹ SNNS (Stuttgart Neural Network Simulator) is copyright University of Stuttgart. The SNNS homepage is located at http://www-ra.informatik.uni-tuebingen.de/SNNS.

Table 6.2: Commonly used classification methods (reproduced from Jain et al. (2000,p. 20)).

Method	Property	Comments
Template matching	Assign patterns to the most similar	The templates and the metric have to be supplied
	template.	by the user; the procedure may include nonlinear
		normalizations; scale (metric) dependent.
Nearest mean classifier	Assign patterns to the nearest class	Almost no training needed; fast testing; scale
	mean.	(metric) dependent.
Subspace method	Assign patterns to the nearest class	Instead of normalizing on invariants, the sub-
	subspace.	space of the invariants is used; scale (metric) de-
		pendent.
1-Nearest neighbour rule	Assign patterns to the class of the	No training needed; robust performance; slow
	nearest training pattern.	testing; scale (metric) dependent.
k-Nearest neighbour rule	Assign patterns to the major-	Asymptotically optimal; scale (metric) depen-
	ity class among k nearest neigh-	dent; slow testing.
	bours using a performance opti-	
	mized value for k .	
Bayes plug-in	Assign pattern to the class which	Yields simple classifiers (linear or quadratic) for
	has the maximum estimated poste-	Gaussian distributions; sensitive to density esti-
	rior probability.	mation errors.
Logistic classifier	Maximum likelihood rule for logis-	Linear classifier; iterative procedure; optimal for
	tic (sigmoidal) posterior probabili-	a family of different distributions (Gaussian);
	ties.	suitable for mixed data types.
Parzen classifier	Bayes plug-in rule for Parzen den-	Asymptotically optimal; scale (metric) depen-
	sity estimates with performance op-	dent; slow testing.
	timized kernel.	
Fisher linear discriminant	Linear classifier using MSE opti-	Simple and fast; similar to Bayes plug-in for
	mization.	Gaussian distributions with identical covariance
		matrices.
Binary decision tree	Finds a set of thresholds for a	Iterative training procedure; overtraining sensi-
	pattern-dependent sequence of fea-	tive; needs pruning; fast testing.
	tures.	
Perceptron	Iterative optimization of a linear	Sensitive to training parameters; may produce
	classifier.	confidence values.
Multi-layer perceptron	Iterative MSE optimization of two	Sensitive to training parameters; slow training;
(Feed-forward neural network)	or more layers of perceptrons (neu-	nonlinear classification function; may produce
	rons) using sigmoid transfer func-	confidence values; overtraining sensitive; needs
	tions.	regularization.
Radial basis network	Iterative MSE optimization of a	Sensitive to training parameters; nonlinear classi-
	feed-forward neural network with	fication function; may produce confidence values;
	at least one layer of neurons using	overtraining sensitive; needs regularization; may
	Gaussian-like transfer functions.	be robust to outliers.
Support vector classifier	Maximizes the margin between the	Scale (metric) dependent.; iterative; slow train-
	classes by selecting a minimum	ing; nonlinear; overtraining insensitive; good gen-
	number of support vectors.	eralization performance.

 $\mathbf{228}$

in order for this estimate to be reliable in predicting the future classification performance, not only should the training set and the test set be sufficiently large, but the training samples and the test samples must be independent. This requirement of independent training and test samples is still often overlooked in practice.

Independence is essential because when the same set is used for both training and testing, called *re-substitution* or *test-on-train*, the resulting estimate of classifier performance will be optimistically biased. To guarantee independence the available data should be partitioned into two disjoint sets. Unfortunately "there are no good guidelines available on how to divide the available samples into training and test sets" (Jain et al., 2000, p. 25). A small training set results in a classifier that is not likely to generalise to new unseen data. A small test set results in an error estimate that has a large variance; i.e. if the process is repeated over and over using different random splits of the available data, the variance of the error rate will be large. Table 6.3 summarises common strategies for splitting the available data into training and test sets and estimating the error rate.

A more detailed account of classifier performance, than the overall error rate, is provided by the class-conditional error rates; i.e. the proportion of observations from each class that are misclassified. A more detailed account still is provided by the *confusion matrix* (Ripley, 1996, p. 75). Element n_{ij} of this matrix is a count of the number of observations of class *i* that are classified as belonging to class *j*. In the two-class case, where the two classes are denoted π_1 and π_2 and a classification rule is known or has been estimated, the problem of classifying a new observation can be formulated as a statistical hypothesis test (Fukunaga, 1990, p. 51). The null and alternate hypotheses are:

 H_0 : The new observation belongs to class π_1

 H_A : The new observation does not belong to class π_1 .

There are two types of error that can be made in deciding whether to accept or reject the null hypothesis. These are summarised as follows (Freund, 1984, p. 282):

	Do not reject H_0	Reject H_0
H_0 is actually true	Correct decision	Type I error
H_0 is actually false	Type II error	Correct decision

Method	Property	Comments
Resubstitution	All the available data	Optimistically biased estimate,
method	is used for training as	especially when the ratio of
	well as testing; train-	sample size to dimensionality
	ing and test sets are	is small.
	the same.	
Holdout method	Half the data is used	Pessimistically biased esti-
	for training and the re-	mate; different partitionings
	maining data is used	will give different estimates.
	for testing; training	
	and test sets are inde-	
	pendent.	
Leave-one-out method	A classifier is designed	Estimate is unbiased but it has
	using $(n-1)$ samples	a large variance; large compu-
	and evaluated on the	tational requirement because n
	one remaining sam-	different classifiers have to be
	ple; this is repeated	designed.
	n times with differ-	
	ent training sets of size	
	(n-1).	
Rotation method, <i>n</i> -	A compromise be-	Estimate has lower bias than
fold cross validation	tween holdout and	the holdout method and is
	leave-one-out meth-	cheaper to implement than
	ods; divide the	leave-one-out method.
	available samples into	
	P disjoint subsets,	
	$1 \leq P \leq n$. Use	
	(P-1) subsets for	
	training and the re-	
	maining subset for	
Destates setting 1	test.	Destate set in the set in the
Bootstrap method	Generate many boot-	Bootstrap estimates can have
	strap sample sets of	one out mathed, computation
	size <i>n</i> by sampling	ally more demanding yearshiling
	oral ostimators of the	any more demanding; useful m
	eral estimators of the	sman sample size situations.
	fined (or F0 and	
	F632) using the best	
	strap samples	
	suap samples.	

Table 6.3: Commonly used error estimation methods (reproduced from Jain et al. (2000,p. 26)).

The probability associated with making a type I error is denoted α and the probability associated with making a type II error is denoted β . The probability $1 - \beta$ is called the *power* or *sensitivity* of the test. The probability $1 - \alpha$ is called the *specificity* of the test. In the general epidemiologic setting, where the interest is in predicting *disease*, the two classes of interest π_1 and π_2 are *negative* (without disease) and *positive* (with disease) respectively. In this setting the confusion matrix has the form:

	Predicted class								
		negative	positive						
True	negative	a	b						
class	positive	С	d						

and can be used to estimate the sensitivity and specificity as well as several other probabilities associated with classifier performance including (Bradley, 1996, p. 137):

1. The *accuracy* or *correct classification rate* (CCR) which is the proportion of the total number of predictions that are correct:

accuracy = CCR =
$$\frac{a+d}{a+b+c+d}$$
.

2. The *sensitivity* or *true positive rate* (TPR) which is defined to be the proportion of positives that are correctly classified:

sensitivity = TPR =
$$\frac{d}{c+d}$$

3. The *specificity* which is defined to be the proportion of negatives that are correctly classified:

specificity
$$= \frac{a}{a+b}.$$

4. The *false positive rate* (FPR) which is defined to be the proportion of negatives that are incorrectly classified:

$$FPR = \frac{b}{a+b} = 1 - specificity.$$

The confusion matrix and related measures summarise the performance of the classifier for a given *cutpoint*. The cutpoint is the threshold at which a decision of positive

or negative is made. A more complete summary of the performance of the classifier is given by the area under the ROC (receiver operating characteristic) curve (Hosmer & Lemeshow, 2000, p. 160). The ROC curve (see Figure 6.2), which originates from signal detection theory, is a plot of the probability of detecting a false positive (1 - specificity) against the probability of detecting a true positive (sensitivity) over the entire range of possible cutpoints. The line from (0,0) to (1,1) is the ROC curve of a completely random classifier, one that randomly allocates a new observation to one of the two classes; e.g. in the case of equal prior probabilities¹⁰ the behaviour of the classifier is equivalent to tossing a coin to predict class membership. The area under the ROC curve (AUC) is a measure of the discrimination provided by a classifier. A value of 0.5 corresponds to a random classifier and a value of 1 corresponds to the perfect classifier. A qualitative description for other values of the AUC is given in Table 6.4. An empirical ROC curve is obtained from the test set by plotting the FPR against the TPR for a range of cutpoints of the classifier. This empirical ROC curve has a stepped appearance (for example see Figure 6.16) because of the finite number of observations in the test set. Numerical integration, e.g. the trapezoidal rule (Bradley, 1997, p. 1146), provides an estimate of the AUC. A commonly used method of estimating the variability of this estimate is to compute the standard error of the Wilcoxon statistic (Bradley, 1997, p. 1147):

$$\hat{SE}(\theta) = \sqrt{\frac{\theta (1 - \theta) + (n_p - 1) (Q_1 - \theta^2) + (n_n - 1) (Q_2 - \theta^2)}{n_p n_n}},$$
(6.1)

where $\theta = A\hat{U}C$ (the estimate of the AUC), $Q_1 = \theta/(2-\theta)$, $Q_2 = 2\theta^2/(1+\theta)$, n_p is the number of positive examples, and n_n is the number of negative examples.

The AUC is "the probability that a randomly chosen positive example will be ranked with greater suspicion than a randomly chosen normal [negative] example" (Bradley, 1996, p. 160). It is a measure of classifier performance that is invariant to the prior probabilities of class membership (Bradley, 1997, p. 1145). This is in stark contrast to the confusion matrix and associated measures. If two classifiers are being compared on the basis of sensitivity and specificity, for example, differences in performance between them might be due entirely to the population priors (or estimates thereof) used to train each.

¹⁰ Prior probabilities, or priors, are the "probabilities specified before seeing the data, and so based on prior experience or belief. Commonly these are the prior probabilities... of the classes" (Ripley, 1996, p. 352).



Figure 6.2: An example of an ROC (receiver operating characteristic) curve.

AUC	Meaning
0.5	no discrimination
$0.7 \le AUC < 0.8$	acceptable discrimination
$0.8 \le AUC < 0.9$	excellent discrimination
$AUC \ge 0.9$	outstanding discrimination

Table 6.4: Qualitative interpretation of the AUC (Hosmer & Lemeshow, 2000, p. 162).

6.4 Case study: A MACs-based classifier for Pap smear screening

This section describes a study, the purpose of which is to demonstrate the practical application of the methods described in the preceding chapters. The study comprises two experiments. The first experiment evaluates the performance of a classifier trained and evaluated using features purposefully designed to measure *chromatin margination*. The second experiment determines a subset of chromatin features, from amongst those computed, that have the most discriminatory power for the Pap smear classification problem. The order in which the experiments were conducted is the order in which they are reported here. The decision to measure chromatin margination is based on a pilot study—not reported here—based on a completely different set of data.

6.4.1 Cancer of the uterine cervix

Cancers of the uterine cervix (neck of the womb) are "thought to derive from the epithelium near the opening of the cervix" (Alberts et al., 2002, p. 1318). Epithelia are the tissues that line all the cavities and free surfaces of the body (Alberts, Johnson, Lewis, Raff, Roberts & Walter, 2002, p. 1066). As shown in Figure 6.3, the epithelia of the cervix can be divided into two types: the *squamous epithelium* of the ectocervix and the *columnar epithelium* of the endocervix. The area of the junction between them is called the *transformation zone* and "is of considerable importance in the genesis of carcinoma of the uterine cervix" (Koss, 1992, p. 265). The anatomic location of the transformation zone may vary considerably and is, to a significant extent, age dependent (Koss, 1992, p. 266).

The squamous epithelium is multi-layered. Three principal layers can be recognised (Koss, 1992, p. 91): (1) the basal layer; (2) the intermediate layers; and (3) the superficial layers (see Figure 6.4). Superficial cells are the most mature squamous cells. Parabasal and basal cells, on the other hand, are immature squamous cells. In a normal Pap smear usually only the upper few layers of the squamous epithelium are removed and so the immature cells near the base of the epithelium are not sampled (Cibas, 2003).

Squamous cell carcinoma is a form a cervical cancer that originates in the squamous epithelium. Adenocarcinoma is a form that originates in the glandular (columnar) epithelium. Eighty-five percent of all cervical cancers are squamous cell carcino-



Figure 6.3: Epithelia of the cervix.



Figure 6.4: Cells of the squamous epithelium (freehand adaptation of Koss (1992, Figure 3-4)).

mas, whilst the remaining 15 percent are glandular or mixed (AHTAC, 1998, p. 4). "Screening using the Pap smear test has a much greater chance of preventing squamous than glandular cancer" (AHTAC, 1998, p. 4).

There is excellent evidence to suggest that invasive squamous cell carcinoma develops from non-invasive lesions known as *carcinoma in situ* (CIS) (Koss, 1992, p. 371). It is also thought that it may develop from related precancerous intraepithelial abnormalities known as *cervical intraepithelial neoplasia* (CIN). These lesions are graded according to the degree of abnormality. Grade I corresponds to mild dysplasia, grade II to moderate dysplasia, and grade III to both severe dysplasia and carcinoma in situ (CIS)(Koss, 1992, p. 390). Precancerous intraepithelial abnormalities, regardless of grade, do not endanger the life of the individual per se. The onset of danger is the moment when the process breaks out of the epithelium and invades the surrounding tissues, and in particular the lymphatic and blood vessels, leading to metastatic spread of the disease (Koss, 1992, p. 387). It is worth noting that for all precancerous intraepithelial abnormalities (Koss, 1992, p. 390):

there are three possible outcomes: (1) the lesion may progress directly to invasive epidermoid carcinoma; (2) the lesion may remain confined to the epithelium...or (3) the lesion may disappear either after a minor diagnostic procedure or spontaneously.... The chances for disappearance are far higher for the CIN grade I...than for CIN III lesions.

In the United States, more than 90% of laboratories use the Bethesda System¹¹, in some form, rather than the CIN system for reporting the results of cervical cytology (Solomon et al., 2002, p. 2114). In this system precancerous squamous lesions are called *squamous intraepithelial lesions*. Mild dysplasias (CIN I) are classified as low-grade squamous intraepithelial lesions (LSIL), whilst moderate and severe dysplasias, and carcinoma in situ (CIN II and CIN III) are collectively classified as high-grade squamous intraepithelial lesions (HSIL) (Solomon et al., 2002, p. 2116).

¹¹ The Bethesda System "was developed at a National Cancer Institute (NCI)-sponsored workshop in December 1988 to provide uniform diagnostic terminology that would facilitate communication between the laboratory and the clinician.... Subsequently, a second workshop was held in April 1991 to evaluate the impact of TBS in actual practice and to amend and modify it where needed" (Kurman & Solomon, 1994, p. ix). In April 2001 a third workshop was held which reviewed issues regarding terminology and reporting of cervical cytology.

6.4.2 Cells of interest for MAC analysis

Recall from Section 1.1 that a typical Pap smear contains cells sampled from in and around the cervix. Thus the smear may or may not contain endocervical cells. Recent studies suggest that an endocervical component is not essential to making a diagnosis (Cibas, 2003). Indeed, according to the 2001 Bethesda system for reporting cervical/vaginal cytologic diagnoses, "a smear without endocervical cells is not considered unsatisfactory, although the absence of an endocervical/transformation zone component is mentioned as a 'quality indicator'" (Cibas, 2003).

With regard to the MACs phenomenon Bengtsson & Nordin (1994, p. 38) state that:

there is substantial evidence in the quantitative cytology literature that... malignancy-associated changes...do exist.... They are found in intermediate cells, metaplastic cells, and endocervical cells.

Given that endocervical cells are likely to be absent or present only in small numbers on a typical Pap smear, this suggests that intermediate cells offer the best opportunity for building a MACs-based classifier for automated Pap smear screening. Indeed, in a cervical MACs study reported by Isenstein et al. (1995, p. 90) the authors state that:

intermediate cells were measured for MAC analysis because they are abundant on most slides and it should be relatively easy to devise a very accurate automatic classification scheme to detect intermediate cells.

Koss (1992, p. 258) states that:

the nuclei of the intermediate cells measures about 8μ m in average diameter, are round or oval, with a clearly defined nuclear membrane surrounding well-preserved homogeneous nucleoplasm. Chromocenters and sex chromatin may be observed within such nuclei.

In comparison the nuclei of superficial cells are pyknotic—i.e. condensed (dark) and shrunken—with a nuclear diameter that is rarely more than 5μ m (Koss, 1992, p. 257), and the nuclei of parabasal cells are variably sized but "usually larger than that of an intermediate cell" (Cibas, 2003).

6.4.3 Cytology slides

The image data used in this study originate from a set of 148 slides obtained in late 2001 from Jenny Halford, Manager of the Cytology Department, Queensland Medical Laboratory¹² (QML). The slides are Papanicolaou-stained cervical smears. One of the slides is shown in Figure 6.5. The slides are a sample of routine smears processed by the QML for 143 different patients: 140 patients ×1 slide, 2 patients ×3 slides, and 1 patient ×2 slides. The slides were prepared using the AutoCyte PREPTM monolayer technology rather than from direct smears of the sampled cells¹³. The technology requires that, at the time of sampling, the cells on the collection device be eluted into a phial of preservative liquid. The phial is then processed by the AutoCyte PREPTM system. The system removes a large portion of blood, mucus, and other debris and deposits cells onto a slide in a mono/thin-layer. From the point of view of screening, mono/thin-layer prepared slides have several advantages (Grohs, Zahniser & Geyer, 1994, p. 182):

In thinly smeared preparations, cell overlap is minimal and the cells are more likely to be in the same focal plane, thus requiring little focusing even when using a $40 \times$ objective. This speeds up slide examination by both visual and automatic techniques. The more uniform distribution of cellular material in a thin layer makes the screening process easier.

QML's diagnosis for each slide is given in Appendix G. In summary, of the 148 slides 101 are negative, 1 is CIN I, 34 are CIN II, 3 are CIN II/III, and 9 are CIN III. A slide is given the diagnosis *negative* if it is deemed to contain wholly normal cells. If, on the other hand, the slide is found to contain abnormal (dysplastic) cells then it is given the diagnosis CIN and a grade from I to III. For the purpose of this study each slide was assigned to one of two classes: class 0 (normal class) if the slide is negative or class 1 (abnormal/suspicious class) if the slide is CIN I+.

6.4.4 Image acquisition

The Cytometrics Project custom cytometer was used to capture all of the cell nuclei images for this study. The cytometer is an AcCell-SAVANT^M/research system

¹² "QML is one of the largest pathology practices in Australia. Wholly Australian owned, QML services an area throughout Queensland, northern New South Wales and the Northern Territory and offers a comprehensive range of testing services" (http://www.qml.com.au).

¹³ In 1999, the AutoCyte PREP[™] System was approved by the FDA as a replacement for the conventional Pap smear as a method for use in the screening of cervical slides.



Figure 6.5: One of the AutoCyte slides used in the study.

installed with proprietary CSSIP segmentation software. The underlying hardware is an AcCell[™] 2000 Workstation (see Figure 6.6). The system hardware components include a robotic slide loader, a barcode reader, an Olympus¹⁴ BX40 microscope fitted with a motor to control focus, an automated stage, a digital video camera¹⁵, and a computer. The AcCell-SAVANT[™] was purchased from AccuMed¹⁶ International Inc. by CSSIP in late 1999. The purchase agreement also granted CSSIP a license to use AccuMed's proprietary software for cell capture and analysis. The image capture and measurement component of the software, called ACQUIRE, is designed to scan and capture images from Thionin-Feulgen stained slides only (brightfield microscopy). To enable CSSIP to replace the segmentation routines in ACQUIRE with its own, AccuMed provided CSSIP with a special version of the ACQUIRE executable. In this version all of the AccuMed segmentation code is located in a dynamic link library (DLL)¹⁷ separate from the main executable. In 2000 CSSIP wrote a replacement for this DLL based on its own proprietary algorithms (Bamford

¹⁴ Olympus Optical Co., Ltd, headquartered in Tokyo, Japan.

¹⁵ DALSTAR 1M15 CCD camera. The camera can capture 12-bit images with $1k \times 1k$ spatial resolution. The pixel size is $14\mu m \times 14\mu m$. The camera is marketed by DALSA Corporation headquartered in Waterloo, Ontario, Canada.

¹⁶ AccuMed International Inc. was acquired by Ampersand Medical Corporation in February 2001.

¹⁷ A Dynamic Link Library (DLL) is a file of code containing functions that can be called from other executable code (either an application or another DLL). Programmers use DLLs to provide code that they can reuse and to parcel out distinct jobs. Unlike an executable (EXE) file, a DLL cannot be directly run. DLLs must be called from other code that is already executing.



Figure 6.6: AcCell[™] 2000 workstation (http://www.accumed.com).

& Lovell, 1999; Bamford & Jackway, 2001). The new DLL enables the cytometer to scan and measure Papanicolaou stained slides using a $40\times$ objective lens with a numerical aperture¹⁸ of 0.75. The segmentation code requires that the cytometer microscope be fitted with a Balzers¹⁹ FILTRAFLEX-K K55 broad bandpass filter. The purpose of the filter is to increase the contrast of the cells. The filter has a peak transmission of $\geq 70\%$ at a wavelength of 550nm (green part of the visible light spectrum) and a half bandwidth of approximately 50nm.

In January 2002 the Cytometrics Project cytometer was programmed to automatically perform an exhaustive scan of each QML slide, at $40\times$, and to capture and archive digitised images of the nuclei of intermediate cells. To eliminate any bias that might be introduced by the order of scanning, each slide was assigned a unique random integer (barcode sticker) between 0 and 147, and the slides were scanned in barcode order: $0, 1, \ldots 147$ (see Appendix G). Each slide took on the order of 4 or 5 hours to scan depending on the density of the cell deposition. A histogram showing the number of nucleus-like objects archived by the cytometer for each slide is shown in Figure 6.7. The ACQUIRE software limited²⁰ the maximum number of nucleus-like objects collected per slide to 10080. Figures 6.8 and 6.9 show sample galleries of objects collected by the cytometer for slides 0 and 1 respectively.

6.4.5 Image processing and analysis

DImPAL (see Appendix A) was used to perform all of the image processing and analysis of the images acquired by the cytometer. In particular it was used to perform chromatin segmentation, artefact rejection, and feature measurement. The

¹⁸ The minimum distance δ that can be resolved by the microscope (i.e. the resolving power of the microscope) is dictated by the numerical aperture (NA) of the objective and the wavelength λ of the light used (Lacey, 1999, p. 6): $\delta = 0.61\lambda/NA$.

¹⁹ Unaxis Balzers Ltd. Optics Div. (Formerly Balzers Thin Films) PO Box 1000 FL-9496 Balzers Liechtenstein.

 $^{^{20}}$ This appears to be a bug in the licensed portion of the software.



Figure 6.7: Number of nuclei-like objects captured per slide by the cytometer (red bars) and the number of objects retained after artefact rejection (blue bars).



Figure 6.8: Sample images captured by the cytometer: the first 64 images captured from slide 0.



Figure 6.9: Sample images captured by the cytometer: the first 64 images captured from slide 1.

whole process of segmenting the chromatin particles within every nucleus-like object archived by the cytometer, performing artefact rejection, and computing the chromatin and nucleus features required approximately 75.7 hours of execution time on a 450 MHz Intel²¹ Pentium III computer with 256MB of RAM.

6.4.5.1 Artefact rejection

"In Papanicolaou-stained cells, overlapping and/or folded cytoplasm, cellular debris, and so on present a formidable problem in object segmentation" (Palcic & MacAulay, 1994*a*, p. 57). Consequently many of the objects collected by the cytometer are not in fact intermediate cell nuclei. The cytometer guarantees only that the objects are of approximately the right size and shape (8μ m in diameter, and oval-shaped). Bengtsson & Nordin (1994, p. 40) note that "ensuring that the object which is analyzed really is a cell and not something else is one of the most important and difficult problems in automated cytology". One of the advantages of the MACs approach to screening, as compared to the RE approach, is that it is not necessary to analyse all of the cells collected from the slide. It is possible to reject doubtful objects (artefacts) because of the large number of cells available for analysis (Palcic & MacAulay, 1994*a*, p. 59).

In this study, a very simple strategy for artefact rejection was adopted based on (i) the shape of the object and (ii) the number and size of dark particles it contains. The second criterion is motivated by the observation of Koss (1992, p. 258) that within the nuclei of intermediate cells "chromocenters and sex chromatin may be observed". Specifically the rejection strategy involves:

- 1. Computing the G-shape factor (see Appendix H) for each nucleus-like object and rejecting those objects that do not satisfy a prescribed range of values; and
- 2. Applying the (preferred embodiment of the) chromatin segmentation algorithm (see Section 4.4.3) to each nucleus-like object in turn to segment the dark particles (chromocentres), and discarding those objects that do not contain a prescribed number of particles, or that contain very large particles.

In this study the minimum number of particles was chosen to be 5, the largest permissible particle was chosen to be no more than 30% of the area of the nucleus,

²¹ Intel Corporation, headquartered in Santa Clara, California.

and the value G had to satisfy: $0.8 \le G \le 1.3$. These parameters were determined by examining a few hundred objects, by eye, collected by the cytometer for slide 0.

The number of nucleus-like objects retained for each slide after applying this artefact rejection scheme is shown in Figure 6.7. Figures 6.10 and 6.11 show sample galleries of objects retained for slides 0 and 1 respectively. This simple artefact rejection scheme is not perfect because it still admits a number of leukocytes (blood cells); e.g. objects 3 and 5 in Figure 6.10. However, the majority of retained objects do appear to be intermediate cell nuclei.

6.4.5.2 Nucleus and chromatin features

Table 6.5 lists the features measured by DImPAL for each nucleus. Table 6.6 lists the features measured for each chromatin particle within a nucleus.

6.4.5.3 Nucleus statistics

Table 6.7 lists the blob statistics computed for each nucleus by DImPAL. For statistics N21 and N22 the median and interquartile range (IQR) were computed rather than the mean and standard deviation. The reason for this is as follows. The maximum dynamics value (Grimaud, 1992, p. 297) is defined to be the difference between the highest grey-value and the lowest grey-value in the image. Thus small errors in the segmentation of the nucleus (performed by the cytometer) are likely to lead to highly variable estimates of this dynamics value. This in turn can significantly affect the value of the mean and variance. It does not, however, affect the median and IQR.

6.4.6 Culling of slides unsuitable for MAC analysis

Palcic & MacAulay (1994b, p. 159) state in relation to the MAC phenomenon that

for most [slide] features, a large number of cells must be measured to achieve a constant value of the feature mean and standard deviation. The exact number of cells that must be measured to achieve these constant values depends on the class of cells and the nuclear feature.

They have determined experimentally that "500 cells is a minimum number of nuclei that must be measured to achieve a relatively constant MAC determination" (Palcic



Figure 6.10: Sample images after artefact rejection: the first 64 images retained from slide 0.



Figure 6.11: Sample images after artefact rejection: the first 64 images retained from slide 1.

	Measurement	Inputs	Method
N1	Area	Binary mask of the nucleus.	Equation 5.4
N2	Perimeter	Binary mask of the nucleus.	Equation 5.7
<i>N</i> 3	3D connectivity num- ber	Grey-scale image cor- responding to the bi- nary mask of the nu- cleus.	(Serra, 1988 <i>c</i> , p. 314)
<i>N</i> 4	Surface area	Grey-scale image cor- responding to the bi- nary mask of the nu- cleus.	Equation 5.11
N5	Volume	Grey-scale image cor- responding to the bi- nary mask of the nu- cleus.	Sum of the grey values.
<i>N</i> 6	Mean grey-level	Grey-scale image cor- responding to the bi- nary mask of the nu- cleus.	Mean of the grey values.
N7	G shape factor	Binary mask of the nucleus.	Appendix H

 Table 6.5:
 Nucleus measurements.

	Measurement	Inputs	Method
<i>B</i> 1	Area	Binary mask of the blob.	Equation 5.4
B2	Perimeter	Binary mask of the blob.	Equation 5.7
<i>B</i> 3	3D connectivity num- ber	Grey-scale image cor- responding to the bi- nary mask of the blob.	(Serra, 1988 <i>c</i> , p. 314)
<i>B</i> 4	Surface area	Grey-scale image cor- responding to the bi- nary mask of the blob.	Equation 5.11
B5	Volume	Grey-scale image cor- responding to the bi- nary mask of the blob.	Sum of the grey values.
B6	Mean grey-level	Grey-scale image cor- responding to the bi- nary mask of the blob.	Mean of the grey values.
<i>B</i> 7	Dynamics value	Grey-scale image cor- responding to the bi- nary mask of the nu- cleus.	The dynamics (Gri- maud, 1992) of the re- gional minimum asso- ciated with the blob.
<i>B</i> 8	Mean distance to nu- cleus boundary	Binary mask of the blob plus the distance transform of the bi- nary mask of the nu- cleus.	Mean of the values in the distance transform of the nucleus that correspond to the bi- nary mask of the blob.

 Table 6.6:
 Blob measurements.

	Statistic	Inputs
N8	number of blobs	
N9, N10	mean and standard deviation	blob areas $(B1)$
N11, N12	mean and standard deviation	blob perimeters $(B2)$
N13, N14	mean and standard deviation	blob 3D connectivity numbers (B3)
N15, N16	mean and standard deviation	blob surface areas $(B4)$
N17, N18	mean and standard deviation	blob volumes $(B5)$
N19, N20	mean and standard deviation	blob mean grey-levels $(B6)$
N21, N22	median and in- terquartile range	blob dynamics $(B7)$
N23, N24	mean and standard deviation	blob mean distances to boundary (B8)
N25	sum	blob areas $(B1)$
N26	sum	blob surface areas $(B4)$
N27	sum	blob volumes $(B5)$
N28	sum	blob mean grey-levels $(B6)$

 Table 6.7: Blob statistics computed for each nucleus.

& MacAulay, 1994*b*, p. 160). Consequently, in the following experiments it was decided to use only those slides for which at least 500 nuclei were measured. The nuclei counts (after artefact rejection) for each slide are as follows (these correspond to the blue bars on the histogram shown in Figure 6.7):

[0]	643	5169	2786	1422	1251	3792	2670	1642	1455	1145	840	1176	2823	2581	2392
[15]	1843	3305	1422	748	2911	3641	256	2808	3602	661	2442	3465	2310	5885	2478
[30]	222	2668	4102	2425	2488	2408	4051	223	1269	4685	4173	901	1476	56	5283
[45]	2152	3069	1196	1161	1303	3927	1858	3131	973	2091	1639	3510	2015	1922	2456
[60]	2263	2315	1137	4382	257	1578	2255	3301	3388	4788	2506	2593	1357	2508	3421
[77]	3522	2237	5438	4600	2254	2191	2506	1221	2787	2736	1598	3919	4664	2400	2255
[90]	3923	781	1558	1389	2023	5975	4324	2154	1028	2987	713	383	1332	2712	1669
[105]	2079	2482	4780	2472	1778	1068	3826	2785	1189	2329	999	2493	3004	1513	2276
[120]	1260	1620	1456	751	958	81	1432	1592	2972	1273	3599	3033	1362	3811	1656
[135]	2904	1527	1756	3177	1282	817	2981	4983	1103	1850	3625	1467	1511		

This suggests that slides 21 (CIN 2), 30 (CIN 2/3), 37 (CIN 2), 43 (CIN 2), 64 (CIN 2), 101 (Negative), and 125 (Negative) be excluded from the experiments. Each of these slides was visually reviewed and all, with the exception of 101, were found to contain a scanty number of cells. The QML slides were specifically prepared for rare event (diagnostic cell) screening and not for a MACs-based review. In this scenario, if one or more diagnostic cells are found on the slide it does not matter that the slide is scanty. However, if the slide is negative then scantiness suggests that the specimen is inadequate even for rare event screening. This appears to be the case for slide 125. In the case of slide 101 the slide is not scanty. However, an intermittent bug in the focusing algorithm in the cytometer appears to have resulted in the capture of several thousand poorly focused objects for this slide²². Consequently very few objects (81) were retained after artefact rejection.

Of the remaining slides, it was decided to exclude 84 (patient 99-90006) and 123 (patient 00-22697) so that the data consists of only one slide per patient (statistically independent observations). These two slides were chosen simply because the remaining slide for each patient, 14 and 120 respectively, bears an official QML sticker. To summarise, of the 148 slides scanned by the cytometer the following 9 slides were excluded from the experiments: 21, 30, 37, 43, 64, 84, 101, 123, and 125. This leaves 40 abnormals and 99 normals.

²² Ideally, the slide should have been re-scanned. However, given that the slide data were processed several months after the initial scanning, and that by this time the cytometer software had been significantly changed, it was decided that it was simpler to omit the slide from the study.

6.4.7 Classification using logistic regression

The *sampling paradigm* is the traditional view of statistical pattern recognition. In this view (Ripley, 1996, p. 6)

the training set is regarded as a sample from a population of possible examples, and the statistical similarities of each class extracted, or more precisely the significant differences between classes are found. A parametric or non-parametric model is constructed for the distribution of features for examples from each class, and statistical decision theory used to find an optimal classification.

Another view of statistical pattern recognition is the *diagnostic paradigm*. In this view the interest is not in what the classes look like but rather, given an example, (Ripley, 1996, p. 7)

in what the distribution over the classes is for similar examples. The main method of this approach became known as *logistic discrimina*tion..., but was never widely known even in statistics and...[before 1996] appears in no pattern recognition text. This is [also] the main approach of the neural network school.

Logistic discrimination is the approach adopted for this study. More specifically, the *logistic regression* model is used in this study as the basis for classification. Logistic regression is popularly used in the analysis of epidemiologic data (Kleinbaum, 1994, p. 4). There are several reasons for the attractiveness of logistic discrimination (Anderson, 1982, p. 169):

- 1. few distributional assumptions are made;
- 2. it is applicable when the predictor variables are continuous, discrete, or both; and
- 3. it is very easy to use because once the model parameters have been estimated, the classification of a new observation requires only the calculation of a linear function.

Logistic regression analysis is concerned with describing the relationship between a binary or dichotomous response variable and one or more predictor (explanatory)
variables. The fact that the response variable is dichotomous is what distinguishes logistic regression from ordinary linear regression. In any regression problem the key quantity is the mean value of the response variable given the values of the predictor variables (Hosmer & Lemeshow, 2000, p. 4). Let Y be a random variable denoting the response and let the vector $\mathbf{x}^{\mathsf{T}} = (x_1, x_2, \ldots, x_r)$ denote a collection of r independent predictor variables. In linear regression the assumption is that the mean of the response variable has the form

$$E(Y \mid \mathbf{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r$$

where the β_i are the parameters to be estimated. The term *linear* comes from the fact that the mean is a linear function of the unknown parameters (Johnson & Wichern, 1988, p. 274). Given n independent observations $(\mathbf{x}_i, Y_i), i = 1, 2, \ldots, n$, the estimation of the parameters β_i is done using the least squares estimation method (Johnson & Wichern, 1988, p. 274). Under the tentative assumption that the error terms associated with the fit have normal distribution, this coincides with the maximum likelihood estimate (Johnson & Wichern, 1988, p. 284). Although it is possible to fit this model when Y is dichotomous this violates the underlying model assumptions and leads to impossible predicted values (see Hosmer & Lemeshow (2000, Chapter 1) for details). Rather than trying to predict the value of the mean response, logistic regression seeks to predict the *log-odds* of the response having one particular value versus the other value. If the response variable Y is coded such that it only takes on the values 0 and 1 then the *odds* of the response having the value 1 is P(Y = 1) / [1 - P(Y = 1)]. Taking the logarithm of this ratio produces a response that can in principle vary between $-\infty$ and ∞ . This suggests the model (Hosmer & Lemeshow, 2000, p. 31):

$$\operatorname{logit}\left[P\left(Y=1 \mid \mathbf{x}\right)\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r \tag{6.2}$$

where the *logit* transform is defined

$$\operatorname{logit}\left(x\right) = \ln\left(\frac{x}{1-x}\right). \tag{6.3}$$

Equation 6.2 can be written

$$P(Y = 1 | \mathbf{x}) = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r)}}.$$
(6.4)



Figure 6.12: The logistic function $f(x) = e^x/(1+e^x)$ restricted to the domain [-10, 10].

Equation 6.4 is called the logistic regression model. Unfortunately the least squares method cannot be used to fit this model (Hosmer & Lemeshow, 2000, p. 8). Instead maximum likelihood estimation is used (see Section 6.3.2.2).

The function

$$f(x) = \frac{e^x}{(1+e^x)}$$
(6.5)

appearing on the right-hand-side of equation 6.4 is called the *logistic function*. According to Kleinbaum (1994, p. 5), one of the reasons why logistic regression is appealing to epidemiologists is the S-shape of the logistic function (see Figure 6.12). If the value 1 denotes *disease* and the value 0 *no disease* then f(x) represents the risk for a given value of x. For low values of x the risk factor is minimal, but once x exceeds some lower threshold the risk begins to sharply increase over a certain range and to attain a maximum. "This threshold idea is thought by epidemiologists to apply to a variety of disease conditions" (Kleinbaum, 1994, p. 7).

6.4.7.1 Using a fitted logistic regression model as a classifier

For any given observation \mathbf{x}_i the fitted logistic regression model returns an estimate of the probability that the response is in fact 1. This estimated probability can be used to classify the observation. To do this it necessary to define a value c, called the *cutpoint*, and to compare the estimated probability with this value. If the estimated probability exceeds c then the observation is deemed to belong to class 1; otherwise it is deemed to belong to class 0. The most commonly used cutpoint is 0.5 (Hosmer & Lemeshow, 2000, p. 156). By varying the cutpoint between 0 and 1 an empirical ROC curve can be estimated from a given set of observations.

6.4.7.2 Comparison with Fisher's linear discriminant function

Another, widely-used, statistical approach to the two-class classification problem is *Fisher's linear discriminant function* (Johnson & Wichern, 1988, p. 473). The method requires no distributional assumptions but does assume that the two classes have the same covariance matrix. It is known to be optimal for multivariate normal and some other distributions (Anderson, 1982, p. 170). However, it is also known that Fisher's linear discriminant function can behave very poorly when the predictor variables are a mix of discrete and continuous variables (Johnson & Wichern, 1988, p. 527). Consequently it can be argued, e.g. Press & Wilson (1978), that logistic regression is more statistically robust than linear discriminant analysis.

6.4.7.3 Logistic regression analysis software

The software used to perform the logistic regression analysis in this study is R Version 1.7.1. More specifically several R programs were written, by the author, to perform the required analyses. Listings for these programs appear in Appendix I (Experiment 1), Appendix J (Experiment 2), and Appendix K (ROC curve analysis).

R is very similar to S-PLUS (both have already been mentioned in Section 6.3.3). Although there are some important differences between the two, much of the code written for S also runs unaltered in R. *Modern Applied Statistics with S-PLUS* (Venables & Ripley, 1999) is the canonical reference for statistical analysis using S-PLUS and R. 'R' Complements to Modern Applied Statistics with S-PLUS (Venables & Ripley, 2001) is a supplement to this reference written for users of R. The software written to accompany the book and the data referenced in the book are freely available for download as a package/library called MASS. Both the R software and the MASS library can be downloaded from the CRAN (comprehensive R archive network) mirror sites: http://cran.r-project.org/mirrors.html.

6.4.8 Sample size considerations

If all of the data (40 abnormals and 99 normals) were used to train a classifier then according to the rule of thumb stated at the end of Section 6.3.1, at most

$$n = \left\lfloor \frac{40}{10} \right\rfloor = 4$$

features could be used in the classifier. Unfortunately then there would be no way to effectively evaluate the performance of the classifier. Testing it on the same data used for training (the resubstitution method) is of course unsatisfactory because the estimate of performance will be optimistically biased. An unbiased method for estimating the performance that permits nearly all of the data to be used for training, is the leave-one-out method. Using this method, at most

$$n = \left\lfloor \frac{40 - 1}{10} \right\rfloor = 3$$

features can be used in the classifier (to avoid the curse of dimensionality). This is the method used in the first of two experiments described below. The experiment evaluates the performance of a classifier defined in terms of three a priori selected features.

In the second experiment, for reasons explained later, the holdout procedure rather than the leave-one-out procedure is used. In fact multiple repetitions of the holdout procedure (called the repeated holdout method) with stratified random sampling are used. In any given repetition a proportion h of the normals and the same proportion h of the abnormals are held out (stratified sample). This suggests then that at most

$$n = \left\lfloor (1-h) \times \frac{40}{10} \right\rfloor$$

features can be used in the classifier. If $h = \frac{1}{4}$ then n = 3. The larger the holdout proportion the smaller is n; e.g. for $h = \frac{1}{3}$, n = 2.

6.4.9 Experiment 1: Classification with a single nucleusfeature histogram

The aim of this experiment was to build and evaluate a logistic classifier based on slide features derived from a single chromatin feature purposefully designed to measure chromatin margination. The margination of chromatin is one of the morphological changes associated with apoptosis (a form of cell death) and is characterised by the packing of chromatin into smooth masses applied against the nuclear membrane (Majno & Joris, 1995, p. 7). Young et al. (1986, p. 467) state that chromatin "margination is a relevant indicator of the state of [a] cell in certain disease processes". Of particular interest is the fact that chromatin margination is also observed in the nuclei of cancer cells. Koss (1992, p. 132) states that:

fixed and stained nuclei of cancer cells examined in light microscopy often display *coarse, dense granularity* that is not usually present in normal cells.... The granular material may be distributed along the nuclear envelope.

Given that disturbances in mitotic²³ activity are also characteristic of precancerous lesions (Koss, 1992, p. 142, 390), this suggests that margination might be a useful feature for discriminating between normal and abnormal Pap smear slides. Young et al. (1986, p. 470) proposed a method for quantifying chromatin margination based on:

computing the average optical density per pixel in a series of concentric "rings" that begin at the outside boundary and move inward toward the center of the nucleus. These rings are formed by beginning with the original nuclear contour and then considering the successive differences between the contour and its eroded versions.

A more direct measure of margination is possible based on chromatin segmentation. This is the motivation behind the definition of feature B8. It is computed from the *cookie-cutting-distance* illustrated in Figure 6.13. This chromatin feature is a relational/contextual feature characterising the distance of a chromatin blob to the nucleus boundary. The median of B8 is a nucleus feature, hereinafter referred to as feature M, characterising the chromatin margination within the nucleus. A typical box-and-whisker plot²⁴ of M for a slide is shown in Figure 6.14.

²³ Mitosis refers to the process of "division of the nucleus of a eucaryotic cell, involving condensation of the DNA into visible chromosomes, and separation of the duplicated chromosomes to form two identical sets" (Alberts et al., 2002, p. G:23).

²⁴ A box-and-whisker plot "is a way to look at the overall shape of a set of data. The central box shows the data between the 'hinges' (roughly quartiles), with the median represented by a line. 'Whiskers' go out to the extremes of the data, and very extreme points are shown by themselves" (Venables & Ripley, 1999, p. 122).



Figure 6.13: Obtaining the *cookie-cutting-distance* used to compute B8. (a) Nucleus image from the cytometer. (b) Corresponding mask from the cytometer. (c) Chromatin segmentation. (d) Chromatin blob masks. (e) Distance transform of the nucleus mask displayed using a *heat* colour map. (f) Portions of *(e) cut out* by the blob masks.

(e)

(f)

(d)



Figure 6.14: Box-and-whisker plot of the feature M computed for each nucleus on slide 6.

Rather than computing summary statistics of a nucleus feature for a slide to obtain slide features, an alternative is to:

- 1. compute a relative frequency histogram for a given feature (choosing appropriate bin widths); and
- 2. to use each bin as a slide feature.

This was done for the feature M. A relative frequency histogram was computed for each slide using a bin width of 64 to ensure good coverage in at least three bins. The following code fragment shows how this was done in DImPAL for slide 6:

For the 139 slides used in this experiment this yielded frequencies in up to five bins. These bins are hereinafter referred to as slide features F1, F2, F3, F4, and F5 respectively. Split box-and-whisker plots for these features (split by class) are shown in Figure 6.15. The first three features were then used as the predictor variables in a logistic regression classifier. To evaluate the performance of the classifier the leave-one-out method was used. This involves:

- 1. Starting with all of the 139 observations;
- 2. Omitting a single observation (the *holdout* observation);
- 3. Fitting the logistic regression model using the remaining 138 observations (on three variables);
- 4. Using the fitted model to predict the probability for the holdout observation; and
- 5. Repeating steps 1 to 4 until all 139 observations have been classified.

A listing of the R program, written by the author, to perform this analysis is show in Appendix I. The logistic regression model is fitted in R as a *generalised linear model* (glm) based on the binomial distribution with the logit transform as the *canonical link* (see Venables & Ripley (1999, p. 212)). The following code fragment shows how the logistic regression classifier is fitted:

The *weights* argument permits a vector of weights to be passed to the fitting procedure, one for each observation in the data. These are used to weight the contributions of each observation to the maximum-likelihood estimate. When the weights are all 1 then the model is implicitly fitted using the class proportions in the training set as estimates of the prior probabilities and with equal misclassification costs; i.e. the penalty cost of classifying a normal as an abnormal is the same as that for classifying an abnormal as normal. Figure 6.16 shows the resulting empirical ROC curve and tabulated sensitivity, specificity, and correct classification rates (CCR) for a range of cutpoints of the logistic classifier. A listing of the R program, written by the author, to perform the ROC curve analysis is shown in Appendix K.

In reality the cost of misclassifying an abnormal is higher than that of misclassifying a normal. Ripley (1996, p. 58) states that:

it is quite common in medical diagnosis for the abundance of the classes in the training set not to reflect their importance in the problem. Often when the training data are a random sample from the population, the vast majority of cases are 'normals' yet the cost of mis-classifying a diseased case as normal is ℓ times higher than that of a false positive. In screening problems ℓ can be ten or more.

To introduce this penalty in the model fitting process it is necessary to weight the contributions of all of the normal observations (in the training set) to the loglikelihood by a factor $\omega = 1/\ell$ (Ripley, 1996, p. 59). Figure 6.17 shows the resulting ROC curve and associated classification summary when $\ell = 10$.

In reality the population (natural) priors are different from the proportions of normals and abnormals in the experimental data set. For example, in a report published by the National Coordinating Centre for Health Technology Assessment in the United Kingdom it is stated that "currently (data for England, 1997-98) about 8-9% of smears are considered 'abnormal' (any grade)" (Payne et al., 2000, p. 2). To introduce population priors into the model fitting process it is necessary to weight the contributions of the observations in class k (in the training set) by

$$\omega_k = N\pi_k/n_k$$

where N is the total number of observations in the training set, π_k is the prior probability for the class, and n_k is the number of observations for this class (Ripley, 1996, p. 111). Figure 6.18 shows the resulting ROC curve and associated classification summary when the prior probability of a slide being abnormal is taken to be 8% and the misclassification costs are assumed to be equal. However, it must be stressed that this ROC curve was estimated from the test data which, like the unweighted training data, does not contain a proportion of normals and abnormals consistent with the natural priors. What this means is that whilst the estimate of the AUC is admissible (because the AUC is independent of priors), the shape of the ROC curve is not. In particular, for any given cutpoint the CCR is not correct for the natural priors. An estimate of the correct CCR is given by

$$\sum_k \pi_k c_k;$$

where c_k is the proportion of observations in class k correctly classified in the test set (Ripley, 1996, p. 67). This estimate is shown as *Corrected CCR* in Figure 6.18. Figure 6.19 shows the resulting ROC curve and associated classification summary when, in addition to using the natural priors, the cost of misclassifying an abnormal is taken to be ten times that of misclassifying a normal. A comparison of the AUC estimate in Figure 6.16 with that in Figure 6.18, and of the AUC estimate in



Figure 6.15: Split box-and-whisker plots for the features F1, F2, F3, F4, and F5.

Figure 6.17 with that in Figure 6.18 confirms that the AUC is indeed independent of priors.

6.4.9.1 Interpretation of the results of Experiment 1

Table 6.8 summarises the performance of the classifiers in this experiment with that of several other MACs-based classifiers for cervical cancer reported in the literature. It is difficult to compare the various results for several reasons:

- 1. Whilst all of the studies are based on intermediate cells, some of the studies— Jones (2001), Kasper et al. (1997), Kemp et al. (1997)—have deliberately excluded diagnostic cells (i.e. only normal-looking cells have been sampled from abnormal slides), whilst others have not.
- 2. The composition of diagnoses for the abnormal class is not the same for each study.
- 3. Different stains and preparation methods have been used.
- 4. Classifier performance is reported in several different ways. Two of the studies— Kasper et al. (1997), Isenstein et al. (1995)—report classifier performance only



Figure 6.16: The empirical ROC curve and associated classification summary for the logistic classifier built using the sample priors and equal costs of misclassification.



Figure 6.17: The empirical ROC curve and associated classification summary for the logistic classifier built using the sample priors and assuming that the cost of misclassifying an abnormal is 10 times worse than that of misclassifying a normal.



Cutpoint	Sensitivity	Specificity	CCR (%)	Corrected CCR $(\%)$
0	1.00	0.00	28.8	8.0
0.1	0.62	0.84	77.7	82.1
0.2	0.45	0.95	80.6	91.0
0.3	0.32	0.97	78.4	91.8
0.4	0.28	0.99	78.4	93.3
0.5	0.18	0.99	75.5	92.5
0.6	0.15	0.99	74.8	92.3
0.7	0.12	0.99	74.1	92.1
0.8	0.10	0.99	73.4	91.9
0.9	0.10	0.99	73.4	91.9
1	0.00	1.00	71.2	92.0

Figure 6.18: The empirical ROC curve and associated classification summary for the logistic classifier built using population priors and equal costs of misclassification.



Cutpoint	Sensitivity	Specificity	CCR (%)	corrected CCR (%)
0	1.00	0.00	28.8	8.00
0.1	1.00	0.03	30.9	10.8
0.2	0.92	0.36	52.5	40.9
0.3	0.88	0.61	68.3	62.8
0.4	0.70	0.72	71.2	71.6
0.5	0.60	0.83	76.3	81.0
0.6	0.60	0.92	82.7	89.4
0.7	0.52	0.95	82.7	91.6
0.8	0.35	0.98	79.9	92.9
0.9	0.18	0.98	74.8	91.5
1	0.00	1.00	71.2	92

Figure 6.19: The empirical ROC curve and associated classification summary for the logistic classifier built using population priors and assuming that the cost of misclassifying an abnormal is 10 times worse than that of misclassifying a normal.

in terms of the CCR. The CCR corresponds to one particular operating point on the ROC curve. Moreover this figure does not indicate what proportion of abnormals and what proportion of normals were misclassified. Another two studies—Kemp et al. (1997), Garner et al. (1994)—present an ROC curve but do not report the AUC and associated standard error. Jones (2001) reports the AUC but no standard error.

- 5. Two of the studies combine MACs features with other features: Isenstein et al. (1995) combines MACs features with cell-based contextual features, and Garner et al. (1994) combines MACs features with the detection of diagnostic cells.
- 6. None of the studies, except the present study, state misclassification costs. This suggests that they have assumed equal costs of misclassification.
- 7. None of the studies, except the present study, state the priors assumed. This suggests that they have used sample proportions as estimates of the priors.

The only other study to use the Papanicolaou stain is that of Jones (2001, Chapter 6), hereinafter called the DJ study. The reported AUC (unfortunately no SE is reported) is very close to those AUC values reported in the present study. However Jackway & Bamford (2000, p. 9) and Mehnert & Bamford (2002, p. 5) have noted several problems with the slides used in the DJ study:

- 1. The slides were 6 years old at the time they were scanned by the Cytometrics Project cytometer and the stain may have faded;
- Many of the slides are scanty (with very few cells), possibly because they were prepared with a beta-version ThinPrep[™] machine; and
- 3. Most of the abnormal slides were used in several in-house imaging experiments and were potentially left exposed to light and dust more than the normal slides. In contrast most of the normal slides were kept in boxes out of the light.

It is known in quantitative histochemistry that stains can fade over time when exposed to heat and light (Pearse, 1980). The DJ study derives features from the granold spectrum which is in turn derived from a thresholding of each nucleus image over all possible grey-levels. As a consequence it is possible that it is the grey-tone difference due to fading that was detected in the DJ study. The validity of using grey-level features to detect MACs in Papanicolaou stain is, in any case, questionable because the Papanicolaou stain is not stoichiometric. In the present study no greylevel features have been used.

A comparison of the present study with the remaining studies indicates a similar level of performance. However, the present study is unique in that:

- 1. it demonstrates that the MACs phenomenon can be detected in Papanicolaou stain; and
- 2. it does not use grey-level features.

6.4.10 Experiment 2: Feature selection

Based on the number of nuclei sampled from a slide (after artefact rejection), and the means and standard deviations of the nucleus features N1 to N28, excluding $N7^{25}$, it is possible to define a set of 55 slide features. These are listed in Appendix L. A graphical summary—a split box-and-whisker plot—of the distribution of each feature for both the normal and abnormal slides is shown in Appendix M. The aim of this experiment was to determine a subset of these slide features (feature selection) that have the most discriminatory power for the Pap smear classification problem. The principal reasons for wanting to do this are:

- 1. to reduce the dimensionality of feature space in order to avoid the curse of dimensionality; and
- 2. to attempt to identify biologically significant/plausible features.

In this experiment the method chosen for feature selection is that implemented by the *stepAIC* procedure in the MASS library of Venables & Ripley (1999) with the argument *direction* set to "both". Used in this way the procedure, given a starting model, iteratively adds a variable to the current model or removes a variable (feature) from it, the choice being determined by a penalised measure of fit (the default is the AIC). The procedure seeks to minimise the measure of fit. Johnson & Wichern (1988, p. 529) note that:

 $^{^{25}}$ Feature N7 was excluded because the artefact rejection strategy ensures that it has only a small range of values.

Study	Preparation /	Abnormal	Normals /	Cells per	CCR	AUC±	
	Stain	Class	Abnormals	slide	(%)	standard error	
Experi-	AutoCyte TM /	CIN I,	101/47	> 600	81.3	0.791 ± 0.046	
ment 1	Papanicolaou	CIN II,			84.2	0.811 ± 0.044	
		CIN III			93.3	0.785 ± 0.047	
_		0.000 0.000			92.9	0.807 ± 0.045	
Jones	ThinPrep [™] /	CIN II,	25/9	96	82.4	$0.834 \pm 0.090^{*}$	
(2001)	Papanicolaou	CIN III					
Kasper et	LBP /	LSIL,	78/53	36 on	82.4	NA	
al. (1997)	Feulgen	HSIL,		average			
		invasive					
		cancer					
Kemp et	conventional	severe	251/144	75 from	76.2	$0.818 \pm 0.024^{**}$	
al. (1997)	smear /	dysplasia		normals,			
	Feulgen-			maxi-			
	Thionin			mum of			
				150 from			
				abnor-			
				mals			
Isenstein	$\mathrm{ThinPrep}^{\mathrm{TM}} \; / \;$	HGSIL,	70/76	40	85.7	NA	
et al.	Thionin-	carcinoma					
(1995)	Feulgen						
Garner et	conventional	low grade	training	500	NA	$0.770 {\pm} 0.016^{**}$	
al. (1994)	smear /	lesions	set:				
	Thionin- SO_2 -	(mild	330/190				
	Feulgen	dysplasia)	test set:				
			676/383				
Garner et	as above	high grade	training	500	NA	$0.834 \pm 0.012^{**}$	
al. (1994)		lesions	set:				
		(moderate	330/240				
		and severe	test set:				
		dysplasia)	676/501				

 Table 6.8: Comparison of the present study with similar studies published in the literature.

 \ast Standard error not reported. Equation 6.1 has been used to estimate it.

** Not reported. The AUC has been estimated by optically scanning the published ROC curve into a computer, cropping the image to the region enclosed by the axes, scaling the resulting image so that both axes had the same scale, flood-filling the area of interest, and dividing the number of pixels in the area of interest by the total number of pixels. The standard error has been estimated using Equation 6.1.

choosing a subset of variables that seems optimal for a given data set is especially disturbing if classification is the objective. At the very least, the derived classification function should be evaluated with a validation sample. As Murray [(1977)]... suggests, a better idea might be to split the sample into a number of batches and determine the "best" subset for each batch. The number of times a given variable appears in the best subsets provides a measure of the worth of that variable for future classification.

The latter idea is essentially the approach used in this experiment. The R program holdout.R, listed in Appendix J, was written by the author to perform multiple repetitions of model fitting, feature selection, and testing for different random splits of the data into training and test data sets (repeated holdout method). In each iteration:

- 1. a specified proportion of the abnormals and of the normals are randomly selected (stratified random sample) and *held out* as a test set, leaving the remaining data to be used as a training set;
- 2. a logistic regression model consisting of the intercept only (no variables) is fitted to the training data;
- 3. the BIC (see the end of Section 6.3.2.2) of the fitted model is computed;
- 4. a variable is then either added to the model or removed from the model, the choice depending on which yields a smaller BIC;
- 5. step 4 is repeated until a model is found for which the addition or omission of a variable does not reduce the BIC; and
- 6. the selected model is then used to classify the observations in the test set and to compute the AUC of the empirical ROC curve.

At the end of the iterations:

- 1. the mean and variance of the computed AUC values are computed; and
- 2. a frequency table is output listing the number of times each feature made an appearance in a selected model.

Without prior knowledge of the variables that are likely to be important, there exist two possibilities for the initial model to use in stepwise selection: (i) a model including all of the variables; and (ii) a model including no variables (intercept only). Epidemiologic methodologists prefer the first option because this makes it possible to control *confounding*²⁶ that may exist in the data set (this strategy is discussed in detail in Kleinbaum (1994)). However, Hosmer & Lemeshow (2000, p. 92) state that:

the major problem with this approach is that the model may be "overfit," producing numerically unstable estimates.

Hosmer & Lemeshow (2000, p. 121) also state that caution is urged when considering a model with many variables because:

significant linear regressions may be obtained from "noise" variables, completely unrelated to the outcome variable.

For these reasons, and given that the amount of data available in this experiment for training permits the building of a classifier with three or fewer variables (depending on the holdout proportion chosen), it was decided to use an intercept only initial model (no variables). In addition, it was decided to use BIC rather than AIC for feature selection because, as noted in Section 6.3.2.2, the AIC has a tendency to choose more and more variables as the size of the training set increases.

Ripley (1996, p. 6) notes that "sometimes good features can be found by training a classifier on a large number of features and extracting the good ones..., but most often problem-specific insights are used". With this in mind it was decided to exclude all of the slide features based on nucleus grey-level measurements (these are the features not marked with an asterisk in Appendix L) based on the knowledge that the Papanicolaou stain is not stoichiometric. To gauge the efficacy of the feature selection it was decided to include an additional random feature, X56. This feature consists of 139 observations drawn from a uniform distribution (generated using the *runif()* command in R). The *holdout*.*R* program (Appendix J) was executed using a holdout proportion of $\frac{1}{3}$ and 100 iterations. A summary of the resulting AUC values is as follows:

²⁶ A predictor variable that is associated with both the response variable and a primary predictor variable is called a *confounder* by epidemiologists (Hosmer & Lemeshow, 2000, p. 70). The relationship between this predictor variable and the response variable is then said to be confounded.

Table 6.9: The frequency with which features were selected in the 100 repetitions of stepwise model fitting.

X1	X2	X4	X5	X6	X7	X16	X17	X25	X26	X27	X28	X31	X53	X54	X55	X56
1	3	2	71	15	11	32	3	1	1	2	8	1	21	2	36	2

> summary(AUC)
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.5361 0.7244 0.7890 0.7753 0.8293 0.9161

The mean and standard deviation of the AUC values is: 0.775 ± 0.075 . The frequency with which each feature appeared in a selected model is shown in Table 6.9.

6.4.10.1 Interpretation of the results of Experiment 2

The fact that the random feature, X56, was selected only 2 times is reassuring because it does not have any discriminatory power whatsoever. Features X5, X16, and X55 are of particular interest because they are the only features to appear in more than 30% of the selected models. Feature X5 is based on nuclear area. That this should be an important feature seems reasonable because a significant variability in nuclear size is one of several known principal morphologic differences between normal and cancer cells (Koss, 1992, p. 129) and because nuclear enlargement is also characteristic of precancerous lesions (Koss, 1992, p. 390). The second feature is based on the total blob area within a nucleus. This too seems reasonable because *hyperchromasia*²⁷ is another characteristic shared by cancerous and precancerous lesions (Koss, 1992, p. 132, 390). The size of the total blob area relative to the nucleus area is a measure of hyperchromasia. The last feature is a measure of margination. Again this feature seems reasonable given the results of Experiment 1.

At this point the reader might be wondering why the three most "important" features identified in this experiment were not used in place of F1, F2, and F3 in a repetition of Experiment 1 (to determine the performance that could be expected from a classifier based on the selected features). The reason is that the resulting estimate of the AUC would be optimistically biased because the same data has already been used for feature selection. An additional *validation* data set is needed to obtain an unbiased estimate.

²⁷ A stained nucleus is said to be hyperchromatic if it is excessively stained. Such nuclei are diffusely dark in appearance (Koss, 1992, p. 132).

6.5 Summary and discussion

This chapter has:

- Discussed the rationale for designing a MACs-based classifier for cytological screening, and examined how such a classifier can be designed.
- Presented a detailed overview of statistical pattern recognition, addressing important issues such as the curse of dimensionality, dimensionality reduction, choice of classifier, and evaluation of classifier performance.
- Demonstrated the practical application of statistical pattern recognition and chromatin segmentation features to the problem of automated Pap smear screening. In particular it was demonstrated that it is possible to quantify a qualitative description of chromatin used by cytoprofessionals—margination—using the chromatin segmentation and representation and description methods described in the preceding chapters.
- Provided empirical evidence that it is possible to detect differences in the pattern of nuclear chromatin between samples of cells from a normal Pap smear and those from an abnormal Pap smear. These nuclear texture differences are supportive of the existence of the MACs (malignancy associated changes) phenomenon.
- Identified several nuclear features, including margination, that empirical evidence suggests have the most discriminatory power for the Pap smear screening problem.

The experimental results reported in this chapter are very encouraging. Nevertheless, there are several caveats:

1. It is known that the quality of the staining pattern produced by the Papanicolaou stain can be quite variable. For example, air drying of the cytological material prior to staining leads to reduced nuclear staining and loss of transparency of the cytoplasm (Schulte & Wittekind, 1994, p. 202). Consequently, until such time as the trained classifiers are tested on different batches of slides from a single pathology laboratory, and between different laboratories, the generalisability of the reported classifiers remains an open question.

- 2. A number of leukocytes were not rejected by the artefact rejection procedure and hence their presence in the data set may confound the results. Further research is needed to determine whether this is the case. In particular this requires the development of an improved artefact rejection algorithm.
- 3. Although it has been reported in the literature that at least 500 nuclei per slide are needed to obtain stable estimates of the means and standard deviations of nucleus features (based on experimental results), this needs to be verified independently.
- 4. Diagnostic cells were not excluded from the data used in the experiments. On the other hand they were not deliberately sought either. Consequently it is possible that the samples of cells from the abnormal slides include both MACaffected and diagnostic cells. To build a classifier specifically for the purpose of detecting MACs, it would be necessary to assemble a database of cells from normal slides and normal-looking cells from abnormal slides. Methods for constructing such a database include:
 - (a) manually selecting normal-looking nuclei from abnormal slides;
 - (b) developing a set of features to automatically detect and reject abnormal cells as artefacts; and
 - (c) examining individual patient histories and selecting apparently normal Pap smears preceding an abnormal one.

Chapter

Summary and Conclusions

The truth is rarely pure and never simple

Oscar Wilde

This chapter reviews the thesis, summarises its key contributions and findings, and discusses the implications of these results. It also outlines the limitations of the research undertaken, and the opportunities for further research.

7.1 Thesis review

Chapter 1 This chapter explained that the research described in this thesis constitutes part of a larger research initiative called the Cytometrics Project. A major goal of the project is to develop an automated image analysis system (cytometer) for screening Papanicolaou-stained cervical smears. It was noted that the conventional method of screening, the Papanicolaou (Pap) test, is a highly labour-intensive complex process and that at least 1 in every 10 to 20 positive cases are missed in conventional routine screening. The two principal causes of false negatives are: (i) human misinterpretation of smears, and (ii) sampling error. It was noted that automated screening, based on the same *rare event* approach used in the conventional Pap test, can reduce false negatives attributable to misinterpretation but cannot address sampling error. It was noted that a phenomenon known as malignancy associated changes (MACs) may offer a solution. In modern usage MACs refers to subtle subvisual changes—predominantly textural changes—in otherwise normalappearing cells on cervical atypical smears. It was noted that measurements (features) characterising nuclear texture are reported, in the quantitative cytology literature, to have the most discriminatory power. The chapter then presented an overview of statistical pattern recognition in computer vision, and a review of the different approaches to texture feature extraction published in the literature. It was noted that the majority of methods published in the literature for quantitatively characterising nuclear texture (i.e. chromatin distribution) stem from the stochastic definition of texture. The chapter argued that a structural approach to quantifying nuclear texture is more appropriate because: (i) cytoprofessionals describe chromatin distribution using terms such as clumping, margination, granulation, condensation, and clearing; and (ii) the original qualitative description of MACs, first described by Nieburgs et al. (1959), is of visible structural changes in chromatin distribution. It was pointed out that the key to the structural approach is chromatin segmentation, and that all of the chromatin segmentation algorithms published in the literature suffer from one or both of the following drawbacks: (i) the need to specify, a priori, one or more subjective operating parameters (thus affecting robustness to variations in illumination and staining); and (ii) segmentations that are not consistent with what a human is likely perceive to be blobs or particles. The chapter then stated the primary and secondary aims of the thesis:

to develop a structural model of chromatin (as visualised by light microscopy) from which features can be defined that can be directly related to the terms and adjectives used by cytoprofessionals to describe chromatin distribution/texture,

and

to demonstrate that such features can be used to detect nuclear changes during neoplasia, and malignancy associated changes.

The chapter also presented a list of five objectives deemed necessary to meet these aims:

1. To develop a class of non-linear self-dual filters for the purpose of attenuating impulse-type noise in digital images;

- 2. To develop a robust algorithm for segmenting the chromatin in a digitised image of a cell nucleus (as visualised by light microscopy);
- 3. To develop a representation and description of the segmented chromatin that characterises the spatial relationship between chromatin regions and additionally incorporates scalar feature parameters associated with these regions;
- 4. To demonstrate that features derived from this representation and description can be related to the descriptive terms used by cytoprofessionals to describe chromatin distribution; and
- 5. To demonstrate that features so-derived can be used to discriminate between normal and abnormal Papanicolaou-stained cervical cytology slides.
- Chapter 2 This chapter reviewed the theoretical framework used throughout the thesis. In summary, the chapter reviewed: sets and ordering; complete lattices; metric spaces; mathematical morphology for complete lattices, binary and grey-scale images, and graphs.
- **Chapter 3** This chapter addressed the first objective as follows. The chapter discussed the rationale behind the desire to construct non-linear self-dual filters. It was noted that whilst all linear filters are self-dual, self-duality is not an intrinsic property of non-linear filters. It was pointed out that the generalisation of the theory of mathematical morphology to the complete lattice algebraic framework in the 1980s, was the catalyst for much of the research into the question of how to construct non-linear self-dual filters. The chapter then reviewed the principal lattice-theoretical approaches to constructing non-linear self-dual operators/filters: the activity ordering, centre and anti-centre, iterations of the centre and the middle filter, self-dual toggle mappings, self-dual operators based on the switch operator, self-dual operators based on folded ordering, and self-dual annular filters. The chapter then presented a new method based on the generalisation of folded ordering by means of the folding operator and fold-space. It was shown that any operator defined on this space, that is self-dual with respect to the second component, leads to a self-dual operator called a folding induced self-dual (FISF) on the original space. Moreover, it was shown that although fold-space is not itself a complete lattice,

it is possible to define morphological meta-operators on this space which are morphological operators on the first component. It was shown that the folded closing, originally devised by Evans et al. (1997), is a particular type of FISF and that other types of FISF can be designed with improved salt-and-pepper noise attenuation properties. It was noted that Heijmans & Keshet (2002) have related the FISF approach to their, more recent, theoretical framework for morphological image processing based on complete inf-semilattices.

- **Chapter 4** This chapter addressed the second objective as follows. The chapter presented an overview of the various approaches to grey-scale image segmentation published in the literature. The chapter then described the nature of chromatin as visualised by light microscopy. This was followed by a critical review of previous approaches to chromatin segmentation published in the literature. It was noted that all of the algorithms suffer from one or more drawbacks. The chapter then presented a new algorithm for chromatin segmentation, based on seeded region growing, that (in its preferred embodiment) overcomes these drawbacks. The chapter then presented a critical review of the seeded region growing algorithm of Adams & Bischof (1994), concluding that the algorithm is inherently dependent on the order of pixel processing. A new improved seeded region growing algorithm was then presented that overcomes this drawback. The chapter also presented a new implementation of an ascending priority queue for use in implementing the watershed transform (a particular case of seeded region growing). It was noted that the algorithm permits the implementation of a fast watershed transform suitable for use in automated cytometry where near real-time processing is required for an economically viable screening device.
- Chapter 5 This chapter addressed the third objective as follows. The chapter presented an overview of (geometric) adjacency graphs used in image processing: connectivity grid, region adjacency graph, and graphs related to the Voronoi diagram. This was followed by a review of the ordinary Voronoi diagram and related graphs—Delaunay graph, Gabriel graph, relative neighbour graph, β -skeletons—used to characterise the geometric adjacency of points in the plane. The chapter then reviewed the area Voronoi diagram, a generalisation of the ordinary Voronoi diagram, where the generators are areas rather than points. It was noted that the distance transform is the key to implementing the area Voronoi diagram in the digital setting. The chapter then presented a review of distance transform algorithms published in the literature.

The chapter then proved an equivalence between the distance transform of a binary image, where the underlying distance is based on a positive definite quadratic form, and its erosion by an elliptic poweroid structuring element. The chapter also presented a new algorithm, based on this result, for computing the exact Euclidean distance transform of a binary image manifested on the hexagonal grid. The chapter then described the generalisation, by means of the distance transform, of the area Voronoi diagram, the Delaunay graph, and the Gabriel graph to connected components of a binary image. Next, the chapter presented a generalisation of the well-known grey-level co-occurrence matrix method to vertex-weighted adjacency graphs (grey-scale graphs). The generalisation involves: (i) reducing the image under study to an adjacency graph with vertices corresponding to individual objects/regions and edges corresponding to an adjacency relationship between regions; (ii) assigning region attributes—average grey-level, area, perimeter, etc.—to each vertex of the adjacency graph; and (iii) computing a co-occurrence matrix—called an adjacency graph attribute co-occurrence matrix (AGACM)—for each attribute. It was noted that AGACM features can be used to quantitatively characterise blob-like and mosaic patterns in the plane such as chromatin particles. The chapter finally presented an overview of the types of parameters (attributes) that can be measured for image objects. It was noted that the convex ring—the class of sets in \mathbb{R}^n whose elements are finite unions of compact convex sets provides a realistic Euclidean model for digital images. It was also noted that the Minkowski functionals form the basis of any valid measurement that can be made on compact convex sets. Estimators for the Minkowski functionals for two-dimensional binary and grey-scale images were presented for both the square and hexagonal grids. Corrections to the literature, as well as new estimators for the perimeter were also proffered.

Chapter 6 This chapter addressed the last two objectives as follows. The chapter discussed the rationale for designing a MACs-based classifier for cytological screening, and examined how such a classifier can be designed. Next it presented a detailed overview of statistical pattern recognition, addressing important issues such as the curse of dimensionality, dimensionality reduction, choice of classifier, and evaluation of classifier performance. The chapter then presented a case study demonstrating the practical application of statistical pattern recognition and chromatin segmentation features to the problem of automated cervical cancer screening. It was shown that it is possible to quantify a qualitative description of chromatin used by cytoprofessionals, namely

margination, using the chromatin segmentation, and representation and description methods presented in the preceding chapters. It was noted that the results provide empirical evidence that it is possible to detect differences in the pattern of nuclear chromatin between samples of cells from a normal Papanicolaou-stained cervical smear and those from an abnormal smear. It was noted that these nuclear texture differences are supportive of the existence of the MACs (malignancy associated changes) phenomenon. It was also noted that the experimental results compare favourably with those reported in the literature for other stains developed specifically for automated cytometry.

7.2 Key contributions and findings

- A new and general method for constructing non-linear self-dual operators. The operators, called folding induced self-dual filters (FISFs), are constructed from arbitrary morphological (meta-) operators defined on an abstract space called fold-space. The folded closing, originally devised by Evans et al. (1997), is a particular type of FISF. Importantly, however, other types of FISF can be designed with improved salt-and-pepper noise attenuation properties.
- A new chromatin segmentation algorithm based on seeded region growing that is, in its preferred embodiment, parameter-free. Moreover the algorithm yields a segmentation consistent with what a human would perceive to be chromatin particles.
- A new seeded region growing algorithm that is independent of the order of pixel processing.
- A new implementation of an ascending priority queue that permits the implementation of a fast version of the watershed transform.
- A new theoretical result establishing that the erosion of the characteristic function of a binary image $X \in \mathcal{P}(\mathbb{R}^n)$ by the elliptic poweroid structuring element $g(x) = -(x^{\mathsf{T}}\mathbf{A}x)^{\frac{\alpha}{2}}$, where \mathbf{A} is $n \times n$ positive definite and $\alpha \in \mathbb{R}^+$, is equivalent to finding the distance transform of X, based on the distance $d(x,y) = \sqrt{(x-y)^{\mathsf{T}}\mathbf{A}(x-y)}$, raised to the power α .
- A new algorithm for computing the exact Euclidean distance transform of a binary image manifested on a hexagonal grid. When originally published, the algorithm had two major advantages over the only other known algorithm

devised by Vincent (1991*a*): (i) simplicity and (ii) suitability for hardware implementation using a pipeline architecture (because it is based on the algorithm of Huang & Mitchell (1994)).

- A new generalisation of the well-known grey-level co-occurrence matrix method to vertex-weighted adjacency graphs. The method, called the adjacency graph attribute co-occurrence matrix (AGACM) method, can be used to characterise blob-like and mosaic patterns in the plane.
- Corrections to formulae published in the literature and two new formulae for estimating the Minkowski functional $W_1^{(2)}$ (perimeter) for digital images manifested on square and hexagonal grids.
- A structural model of chromatin texture from which it is possible to define features that can be directly related to the terms and adjectives, used by cytoprofessionals, to describe chromatin distribution.
- New features—in particular *margination*—for the purpose of quantifying the distribution of nuclear chromatin.
- Empirical evidence that it is possible to detect differences in the pattern of nuclear chromatin between samples of cells from a normal Papanicolaou-stained cervical smear and those from an abnormal smear using these new features. These differences are supportive of the existence of the malignancy associated changes (MACs) phenomenon. The classification results are based on 99 normal and 40 abnormal Pap-stained cervical slides collected as part of the (Australian) National Cervical Screening Program. Moreover the results compare favourably with those reported in the literature for other stains developed specifically for automated cytometry. To the author's knowledge this is the first time that MACs have been demonstrated in Papanicolaou stain based on a sizable and uncontaminated data set.

7.3 Implications

New EDT algorithm

The new EDT algorithm presented in Chapter 5 is based on the idea of first embedding the hexagonal grid image in a rectangular grid and then applying the algorithm of Huang & Mitchell (1994) with the appropriate aspect ratio correction. Consequently, as noted in Chapter 5, (i) a highly efficient serial implementation is possible based on the fast implementation of Huang and Mitchell's algorithm devised by Lotufo & Zampirolli (2001); and (ii) an O(n) exact EDT should also be possible by replacing Huang and Mitchell's algorithm with the anisotropic version of the recent algorithm of Maurer et al. (2003). Hence the new EDT algorithm presented in this thesis is also the fastest available for the hexagonal grid.

MACs in Papanicolaou stain

The experimental results of Chapter 6 provide empirical evidence that it is possible to detect malignancy associated changes in Papanicolaou-stained cervical slides. Moreover, the reported classifier performance is on a par with the performance of similar classifiers reported in the literature for stoichiometric stains. Unfortunately these stoichiometric stains, whilst they are useful in automated cytometry, are are not acceptable to cytoprofessionals for visual screening. This is because "the Papanicolaou staining pattern contains a great deal of diagnostic information which is useful for the cytotechnologist, even if the computer does not always use that information" (Schulte & Wittekind, 1994, p. 208). Consequently there is significant advantage in being able to automate the screening of Papanicolaou-stained slides because there is no need for additional preparatory steps necessary to permit both automated screening using a stoichiometric stain and manual interpretation using the Pap stain.

7.4 Limitations

Folding induced self-dual filters

With regard to noise filtering, folding induced self-dual filters (FISFs) are, by virtue of the notion of folding, limited to filtering salt-and-pepper impulse noise. Referring to Figure 3.2, it is clear that positive impulse spikes must be of value greater than the crease and that negative impulse spikes must be of value less than the crease (fold point) so that folding will cause all of the impulse spikes to be negative.

Again, by virtue of the notion of folding, FISFs are generally not increasing.

Fold-space, as defined in Chapter 3, is not a complete lattice. This means that it is not possible to define mathematical morphology operators on this space, but only to define meta-operators that are morphological operators on the embedded space of folded values. However, the meta-supremum and meta-infimum operators associated with type 2 FISFs suggest the following approach to constructing a fold-space that is a complete lattice. The approach involves extending the set of indicator values from three elements $\{-1, 0, 1\}$ to four elements $\{\mp, -, +, \pm\}$ and defining the partial ordering on the set $\tilde{\mathcal{T}} = \mathcal{T} \times \{\mp, -, +, \pm\}$ shown in Figure 7.1. The space Fun $(E, \tilde{\mathcal{T}})$ is then a complete lattice. The folding operator β : Fun $(E, \mathcal{T}) \to$ Fun $(E, \tilde{\mathcal{T}})$ is defined as before (but with the indicator values -1, 0, 1 replaced with $-, \mp, +$ respectively):

$$\beta(f)(x) = \begin{cases} (f(x), +), & \text{if } f(x) < f^*(x) \\ (f(x), \mp), & \text{if } f(x) = f^*(x) \\ (f^*(x), -), & \text{if } f(x) > f^*(x). \end{cases}$$

The ordered pairs of the form (\cdot, \pm) result from application of the supremum or infimum on Fun (E, \tilde{T}) . In keeping with the idea underlying type 2 FISFs, the supremum or infimum of a pair of distinct but equivalent elements, i.e. two elements with the same folded value but different indicator values, is the equivalent element that will map back to the crease. This suggests the unfolding operator Υ : Fun $(E, \tilde{T}) \to$ Fun (E, T) defined

$$\Upsilon((f_1, f_2))(x) = \begin{cases} f_1(x), & \text{if } f_2(x) = + \\ c, & \text{if } f_2(x) = \mp \text{ or } f_2(x) = \pm \\ f_1^*(x), & \text{if } f_2(x) = -. \end{cases}$$

A similar approach is not possible for type 3 FISFs.

Chromatin segmentation

The implementation of the preferred embodiment of the chromatin segmentation algorithm, as described in Chapter 4, uses the divide lines of the watershed transform to determine a zone of influence (ZOI) around each regional minimum. The nature of digital space dictates that there will be slight discrepancies in the placement of the divide lines when the watershed transform is applied to the same input image with a different grid orientation, e.g. if the image is first rotated 90°, 180°, or 270° on the square grid (see Section 4.6.2). An alternative approach, that would permit independence of pixel order processing, would be to determine the ZOIs without rendering



Figure 7.1: Hasse diagram showing the partial ordering on $\mathcal{T} \times \{\mp, -, +, \pm\}$ for the case $\mathcal{T} = \{0, 1, 2\}$.

boundary lines—i.e. label the catchment basins of the watershed transform—and then sequentially and independently segment a chromatin particle in each.

Experimental results

Features

The size of the data set used in Experiment 1 of Chapter 6 limited the number of features that could be used to train and evaluate a classifier, for the purpose of discriminating between normal and abnormal Papanicolaou-stained cervical slides, to three features (using the leave-one-out method). Consequently only one nucleus feature, margination, was used to define three slide features. With more data, e.g. an order of magnitude larger, it would be possible to evaluate other features based on the chromatin segmentation, and representation and description methods presented in this thesis.

Generalisability

It is known that the quality of the staining pattern produced by the Papanicolaou stain can be highly variable. The Papanicolaou staining method uses a natural dye called hematein to stain the nucleus. Hematein is currently not synthesised in chemical laboratories and as a consequence "the quality of the Pap staining pattern depends largely on the quality of the commercially available hematein batches" (Schulte & Wittekind, 1994, p. 202). In addition, hematein solutions have a limited shelf life and deteriorate on standing (Schulte & Wittekind, 1994, p. 203). Consequently it is not yet clear whether the proposed chromatin segmentation features are robust to variations between batches of slides from a single pathology laboratory, and between different laboratories.

7.5 Opportunities for further research

Additional features

Of particular interest is the possibility of segmenting both dark and light particles in a nucleus and to compute an adjacency graph for the dark particles only (see Figure 7.2), the light particles only, and for both types of particle. Both adjacency graph attribute co-occurrence matrix (AGACM) features (introduced in Section 5.7) and other graph-based features (discussed below) could then be computed from these. An obvious choice of attribute for constructing an AGACM is the type of particle: dark or light. Based on an adjacency graph defined on both dark and light particles, for example, a 2×2 AGACM could be constructed for each nucleus. The elements of this AGACM would represent: the relative frequency with which two light particles are adjacent, two dark particles are adjacent, and a dark and a light particle are adjacent. In fact several such matrices could be defined for various orders of adjacency—1-adjacency, 2-adjacency, etc.—and for different adjacency graphs; e.g. perceptual and Delaunay. Other features that could be derived from the adjacency graphs include:

- the number of dark particle neighbours a dark particles has;
- the mean distance between a dark particle and its dark particle neighbours;
- the number of light particle neighbours a light particle has;
- the mean distance between a light particle and its light particle neighbours;
- the number of dark particle neighbours;
- the mean distance to dark particle neighbours;
- the number of light particle neighbours;
- the mean distance to light particle neighbours;
- the number of particle neighbours; and
- the mean distance to particle neighbours.

Another source of features stems from the histogram of the values in the distance transform of the binary mask of the background between segmented particles (see Figure 7.2). Three masks are possible: (i) the background between dark particles, (ii) the background between light particles, and (iii) the background between both types of particle. An interesting possibility here is to use the method of Russ (1999, p. 504) to extract a feature that characterises the clustering of the particles. The method involves:

- 1. computing the cumulative frequency histogram of the Euclidean distance transform of the background between the particles; and
- 2. fitting a least-squares straight line to the central 80% of the values of the distribution.

The slope of the fitted line is then a feature characterising the clustering of the particles.

The segmentation of chromatin into dark and light particles partitions the nucleus into three sets of pixels: (i) pixels belonging to dark particles, (ii) pixels belonging to light particles, and (iii) pixels belonging to the background between the particles. This is conceptually similar to the segmentation produced by the algorithm of Young et al. (1986) (described in Section 4.3). Hence the features proposed by Young et al. can be also be used for the chromatin segmentation algorithm proposed in this thesis. For example, Young et al. measure the *homogeneity* of the chromatin distribution using the following feature:

$$hetero = \frac{N_{\rm B} + N_{\rm W}}{N_{\rm B} + N_{\rm G} + N_{\rm W}},$$

where $N_{\rm B}$ is the number of pixels belonging to the dark (black) regions, $N_{\rm W}$ is the number of pixels belonging to the light (white) regions, and $N_{\rm G}$ is the number of pixels belonging to the background (grey) between the dark and light regions. It is important to realise, of course, that the exact meaning of this feature is different for the two segmentation methods, because the two segmentation approaches are fundamentally different.

Variations on these approaches to feature extraction, as well as several other possibilities for computing features are described in Mehnert & Jackway (2002).

Invariance to Papanicolaou staining quality

Further research is needed to determine whether the proposed chromatin segmentation features are robust to variations in Papanicolaou-staining between batches of slides from a single pathology laboratory, and between different laboratories.

Quantitative stains

The experiments in this thesis dealt solely with the Papanicolaou stain. It is likely that the chromatin segmentation method will perform at least as well on a stoichio-



Figure 7.2: Compendium of images and graphs derived from the chromatin segmentation algorithm.
metric stain such as Thionin-Feulgen. Moreover, stoichiometric stains open up the possibility of investigating the discriminatory power of grey-level features; e.g. the median and interquartile range of the distribution of dynamics values of the dark particles in the nucleus.

Number of nuclei sampled from each slide

The slide-based features used in the experiments were computed from all of the nuclei-like objects retained for each slide after artefact rejection. In the worst case, features were computed from 643 nuclei (slide 0), and in the best case, from 5975 nuclei (slide 95). As noted in Section 6.4.6, the literature suggests that at least 500 nuclei are necessary for robust estimation of means and standard deviations. Further research is needed to determine what is the minimum number needed. This will impact on the speed with which a slide can be automatically screened.

Multi-class classification

The experiments in this thesis considered only two classes: normals and abnormals. Further research is needed to determine whether it is possible to classify a slide according to the grade of abnormality; e.g. CIN I, CIN II, and CIN III.

Combination of features

As noted in Chapter 1, the research described in this thesis constitutes part of a larger initiative within the Cytometrics Project. Further research is needed, on a larger data set, to determine whether better discrimination is possible using a combination of chromatin segmentation features, other novel features within the Cytometrics Project such as *granold* features, and/or conventional features published in the literature.

Other cancers

The experiments in this thesis dealt only with the problem of screening for cervical cancer. However, as noted in Section 6.1, MACs have been detected in other tissues. Two particularly interesting avenues for further research include: (i) lung cancer screening based on sputum samples (see, e.g. Payne et al. (1997)), and (ii) oral cancer screening based on buccal smears.

Chromatin patterns in living cells

Recent research by Rousselle et al. (1999) suggests that chromatin segmentation features might also be used to assess the chromatin patterns in living cells during the cell cycle. This would make it possible to measure the alterations in the evolving chromatin patterns that result from pathological or environmental influences.

Novel new research

The encouraging experimental results obtained in Chapter 6, based on chromatin segmentation features, suggests that there is merit in returning to the original qualitative descriptions of MACs. Recall from Section 1.5 that the findings of a MAC study group, reported at the 16th Annual Meeting of the American Society of Cytology in 1968, were that a MAC positive nucleus must possess eight characteristics, one of which is that: "four of the circular areas are present in a row and two appear together with two of the circular areas of the row in a quadrant formation of four circular areas of clusters of circular areas occupying an entire lobule" (Meisels, 1969, p. 476). This suggests the possibility of employing the concept of a structuring graph (see Section 2.10.1) to detect such configurations in a geometric adjacency graph constructed on both the light and dark particles.

Bibliography

- Abutaleb, A. S. (1989), 'Automatic thresholding of gray-level pictures using twodimensional entropy', Computer Vision, Graphics, and Image Processing 47, 22– 32.
- ACS (2002), 'Cancer facts & figures 2002', http://www.cancer.org. American Cancer Society.
- Adams, R. & Bischof, L. (1994), 'Seeded region growing', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(6), 641–647.
- AHTAC (1998), Review of automated and semi-automated cervical screening devices, Technical report, The Australian Health Technology Advisory Committee (AHTAC), Commonwealth of Australia, Canberra. ISBN 0 642 36755 8.
- Al, I., Cornelisse, C. J., Pearson, P. L. & Ploem, J. S. (1978), 'Automated chromatin analysis using the Leyden Television Analysis System (LEYTAS)', Acta Histochemica. Supplementband 20, 211–215.
- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. & Walter, P. (2002), Molecular Biology of the Cell, fourth edn, Garland Science, New York.
- Albregtsen, F., Schulerud, H. & Yang, L. (1995), Texture classification of mouse liver cell nuclei using invariant moments of consistent regions, in S. Hlavác, ed., 'CAIP '95 Proceedings', Vol. 970 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 496–502.
- Allenby, R. B. J. T. (1991), Rings, Fields and Groups: An Introduction to Abstract Algebra, second edn, Edward Arnold, London.
- Anderson, G., MacAulay, C., Matisic, J., Garner, D. & Palcic, B. (1997), 'The use of an automated image cytometer for screening and quantitative assessment of

cervical lesions in the British Columbia Cervical Smear Screening Programme', *Cytopathology* **8**, 298–312.

- Anderson, J. A. (1982), Logistic discrimination, in P. R. Krishnaiah & L. N. Kanal, eds, 'Classification, Pattern Recognition, and Reduction of Dimensionality', Vol. 2 of *Handbook of Statistics*, North-Holland Publishing Company, Amsterdam, chapter 7, pp. 169–191.
- Anderson, P. S. & Runowicz, C. D. (2001), 'Beyond the Pap test: New techniques for cervical cancer screening', Women's Health in Primary Care 4(12), 753–758.
- Ayres, Jr., F. (1965), Modern Abstract Algebra, Schaum's Outline Series, McGraw-Hill, New York.
- Bamford, P. C. (1999), Segmentation of Cell Images with Application to Cervical Cancer Screening, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Bamford, P. C. & Jackway, P. T. (2001), 'Unsupervised scene segmentation', *Inter*national Patent Application. Number PCT/AU01/00787. Priority date 30 June 2000.
- Bamford, P. C. & Lovell, B. C. (1999), 'Method of unsupervised cell nuclei segmentation', *International Patent Application*. Number PCT/AU99/00231. Priority date 3 April 1998.
- Barnett, V. (1976), 'The ordering of multivariate data', Journal of The Royal Statistical Society Series A 139(3), 318–354.
- Bartels, P. H. & Vooijs, G. P. (1999), 'Automation of primary screening for cervical cancer: Sooner or later?', Acta Cytologica 43(1), 7–12.
- Bartels, P. H., Bahr, G. F. & Wied, G. L. (1969), 'Cell recognition from line scan transition probability profiles', Acta Cytologica 13(4), 210–217.
- Bartels, P. H., Wied, G. L. & Bahr, G. F. (1968), 'Cell recognition from equiprobable extinction range contours', Acta Cytologica 12(3), 205–217.
- Beare, R. & Talbot, H. (1999), Exact seeded region growing for image segmentation, in 'Proceedings of the Fifth International/National Biennial Conference on Digital Image Computing, Techniques, and Applications', Curtin University, Perth, Western Australia, pp. 132–137.

- Beil, M. (1992), 'Description of chromatin structures in cell nuclei', Acta Stereol 11, 129–134.
- Beil, M., Irinopoulou, T., Vassy, J. & Wolf, G. (1995), 'A dual approach to structural texture analysis in microscopic cell images', *Computer Methods and Programs in Biomedicine* 48(3), 211–219.
- Beineke, L. W. & Wilson, R. J., eds (1997), Graph Connections: Relationships Between Graph Theory and Other Areas of Mathematics, Vol. 5 of Oxford Lecture Series in Mathematics and its Applications, Clarendon Press, Oxford.
- Bengtsson, E. & Nordin, B. (1994), Densitometry, morphometry, and textural analysis as tools in quantitative cytometry and automated cancer screening, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 3, pp. 21–51.
- Berge, C. (1962), *The Theory of Graphs and its Applications*, John Wiley & Sons, New York.
- Beucher, S. & Lantuejoul, C. (1979), Use of watersheds in contour detection, in 'International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation', Rennes, France, pp. 2.1–2.12.
- Beucher, S. & Meyer, F. (1993), The morphological approach to segmentation: The watershed transformation, in E. R. Dougherty, ed., 'Mathematical Morphology in Image Processing', Marcel Dekker, New York, chapter 12, pp. 433–481.
- Birkhoff, G. (1948), Lattice Theory, Vol. 25 of American Mathematical Society Colloquium Publications, revised edn, American Mathematical Society, New York.
- Borgefors, G. (1984), 'Distance transformations in arbitrary dimensions', Computer Vision, Graphics, and Image Processing 27, 321–345.
- Borgefors, G. (1986), 'Distance transformations in digital images', Computer Vision, Graphics, and Image Processing 34, 344–371.
- Borgefors, G. (1988), Distance transformations in hexagonal grids, in V. Cantoni, V. Di Gesù & S. Levialdi, eds, 'Image Analysis and Processing II', Plenum Press, New York, pp. 213–220.
- Borgefors, G. (1989), 'Distance transformations on hexagonal grids', Pattern Recognition Letters 9, 97–105.

- Borgefors, G. & Sanniti di Baja, G. (1988), Skeletonizing the distance transform on the hexagonal grid, in 'Proceedings of the 9th International Conference on Pattern Recognition', IEEE Computer Society Press, Washington, pp. 504–507.
- Borisenko, V. I., Zlatopol'skii, A. A. & Muchnik, I. B. (1987), 'Image segmentation (state-of-the-art survey)', Automation and Remote Control 48(7), 837–879.
- Bradley, A. P. (1996), Machine Learning for Medical Diagnostics: Techniques for Feature Extraction, Classification, and Evaluation, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Bradley, A. P. (1997), 'The use of the area under the ROC curve in the evaluation of machine learning algorithms', *Pattern Recognition* **30**(7), 1145–1159.
- Breen, E. J. & Monro, D. H. (1994), An evaluation of priority queues for mathematical morphology, in J. Serra & P. Soille, eds, 'Mathematical Morphology and its Applications to Image Processing', Vol. 2 of Computational Imaging and Vision, Kluwer Academic Publishers, Dordrecht, pp. 249–256.
- Breu, H., Gil, J., Kirkpatrick, D. & Werman, M. (1995), 'Linear time Euclidean distance transform algorithms', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(5), 529–533.
- Butt, M. A. & Maragos, P. (1998), 'Optimum design of chamfer distance transforms', IEEE Transactions on Image Processing 7(10), 1477–1484.
- Castleman, K. R. (1996), Digital Image Processing, Prentice Hall, New Jersey.
- Chen, L. & Chuang, H. Y. H. (1994), 'A fast algorithm for Euclidean distance maps of a 2-D binary image', *Information Processing Letters* 51, 25–29.
- Chen, Y. Q., Nixon, M. S. & Thomas, D. W. (1995), 'Statistical geometrical features for texture classification', *Pattern Recognition* 28(4), 537–552.
- Chomet, B., La Porte, J. & McGrew, E. A. (1966), 'Atypical monocytes in patients with malignant tumors', Acta Cytologica 10(3), 197–199.
- Cibas, E. S. (2003), Cervical and vaginal cytology, *in* 'Cytology: Diagnostic Principles and Clinical Correlates', second edn, W.B. Saunders, London.
- Cohen, L. D. (1991), 'On active contour models and balloons', CVGIP: Image Understanding 53(2), 211–218.

- Cotsaces, C. & Pitas, I. (1998), Computing the watersheds of large three-dimensional images using limited random access memory, in H. J. A. M. Heijmans & J. B. T. M. Roerdink, eds, 'Mathematical Morphology and its Applications to Image and Signal Processing', Vol. 12 of Computational Imaging and Vision, Kluwer Academic Publishers, Dordrecht, pp. 239–246.
- Crombez, G. (1990), 'The Matheron representation theorem for gray-scale morphological filters', *Proceedings of the American Mathematical Society* **108**(3), 795–799.
- Cross, G. R. & Jain, A. K. (1983), 'Markov random field texture models', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5(1), 25–39.
- Cuisenaire, O. (1997), Region growing Euclidean distance transforms, in A. Del Bimbo, ed., 'Image Analysis and Processing', Vol. 1310 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 263–270.
- Cuisenaire, O. (1999), Distance Transformations: Fast Algorithms and Applications to Medical Image Processing, PhD thesis, Communications and Remote Sensing Laboratory (TELE), Université Catholique de Louvain, Belgium.
- Cuisenaire, O. & Macq, B. (1999), 'Fast Euclidean distance transformation by propagation using multiple neighborhoods', Computer Vision and Image Understanding 76(2), 163–172.
- Danielsen, H. E., Farrants, G. & Reith, A. (1989), 'Characterization of chromatin structure by image analysis – a method for the assessment of changes in chromatin organization', *Scanning Microscopy Supplement* 3, 297–302.
- Danielsson, P.-E. (1978), 'A new shape factor', Computer Graphics and Image Processing 7, 292–299.
- Danielsson, P.-E. (1980), 'Euclidean distance mapping', Computer Graphics and Image Processing 14, 227–248.
- Dash, M. & Liu, H. (1997), 'Feature selection for classification', Intelligent Data Analysis 1, 131–156.
- de Campos Vidal, B., Schlüter, G. & Moore, G. W. (1973), 'Cell nucleus pattern recognition: Influence of staining', *Acta Cytologica* **17**(6), 510–521.
- DeMay, R. M. (1997), 'Common problems in Papanicoaou smear interpretation', Archives of Pathology & Laboratory Medicine **121**(3), 229–238.

- DePree, J. D. & Swartz, C. W. (1988), *Introduction to Real Analysis*, John Wiley & Sons, New York.
- Devijver, P. A. & Kittler, J. (1982), *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London.
- Digabel, H. & Lantuejoul, C. (1978), Iterative Algorithms, Vol. 8 of Practical Metallography. Special Issue, pp. 85–99.
- Dobrin, B. P., Viero, T. & Gabbouj, M. (1994), 'Fast watershed algorithms: Analysis and extensions', Proceedings of the SPIE – The International Society for Optical Engineering 2180, 209–220.
- Donnellan, T. (1968), Lattice Theory, Pergamon Press, Oxford.
- Doudkine, A., MacAulay, C., Poulin, N. & Palcic, B. (1995), 'Nuclear texture measurements in image cytometry', *Pathologica* 87, 286–299.
- Duda, R. O. & Hart, P. E. (1973), Pattern Classification and Scene Analysis, John Wiley & Sons, New York.
- Duda, R. O., Hart, P. E. & Stork, D. G. (2001), Pattern Classification, second edn, John Wiley & Sons, New York.
- Eggers, H. (1998), 'Two fast Euclidean distance transformations in \mathbb{Z}^2 based on sufficient propagation', Computer Vision and Image Understanding **69**(1), 106–116.
- Embrechts, H. & Roose, D. (1996), 'A parallel Euclidean distance transformation algorithm', Computer Vision and Image Understanding 63(1), 15–26.
- Evans, C. J., Svalbe, I. D. & Jones, R. (1997), Nonlinear image processing using alternative orderings, in B. I. Chaplin & W. H. Page, eds, 'DICTA'97 and IVCNZ'97 Proceedings of the First Joint Australia and New Zealand Biennial Conference on: Digital Image & Vision Computing—Techniques and Applications', Massey University, Palmerston North, New Zealand, pp. 177–182. ISBN:0-473-04947-3.
- FDA (1995), 'FDA approves new Pap smear screening system', http://www.fda. gov/. U.S. Food and Drug Administration Talk Paper T95-55.
- FDA (1996), 'Updates', *FDA Consumer* **30**(2), 3.

- Francos, J. M., Meiri, A. Z. & Porat, B. (1993), 'A unified texture model based on a 2-D Wold-like decomposition', *IEEE Transactions on Signal Processing* 41(8), 2665–2678.
- Freund, J. E. (1984), *Modern Elementary Statistics*, sixth edn, Prentice-Hall, Englewood Cliffs, New Jersey.
- Fu, K. S. & Mui, J. K. (1981), 'A survey on image segmentation', *Pattern Recognition* 13, 3–16.
- Fukunaga, K. (1990), Introduction to Statistical Pattern Recognition, Computer Science and Scientific Computing, second edn, Academic Press, San Diego.
- Gabriel, K. R. & Sokal, R. R. (1969), 'A new statistical approach to geographic variation analysis', *Systematic Zoology* **18**(3), 259–278.
- Garner, D. M., Harrison, A., MacAulay, C. & Palcic, B. (1994), Cyto-Savant[™] and its use in automated screening of cervical smears, in G. L. Wied, P. H. Bartels, D. L. Rosenthal & U. Schenck, eds, 'Compendium on the Computerized Cytology and Histology Laboratory', Tutorials of Cytology, Chicago, Illinois, pp. 346–352.
- Giardina, C. R. & Dougherty, E. R. (1988), Morphological Methods in Image and Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey.
- Giménez-Mas, J. A., Sanz-Moncasi, M. P., Remón, L., Gambó, P. & Gallego-Calvo, M. P. (1995), 'Automated textural analysis of nuclear chromatin', Analytical and Quantitative Cytology and Histology 17(1), 39–47.
- Giroud, F. (1994), Standardization of specimen fixation in preparation for cytometric evaluation, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 13, pp. 186– 199.
- Gonzalez, R. C. & Woods, R. E. (1992), Digital Image Processing, Addison-Wesley, Reading, Massachusetts.
- Greening, S. E. (1994), The impact of computerized microscopy and automated cancer screening on the cytoprofessional, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 25, pp. 328–337.

- Greer, B. E. (1997), 'The gynecologist's perspective of liability and quality issues with the Papanicolaou smear', Archives of Pathology & Laboratory Medicine 121(3), 246–249.
- Grimaud, M. (1992), 'A new measure of contrast: The dynamics', Proceedings of the SPIE – The International Society for Optical Engineering 1769, 292–305.
- Grohs, H. K. (1994), Reflection on a changing environment: Gynecological screening in the age of automation, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, pp. xi–xv.
- Grohs, H. K., Zahniser, D. J. & Geyer, J. W. (1994), Standardization of specimen preparation through mono/thin-layer technology, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 12, pp. 176–185.
- Gruner, O. C. (1916), 'A study of the changes met with in the leucocytes in certain cases of malignant disease', *The British Journal of Surgery* **3**(11), 506–525. Cited in Nieburgs & Goldberg (1968).
- Guan, W. & Ma, S. (1998), 'A list-processing approach to compute Voronoi diagrams and the Euclidean distance transform', *IEEE Transactions on Pattern Analysis* and Machine Intelligence 20(7), 757–761.
- Guillaud, M., Doudkine, A., Garner, D., MacAulay, C. & Palcic, B. (1995), 'Malignancy associated changes in cervical smears: Systematic changes in cytometric features with the grade of dysplasia', *Analytical Cellular Pathology* 9(3), 191–204.
- Hadwiger, H. (1950), 'Minkowskische addition und subtraktion beliebiger punktmengen und die theoreme von Erhard Schmidt', *Math Zeitschrift* 53, 210–218.
- Hadwiger, H. (1957), Vorlesungen über Inhalt, Oberfläche und Isoperimetrie, Springer-Verlag, Berlin.
- Hale, E. (1989), 'The controversial Pap test', FDA Consumer 23(7), 20–23.
- Hallinan, J. S. (2000), Detection of Malignancy Associated Changes in Cervical Cells Using Statistical and Evolutionary Computation Techniques, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.

- Hanselaar, A. G. J. M., Poulin, N., Pahlplatz, M. M. M., Garner, D., MacAulay, C., Matisic, J., LeRiche, J. & Palcic, B. (1998), 'DNA-cytometry of progressive and regressive cervical intraepithelial neoplasia', *Analytical Cellular Pathology* 16, 11– 27.
- Hao, X., Bruce, C. J., Pislaru, C. & Greenleaf, J. F. (2001), 'Segmenting highfrequency intracardiac ultrasound images of myocardium into infarcted, ischemic, and normal regions', *IEEE Transactions on Medical Imaging* 20(12), 1373–1383.
- Haralick, R. M. (1979), 'Statistical and structural approaches to texture', *Proceedings of the IEEE* 67(5), 786–804.
- Haralick, R. M. (1989), Mathematical morphology and computer vision, in 'Conference Record. Twenty Second Asilomar Conference on Signals, Systems and Computers', Maple Press, San Jose, CA, pp. 468–479. Cat. No.88CH2835-7.
- Haralick, R. M. & Shapiro, L. G. (1985), 'Image segmentation techniques', Computer Vision, Graphics, and Image Processing 29, 100–132.
- Haralick, R. M., Shanmugam, K. & Dinstein, I. (1973), 'Textural features for image classification', *IEEE Transactions on Systems, Man, and Cybernetics* 3(6), 610– 621.
- Heijmans, H. J. A. M. (1994a), Morphological Image Operators, Advances in Electronics and Electon Physics. Supplement 25, Academic Press, Boston.
- Heijmans, H. J. A. M. (1994b), 'On the construction of morphological operators which are self-dual and activity-extensive', Signal Processing 38, 13–19.
- Heijmans, H. J. A. M. (1996), 'Self-dual morphological operators and filters', Journal of Mathematical Imaging and Vision 6, 15–36.
- Heijmans, H. J. A. M. & Keshet, R. (2001), Inf-semilattice approach to self-dual morphology, Technical Report PNA-R0101, Centrum voor Wiskunde en Informatica (CWI), Stichting Mathematisch Centrum (SMC), Amsterdam, The Netherlands. ISSN 1386-3711.
- Heijmans, H. J. A. M. & Keshet, R. (2002), 'Inf-semilattice approach to self-dual morphology', Journal of Mathematical Imaging and Vision 17(1), 55–80.
- Heijmans, H. J. A. M. & Ronse, C. (1990), 'The algebraic basis of mathematical morphology: I. dilations and erosions', *Computer Vision, Graphics, and Image Processing* 50(3), 245–295.

- Heijmans, H. J. A. M. & Ronse, C. (1999), 'Annular filters for binary images', *IEEE Transactions on Image Processing* 8(10), 1330–1340.
- Heijmans, H. J. A. M. & Vincent, L. (1993), Graph morphology in image analysis, in E. R. Dougherty, ed., 'Mathematical Morphology in Image Processing', Marcel Dekker, New York, pp. 171–203.
- Heijmans, H. J. A. M., Nacken, P., Toet, A. & Vincent, L. (1992), 'Graph morphology', Journal of Visual Communication and Image Representation 3(1), 24–38.
- Hirata, T. (1996), 'A unified linear-time algorithm for computing distance maps', Information Processing Letters 58, 129–133.
- Hosmer, Jr., D. W. & Lemeshow, S. (2000), *Applied Logistic Regression*, Wiley Series in Probability and Statistics, second edn, John Wiley & Sons, New York.
- Huang, C. T. & Mitchell, O. R. (1994), 'A Euclidean distance transform using grayscale morphology decomposition', *IEEE Transactions on Pattern Analysis* and Machine Intelligence 16(4), 443–448.
- Husain, O. A. N. (1994), The history of automated cell scanners, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 1, pp. 3–14.
- Ikeda, N., MacAulay, C., Lam, S., LeRiche, J., Payne, P., Garner, David; Konaka, C., Kato, H. & Palcic, B. (1998), 'Malignancy associated changes in bronchial epithelial cells and clinical application as a biomarker', *Lung Cancer* 19, 161–166.
- Isenstein, L. M., Zahniser, D. J. & Hutchinson, M. L. (1995), 'Combined malignancy associated change and contextual analysis for computerized classification of cervical cell monolayers', *Analytical Cellular Pathology* 9, 83–93.
- Jackway, P. & Bamford, P. (2000), Modifications to the ACCELL-SAVANT cytometer to enable the classification of Pap smears by the method of malignancy associated changes, Technical report, Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), SPRI Building, Technology Park, Mawson Lakes, South Australia. Commercial-in-confidence.
- Jackway, P. T. (1995), 'On dimensionality in multiscale morphological scale-space with elliptic poweroid structuring functions', Journal of Visual Communication and Image Representation 6(2), 189–195.

- Jackway, P. T. (1996), 'Gradient watersheds in morphological scale-space', IEEE Transactions on Image Processing 5(6), 913–921.
- Jain, A. K., Duin, R. P. W. & Mao, J. (2000), 'Statistical pattern recognition: A review', IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4– 37.
- Jain, A. K., Zhong, Y. & Dubuisson-Jolly, M.-P. (1998), 'Deformable template models: A review', Signal Processing 71, 109 –129.
- James, J. & Tanke, H. J. (1991), Biomedical Light Microscopy, 2nd edn, Kluwer Academic Publishers, Dordrecht.
- Johnson, R. A. & Wichern, D. W. (1988), Applied Multivariate Statistical Analysis, second edn, Prentice-Hall, Englewood Cliffs, New Jersey.
- Jones, D. G. (2001), Granolds: A Novel Method for Image Texture Analysis with Application to the Detection of Malignancy Associated Changes in Pap Smears, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Jones, D. G. & Jackway, P. T. (2000), 'Granolds: A novel texture representation', *Pattern Recognition* **33**, 1033–1045.
- Kasper, H. U., Haroske, G., Geissler, U., Meyer, W. & Kunze, K. D. (1997), 'Diagnostic and prognostic relevance of malignancy-associated changes in cervical smears', Analytical and Quantitative Cytology and Histology 19(6), 482–488.
- Kass, M., Witkin, A. & Terzopoulos, D. (1987), 'Snakes: Active contour models', International Journal of Computer Vision 1(4), 321–331.
- Kemp, R. A., MacAulay, C., Garner, D. & Palcic, B. (1997), 'Detection of malignancy associated changes in cervical cell nuclei using feed-forward neural networks', *Analytical Cellular Pathology* 14, 31–40.
- Kervrann, C. (2001), 'Bayesian image segmentation through level lines selection', Image Analysis & Stereology 20, 163–168.
- Keshet, R. (2000), 'Mathematical morphology on complete semilattices and its applications to image processing', *Fundamenta Informaticae* **41**, 33–56.

- Kimmel, R., Kiryati, N. & Bruckstein, A. M. (1996), 'Sub-pixel distance maps and weighted distance transforms', *Journal of Mathematical Imaging and Vision* 6, 223–233.
- Kirkpatrick, D. G. & Radke, J. D. (1985), A framework for computational morphology, in G. T. Toussaint, ed., 'Computational Geometry', Elsevier Science, Amsterdam, pp. 217–248.
- Klawe, H. & Rowiński, J. (1974), 'Malignancy associated changes (MAC) in cells of buccal smears detected by means of objective image analysis', *Acta Cytologica* 18, 30–33.
- Kleinbaum, D. G. (1994), *Logistic Regression: A Self-Learning Text*, Statistics in the Health Sciences, Springer-Verlag, New York.
- Kline, T. S. (1997), 'The Papanicolaou smear: A brief historical perspective and where we are today', Archives of Pathology & Laboratory Medicine 121(3), 205– 209.
- Knox, B., Ladiges, P. & Evans, B., eds (1994), Biology, McGraw-Hill, Sydney.
- Komitowski, D. & Zinser, G. (1985), 'Quantitative description of chromatin structure during neoplasia by the method of image processing', Analytical and Quantitative Cytology and Histology 7(3), 178–182.
- Kondo, A. & Taniguchi, K. (1986), 'Evaluation of the chromatin for cell images', Systems and Computers in Japan 17(9), 11–19.
- Koss, L. G. (1989), 'The Papanicolaou test for cervical cancer detection: A triumph and a tragedy', *Journal of the American Medical Association* **261**(5), 737–743.
- Koss, L. G. (1992), Diagnostic Cytology and its Histopathologic Bases, Vol. 1, fourth edn, J.B. Lipincott Company, Philadelphia.
- Kresch, R. (1998), Extension of morphological operations to complete semilattices and its applications to image and video processing, in H. J. A. M. Heijmans & J. B. T. M. Roerdink, eds, 'Mathematical Morphology and its Applications to Image and Signal Processing', Vol. 12 of *Computational Imaging and Vision*, Kluwer Academic Publishers, Dordrecht, pp. 35–42.
- Kurman, R. J. & Solomon, D. (1994), The Bethesda System for Reporting Cervical/Vaginal Cytologic Diagnoses: Definitions, Criteria and Explanatory Notes for Terminology and Specimen Adequacy, Springer-Verlag, New York.

- Lacey, A. J., ed. (1999), *Light Microscopy in Biology: A Practical Approach*, The Practical Approach Series, second edn, Oxford University Press, Oxford, England.
- Lamport, L. (1994), *PT_EX: A Document Preparation System: User's Guide and Reference Manual*, second edn, Addison-Wesley, Reading, Massachusetts.
- Levine, M. D. (1985), Vision in Man and Machine, McGraw-Hill, New York, chapter 9, pp. 422–479.
- Li, W. & Nyholt, D. R. (2001), 'Marker selection by Akaike information criterion and Bayesian information criterion', *Genetic Epidemiology* **21 Suppl 1**, S272–277.
- Linder, J. (1994), Considerations in automated cytology, in G. L. Wied, P. H. Bartels, D. L. Rosenthal & U. Schenk, eds, 'Compendium on the Computerized Cytology and Histology Laboratory', Tutorials of Cytology, Chicago, pp. 25–31.
- Linder, J. (1997), 'Automation of the Pap smear: A technology assessment perspective', Archives of Pathology & Laboratory Medicine 121(3), 282–286.
- Lønnestad, T. (1988), Connected filters for noise removal, in 'Proceedings of the 9th International Conference on Pattern Recognition', IEEE Computer Society Press, Washington, pp. 848–850.
- Lotufo, R. A. & Zampirolli, F. A. (2001), Fast multidimensional parallel Euclidean distance transform based on mathematical morphology, *in* D. L. Borges & W. Shing-Ting, eds, 'Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing', IEEE Computer Society, Los Alamitos, CA, pp. 100–105.
- Luczak, E. & Rosenfeld, A. (1976), 'Distance on a hexagonal grid', *IEEE Transac*tions on Computers C-25(5), 532–533.
- Madachy, R. J. & Fu, Y. S. (1988), Image analysis for automatic classification of mitotic cervical cells, in G. Harris & C. Walker, eds, 'Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society', IEEE, New York, pp. 372–374.
- Mairinger, T., Mikuz, G. & Gschwendtner, A. (1999), 'Nuclear chromatin texture analysis of nonmalignant tissue can detect adjacent prostatic adenocarcinoma', *The Prostate* **41**, 12–19.
- Majno, G. & Joris, I. (1995), 'Apoptosis, oncosis, and necrosis: An overview of cell death', American Journal of Pathology 146(1), 3–15.

- Maragos, P. (1989), 'A representation theory for morphological image and signal processing', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(6), 586–599.
- Matheron, G. (1975), Random Sets and Integral Geometry, Wiley, New York.
- Maurer, Jr., C. R., Qi, R. & Raghavan, V. (2003), 'A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(2), 265–270.
- McLachlan, G. J. (1992), Discriminant Analysis and Statistical Pattern Recognition, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York.
- Medioni, G., Lee, M.-S. & Tang, C.-K. (2000), A Computational Framework for Segmentation and Grouping, first edn, Elsevier Science B.V., Amsterdam.
- Mehnert, A. (1994), Digital morphometry: A taxonomy of morphological filters and feature parameters with application to Alzheimer's disease research, Master's thesis, Faculty of Science & Technology, Edith Cowan University, Perth, Western Australia.
- Mehnert, A. J. H. (2002), An independent review and evaluation of the Cytometrics Project's classifiers for Papanicolaou stained cervical slides, Technical Report TI009, Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), SPRI Building, Technology Park, Mawson Lakes, South Australia. Commercial-in-confidence.
- Mehnert, A. J. H. & Bamford, P. C. (2002), Cell-image data sets, Technical Report TI013, Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), SPRI Building, Technology Park, Mawson Lakes, South Australia. Commercial-in-confidence.
- Mehnert, A. J. H. & Jackway, P. T. (1997), 'An improved seeded region growing algorithm', *Pattern Recognition Letters* 18, 1065–1071.
- Mehnert, A. J. H. & Jackway, P. T. (1999a), Inducing self-duality through grey-scale folding, in 'Proceedings DICTA'99: The Fifth International/National Biennial Conference on Digital Image Computing, Techniques, and Applications', Curtin University, Perth, Western Australia, pp. 176–180. ISBN: 1863428380.

- Mehnert, A. J. H. & Jackway, P. T. (1999b), 'On computing the exact Euclidean distance transform on rectangular and hexagonal grids', *Journal of Mathematical Imaging and Vision* 11, 223–230.
- Mehnert, A. J. H. & Jackway, P. T. (2000), Folding induced self-dual filters, in J. Goutsias, L. Vincent & D. S. Bloomberg, eds, 'Mathematical Morphology and its Applications to Image and Signal Processing', Vol. 18 of Computational Imaging and Vision, Kluwer Academic Publishers, Boston, pp. 99–108.
- Mehnert, A. J. H. & Jackway, P. T. (2001), 'Chromatin segmentation', Australian Provisional Patent Application. Number PR6478.
- Mehnert, A. J. H. & Jackway, P. T. (2002), 'Chromatin segmentation', International Patent Application. Number PCT/AU02/00969. Priority date 19 July 2001.
- Meisels, A. (1969), 'Summary of investigational seminar on malignancy associated changes (MAC)', Acta Cytologica 13(8), 473–478.
- Meyer, F. (1978), Contrast features extraction, in J. L. Chermant, ed., 'Quantitative Analysis of Microstructures in Materials Science, Biology and Medicine', Dr. Riederer-Verlag GmbH, Stuttgart, pp. 374–380. Proceedings of the Symposium "Quantitative Analysis of Microstructures in Materials Science, Biology and Medicine", Caen, France, October, 1977.
- Meyer, F. (1979), 'Iterative image transformations for an automatic screening of cervical smears', The Journal of Histochemistry and Cytochemistry 27(1), 128– 135.
- Meyer, F. (1980), 'Quantitative analysis of the chromatin of lymphocytes: An essay on comparative structuralism', *Blood Cells* **6**, 159–172.
- Meyer, F. (1982), The perceptual graph: A new algorithm, in 'Proceedings of ICASSP 82. IEEE International Conference on Acoustics, Speech and Signal Processing', IEEE, New York, pp. 1932–1935.
- Meyer, F. (1986), 'Automatic screening of cytological specimens', Computer Vision, Graphics, and Image Processing 35, 356–369.
- Meyer, F. (1988), Skeletons in digital spaces, in 'Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances', Academic Press, London, chapter 13, pp. 257–296.

Meyer, F. (1989), 'Skeletons and perceptual graphs', Signal Processing 16, 335–363.

- Meyer, F. (1991), Un algorithme optimal de ligne de partage des eaux, *in* 'Proceedings of the 8^e Congres Reconnaissance des Formes et Intelligence Artificielle', Lyon-Villeurbanne, pp. 847–857.
- Meyer, F. & Beucher, S. (1990), 'Morphological segmentation', Journal of Visual Communication and Image Representation 1(1), 21–46.
- Meyer, F. & Serra, J. (1989), 'Contrasts and activity lattice', Signal Processing 16, 303–317.
- Minkowski, H. (1903), 'Volumen und oberfläche', Math. Ann. 57, 447–495.
- Morel, J.-M. & Solimini, S. (1995), Variational Methods in Image Segmentation: With Seven Image Processing Experiments, Vol. 14 of Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, Boston.
- Mullikin, J. C. (1992), 'The vector distance transform in two and three dimensions', CVGIP: Graphical Models and Image Processing 54(6), 526–535.
- Mumford, D. (1994), The Bayesian rationale for energy functionals, in B. M. ter Haar Romeny, ed., 'Geometry-Driven Diffusion in Computer Vision', Vol. 1 of *Computational Imaging and Vision*, Kluwer Academic Publishers, Dordrecht, Netherlands, pp. 135–146.
- Murray, G. D. (1977), 'A cautionary note on selection of variables in discriminant analysis', *Applied Statistics* **26**(3), 246–250.
- Nacken, P. F. M. (1996), 'Chamfer metrics, the medial axis and mathematical morphology', Journal of Mathematical Imaging and Vision 6, 235–248.
- Narendra, P. M. & Goldberg, M. (1980), 'Image segmentation with directed trees', IEEE Transactions on Pattern Analysis and Machine Intelligence 2(2), 185–191.
- Nieburgs, H. E. & Goldberg, A. F. (1968), 'Changes in polymorphonuclear leukocytes as a manifestation of malignant neoplasia', *Cancer* **22**, 35–42.
- Nieburgs, H. E., Zak, R. G., Allen, D. C., Reisman, H. & Clardy, T. (1959), Systemic cellular changes in material from human and animal tissues in presence of tumors, *in* 'Transactions of the Seventh Annual Meeting of the Inter-Society Cytology Council', pp. 137–144. Cited in Nieburgs & Goldberg (1968).

- Nordin, B. & Bengtsson, E. (1994), Specimen analysis by rare event, cell population, and/or contextual evaluation, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 4, pp. 44–51.
- Okabe, A., Boots, B. & Sugihara, K. (1992), Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester, England.
- O'Rourke, J. (1993), *Computational Geometry in C*, Cambridge University Press, Cambridge.
- Osher, S. & Fedkiw, R. (2003), Level Set Methods and Dynamic Implicit Surfaces, Vol. 153 of Applied Mathematical Sciences, Springer, New York.
- Paget, R. (1999), Nonparametric Markov Random Field Models for Natural Texture Images, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Paget, R. & Longstaff, I. D. (1998), 'Texture synthesis via a noncausal nonparametric multiscale Markov random field', *IEEE Transactions on Image Processing* 7(6), 925–931.
- Paglieroni, D. W. (1992), 'Distance transforms: Properties and machine vision applications', CVGIP: Graphical Models and Image Processing 54(1), 56–74.
- Pal, N. R. & Pal, S. K. (1993), 'A review on image segmentation techniques', Pattern Recognition 26(9), 1277–1294.
- Palcic, B. (1994), 'Nuclear texture: Can it be used as a surrogate endpoint biomarker?', Journal of Cellular Biochemistry, Supplement 19, 40–46.
- Palcic, B. & MacAulay, C. (1994a), Algorithms in preparation for decision making, in H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 5, pp. 52–61.
- Palcic, B. & MacAulay, C. (1994b), Malignancy associated changes: Can they be employed clinically?, in G. L. Wied, P. H. Bartels, D. L. Rosenthal & U. Schenk, eds, 'Compendium on the Computerized Cytology and Histology Laboratory', Tutorials of Cytology, Chicago, pp. 157–165.

- Palcic, B., MacAulay, C. E., Harrison, S. A., Lam, S., Payne, P. W., Garner, D. M. & Doudkine, A. (2000), 'System and method for automatically detecting malignant cells and cells having malignancy-associated changes', *United States Patent*. Patent number 6026174.
- Papanicolaou, G. N. (1928), New cancer diagnosis, in 'Proceedings of the Third Race Betterment Conference', Battle Creek, Michigan, p. 538. Cited in Papanicolaou & Traut (1941).
- Papanicolaou, G. N. & Traut, H. F. (1941), 'The diagnostic value of vaginal smears in carcinoma of the uterus', American Journal of Obstetrics and Gynecology 42(2), 193–206.
- Papanicolaou, G. N. & Traut, H. F. (1943), Diagnosis of Uterine Cancer by the Vaginal Smear, Commonwealth Fund, New York. Cited in Kline (1997).
- Patlak, M. (1996), 'New devices aim at improving Pap test accuracy', FDA Consumer 30(8), 9–13.
- Payne, N., Chilcott, J. & McGoogan, E. (2000), 'Liquid-based cytology in cervical screening: A rapid and systematic review', *Health Technology Assessment* 4(18), 1–73.
- Payne, P. W., Sebo, T. J., Doudkine, A., Garner, D., MacAulay, C., Lam, S., LeRichie, J. C. & Palcic, B. (1997), 'Sputum screening by quantitative microscopy: A reexamination of a portion of the National Cancer Institute cooperative early lung cancer study', *Mayo Clinic Proceedings* 72(8), 697–704.
- Pearse, A. G. E. (1980), *Histochemistry: Theoretical and Applied*, Vol. 3, Churchill Livingstone, Edinburgh.
- Picard, R. W. (1991), Texture Modeling: Temperature Effects on Markov/Gibbs Random Fields, PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
- Pieris-Caldwell, I., Stevenson, C. & Harding, J. (2002), Cervical screening in Australia 1998–1999, Technical Report CAN 11, The Australian Institute of Health and Welfare (AIHW), Commonwealth of Australia, Canberra. ISBN 0 642 82088 0.
- Pitkänen, J. (2001), 'Individual tree detection in digital aerial images by combining locally adaptive binarization and local maxima methods', *Canadian Journal of Forest Research* **31**(5), 832–844.

- Poulin, N., Susnik, B., Guillaud, M., Doudkine, A., Worth, A. & Palcic, B. (1995), 'Histometric texture analysis of DNA in thin sections from breast biopsies: Application to the detection of malignancy-associated changes in carcinoma in situ', *Analytical and Quantitative Cytology and Histology* 17(5), 291–299.
- Pratt, W. K. (1991), Digital Image Processing, second edn, John Wiley & Sons, New York.
- Preparata, F. P. & Shamos, M. I. (1985), Computational Geometry: An Introduction, Texts and Monographs in Computer Science, Springer-Verlag, New York.
- Press, S. J. & Wilson, S. (1978), 'Choosing between logistic regression and discriminant analysis', Journal of the American Statistical Association 73(364), 699–705.
- Preteux, F. & Merlet, N. (1991), New concepts in mathematical morphology: The topographical and differential distance functions, *in* 'Proceedings of the SPIE – The International Society for Optical Engineering', Vol. 1568, pp. 66–77.
- Ragnemalm, I. (1992), 'Neighborhoods for distance transformations using ordered propagation', *CVGIP: Image Understanding* **56**(3), 399–409.
- Reed, T. R. & du Buf, J. M. H. (1993), 'A review of recent texture segmentation and feature extraction techniques', CVGIP: Image Understanding 57(3), 359–372.
- Ripley, B. D. (1996), Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge.
- Rivest, J.-F. & Soille, P. (1995), 'Physical significance of image measurements', *IEEE Transactions on Instrumentation and Measurement* 44(3), 751–754.
- Rivest, J.-F., Serra, J. & Soille, P. (1992), 'Dimensionality in image analysis', *Journal* of Visual Communication and Image Representation **3**(2), 137–146.
- Rivest, J.-F., Soille, P. & Beucher, S. (1993), 'Morphological gradients', Journal of Electronic Imaging 2(4), 326–336.
- Rodenacker, K. (1992), Applications of topology for evaluating pictorial structures, in R. Klette & W. G. Kropatsch, eds, 'Theoretical Foundations of Computer Vision: Proceedings of the Vth Workshop 1992, Buckow (Markische Schweiz), March 30-April 3, 1992', Vol. 5, Akademie Verlag, Berlin, pp. 35–46.
- Rodenacker, K., Bischoff, P. & Chaudhuri, B. B. (1987), 'Featuring of topological characteristics in digital images', Acta Stereol 6(3), 945–950.

- Rodenacker, K., Gais, P., Jütting, U. & Burger, G. (1983), Mathematical morphology in grey-images, in H. W. Schüssler, ed., 'Signal Processing II: Theories and Applications', Elsevier Science Publishers, Amsterdam, The Netherlands, pp. 131– 134.
- Roerdink, J. B. T. M. & Meijster, A. (2001), 'The watershed transform: Definitions, algorithms and parallelization strategies', *Fundamenta Informaticae* **41**, 187–228.
- Romsdahl, M. M., McGrew, E. A., McGrath, R. G. & Valaitis, J. (1964), 'Hematopoietic nucleated cells in the peripheral venous blood of patients with carcinoma', *Cancer* 17, 1400–1404.
- Ronse, C. & Heijmans, H. J. A. M. (1998), 'A lattice-theoretical framework for annular filters in morphological image processing', *Applicable Algebra in Engineering*, *Communication and Computing* 9(1), 45–89.
- Rosenfeld, A. (2000), 'Classifying the literature related to computer vision and image analysis', *Computer Vision and Image Understanding* **79**(2), 308–323.
- Rosenfeld, A. & Kak, A. C. (1981), Digital Picture Processing. Volume 2, Computer Science and Applied Mathematics, second edn, Academic Press, New York.
- Rosenfeld, A. & Pfaltz, J. L. (1966), 'Sequential operations in digital picture processing', Journal of the Association for Computing Machinery 13(4), 471–494.
- Rosenfeld, A. & Pfaltz, J. L. (1968), 'Distance functions on digital pictures', Pattern Recognition 1, 33–61.
- Rousselle, C., Paillasson, S., Robert-Nicoud, M. & Ronot, X. (1999), 'Chromatin texture analysis in living cells', *The Histochemical Journal* **31**, 63–70.
- Rowiński, J. M., Pieńkowski, M. & Abramczuk, J. (1972), 'Area representation of optical density of chromatin in resting and stimulated lymphocytes as measured by means of Quantimet', *Histochemie* **32**, 75–80.
- Russ, J. C. (1999), *The Image Processing Handbook*, third edn, CRC Press, Boca Raton, Florida.
- Saito, T. & Toriwaki, J.-I. (1994), 'New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with applications', Pattern Recognition 27(11), 1551–1565.

- Sandritter, W., Kiefer, G., Kiefer, R., Salm, R., Moore, G. W. & Grimm, H. (1974), 'DNA in heterochromatin cytophotometric pattern recognition image analysis among cell nuclei in duct epithelium and in carcinoma of the human breast', *Beitrage zur Pathologie* 151, 87–96.
- Schulte, E. K. W. & Wittekind, D. H. (1994), Standardization of the staining process as a prerequisite for successful quantitative microscopy and automated cytology screening, *in* H. K. Grohs & O. A. N. Husain, eds, 'Automated Cervical Cancer Screening', IGAKU-SHOIN Medical Publishers, New York, chapter 14, pp. 200– 216.
- Serra, J. (1982), Image Analysis and Mathematical Morphology, Academic Press, London.
- Serra, J. (1986), 'Introduction to mathematical morphology', Computer Vision, Graphics, and Image Processing 35, 283–305.
- Serra, J. (1988a), The centre and self-dual filtering, in J. Serra, ed., 'Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances', Academic Press, London, pp. 159–180.
- Serra, J. (1988b), Introduction to morphological filters, in J. Serra, ed., 'Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances', Academic Press, London, pp. 101–114.
- Serra, J. (1988c), Measurements on numerical functions, in J. Serra, ed., 'Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances', Academic Press, London, pp. 297–315.
- Serra, J. (1989), Toggle mappings, in J. C. Simon, ed., 'From Pixels to Features', Elsevier Science, North-Holland, pp. 61–72.
- Serra, J. & Laÿ, B. (1985), 'Square to hexagonal lattices conversion', Signal Processing 9, 1–13.
- Serra, J. & Vincent, L. (1992), 'An overview of morphological filtering', Circuits, Systems and Signal Processing 11(1), 47–108.
- Serra, J., ed. (1988d), Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances, Academic Press, London.

- Sethian, J. A. (1999), Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Vol. 3 of Cambridge Monographs on Applied and Computational Mathematics, second edn, Cambridge University Press, Cambridge, U.K.
- Shih, F. Y.-C. & Mitchell, O. R. (1991), 'Decomposition of gray-scale morphological structuring elements', *Pattern Recognition* 24(3), 195–203.
- Shih, F. Y.-C. & Mitchell, O. R. (1992), 'A mathematical morphology approach to Euclidean distance transformation', *IEEE Transactions on Image Processing* 1(2), 197–204.
- Smeulders, A. W. M., Cornelisse, C. J., Vossepoel, A. M. & Ploem, J. S. (1978), 'An image segmentation approach to the analysis of nuclear texture of cervical cells', *Acta Histochemica. Supplementband* 20, 217–222.
- Smeulders, A. W. M., Leyte-Veldstra, L., Ploem, J. S. & Cornelisse, C. J. (1979), 'Texture analysis of cervical cell nuclei by segmentation of chromatin patterns', *The Journal of Histochemistry and Cytochemistry* 27(1), 199–203.
- Smith, G. M. (1998), Image Texture Analysis Using Zero Crossings Information, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Soille, P. (2003), Morphological Image Analysis: Principles and Applications, second edn, Springer-Verlag, Berlin.
- Soille, P. & Rivest, J.-F. (1996), 'On the validity of fractal dimension measurements in image analysis', Journal of Visual Communication and Image Representation 7(3), 217–229.
- Soille, P., Serra, J. & Rivest, J.-F. (1992), 'Dimensional measurements and operators in mathematical morphology', *Proceedings of the SPIE – The International Society* for Optical Engineering 1658, 127–138.
- Solomon, D., Davey, D., Kurman, R., Moriarty, A., O'Connor, D., Prey, M., Raab, S., Sherman, M., Wilbur, D., Wright, Jr., T. & Young, N. (2002), 'The 2001 Bethesda System: Terminology for reporting results of cervical cytology', *Journal* of the American Medical Association 287(16), 2114–2119.
- Sprenger, E., Moore, G. W., Naujoks, H., Schlüter, G. & Sandritter, W. (1973), 'DNA content and chromatin pattern analysis on cervical carcinoma in situ', Acta Cytologica 17(1), 27–31.

- Staunton, R. C. (2001), The processing of hexagonally sampled images, in P. W. Hawkes, B. Kazan & T. Mulvey, eds, 'Aspects of Image Processing and Compression', Vol. 119 of Advances in Imaging and Electron Physics, Academic Press, San Diego, pp. 191–265.
- Sternberg, S. R. (1979), Parallel architecture for image processing, in 'Proceedings of the IEEE Computer Society's Third International Computer Software and Applications Conference. COMPSAC 79.', Chicago, pp. 712–717.
- Sternberg, S. R. (1986), 'Grayscale morphology', Computer Vision, Graphics, and Image Processing 35, 333–355.
- Szász, G. (1963), *Introduction to Lattice Theory*, third edn, Academic Press, New York.
- Talbot, H. & Appleton, B. C. (2002), Elliptical distance transforms and the object splitting problem, in H. Talbot & R. Beare, eds, 'Proceedings of the 6th International Symposium on Mathematical Morphology', CSIRO Publishing, Australia, pp. 229–240. ISBN: 0 643 06804 X.
- Tanaka, N., Ikeda, H., Ueno, T., Mukawa, A., Watanabe, S., Okamoto, K., Hosoi, S. & Tsunekawa, S. (1987a), 'Automated cytologic screening system (CYBEST model 4): An integrated image cytometry system', *Applied Optics* 26(16), 3301– 3307.
- Tanaka, N., Ueno, T., Ikeda, H., Ishikawa, A., Yamauchi, K., Okamoto, Y. & Hosoi, S. (1987b), 'CYBEST model 4 automated cytologic screening system for uterine cancer utilizing image analysis processing', *Analytical and Quantitative Cytology* and Histology 9(5), 449–454.
- Tenenbaum, A. M. & Augenstein, M. J. (1986), Data Structures Using Pascal, second edn, Prentice-Hall International, Englewood Cliffs, NJ.
- Tomori, Z., Marcin, J. & Vilim, P. (1999), 'Pyramidal seeded region growing algorithm and its use in image segmentation', *Lecture Notes in Computer Science* 1689, 395–402.
- Toriwaki, J. & Yokoi, S. (1988), Voronoi and related neighbors on digitized twodimensional space with applications to texture analysis, in G. T. Toussaint, ed., 'Computational Morphology: A Computational Geometric Approach to the Analysis of Form', Vol. 6 of Machine Intelligence and Pattern Recognition, Elsevier Science Publishers, pp. 207–228.

- Toussaint, G. T. (1980), 'The relative neighbourhood graph of a finite planar set', *Pattern Recognition* **12**, 261–268.
- Tüceryan, M. & Chorzempa, T. (1991), 'Relative sensitivity of a family of closestpoint graphs in computer vision applications', *Pattern Recognition* 24(5), 361–373.
- Tüceryan, M. & Jain, A. K. (1990), 'Texture segmentation using Voronoi polygons', IEEE Transactions on Pattern Analysis and Machine Intelligence 12(2), 211–216.
- Tüceryan, M. & Jain, A. K. (1999), Texture analysis, in C. H. Chen, L. F. Pau & P. S. P. Wang, eds, 'Handbook of Pattern Recognition and Computer Vision', second edn, World Scientific Publishing, Singapore, chapter 2.1, pp. 207–248.
- Vachier, C. & Vincent, L. (1995), 'Valuation of image extrema using alternating filters by reconstruction', Proceedings of the SPIE – The International Society for Optical Engineering 2568, 94–103.
- van der Heijden, F. (1994), Image Based Measurement Systems: Object Recognition and Parameter Estimation, John Wiley & Sons, Chichester, England.
- van der Ploeg, M., van den Broek, K., Smeulders, A. W. M., Vossepoel, A. M. & van Duijn, P. (1977), 'HIDACSYS: Computer programs for interactive scanning cytophotometry', *Histochemistry* 54(4), 273–288.
- Venables, W. N. & Ripley, B. D. (1999), Modern Applied Statistics With S-PLUS, third edn, Springer-Verlag, New York.
- Venables, W. N. & Ripley, B. D. (2001), 'R' complements to modern applied statistics with S-plus third edition', http://www.stats.ox.ac.uk/pub/MASS3/Compl. shtml. ISBN 0-387-98825-4.
- Vincent, L. (1989), 'Graphs and mathematical morphology', Signal Processing 16, 365–388.
- Vincent, L. (1991a), Exact Euclidean distance function by chain propagations, in 'Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition', pp. 520–525.
- Vincent, L. (1991b), New trends in morphological algorithms, in 'Proceedings of the SPIE – The International Society for Optical Engineering', Vol. 1451, pp. 158–170.

- Vincent, L. (1993), 'Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms', *IEEE Transactions on Image Processing* 2(2), 176–201.
- Vincent, L. & Soille, P. (1991), 'Watersheds in digital spaces: An efficient algorithm based on immersion simulations', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6), 583–598.
- Walker, R. F. (1997), Adaptive Multi-Scale Texture Analysis with Application to Automated Cytology, PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland, Australia.
- Walker, R. F. & Jackway, P. T. (1996), Statistical geometric features extensions for cytological texture analysis, *in* 'Proceedings of the 13th International Conference on Pattern Recognition', IEEE Computer Society Press, Los Alamitos, California, pp. 790–794.
- WHO (1997), 'Pap cytology screening: Most of the benefits reaped?', http://www. who.int/. World Health Organization Press Release WHO/25.
- Wied, G. L., Bartels, P. H., Bahr, G. F. & Oldfield, D. G. (1968), 'Taxonomic intra-cellular analytic system (TICAS) for cell identification', Acta Cytologica 12(3), 180–204.
- Wilson, R. J. (1985), *Introduction to Graph Theory*, third edn, Longman, Essex, England.
- Wolf, G., Beil, M. & Guski, H. (1995), 'Chromatin structure analysis based on a hierarchic texture model', Analytical and Quantitative Cytology and Histology 17, 25–34.
- Yamada, H. (1984), Complete Euclidean distance transformation by parallel operation, in 'Proceedings of the 7th International Conference on Pattern Recognition', IEEE Computer Society Press, Silver Spring, MD, USA, pp. 69–71.
- Young, I. T., Verbeek, P. W. & Mayall, B. H. (1986), 'Characterization of chromatin distribution in cell nuclei', *Cytometry* 7, 467–474.
- Zhang, Y. J. (1996), 'A survey on evaluation methods for image segmentation', Pattern Recognition 29(8), 1335–1346.

- Zhu, S. C. & Yuille, A. (1996), 'Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(9), 884–900.
- Zinser, G. & Komitowski, D. (1983), 'Segmentation of cell nuclei in tissue section analysis', *The Journal of Histochemistry and Cytochemistry* **31**(1), 94–100.



DImPAL 4.0



DImPAL¹—an abbreviation for **D**igital **Im**age **P**rocessing and **A**nalysis **L**anguage is an interpreted language, developed by the author², that can be used to process and analyse digital images and graphs. DImPAL can operate on images sampled on either square or hexagonal grids. Boolean, byte, integer and real data types are supported for image pixels and for the vertex weights of graphs. DImPAL can be used interactively to execute statements, one at a time, entered at the command prompt, or it can be used to execute a program (written as a plain text file). DImPAL supports four types of statement: command, assignment, while-endwhile, and ifthen-else. The latter two statements, however, can only be used in programs.

Variables are used to represent images (bound matrices), graphs (sets of vertices and edges), and matrices (scalars are single element matrices). Complex expressions can be constructed using variables, functions, constants, and arithmetic and logic operators. Such expressions can appear on the right-hand-side of an assignment

 $^{^1}$ Copyright 1991-2003, Andrew Mehnert. All rights reserved.

 $^{^2}$ Version 1.2 is documented in Mehnert (1994).

statement, in the actual parameters of a command or function, and as the conditional expression associated with the while-endwhile and the if-then-else statements. Assignment statements are used to create new variables or to overwrite old ones. DImPAL provides a suite of functions including the C language mathematics library functions, data type casting functions, grid conversion functions, image and graph mathematical morphology functions, image measurement functions, neighbourhood filters, radiometric enhancement functions, isometries, and file import functions. Commands are also provided for variable management, display, plotting histograms and surfaces, writing images to a variety of file formats, and executing programs.

At the time of writing, DImPAL (version 4.0) implements more than sixty functions and a dozen commands. Its novel features include:

- 1. The concept of a missing value. Missing values are used to denote *don't care* values of parameters in function and command calls, and to denote undefined real and integer values in matrices, images, and graphs. In the latter case such values can be set explicitly, e.g. by assignment, or occur as the result of undefined arithmetic operations such as division by zero.
- 2. Multi-layer variables. A variable is a collection of one or more *layers*. A layer can represent an image manifested on a square grid, an image manifested on a hexagonal grid, a matrix, a set of graph vertices, or a set of graph edges. A grey-scale graph, for example, is defined to be a three layer variable for which the first layer contains the vertices, the second contains the edges, and the third contains the vertex weights.
- 3. Multi-layer processing convention. When a monadic operator is applied to a variable it is actually applied to each layer in turn thus producing a result with the same number of layers. For example, if **image** is a variable with two layers then **-image** also has two layers.

In the case of dyadic operators (taking two arguments) only the following layer combinations are permissible:

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

In each case DImPAL produces an n layer result. In the first case DImPAL applies the operator n times, each time with a different layer of the first operand (the second operand—a single layer—is the same each time):

```
layer 1 <operator> layer 1, ..., layer n <operator> layer 1.
```

In the second case DImPAL applies the operator n times, each time with a different layer of the second operand (the first operand—a single layer—is the same each time):

layer 1 <operator> layer 1, ..., layer 1 <operator> layer n.

In the last case DImPAL applies the operator pairwise:

layer 1 <operator> layer 1, ..., layer n <operator> layer n.

- 4. Function overloading. What this means is that a single function name (identifier) can be used to represent several different functions each valid for a particular layer type and data type. For example, the function dilate() is used to compute the metric dilation of an image, of a graph, or of an image by another image (structuring element).
- 5. Default parameters. Many functions and commands can be called without specifying all of their parameters. For example the rotate() function requires four parameters: a variable, an angle, the x coordinate of the centre of rotation, and the y coordinate of the centre of rotation. However, the latter three parameters have defaults and so do not need to be specified.

The material presented in the remainder of this appendix is taken from the *DImPAL* version 4.0 user manual.

A.1 Technical details / history

DImPAL comprises several distinct software components. These are listed in Table A.1.

Component	Description
console handler	The primary interface to DImPAL
	which accepts statements typed in by
	the user.
lexical analyser	Extracts tokens such as variable and
	function names, operators, and con-
	stants from user input.
parser	Uses recursive descent parsing to pro-
	cess command, assignment, decision,
	and looping statements.
file manager	Handles all I/O between secondary
	storage and memory.
symbol table manager	Maintains a list of variable names and
	descriptions.
command dictionary	List of all commands and associated
	syntax.
function dictionary	List of all functions and associated syn-
	tax.

 Table A.1: DImPAL's software components.

Prototypes of DImPAL's core software components—lexical analyser, parser, and file manager—were developed in ANSI C³ under AmigaDOS⁴, and then MS-DOS⁵ in 1991 and 1992. During 1993 the software was ported to and then developed under OS/2 version 1.3^6 . This constituted version 1.0 of DImPAL. In 1994 DImPAL was ported to OS/2 version 2.0^7 and a large number of functions and commands were added to it. This version, version 1.2, is documented in Mehnert (1994).

In 1995 and 1996 DImPAL was ported to UNIX⁸—AIX⁹, ULTRIX¹⁰ and Digital UNIX¹¹—and much of the code was revised, redesigned, and rewritten. Most no-tably:

1. internal support for the loading of images one or several lines at a time was dropped (the virtual memory management provided by modern operating sys-

³ American National Standards Institute defined standard of the C programming language.

 $^{^{\}rm 4}$ Commodore International Limited.

⁵ Microsoft Corporation.

⁶ Microsoft Corporation and International Business Machines Corporation.

 $^{^{7}}$ International Business Machines Corporation.

 $^{^8}$ Unix is a registered trademark of AT&T.

 $^{^{9}}$ UNIX variant developed by International Business Machines Corporation.

 $^{^{10}}$ UNIX variant developed by Digital Equipment Corporation.

¹¹ UNIX variant developed by Digital Equipment Corporation.

tems and compilers made this scheme for conserving memory redundant);

- 2. the number of data types was reduced from six to four;
- 3. support for hexagonal grid images was added;
- 4. layer types were introduced: matrix, points sampled on a square grid, points sampled on a hexagonal grid, graph edges, and graph vertices;
- 5. scalar variables were introduced; and
- 6. the OS/2 specific graphics and window management routines were discarded in favour of calls to third party software such as xv^{12} and $Gnuplot^{13}$ (making DImPAL highly portable).

This constituted version 2.0.

Version 3.0 was developed during 1997 and 1998. Changes and extensions of note include:

- 1. internal support for two-dimensional array style addressing of images;
- 2. internal support for padding an image with a border of pixels of specified width and value as it is loaded into memory (simplifies internal code when dealing with edge effects);
- 3. a rewrite of the left-recursive production rules so that the parser uses leftto-right associativity for all dyadic operators (rather than right-to-left as in preceding versions);
- 4. the retiring of the . operator of version 2.0 in favour of two new operators: . operator for specifying layer references with an identifier and the **#** operator for specifying layer references with an integer constant or expression;
- 5. the introduction of the ~ operator (for joining several layers to produce a single variable);
- 6. the introduction of * as a missing value symbol that can be used as a constant in expressions; and

¹² Freely available via ftp://ftp.cis.upenn.edu/pub/xv.

¹³ Freely available from http://www.gnuplot.info.

7. the introduction of the while-endwhile and if-then-else statements and the necessary functions to support program execution (previous versions of DImPAL provided only batch execution with no flow control statements).

Several commands and functions—most notably seeded region growing, watershed transform, dynamics, reconstruction filtering, and graph morphology—were added to this version yielding several sub-versions (3.x series). Version 4.0 was developed in 2002 and includes several bug fixes, speed improvements, and minor customisations for processing cytometer files. Changes of note include:

- 1. bug fixes to the internal code supporting nested looping and conditional statements;
- 2. implementation of fast watershed and contrast dynamics routines;
- 3. implementation of *character string* assignment, and the passing of string variables as parameters to functions and commands; and
- 4. extension of the *declare()* command to import *.img* files from the Cytometrics Project cytometer.

DImPAL (versions 3.x and greater) is known to compile, using gcc^{14} , and run under Linux¹⁵, Tru64 UNIX 5.0¹⁶, and SunOS 5.8¹⁷. It also compiles, using Microsoft Visual C++ Compiler, and runs under Windows 98¹⁸.

Although DImPAL is written in ANSI C it does use the following non-ANSI UNIX functions: tempnam(), remove(), and fork(). The tempnam() function is used in both the file manager and graphics routines to generate temporary file names. The remove() function is used in the parser, file manager, and graphics routines to delete files. The fork() function is used in the graphics routines to execute multiple instances of xv and Gnuplot.

¹⁴ GCC is the GNU Compiler Collection which contains front ends and libraries for several different languages including: C, C++, Objective-C, Fortran, Java, and Ada. The GCC homepage is located at http://gcc.gnu.org/.

¹⁵ Red Hat Linux Version 8.0, Red Hat, Inc.

¹⁶ Hewlett-Packard Company.

 $^{^{17}}$ Sun Microsystems, Inc.

¹⁸ Microsoft Corporation.

A.2 Lexical conventions

A DImPAL statement is fundamentally a collection of small syntactic units called *tokens*. There are four types of token:

- 1. identifiers,
- 2. constants,
- 3. operators, and
- 4. separators.

The lexical analyser resolves a statement into its constituent tokens. It ignores any white space, except when it appears between double quotes, and any characters following a semicolon (deemed to be a comment). A particular instance of a token is called a *lexeme*. The stream of lexemes resolved from a statement are passed on to the parser whose job it is to determine the syntactic validity of the statement. If the statement is syntactically valid then it can be executed, otherwise an error message is produced. In the remainder of this section each type of token is formally described.

A.2.1 Identifiers

An identifier is defined to be a letter followed by a mixed sequence of letters, numbers, and underscores. Identifiers are case sensitive, so that *Image* is different from *image*. Identifiers are used to denote variables, keywords, layer names, commands, and functions. There are no restrictions on the length of an identifier except in the case of a variable. A variable is actually stored in a file that has the same name as the variable plus the extension *.lay*. This means that the length of an identifier used for a variable cannot exceed four less than the maximum length of a file name for the underlying file system¹⁹. It is permissible to give a variable an identifier that is the same as that already used for a layer, command, or function. It is also permissible to use the *while, endwhile, if, else,* and *endif* keywords as identifiers for variables (although this is not good programming practice).

¹⁹ The ext2/ext3 filesystems used in Linux, for example, impose a limit of 255 characters on the length of a filename.

A.2.2 Constants

A constant is a number or a character string that can be used as a value or a parameter in a statement. 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 can be 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>
```

A string constant is any sequence of alphanumeric characters surrounded by double quotes (but not containing the double quote character itself).

A.2.3 Missing value constant

The * symbol can be used in DImPAL in the place of any real or integer constant. The * symbol denotes the missing value constant. This constant is used in DImPAL for several purposes including: encoding non-rectangular images, denoting *don't care* parameters in function and command calls, and to signify the result yielded by a mathematically undefined operation.

A.2.4 Operators

All of the DImPAL operators, except ^, ., and # are a subset of the C language operators. Table A.2 lists the DImPAL operators in decreasing order of precedence; e.g. when evaluating an expression DImPAL will perform multiplication before addition. Parentheses can be used to override precedence.

A.2.5 Meta-operator

A pair of square brackets placed around a dyadic operator invokes the clipping metaoperator; e.g. result = image1[+] image2. Ordinarily a dyadic operation between two layers, both representing square grid images, or both representing hexagonal grid images, is only valid if both layers have the same size and location in \mathbb{Z}^2 . If this is
Operator	Associativity	Description
• #	left to right	layer reference operators
~	left to right	group
- !	right to left	unary minus, logical negation
^	left to right	exponentiation
* / %	left to right	multiplication, division, modulus
+ -	left to right	addition, subtraction
== !=	left to right	equal to, not equal to
< > <= >=	left to right	relational operators
&&	left to right	logical AND
	left to right	logical OR

Table A.2: DImPAL operators listed in decreasing order of precedence. Where several operators appear on the same line they have equal precedence.

not the case then the clipping meta-operator can be used. The operator forces the operation to be performed on the intersection of the domains of the operands.

A.2.6 Separators

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

A.3 Language grammar

DImPAL statements are parsed using a method known as *recursive descent parsing*. A collection of recursive procedures, which in total constitute the parser, determines the syntactic validity of a statement based on the sequence of lexemes generated by the lexical analyser. If the statement is also semantically correct then it is evaluated as the recursion unwinds. DImPAL statements can be described by a Class 2, also called context-free, grammar. Class 2 grammars can be expressed in EBNF notation. The production rules for a statement entered at the command prompt (an interactive statement) and for a program are as follows:

```
<program>::=<statement_list></products/
<statement_list>::=<program_statement>
<program_statement>::=<if_statement>
                                                <while_statement>
                                                1
                      <interactive_statement>
<if_statement>::=if <disjunction>
                 <statement_list>
                 {else
                  <statement_list>}
                 endif
<while_statement>::=while <disjunction>
                    <statement_list>
                    endwhile
<interactive_statement>::=<assignment_statement> |
                           <command_statement>
<assignment_statement>::=<identifier> = <disjunction> |
                         <identifier> = <string>
<command_statement>::=<identifier>(<parameter_list>)
<disjunction>::=<disjunction> || <conjunction> |
                <conjunction>
<conjunction>::=<conjunction> && <relational> |
                <relational>
<relational>::=<relational> < <equality> |
               <relational> <= <equality> |
```

```
<relational> >= <equality> |
               <relational> > <equality> |
               <equality>
<equality>::=<equality> == <additive> |
             <equality> != <additive> |
             <additive>
<additive>::=<additive> + <multiplicative> |
             <additive> - <multiplicative> |
             <multiplicative>
<multiplicative>::=<multiplicative> * <exponentiation> |
                   <multiplicative> / <exponentiation> |
                   <multiplicative> % <exponentiation> |
                   <exponentiation>
<exponentiation>::=<exponentiation> ^ <unary> |
                   <unary>
<unary>::= !<unary> |
           -<unary> |
           <group>
<group>::=<group> ~ <postfix> |
          <postfix>
<postfix>::=<postfix> . <identifier> |
            <postfix> # <subfactor> |
            <subfactor>
<subfactor>::=<constant>
                              1
              <variable>
                              <function>
                              (<disjunction>)
<constant>::=<integer> |
```

Data type	Range of values	Comment
boolean	$\{0,1\}$	
byte	$\{0, 1, \cdots, 255\}$	
integer	C language int	INT_MIN is used to represent a missing value
		*
real	C language double	-DBL_MAX is used to represent a missing
		value *

Table A.3: DImPAL data types.

```
<real> |
<string> |
```

```
<variable>::=<identifier>
```

```
<function>::=<identifier>(<parameter_list>)
```

```
<parameter_list>::=<parameter_list> , <parameter> |
        <parameter>
```

```
<parameter>::=<string> |
        <disjunction>
```

A.4 Variables and layers

The fundamental data objects in DImPAL are variables. Variables are used to store images, matrices, and graphs. A variable is a collection of one or more *layers*. A layer is simply a two dimensional array of elements. An individual layer can store either a matrix, bound matrix (image), set of graph vertices, or set of graph edges. Every layer has an identifier, a size (rows \times columns), a *data type*, and a *layer type*.

A.4.1 Data types

There are only four data types in DImPAL. These are listed in TableA.3.

A.4.2 Layer types

Each layer of a variable has an associated layer type. There are five possible layer types:

- 1. MATRIX,
- 2. SQUARE,
- 3. HEXAGONAL,
- 4. VERTICES, and
- 5. EDGES.

A layer of type MATRIX can store a conventional matrix. Elements of the matrix can store values in the range dictated by the data type of the layer.

A layer of type SQUARE is used to store a digital image manifested on a square grid. More specifically the layer type SQUARE defines a *bound matrix*. A bound matrix is simply a conventional matrix (of pixel values) that has location in \mathbb{Z}^2 . In other words, each element of the matrix can be indexed not only by row and column, but also by a coordinate pair. For example, the *disk* function produces a single layer variable with the origin located at the centre of the matrix. DImPAL uses a rectangular coordinate system in which x values increase from left to right and y values increase from bottom to top. Elements (pixels) of a layer of type SQUARE can store values in the range dictated by the data type of the layer.

A layer of type HEXAGONAL is used to store a digital image manifested on a hexagonal grid. A layer of type HEXAGONAL is similar to a layer of type SQUARE in that it can store a bound matrix. However, the way in which DImPAL interprets a HEXAGONAL bound matrix is very different to the way in which it interprets a SQUARE bound matrix. For example:

1	5	3		1	5	3			1	5		3	
7	3	8	>	7	3	8	or	*	7		3		8
1	4	6		1	4	6			1	4		6	
bo	und	matrix		squ	are	gri	id	hexa	ago	nal	L g	ŗi	d
(o:	rig	in at		int	erp	reta	ation	int	erp	ret	tat	io	'n
t	op-	left)											

DIMPAL adopts the convention that the row spacing of pixels in a SQUARE layer and a HEXAGONAL layer is 1 unit. This implies that the horizontal spacing between points on the square grid is 1 unit whilst that for the hexagonal grid is $2/\sqrt{3}$.

A layer of type VERTICES can store an array of vertices for a graph. A layer of type EDGES can store an array of vertex neighbours (edges) for a graph. Each layer type can have only one row. The number of columns for type VERTICES corresponds to the number of vertices. The number of columns for type EDGES corresponds to twice the number of edges. Internally DImPAL uses the data type *integer* for elements of either layer type.

A.4.3 Creating and managing variables

In a DImPAL session or program a variable can be introduced in one of the following ways:

- 1. a variable defined in a previous session can be declared using the *declare* command;
- 2. the *read* function can be used to create a variable from an image file (such as a GIF file or PGM file) or from a description provided in a .*dpl* file (described in the next section);
- 3. a new variable can be created using an assignment statement; e.g. gradient=dilate(image,disk(1))[-]erode(image,disk(1)).

The *list* command prints a list of all of the variables known to DImPAL. A variable is physically stored as a file that has the same name as the variable and the extension *.lay.* For example the variable *fred* would be stored in the file *fred.lay.* The symbol table manager maintains a binary tree of descriptors for each variable. The *describe* command is used to print a description of each layer of a variable. Actual data is only ever loaded into memory when a command or function is executed. A variable can be removed from the symbol table using the *discard* command or the *destroy* command. In the first case the *.lay* file is not deleted which means that it can be reintroduced at a later date using the *declare* command. In the second case the *.lay* file is permanently deleted.

A.4.4 Creating .dpl Files

The *read* function is used to create DImPAL variables from standard image files such as GIF or PGM. In addition, the function can read in a new variable that has been defined in a DImPAL specific .dpl file (i.e. the file name must have the extension .dpl). A .dpl file is a text file containing a description of the variable. Here is an example of a .dpl file:

LAYERS: 7 **IDENTIFIER:** vertices FORMAT: VERTICES SIZE: 1 x 27 IDENTIFIER: edges FORMAT: EDGES SIZE: 1 x 68 IDENTIFIER: x_coordinates FORMAT: MATRIX TYPE: INTEGER SIZE: 1 x 27 IDENTIFIER: y_coordinates FORMAT: MATRIX TYPE: INTEGER SIZE: 1 x 27 IDENTIFIER: attributes FORMAT: MATRIX TYPE: BOOLEAN SIZE: 1 x 27 IDENTIFIER: structuring_element_on_square_grid FORMAT: SQUARE TYPE: INTEGER LOCATION: (1,-1) SIZE: 3 x 3

```
IDENTIFIER: structuring_element_on_hexagonal_grid
FORMAT: HEXAGONAL
TYPE: BOOLEAN
LOCATION: (1,-1)
SIZE: 3 x 3
0 2 4 8 12 13 18 22 24 25 29 34 37 40 41 42 44 47
51 53 56 57 58 60 63 65 67
23
25
0 1 3 5
0 2 4 6
3
1 2 6 9 10
3 5 7 10
6 11
9
5 8 16 17
5 6 15 17 18
7 12 15
11 13 14
12
12
10 11
9 17 19
9 10 16 23
10 20
16 21 22
18
19
19 26
17 24 25
23 25
23 24
22
```

A .dpl file is used to define a variable with an arbitrary number of layers. The first line of the file specifies the number of layers the variable has (using the LAYERS keyword). Next, each layer is described individually. For every layer, an IDEN-TIFIER: and a FORMAT: (one of VERTICES, EDGES, MATRIX, SQUARE, and HEXAGONAL) must be specified. For a layer of type VERTICES or EDGES, a SIZE: (1 row by *n* columns) must be specified. For a MATRIX layer, a TYPE: (one of BOOLEAN, BYTE, INTEGER, or REAL) and SIZE: (rows by columns) must be specified. For SQUARE or HEXAGONAL layers a TYPE:, LOCATION: (integer coordinates of the top-left pixel), and SIZE: must be specified. After the descriptions, the data are given for each layer. The * symbol is used to denote missing values for layers with data type *integer* or *real*.

To define a graph it is necessary to define two layers: one of type VERTICES and one of type EDGES. The layer of type VERTICES must have one row and as many columns as there are vertices. Each element of the array is an offset into a second array defined in the layer of type EDGES. The first value in the VERTICES layer should be zero (pointing to the first element of EDGES). Each edge that radiates from the first vertex is encoded by appending its opposite vertex (vertices are labelled from 0 upward) to the end of the EDGES array. The second value in the VERTICES array must then point to one place after the last vertex value appended to the EDGES array. Then each edge that radiates from the second vertex is encoded, and so on.

A.4.5 Scalar variables

A scalar variable is a variable that has a single layer of type MATRIX and of size 1×1 . Such a variable is created when a constant or an expression involving only constants appears on the right hand side of an assignment statement; e.g. number=3.

A.4.6 String variables

A string variable is a variable that has a single layer of type MATRIX, of byte data type, and of size $1 \times (n+1)$ where n is the length of the string. Such a variable is created by explicit assignment; e.g. filename="file.txt".

A.5 Sample program

```
; SYNOPSIS: VISUAL COMPARISON OF SEVERAL DISCRETE METRICS
; CREATE A 201x201 BOOLEAN IMAGE WITH ALL BUT THE CENTRE PIXEL
; SET TO BINARY '1'
image=cone(100)!=100
result=distance_transform(image, "Euclidean",1)
result=result~distance_transform(image,"cityblock",1)
result=result~(distance_transform(image,"chessboard",1))
result=result~(distance_transform(image,"chamfer34",1)/3.0)
result=result~(distance_transform(image,"chamfer5711",1)/5.0)
; RESAMPLE image TO OBTAIN A REPRESENTATION ON THE HEXAGONAL GRID
image=hexagonal_grid(translate(image,0,0))
; GIVEN THAT THE HORIZONTAL SPACING ON THE HEXAGONAL GRID IS 2/sqrt(3),
; IT IS NECESSARY TO MULTIPLY THE "honeycomb" and "chamfer35" DTs BY
; THIS FACTOR TO YIELD AN APPROXIMATION TO THE "Euclidean" DT.
; THE RESULTS OF THE INDIVIDUAL DTS ARE CONVERTED BACK TO THE SQUARE
; GRID FOR DISPLAY PURPOSES.
result=result~(distance_transform(image,"Euclidean",1))
result=result~(distance_transform(image, "honeycomb",1)*(2.0/sqrt(3)))
result=result~(distance_transform(image,"chamfer35",1)*(2.0/sqrt(3))/3.0)
; DISPLAY QUANTISED VERSIONS OF EACH DT USING THE SAME GREY-SCALE RANGE.
; THE RADII OF THE ANNULI CORRESPOND TO EQUAL DISTANCES
declare("variables/grey.lay")
log_step_grey=saturate(log(grey+1))
layer = 1
while (laver <= 5)
  display(byte(byte(floor(result#layer+0.5)/16)*20),grey,(layer-1)*220,0)
  layer = layer + 1
endwhile
while (layer <= 8)
  display(square_grid(byte(floor(result#layer+0.5)/16)*20)),grey,(layer-
6) * 220.260)
  layer = layer + 1
endwhile
```

Appendix B

The Watershed Transform

Image segmentation by *watersheds* was first proposed by Digabel & Lantuejoul (1978) as a tool for the analysis of serial cross-sections of a petrographic sample (cited in Soille, 2003, p. 2). They imagined that each cross-section—a two-dimensional binary image—is in fact a horizontal cross-section of a topographic relief. If a drop of rain is imagined to fall on this relief then it would, by the law of gravity, follow the steepest path of descent until it reached a minimum. Borrowing from geographical terminology, the set of all possible paths that a raindrop could follow to reach a given minimum defines a *catchment basin*. The boundaries between adjacent catchment basins are called *watersheds*. Digabel & Lantuejoul devised an algorithm to automatically determine these watershed boundaries. Beucher & Lantuejoul (1979) extended the idea to two-dimensional grey-scale images. In this case, a topographic surface is realised if each grey-value of the image is taken to represent elevation rather than intensity.

Unfortunately the *path of steepest descent* definition of catchment basins and watersheds, although intuitive, does not facilitate algorithm design for the computation of watersheds in digital spaces. The reason for this is that there are several situations where the direction of flow at a given pixel cannot be determined; e.g. plateau pixels (Soille, 2003, p. 269). The problem is solved if instead the topographic surface is imagined to be flooded by water coming out of the ground at various sources rather than from falling rain. Meyer & Beucher (1990, p. 23) describe this flooding scheme as follows:

we bore a hole in each minimum of the relief and immerse the surface in a lake with a uniform vertical speed. The water entering through the holes fills up the various catchment basins. We suppose that the immersion speed is slow enough to ensure a constant level in all the basins. In order to avoid the confluence of the floods coming from the different minima, we build a dam along the lines where the floods would merge. After complete immersion only the dams [watersheds] emerge and separate the various catchment basins.

All of the algorithms for computing the watersheds (watershed transform) published before 1991 are notoriously slow or inaccurate (Vincent & Soille, 1991, p. 583). They involve many iterations, each of which involves a pass through all of the image pixels, until convergence. In 1991 Vincent & Soille proposed a fast algorithm for computing watersheds. The algorithm eliminates the need to successively scan all of the image pixels. "It is based on a sorting of the pixels in the increasing order of their gray values, and on fast breadth-first scannings of the plateaus enabled by a first-in-first-out [queue] type data structure" (Vincent & Soille, 1991, p. 583). Subsequently, Meyer (1991) proposed a more general algorithm based on a priority queue of queues. The algorithm permits flooding to be initiated from selected markers, not necessarily the minima. Meyer proposes two variants of the algorithm: one that renders watershed lines and labelled basins, and one that renders only the labelled catchment basins. A description of the algorithms, in English, can be found in Beucher & Meyer (1993). Pseudocode for both implementations can be found in Dobrin et al. (1994). Pseudocode—in the style presented in Section 4.6—for Meyer's watershed line algorithm is as follows:

- 1: add each pixel neighbouring a marker region to the PQ with priority equal to its associated grey-value, and label as IN_QUEUE
- 2: while the PQ is not empty do
- 3: remove pixel from the PQ
- 4: examine all of its neighbours that have a region label
- 5: if they all have the same label then
- 6: assign this label to the pixel
- 7: add each unlabelled neighbour to the PQ with priority equal to its associated grey-value, and label as IN_QUEUE

- 9: label the pixel as WATERSHED
- 10: end if
- 11: end while.

^{8:} else

The reader is referred to Roerdink & Meijster (2001) for a recent review of watershed transform algorithms.

Appendix C

FISF Implementation for MICROMORPH

fold.mi	с		}
			}
AUTHOR	: And	drew Mehnert	}
DATE:	17/6,	/99	}
			}
NOTES:	The	following procedures are defined:	}
			}
	(1)	fold greyinout1 greyinout2	}
		- greyinout1 initially contains the image to be folded,	}
		and after execution contains the folded grey-scale	}
		values (folded about c=127.5).	}
		- greyinout2 can initially contain anything,	}
		but after execution it contains the template	}
		(indicator) values.	}
			}
	(2)	unfold greyinout1 greyinout2	}
		- The pair (greyinout1,greyinout2) should initially be a	}
		fold-space image (as produced by "fold", "foldmax", and	}
		"foldmin").	}
		- After execution, greyinout1 contains the unfolded image $% \left[{{\left[{{\left[{{\left[{\left[{\left[{\left[{\left[{\left[{$	}
		and greyinout2 is empty.	}
-		- NOTE: because 'c=127.5' is not a representable grey-	}
		value, the value 127 is used instead.	}
-			}
	(3)	foldmax greyin1 greyin2 greyout1 greyout2 size	}
		- The pair (greyin1,greyin2) should be a fold-space	}
		image (as produced by "fold", "foldmax", and	}
		"foldmin").	}
		- After execution, the pair (greyout1,greyout2)	}
		will contain the fold-space dilation of	}
		(greyin1,greyin2).	}
		- size specifies the neighbourhood size of the dilation.	}
			}
	(4)	foldmin greyin1 greyin2 greyout1 greyout2 size	}
		- The pair (greyin1,greyin2) should be a fold-space	}
		<pre>image (as produced by "fold", "foldmax", and</pre>	}
		"foldmin").	}
		- After execution, the pair (greyout1,greyout2)	}
		will contain the fold-space erosion of	}

```
{
               (greyin1, greyin2).
                                                                       }
{
             - size specifies the neighbourhood size of the erosion
                                                                       }
Ł
                                                                       }
ł
         (5) imextsupngb and imextinfngb are ancillary functions.
                                                                       }
deproc fold fold s t
syntax "fold greyinout greyinout"
  int g b;
  g := imalloc 8
  b := imalloc 1
  imthresh s 0 127 b {identify all pixels below the crease}
  immask b 0 255 t {t is 255 wherever b is a 1}
  imand t s g
                     {g contains only those pixels of s less than the crease, 0 else-
where}
  iminv s s
                     {now invert the original image}
  iminv t t
                     {invert the mask}
  imand t s s
                     {s now contains the folded pixels}
                     {s contains the folded greyscale image, t the template}
  imsup g s s
  imfree g
  imfree b
end
deproc unfold unfold s t
syntax "unfold greyinout greyin"
  int g1 g2 b;
  g1 := imalloc 8
  g2 := imalloc 8
  b := imalloc 1
  imthresh t 0 0 b
                       {b is 1 wherever t=0}
  immask b 0 255 g1
                       {g1 is 255 wherever t=0, 0 otherwise}
  imand s g1 g1
                       {g1 contains all values that do not need folding, 0 otherwise}
  imthresh t 255 255 b {b is 1 wherever t=255}
  immask b 0 255 g2
                     {g2 is 255 wherever t=255, 0 otherwise}
  iminv s s
                       {g2 contains all the unfolded values, 0 otherwise}
  imand s g2 g2
  imthresh t 127 127 b {b is 1 wherever t=127}
  immask b 0 127 s
                       {s is 127 wherever there is a value on the crease}
  imor g1 g2 g1
  imor g1 s s
  imfree g1
  imfree g2
  imfree b
  clr t
end
deproc imextsupngb imextsupngb s st d dt di
  syntax "imextsupngb greyin1 greyin2 greyout1 greyout2 direction"
  int w1 w2 w3 w4 w5 b;
  w1 := imalloc 8
  w2 := imalloc 8
  w3 := imalloc 8
  w4 := imalloc 8
  w5 := imalloc 8
  b := imalloc 1
  imcopyngb s w1 di 1 0
  imcopyngb st w2 di 1 0
  imdiff s w1 w3
                     {w3 contains all pixels s>w1, 0 otherwise}
  imthresh w3 1 255 b {b is 1 wherever s>w1}
```

```
immask b 0 255 w3 {w3 is 255 wherever s>w1, 0 otherwise}
  imdiff w1 s w4
                     {w4 contains all pixels w1>s, 0 otherwise}
  imthresh w4 1 255 b {b is 1 wherever w1>s}
  immask b 0 255 w4 {w4 is 255 wherever w1>s, 0 otherwise}
                     {w5 is 255 wherever s!=w1, 0 otherwise}
  imor w3 w4 w5
  iminv w5 w5
                     {w5 is 255 wherever s=w1, 0 otherwise}
  imand w3 st w3
                     {w3 contains template values for s such that s>w1}
  imand w4 w2 w4
                     {w4 contains template values for w1 such that w1>s}
                     {w3 contains template values correct for s!=w1}
  imor w3 w4 w3
  imsup s w1 d
                     {d contains the correct values}
  imdiff st w2 w1
                     {w1 contains all pixels st>w2, 0 otherwise}
  imthresh w1 1 255 b {b is 1 wherever st>w2}
  immask b 0 255 w1 {w1 is 255 wherever st>w2, 0 otherwise}
  imdiff w2 st w4
                     {w4 contains all pixels w2>st, 0 otherwise}
  imthresh w4 1 255 b {b is 1 wherever w2>st}
  immask b 0 255 w4 {w4 is 255 wherever w2>st, 0 otherwise}
                      {w1 is 0 wherever st=w2, 255 otherwise}
  imor w1 w4 w1
  iminv w1 w1
                     {w1 is 255 wherever st=w2, 0 otherwise}
  imand w1 w5 w4
                     {w4 is 255 wherever st=w2 and s=(former)w1}
  imand st w4 w4
                     {w4 contains template values such that s=(former)w1 and st=w2}
  imor w3 w4 w3
                      {merge with template values already deter-
mined for s!=(former)w1}
  imthresh w1 1 255 b {b is 1 wherever st=w2}
  immask b 127 0 w1 {w1 is 0 wherever st=w2, 127 otherwise}
  imand w1 w5 w1
                     {w1 is 127 wherever s=(former)w1 and st!=w2}
  imor w1 w3 dt
  imfree w1
  imfree w2
  imfree w3
  imfree w4
  imfree w5
  imfree b
end
deproc imextinfngb imextinfngb s st d dt di
  syntax "imextinfngb greyin1 greyin2 greyout1 greyout2 direction"
  int w1 w2 w3 w4 w5 b;
  w1 := imalloc 8
  w2 := imalloc 8
  w3 := imalloc 8
  w4 := imalloc 8
  w5 := imalloc 8
  b := imalloc 1
  imcopyngb s w1 di 1 0
  imcopyngb st w2 di 1 0
                     {w3 contains all pixels s>w1, 0 otherwise}
  imdiff s w1 w3
  imthresh w3 1 255 b {b is 1 wherever s>w1}
  immask b 0 255 w3 {w3 is 255 wherever s>w1, 0 otherwise}
  imdiff w1 s w4
                      {w4 contains all pixels w1>s, 0 otherwise}
  imthresh w4 1 255 b {b is 1 wherever w1>s}
  immask b 0 255 w4 {w4 is 255 wherever w1>s, 0 otherwise}
  imor w3 w4 w5
                     {w5 is 255 wherever s!=w1, 0 otherwise}
  iminv w5 w5
                     {w5 is 255 wherever s=w1, 0 otherwise}
  imand w3 w2 w3
                     {w3 contains template values for w1 such that s>w1}
  imand w4 st w4
                    {w4 contains template values for s such that w1>s}
  imor w3 w4 w3
                    {w3 contains template values correct for s!=w1}
  iminf s w1 d
                    {d contains the correct values}
```

```
imdiff st w2 w1
                      {w1 contains all pixels st>w2, 0 otherwise}
  imthresh w1 1 255 b {b is 1 wherever st>w2}
  immask b 0 255 w1 {w1 is 255 wherever st>w2, 0 otherwise}
  imdiff w2 st w4
                      {w4 contains all pixels w2>st, 0 otherwise}
  imthresh w4 1 255 b {b is 1 wherever w2>st}
  immask b 0 255 w4 {w4 is 255 wherever w2>st, 0 otherwise}
  imor w1 w4 w1
                      {w1 is 0 wherever st=w2, 255 otherwise}
  iminv w1 w1
                      {w1 is 255 wherever st=w2, 0 otherwise}
                     {w4 is 255 wherever st=w2 and s=(former)w1}
  imand w1 w5 w4
  imand st w4 w4
                     {w4 contains template values such that s=(former)w1 and st=w2}
  imor w3 w4 w3
                      {merge with template values already deter-
mined for s!=(former)w1}
  imthresh w1 1 255 b {b is 1 wherever st=w2}
  immask b 127 0 w1 {w1 is 0 wherever st=w2, 127 otherwise}
                      {w1 is 127 wherever s=(former)w1 and st!=w2}
  imand w1 w5 w1
  imor w1 w3 dt
  imfree w1
  imfree w2
  imfree w3
  imfree w4
  imfree w5
  imfree b
end
deproc foldmax foldmax s st d dt sz
syntax "foldmax greyin greyin greyout greyout size"
  int w wt i;
  imcopy s d
  imcopy st dt
  if(grid = 1) then
    w := imalloc 8
    wt := imalloc 8
    for 1 to sz do
      i := 0
      imcopy d w
      imcopy dt wt
      for 1 to 6 do
        imextsupngb d dt w wt ++ i
      end
      imcopy w d
      imcopy wt dt
    end
    imfree w
    imfree wt
  else
    for 1 to sz do
      imextsupngb d dt d dt 1
      imextsupngb d dt d dt 3
      imextsupngb d dt d dt 5
      imextsupngb d dt d dt 7
    end
  end
end
deproc foldmin foldmin s st d dt sz
syntax "foldmin greyin greyin greyout greyout size"
  int w wt i;
```

```
imcopy s d
  imcopy st dt
  if(grid = 1) then
    w := imalloc 8
    wt := imalloc 8
   for 1 to sz do
     i := 0
     imcopy d w
     imcopy dt wt
     for 1 to 6 do
       imextinfngb d dt w wt ++ i
      end
     imcopy w d
     imcopy wt dt
    end
    imfree w
    imfree wt
  else
    for 1 to sz do
     imextinfngb d dt d dt 1
     imextinfngb d dt d dt 3
     imextinfngb d dt d dt 5
     imextinfngb d dt d dt 7
    end
  end
end
```

Appendix D

DImPAL Programs

D.1 Implementation of the new chromatin segmentation algorithm

```
; ....Segment the chromatin blobs within the nucleus....
image = resize(filter(nuclear_images#nucleus_number,"median",3),3,3)
mask = resize(clip(nuclear_masks#nucleus_number,1,1,1,1,"relative"),3,3)!=0
nucleus = image * mask
gradient = dilate(nucleus)-erode(nucleus)
nucleus = clip(nucleus,1,1,1,1,"relative")
minima = minima(nucleus)
inner_markers = minima>0
outer_marker = watershed(nucleus,label(inner_markers||border(nucleus)),"lines")==-1
labelled_markers=label(inner_markers)
labelled_markers=labelled_markers+outer_marker*(maximum(labelled_markers)+1)
result = watershed(gradient,labelled_markers)
blobs = result>0 && result<maximum(result)</pre>
labelled_blobs = blobs * result
outline = blobs [&&] !erode(blobs)
overlay = image[*]!outline [+] 255*outline
```

D.2 Additional code for constructing the generalised Delaunay graph

coordinates = integer(measurements_of_blobs.centroid_x~measurements_of_blobs.centroid_y)*4 attributes = measurements_of_blobs.connectivity_number==1 graph_image = draw_graph(adjacency_graph~attributes,coordinates) overlay(resize(overlay,4,4),graph_image)

Appendix

Fast Priority Queue Implementation in C

Notes:

- The implementation assumes that the calling (watershed) function takes an 8-bit grey-scale image as input and produces an integer image, containing the numerically labelled catchment basins and/or watershed lines, as output.
- 2. The elements of the FIFO queues within the priority queue are pointers to integers because the output image is an integer image.

```
{
  SIMPLE_QUEUE *queue;
  INTEGER highest_priority;
  highest_priority = priority_queue->highest_priority;
  priority = MAX (priority, highest_priority);
  queue = &priority_queue->queue[priority];
  insert_into_queue (queue, pixel_ptr);
}
static INTEGER *
remove_from_priority_queue (PRIORITY_QUEUE * priority_queue)
{
  SIMPLE_QUEUE *queue;
  INTEGER highest_priority, *pixel_ptr;
  highest_priority = priority_queue->highest_priority;
  queue = &priority_queue->queue[highest_priority];
  while ((pixel_ptr = remove_from_queue (queue)) == NULL &&
         highest_priority < 255)
    {
      priority_queue->highest_priority++;
      highest_priority = priority_queue->highest_priority;
      queue = &priority_queue->queue[highest_priority];
    }
  return (pixel_ptr);
}
void
insert_into_queue (SIMPLE_QUEUE * queue, INTEGER * pixel_ptr)
{
  if (queue->rear == NULL)
    printf ("Overflow\n");
  *queue->rear = pixel_ptr;
  queue->rear--;
}
INTEGER *
remove_from_queue (SIMPLE_QUEUE * queue)
ſ
  if (queue->front == queue->rear)
    return (NULL);
  else
    return (*queue->front--);
}
```

The following code fragment shows how the priority queue is initialised in the calling (watershed) function:

```
{
  SIMPLE_QUEUE queue[256];
  PRIORITY_QUEUE priority_queue;
  long histogram[256];
  int i;
  .
  .
  .
  /* ....INITIALISE PRIORITY QUEUE AND COMPONENT SIMPLE QUEUES.... */
  priority_queue.queue = queue;
  priority_queue.highest_priority = 0;
  queue[0].front = queue[0].rear = heap + histogram[0] - 1;
  for (i = 1; i < 256; i++)
     queue[i].front = queue[i].rear = queue[i - 1].front + histogram[i];
  .
  .
  .
</pre>
```

Appendix

Minkowski Functionals in \mathbb{R}^n

Associated with every compact convex set (ovoid) X in \mathbb{R}^n there exist n+1 Minkowski functionals. The k-th functional is denoted $W_k^{(n)}$. The functionals are defined by a recurrence relation on sub-dimensions of the space as follows (Serra, 1982, p. 104):

when
$$n = 1$$
 $W_0^{(1)}(X) = L(X)$ and $W_1^{(1)}(X) = 2;$
when $n > 1$ $W_0^{(n)}(X) = V^{(n)}(X);$ and
when $1 \le k \le n$ $W_k^{(n)}(X) = \frac{1}{nb_{n-1}} \int_{\Omega_n} W_{k-1}^{(n-1)}\left(\operatorname{proj}_{\Pi_{\omega}^{(n-1)}}(X)\right) d\omega,$

where L(X) is the length of X, $\operatorname{proj}_{\Pi_{\omega}^{(n-1)}}(X)$ is the projection of X onto the hyperplane $\Pi_{\omega}^{(n-1)}$ with normal ω , $V^{(n)}(X)$ is the *n*-volume of X, b_n is the *n*-volume of the unit ball, and Ω_n is the set of directions (i.e. the unit sphere) in \mathbb{R}^n .

Appendix G

QML AutoCyteR Slide Diagnoses

In the following table, HPV indicates the presence of human papillomavirus. For a description of CIN, refer to Section 6.4.1.

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-93398	69	Negative	0	
99-93622	117	Negative	0	
99-93623	29	Negative	0	
99-93625	44	Negative	0	
99-93626	99	Negative	0	
99-93971	72	Negative	0	
99-94040	60	Negative	0	
99-94041	147	Negative	0	
99-94042	136	Negative	0	
99-94167	2	Negative	0	
99-94168	46	Negative	0	
99-94169	5	Negative	0	
99-94170	103	Negative	0	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-94171	25	Negative	0	
99-94172	86	Negative	0	
99-94173	45	Negative	0	
99-94618	141	Negative	0	
99-94619	107	Negative	0	
99-94764	6	Negative	0	
99-94765	75	Negative	0	
99-94767	114	Negative	0	
99-94768	31	Negative	0	
99-97056	95	Negative	0	
99-97057	112	Negative	0	
99-97068	121	Negative	0	
99-97069	94	Negative	0	
99-97074	24	Negative	0	
99-97092	20	Negative	0	
99-97618	145	Negative	0	
99-97615	126	Negative	0	
99-97620	74	Negative	0	
99-97617	39	Negative	0	
99-97619	49	Negative	0	
99-104271	142	Negative	0	
99-104302	3	Negative	0	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-104303	101	Negative	0	
99-104304	104	Negative	0	
99-104372	92	Negative	0	
99-104373	106	Negative	0	
99-104387	82	Negative	0	
99-104388	98	Negative	0	
99-104498	10	Negative	0	
99-104499	90	Negative	0	
99-104500	128	Negative	0	
99-104501	137	Negative	0	
99-104514	139	Negative	0	
99-104516	34	Negative	0	
99-104517	81	Negative	0	
99-107370	70	Negative	0	
99-107369	110	Negative	0	
99-107343	65	Negative	0	
99-107368	93	Negative	0	
99-107339	115	Negative	0	
99-107375	125	Negative	0	
99-107436	118	Negative	0	
99-107664	113	Negative	0	
99-108480	59	Negative	0	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-108484	47	Negative	0	
99-108485	109	Negative	0	
99-108487	16	Negative	0	
99-108488	35	Negative	0	
99-108589	19	Negative	0	
99-108544	108	Negative	0	
99-108584	80	Negative	0	
99-108585	41	Negative	0	
99-108586	0	Negative	0	
99-108590	67	Negative	0	
99-110708	28	Negative	0	
99-110709	40	Negative	0	
99-110875	33	Negative	0	
99-110876	57	Negative	0	
99-110878	7	Negative	0	
99-110879	8	Negative	0	
99-110966	89	Negative	0	
99-111784	78	Negative	0	
99-111894	22	Negative	0	
99-111895	56	Negative	0	
99-111897	127	Negative	0	
99-111898	9	Negative	0	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-111899	143	Negative	0	
99-111900	11	Negative	0	
99-111902	131	Negative	0	
99-111903	130	Negative	0	
99-111904	146	Negative	0	
99-111905	87	Negative	0	
99-111906	17	Negative	0	
99-111936	77	Negative	0	
99-111937	15	Negative	0	
99-122699	129	Negative	0	
99-122700	134	Negative	0	
99-122701	1	Negative	0	
99-122749	53	Negative	0	
99-122750	73	Negative	0	
99-122751	133	Negative	0	
99-122755	91	Negative	0	
99-122756	119	Negative	0	
99-122758	96	Negative	0	
99-122759	71	Negative	0	
99-123555	144	Negative	0	
99-123556	135	Negative	0	
99-123771	26	Negative	0	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
99-87819	124	CIN 1	1	& HPV
99-90006	14	CIN 2	1	x 3 slides
99-90006	84	CIN 2	1	
99-90006	37	CIN 2	1	
99-95914	48	CIN 2	1	
99-94766	52	CIN 2	1	
99-115853	79	CIN 3	1	& glandular cells possible highgrade glan- dular epithelial abnormality
99-122752	58	CIN 2	1	
99-143943	63	CIN 2	1	
99-146273	105	CIN 3	1	
99-149738	76	CIN 2	1	& HPV
99-151440	61	CIN 2	1	& HPV
99-152486	55	CIN 3	1	
00-19171	51	CIN 2	1	& HPV
00-22697	123	CIN 2	1	x 2 slides
00-22697	140	CIN 2	1	
00-26712	32	CIN 2	1	& HPV
00-27411	102	CIN 2	1	
00-42514	36	CIN 2	1	
00-47504	21	CIN 2	1	& HPV x 3 slides

			1	
Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
00-47504	64	CIN 2	1	
00-47504	12	CIN 2	1	
00-49362	18	CIN 2	1	
99-125848	88	CIN 2	1	& HPV
99-128307	27	CIN $2/3$	1	
99-146858	23	CIN 2	1	& HPV
00-018748	13	CIN 2	1	
00-59665	42	CIN $2/3$	1	
00-62235	116	CIN 2	1	& HPV
99-62534	38	CIN 2	1	& HPV
99-107373	85	CIN 3	1	
99-110498	120	CIN 2	1	& HPV
99-122487	50	CIN 3	1	
99-124494	43	CIN 2	1	
00-65141	68	CIN 2	1	
00-67566	122	CIN 3	1	
00-74949	138	CIN 2	1	
00-76940	66	CIN 3	1	
00-79272	111	CIN 2	1	& HPV
00-85617	4	CIN 3	1	
00-101962	54	CIN 2	1	
00-103416	83	CIN 2	1	

Patient number	Imaging order (barcode number)	Diagnosis	Class	Notes
00-114189	30	CIN $2/3$	1	
00-114258	100	CIN 2	1	& HPV
00-114259	97	CIN 2	1	
00-122495	62	CIN 2	1	
00-130282	132	CIN 3	1	


Danielsson's G Shape Factor



Danielsson (1978) devised the following shape factor for a binary image $X \subset \mathbb{R}^2$:

$$G = \frac{A}{9\pi \left(\overline{d}\right)^2}$$

where

$$\overline{d} = \left(\iint_A r \, dA\right) / A,$$

and A is the area of X. The shape factor is dimensionless and takes the value 1 for a perfect circle. For a digital image, the quantity \overline{d} is estimated by computing the mean of the distance transform of X (see Section 5.4).

Appendix

R Program for Experiment 1

```
# leave_one_out.R
#
# AUTHOR: Andrew Mehnert
# DATE: May 2002
#
# HISTORY: July 2003 - added weights and priors
#
rm (list=ls())
#
# Load the ROC curve analysis code
#
source("ROC.R")
#
# Specify the population priors
#
USE.POPULATION.PRIORS <- FALSE
POPULATION.PROPORTION.OF.ABNORMALS = 0.08
POPULATION.PROPORTION.OF.NORMALS = 0.92
#
# Specify misclassification costs
#
COST.OF.MISCLASSIFYING.AN.ABNORMAL <- 1
#
# Import data
#
Class <- factor(read.csv('../AutoCyteImagingOrder.csv')$Classification,</pre>
```

```
levels=c(FALSE,TRUE),labels=c("Normal","Abnormal"))
blob.distance.data.set <- read.table("../measurements/blob_distance.dat")</pre>
working.data.set <- as.data.frame(cbind(blob.distance.data.set[,1:3],</pre>
                                    Class))
attr(working.data.set,"names")<-c(paste("F",1:(length(working.data.set)-1),
                                          sep=""),"Class")
barcode.of.slides.to.omit <- c(21,30,37,43,64,84,101,123,125)
slides.to.omit <- barcode.of.slides.to.omit + 1</pre>
working.data.set <- working.data.set[-slides.to.omit,]</pre>
rm(blob.distance.data.set)
rm(Class)
±
# Create the sequence 1,2,3,... up to the number of observations
sequence <- seq(1,nrow(working.data.set))</pre>
# Create a vector with missing entries to hold the predicted probability
# (of belonging to the class "Abnormal") for each observation held out
probabilities<-numeric(nrow(working.data.set))</pre>
# Apply the leave-one-out (holdout) methodology
for (i in sequence)
    {
    print(i)
    training.set <- working.data.set[-i,]</pre>
    holdout.observation <- working.data.set[i,]</pre>
    weight.for.normals <- 1.0 / COST.OF.MISCLASSIFYING.AN.ABNORMAL</pre>
    weight.for.abnormals <- 1.0</pre>
    if (USE.POPULATION.PRIORS == TRUE)
       {
       weight.for.normals <- weight.for.normals *</pre>
                                nrow(training.set) *
```

```
POPULATION.PROPORTION.OF.NORMALS /
                               sum(training.set$Class=="Normal")
       weight.for.abnormals <- weight.for.abnormals *</pre>
                                nrow(training.set) *
                                POPULATION.PROPORTION.OF.ABNORMALS /
                                sum(training.set$Class=="Abnormal")
       }
   training.set.weights <- (training.set$Class=="Normal") *</pre>
                            weight.for.normals +
                            (training.set$Class=="Abnormal") *
                            weight.for.abnormals
   analysis <- glm(Class~F1+F2+F3, family=binomial(link=logit),</pre>
                   data=training.set, weights=training.set.weights)
   probabilities[i] <- predict(analysis, holdout.observation,</pre>
                                type="response")
   }
#
# (1) Plot the empirical ROC curve and label it with the AUC +/- SE
# (2) Label points on the curve corresponding to cutpoints
      0, 0.1, 0.2, ..., 1
#
# (3) List the CCR, specificity, and sensitivity corresponding to the
#
      cutpoints
attach(working.data.set)
result<-make.ROC.curve(probabilities,Class)
X11()
#postscript(file="rawROC.eps",paper="special",width=8.0,height=8.0,
            horizontal=FALSE)
#
plot(result$rawx,result$rawy,type="l",xlab="P(False +ve)",
     ylab="P(True +ve)")
axis(1,tck=1,lty=2)
axis(2,tck=1,lty=2)
for (cutpoint in seq(0,1,0.1))
    confusion.matrix <- table(Class,factor(probabilities >= cutpoint,
                               levels=c(FALSE,TRUE),
                               labels=c("Normal","Abnormal")))
```

```
sensitivity <- confusion.matrix[2,2] / (confusion.matrix[2,1] +</pre>
```

```
confusion.matrix[2,2])
    specificity <- confusion.matrix[1,1] / (confusion.matrix[1,1] +</pre>
                   confusion.matrix[1,2])
    CCR <- (confusion.matrix[1,1] + confusion.matrix[2,2])/
            nrow(working.data.set)
    if (USE.POPULATION.PRIORS)
       {
       CCR.corrected <- (confusion.matrix[1,1] / (confusion.matrix[1,1] +
                         confusion.matrix[1,2])) *
                         POPULATION.PROPORTION.OF.NORMALS +
                         (confusion.matrix[2,2] / (confusion.matrix[2,1] +
                         confusion.matrix[2,2])) *
                         POPULATION. PROPORTION. OF. ABNORMALS
       cat(round(sensitivity,2),", ", round(specificity,2), ", ",
           round(CCR*100,1),", ",round(CCR.corrected*100,1),"\n")
       } else
       {
       cat(round(sensitivity,2),", ", round(specificity,2), ", ",
           round(CCR*100,1),"\n")
       }
    points(1-specificity,sensitivity,pch="*",cex=3)
    #text(1-specificity+0.03,sensitivity,paste(cutpoint))
    }
text(0.7,0.1,paste("AUC = ",round(result$rawAUC,3),"+/-",
     round(result$rawSE,3)))
#dev.off()
```

l Appendix 🗨

R Program for Experiment 2 $\,$

```
# holdout.R
#
# AUTHOR: Andrew Mehnert
# DATE: May 2002
#
# HISTORY: August 2003 - added table code for features
#
rm (list=ls())
#
# Load the Venables and Ripley MASS library --- stepAIC()
#
library(MASS)
#
# Load the ROC curve analysis code
#
source("ROC.R")
#
# Constants
#
HOLD.OUT.PROPORTION <- 1/3
NUMBER.OF.TESTS <- 100
COST.OF.MISCLASSIFYING.AN.ABNORMAL <- 1
#
# Import data
#
source("import_slide_features_from_DImPAL.R")
```

```
# Specify features (variables) to keep (exclude grey-level features)
features.to.keep <- seq(1,56)</pre>
features.to.keep <- features.to.keep[c(seq(-8,-15),seq(-18,-23),</pre>
                                         seq(-32,-51))]
# Include the "Class" column also
working.data.set <- working.data.set[,c(features.to.keep,</pre>
                                          length(working.data.set))]
# omit unsuitable slides
barcode.of.slides.to.omit <- c(21,30,37,43,64,84,101,123,125)
slides.to.omit <- barcode.of.slides.to.omit + 1</pre>
working.data.set <- working.data.set[-slides.to.omit,]</pre>
#
# Construct the model formula, based on 'features.to.keep', to be
# used in the generalised linear model
# The formula has the form: Class ~ X? + X? + ...
#
my.formula <- formula(paste("Class ~",paste("X",features.to.keep,sep="",</pre>
                              collapse=" + ")))
#
# Perform NUMBER.OF.TESTS trials. Each trial consists of
# (1) Randomly selecting HOLD.OUT.PROPORTION of the normals and
      HOLD.OUT.PROPORTION of the abnormals (to be used as a test set);
#
# (2) Fitting a logistic regression model to the remaining data using
#
      stepwise feature selection based on BIC; and
# (3) Classifying the test set.
#
# Notes: (a) The classifier formulae are accumulated in "list.of.formulae"
         (b) The AUCs are accumulated in "AUC"
#
#
         (c) The classification probabilities for the holdout observations
#
             in each trial are accumulated in "probabilities"
#
list.of.formulae <- list(NUMBER.OF.TESTS)</pre>
list.of.features <- list(NUMBER.OF.TESTS)</pre>
AUC <- numeric(NUMBER.OF.TESTS)
probabilities <- list(NUMBER.OF.TESTS)</pre>
normals <- working.data.set[working.data.set$Class== "Normal",]</pre>
abnormals <- working.data.set[working.data.set$Class== "Abnormal",]
rm(working.data.set)
```

```
number.of.normals <- nrow(normals)</pre>
number.of.abnormals <- nrow(abnormals)</pre>
number.of.normals.to.hold.out <- trunc(number.of.normals *</pre>
                                         HOLD.OUT.PROPORTION)
number.of.abnormals.to.hold.out <- trunc(number.of.abnormals *
                                           HOLD.OUT.PROPORTION)
for (test.number in 1:NUMBER.OF.TESTS)
  ł
  cat("**** Test number",test.number,"\n")
  which.normals.to.use.for.testing <- sample(1:number.of.normals,</pre>
                                            number.of.normals.to.hold.out)
  which.abnormals.to.use.for.testing <- sample(1:number.of.abnormals,</pre>
                                              number.of.abnormals.to.hold.out)
  training.set <- rbind(normals[-which.normals.to.use.for.testing,],</pre>
                         abnormals[-which.abnormals.to.use.for.testing,])
  test.set <- rbind(normals[which.normals.to.use.for.testing,],</pre>
                     abnormals[which.abnormals.to.use.for.testing,])
  weight.for.normals <- 1.0 / COST.OF.MISCLASSIFYING.AN.ABNORMAL</pre>
  weight.for.abnormals <- 1.0
  training.set.weights <- (training.set$Class=="Normal") *</pre>
                           weight.for.normals +
                           (training.set$Class=="Abnormal") *
                           weight.for.abnormals
  # Fit a logistic regression model to the training data and perform
  # stepwise variable selection
  #
  my.lrm <- glm(Class~1, family=binomial(link=logit), data=training.set,</pre>
                 weights=training.set.weights)
  # Use BIC rather than the default AIC for stepwise selection
  my.step <- stepAIC(my.lrm, scope=my.formula, k=log(nrow(training.set)),</pre>
                      direction = "both")
  #
  # Record the model formula, and parse and record the list of features used
  list.of.formulae[[test.number]] <- my.step$formula</pre>
```

```
parsed.features <- textConnection(gsub("\\+","",gsub("X","",</pre>
                                    paste(my.step$formula)[3])))
  list.of.features[[test.number]] <- scan(parsed.features)</pre>
  close(parsed.features)
  #
  # Evaluate the performance of the fitted logistic classifier on the
  # testing data
  #
  probabilities[[test.number]] <- predict(my.step, test.set,</pre>
                                          type="response")
 result<-make.ROC.curve(probabilities[[test.number]],test.set$Class)</pre>
  cat("AUC:",result$rawAUC,"\n")
  AUC[test.number] <- result$rawAUC
 }
cat("\n-----\n\n")
cat(paste("AUC = ",round(mean(AUC),3),"+/-",round(sqrt(var(AUC)),3),"\n"))
cat("\nFeature frequency table:\n")
print(table(unlist(list.of.features)))
```



R program for Empirical ROC Curve Analysis

```
# ROC.R
#
# AUTHOR: Andrew Mehnert
# DATE: 4/2/00
#
# HISTORY: 15/8/2003 (1) Corrected (x,y) pairs for plotting. Does not
#
                          affect AUC.
#
                      (2) Added SE calculations.
#
                      (3) Changed the scale for the smoothed plot.
make.ROC.curve <- function(degree.of.suspicion,class)</pre>
{
total.number.of.negatives <- sum(class=="Normal")</pre>
total.number.of.positives <- sum(class=="Abnormal")</pre>
# Order the "degree.of.suspicion" scores and create a vector
# containing the class designation of each.
sortedclass <- class[order(degree.of.suspicion,decreasing=TRUE)]</pre>
# Assuming that increasing "degree.of.suspicion" scores indicate
# increasing likelihood of being "Abnormal" (positive), record the number
# of false positives and the number of true positives as the
# decision threshold is varied (these are stored as elements
# of the x and y vectors respectively).
number.of.true.positives <- 0</pre>
number.of.false.positives <- 0</pre>
x <- 0
y <- 0
```

```
for (i in seq(1,length(degree.of.suspicion)))
    {
    if (sortedclass[i]=="Abnormal")
       number.of.true.positives <- number.of.true.positives + 1</pre>
    else
       number.of.false.positives <- number.of.false.positives + 1</pre>
    y <- append(y,number.of.true.positives)</pre>
    x <- append(x,number.of.false.positives)</pre>
    }
# The list of x and y coordinates determines a step function.
# Calculate the area under this function.
AUC <- 0
for (i in seq(2,length(degree.of.suspicion)))
    ſ
    AUC <- AUC + (x[i]-x[i-1])*y[i-1]
    }
AUC<-AUC/max(x)/max(y)
SE<-sqrt((AUC*(1-AUC)+(total.number.of.positives-1)*</pre>
         (AUC/(2-AUC)-AUC^2)+(total.number.of.negatives-1)*
         ((2*AUC<sup>2</sup>)/(1+AUC)-AUC<sup>2</sup>))/
         (total.number.of.positives*total.number.of.negatives))
x <- x/max(x)
y <- y/max(y)
# Rescan the x and y coordinates of the step function and
# determine a new pair of coordinates representing the
# midpoints of each vertical or horizontal step.
plotx <- 0
ploty <- 0
start <- 1
if (x[2] != x[1])
   horizontal <- TRUE
else
   horizontal <- FALSE
for (i in seq(2,length(degree.of.suspicion)))
    if ((x[i] == x[i-1]) && horizontal==TRUE)
       {
       horizontal <- FALSE
       plotx <- append(plotx,(x[i-1]+x[start])/2)</pre>
```

```
ploty <- append(ploty,y[start])</pre>
       start <- i-1
       }
    else
       if ((y[i] == y[i-1]) && horizontal==FALSE)
           ſ
          horizontal <- TRUE
          plotx <- append(plotx,x[start])</pre>
          ploty <- append(ploty,(y[i-1]+y[start])/2)</pre>
          start <- i-1
           }
    }
if (max(plotx) != max(x))
   {
   plotx <- append(plotx,max(x))</pre>
   ploty <- append(ploty,max(y))</pre>
   }
# Calculate the area under the "smoothed" function.
# This is done by summing the areas of the individual trapeziums.
plotAUC <- 0
for (i in seq(2,length(plotx)))
    Ł
    plotAUC <- plotAUC + (plotx[i]-plotx[i-1])*(ploty[i]+ploty[i-1])/2</pre>
    }
plotAUC <- plotAUC/max(x)/max(y)</pre>
plotSE<-sqrt((plotAUC*(1-plotAUC)+(total.number.of.positives-1)*</pre>
              (plotAUC/(2-plotAUC)-plotAUC^2)+(total.number.of.negatives-1)*
              ((2*plotAUC^2)/(1+plotAUC)-plotAUC^2))/
              (total.number.of.positives*total.number.of.negatives))
plotx <- plotx/max(plotx)</pre>
ploty <- ploty/max(ploty)</pre>
r <-list(rawx=x,rawy=y,rawAUC=AUC,rawSE=SE,</pre>
         x=plotx,y=ploty,AUC=plotAUC,SE=plotSE)
r
}
```

Appendix

Slide Features

Refer to Table 6.5 and Table 6.7 for a description of the inputs N1 to N28. The features marked with an asterisk are not defined in terms of grey-level.

Feature identifier	Statistic	Inputs
X1*	number of nuclei re- tained after artefact rejection	
X2*, X3*	mean and standard deviation	blob count $(N8)$ for each nucleus
X4*, X5*	mean and standard deviation	nucleus areas $(N1)$
X6*, X7*	mean and standard deviation	nucleus perimeters $(N2)$
X8, X9	mean and standard deviation	nucleus 3D connectivity numbers $(N3)$
X10, X11	mean and standard deviation	nucleus surface areas $(N4)$
X12, X13	mean and standard deviation	nucleus volumes $(N5)$

Feature identifier	Statistic	Inputs
X14, X15	mean and standard deviation	nucleus mean grey-levels $(N6)$
X16*, X17*	mean and standard deviation	total blob area $(N25)$ for each nucleus
X18, X19	mean and standard deviation	total blob surface area $(N26)$ for each nucleus
X20, X21	mean and standard deviation	total blob volume $(N27)$ for each nucleus
X22, X23	mean and standard deviation	sum of blob mean grey-levels $(N28)$ for each nucleus
X24*, X25*	mean and standard deviation	N9 for each nucleus
X26*, X27*	mean and standard deviation	N10 for each nucleus
X28*, X29*	mean and standard deviation	N11 for each nucleus
X30*, X31*	mean and standard deviation	N12 for each nucleus
X32, X33	mean and standard deviation	N15 for each nucleus
X34, X35	mean and standard deviation	N16 for each nucleus
X36, X37	mean and standard deviation	N17 for each nucleus
X38, X39	mean and standard deviation	N18 for each nucleus
X40, X41	mean and standard deviation	N19 for each nucleus

Feature identifier	Statistic	Inputs
X42, X43	mean and standard deviation	N20 for each nucleus
X44, X45	mean and standard deviation	N13 for each nucleus
X46, X47	mean and standard deviation	N14 for each nucleus
X48, X49	mean and standard deviation	N21 for each nucleus
X50, X51	mean and standard deviation	N22 for each nucleus
X52*, X53*	mean and standard deviation	N23 for each nucleus
X54*, X55*	mean and standard deviation	N24 for each nucleus



Slide Feature Box-and-Whisker Plots

























- -

-

+

1.0