

SEGMENTATION AND SYMMETRY MEASURE FOR IMAGE ANALYSIS: APPLICATION TO DIGITAL DERMATOSCOPY

THÈSE N° 2045 (1999)

PRÉSENTÉE AU DÉPARTEMENT D'ÉLECTRICITÉ

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Philippe SCHMID

Ingénieur électricien diplômé EPF

de nationalité suisse et originaire des Bayards et La Côte-aux-Fées (NE)

acceptée sur proposition du jury:

Prof. M. Kunt, directeur de thèse

Prof. G. Coray, rapporteur

Dr D. Squire, rapporteur

Dr J.-P. Thiran, rapporteur

Prof. L. Torres, rapporteur

Lausanne, EPFL
1999

Acknowledgements

My greatest gratitude goes to my parents Edith and Martin for having supported me during all the stages I went through up to now. I am also deeply grateful to my wife Patricia, my sister Anita, and all the friends whose presence was a great aid.

I would like to thank Prof. Murat Kunt who encouraged this work and always had motivating words, Dr. Joël Guillod (MD) for providing the images and discussing the results (and other more general topics !), as well as all the LTS members for their friendship. Finally, I would like to thank myself for having always had the motivation and the energy to carry out this research and to write this thesis.

Contents

List of Figures	vii
List of Tables	xi
Résumé	xiii
Abstract	xv
1 Introduction	1
1.1 Scope of the thesis	1
1.2 Investigated problems	2
1.3 Initial motivation	3
1.4 Major contributions	4
1.5 Organization of the thesis	4
2 Edge preserving image filtering	7
2.1 Introduction	7
2.2 Gaussian filter	8
2.3 Median filter	10
2.4 Morphological filters	11
2.5 Diffusion in the framework of image filtering	15
2.5.1 From physics to image processing	15
2.5.2 The diffusion filter	16
2.6 Nonlinear isotropic diffusion	20
2.6.1 Formulation	20
2.6.2 Conductivity functions	22
2.6.3 Behavior in noisy images	24
2.6.4 Numerical implementation	25
2.7 Extended diffusion schemes	28
2.7.1 Introduction	28
2.7.2 Anisotropic diffusion	28
2.7.3 Automatic setting of k_0 and stopping criterion	30
2.7.4 Pilot for scale dependent diffusion	31
2.7.5 Time varying conductivity function	31
2.7.6 Application to vector-valued images	33
2.8 An alternative to backward-diffusion: shock filters	33

2.9	Other developments	34
2.10	Summary	36
	Bibliography	41
3	Colorimetry and color spaces	43
3.1	Introduction	43
3.2	Colorimetry	43
3.2.1	Principles and definitions	43
3.2.2	Real and imaginary primary stimuli	47
3.2.3	Standard colorimetric observers	49
3.2.4	CIE standard illuminants	51
3.3	Color spaces	52
3.3.1	Linear colorimetric transformations	52
3.3.2	The principal component decomposition	54
3.3.3	Nonlinear transformations and uniform color spaces	55
3.4	Summary	57
	Bibliography	59
4	Image segmentation by color clustering	61
4.1	Introduction	61
4.2	Image segmentation strategies	62
4.3	Data clustering	64
4.3.1	Introduction	64
4.3.2	Probability density estimation	65
4.3.3	Classical clustering techniques	69
4.4	Fuzzy c-means clustering technique	73
4.4.1	Basic algebra of fuzzy sets and fuzzy partition space	73
4.4.2	Fuzzy c-means	74
4.4.3	Orientation sensitive FCM	78
4.4.4	Noise reduction	83
4.5	Cluster detection	84
4.5.1	Introduction	84
4.5.2	Geometrical approach	85
4.6	Preparation and cleaning phases	87
4.6.1	Pre-processing	87
4.6.2	Region selection and filling	87
4.7	Summary	88
	Bibliography	93
5	Topology-unconstrained contour detection	95
5.1	Introduction	95
5.2	Classical edge detection	96
5.3	Canny's edge detector	102
5.3.1	Edge detection and localization criteria	102
5.3.2	Optimal edge detector	103
5.3.3	Two-dimensional implementation	104

5.4	Active contour models	106
5.4.1	Introduction	106
5.4.2	Snakes	106
5.5	A geometrical approach to active contours	110
5.5.1	Contour detection with front propagation	110
5.5.2	A level set formulation of curve evolution	112
5.5.3	Geodesic active contours	114
5.6	Nonlinear isotropic diffusion and morphological flooding	118
5.7	Summary	121
	Bibliography	125
6	Symmetry axes and symmetry quantification	127
6.1	Introduction	127
6.2	Definition and approaches	128
6.2.1	What is symmetry ?	128
6.2.2	Human symmetry perception	128
6.3	Symmetry in computer vision	129
6.4	Principal component approaches	130
6.4.1	Symmetry and orientation	130
6.4.2	Spatial formulation	130
6.4.3	Frequency domain formulation	134
6.4.4	Orientation through ellipse fitting	135
6.4.5	The segmentation problem	137
6.5	Symmetry optimization	137
6.5.1	Symmetry measure and parameter space	137
6.5.2	Color and texture descriptors	139
6.5.3	Genetic algorithm approach	141
6.5.4	Self-organizing maps derived approach	143
6.6	Symmetry map computation	145
6.6.1	Direct computation	145
6.6.2	Multi-resolution approach	146
6.6.3	Pyramidal approach	147
6.6.4	Noise component	148
6.7	Summary	151
	Bibliography	155
7	Digital dermatoscopy	157
7.1	Introduction	157
7.2	Skin cancer diagnosis	158
7.2.1	Introduction	158
7.2.2	Imaging	159
7.2.3	Clinical diagnosis schemes	161
7.2.4	CAD scheme and principles	162
7.2.5	Feature extraction and classification	163
7.3	Hair removal in dermatoscopic images	164

7.4	Segmentation and boundary detection	166
7.4.1	Introduction	166
7.4.2	Visual assessment	166
7.4.3	Validation	168
7.5	Symmetry	173
7.5.1	Introduction	173
7.5.2	Visual assessment	173
7.5.3	Symmetry values for diagnostic classification	175
7.6	Summary	177
	Bibliography	181
8	Conclusions	183
8.1	Summary	183
8.2	Achievements	186
8.3	Possible extensions	187
	Curriculum Vitae	189

List of Figures

1.1	Image analysis scheme	3
2.1	Binary test image with geometrical figures	9
2.2	Test image <i>bicycle</i>	9
2.3	Geometrical and spatial distortion introduced by Gaussian low-pass filtering	10
2.4	Gaussian filtering example	10
2.5	Geometrical distortion introduced by median filtering	11
2.6	Median filtering of <i>bicycle</i>	11
2.7	Erosion and dilation operators	12
2.8	Opening and closing operators	12
2.9	Morphological filter $(f \circ b) \bullet b$	13
2.10	Morphological filter $(f \bullet b) \circ b$	13
2.11	Morphological filter $[(f \bullet b) + (f \circ b)]/2$	14
2.12	Morphological filter $[(f \circ b) \bullet b + (f \bullet b) \circ b]/2$	14
2.13	Morphological filtering of <i>bicycle</i> $[(f \bullet b) + (f \circ b)]/2$	15
2.14	Morphological filtering of <i>bicycle</i> $[(f \circ b) \bullet b + (f \bullet b) \circ b]/2$	15
2.15	Geodesic reconstruction	16
2.16	Scale space representation of 1-D signal	17
2.17	Zero-crossings of the second derivative of a scale-space transform	17
2.18	Transfer function of the diffusion filter	20
2.19	Transfer function of the finite difference approximation to the diffusion filter	20
2.20	Exponentially decreasing Perona-Malik conductivity function	22
2.21	Inverse decreasing Perona-Malik conductivity function	22
2.22	Well-posed inverse decreasing conductivity function	23
2.23	Well-posed piecewise linear conductivity function	23
2.24	Example of well-posed nonlinear isotropic diffusion after 50, 500, and 1000 iterations	25
2.25	Binary test image corrupted by zero mean Gaussian noise	26
2.26	Noisy binary test image after 500 iterations of nonlinear isotropic diffusion	26
2.27	<i>Bicycle</i> corrupted by zero mean Gaussian noise	27
2.28	Noisy <i>bicycle</i> after 1000 iterations of nonlinear isotropic diffusion	27
2.29	Example of diffusion with ill-posed conductivity function	28
2.30	Example of diffusion with ill-posed conductivity function in noisy environment . .	28
2.31	PSNR evolution in noisy environment with Perona-Malik conductivity function . .	29
2.32	PSNR evolution in noisy environment with well-posed conductivity function	29
2.33	Anisotropic diffusion in the direction normal to the gradient	30

2.34	Anisotropic diffusion parallel to the gradient	30
2.35	Example of <i>pilot</i> for scale-dependent diffusion	32
2.36	Sigmoid function used for progressive smoothing-to-sharpening diffusion	32
2.37	Blurred square function before and after edge enhancement with a shock filter	35
2.38	Shock filter with unstable numerical implementation	35
2.39	Edge enhancement using a shock filter after anisotropic diffusion	36
2.40	Edge enhancement using a shock filter after anisotropic diffusion of a noisy image	36
2.41	Gradient of <i>bicycle</i>	37
2.42	Gradient of <i>bicycle</i> after isotropic filtering	37
3.1	(r, g, b) -chromaticity diagram	48
3.2	Chromaticity diagram of CIE 1931 standard colorimetric observer in the system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	48
3.3	Color-matching function of CIE 1931 standard colorimetric observer in the system of real primary stimuli $\mathbf{R}, \mathbf{G}, \mathbf{B}$	49
3.4	Color-matching function of CIE 1964 supplementary standard colorimetric observer in the system of real primary stimuli $\mathbf{R}, \mathbf{G}, \mathbf{B}$	49
3.5	Color-matching function of CIE 1931 standard colorimetric observer in the transformed system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	50
3.6	Color-matching function of CIE 1964 supplementary standard colorimetric observer in the transformed system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	50
3.7	The color matching experiment	51
4.1	Example of Parzen window	66
4.2	Density approximation using the Parzen window	66
4.3	Sparse Gaussian mixture	71
4.4	Dense Gaussian mixture	71
4.5	C-means clustering of sparse Gaussian mixture	72
4.6	C-means clustering of dense Gaussian mixture	72
4.7	C-means clustering of noisy Gaussian mixture	72
4.8	C-means clustering of Gaussian mixture initialized with more centers	72
4.9	Fuzzy c-means clustering of sparse Gaussian mixture	77
4.10	Fuzzy c-means clustering of dense Gaussian mixture	77
4.11	Fuzzy c-means clustering of noisy Gaussian mixture ($m = 2$)	78
4.12	Fuzzy c-means clustering of noisy Gaussian mixture ($m = 2.5$)	78
4.13	Fuzzy c-means clustering of Gaussian mixture with less centers	79
4.14	Fuzzy c-means clustering of Gaussian mixture initialized with more centers	79
4.15	Maximum fuzzy membership of Gaussian mixture	80
4.16	Fuzzy partition of Gaussian mixture	80
4.17	Orientation-sensitive fuzzy c-means clustering of sparse Gaussian mixture	81
4.18	Orientation-sensitive fuzzy c-means clustering of dense Gaussian mixture	81
4.19	Maximum fuzzy membership of Gaussian mixture after OS-FCM clustering	81
4.20	Fuzzy partition of Gaussian mixture after OS-FCM clustering	81
4.21	One-dimensional partition of Gaussian mixture	82
4.22	Quiver plot of gradient orientations with superimposed partition	82

4.23	OS-FCM clustering of noisy Gaussian mixture	83
4.24	Two-dimensional histogram and contour lines	85
4.25	One-dimensional contour lines tree	85
5.1	Smooth edge model and zero-crossings of second-order derivative	97
5.2	Laplacian of Gaussian	97
5.3	Zero-crossings of Laplacian using different numerical approximations	98
5.4	Transfer function of central difference approximation to the Laplacian	99
5.5	Transfer function of the Sobel approximation to the Laplacian	99
5.6	Zero-crossings of Laplacian for noisy images	100
5.7	Test image after nonlinear isotropic diffusion	101
5.8	Zero-crossings of Laplacian after nonlinear isotropic diffusion, using the Sobel approximation	101
5.9	Edge detection with Canny's edge detector	105
5.10	Canny's edge detector on noisy image with $\sigma = 3$	106
5.11	Canny's edge detector on noisy image with $\sigma = 6$	106
5.12	Example of snake convergence for non-convex shape	109
5.13	Propagating curve	111
5.14	Narrow band used in level set approach	111
5.15	Curvature dependent front propagation	114
5.16	Curvature dependent front propagation with constant speed term	114
5.17	Geodesic shape detection with inward motion	116
5.18	Geodesic shape detection with outward motion	116
5.19	Geodesic active contour for multiple objects detection	117
5.20	Example of morphological flooding	120
5.21	Iso-level curves	120
5.22	Energy functional used for contour selection	121
5.23	Minimum energy iso-level contour	121
6.1	Principal component computation	131
6.2	Object with aligned orientation and symmetry axis	132
6.3	Object with orthogonal orientation and symmetry axis	132
6.4	Gray level principal component of symmetrical object with white hole	133
6.5	Gray level principal component of symmetrical object with black hole	133
6.6	Normalized parameters for symmetry map	139
6.7	Sampling grid used to produce a symmetry map	139
6.8	Filter bank derived from Gabor filters	140
6.9	Genetic algorithm scheme	142
6.10	Probability density function used for crossover operator	143
6.11	Example image showing an object with symmetric pattern and circular contour	143
6.12	Example of GA-based symmetry optimization	144
6.13	Example of SOM-based symmetry optimization	144
6.14	Example of symmetry map	145
6.15	Symmetry map with reduced precision	145
6.16	Symmetry map obtained with randomly chosen image pixels	146

6.17	Filtered symmetry map	146
6.18	“Mexican hat” filter and second derivative of Gaussian (normalized).	147
6.19	Transfer function of “Mexican hat” filter and second derivative of Gaussian	147
6.20	Pyramidal image representation	148
6.21	Multi-scale symmetry map	149
6.22	Pyramidal approach to symmetry map computation	149
6.23	Angular deviation vs Gaussian noise variance	149
6.24	Symmetry axis of isotropic shape with symmetric texturized content	150
6.25	Symmetry axis of isotropic shape with symmetric texturized content and noise . .	150
7.1	Death rate evolution for Switzerland and Europe from 1969 to 1994	158
7.2	Death rate versus age for Switzerland in 1991	158
7.3	Macroscopic image of benign melanocytic skin lesion	160
7.4	Dermatoscopic image of benign melanocytic skin lesion	160
7.5	Computer aided diagnosis system for skin cancer	162
7.6	Hair removal scheme	165
7.7	Image corrupted with hair	165
7.8	Image after hair removal	165
7.9	Ratios between variance of the different KL components in the RGB color space .	167
7.10	Ratios between variance of the different KL components in the $L^*u^*v^*$ color space	167
7.11	Pigmented lesion contour detection	168
7.12	Contour detection of complex pigmented lesion	169
7.13	Pigmented skin lesion segmented with the FCM technique	170
7.14	Pigmented skin lesion segmented with the OS-FCM technique	170
7.15	Contour drawn by physician (A)	170
7.16	Contour drawn by physician (B)	170
7.17	Divergence for contours drawn by dermatologist (A)	171
7.18	Divergence for contours drawn by dermatologist (B)	171
7.19	Probability image obtained from the hand drawn and automatic contours	172
7.20	Probability that an image pixel has been misclassified (automatic techniques) . .	172
7.21	Probability that an image pixel has been misclassified (A)	172
7.22	Probability that an image pixel has been misclassified (B)	172
7.23	Symmetry map of symmetric lesion	174
7.24	Axes of symmetric lesion	174
7.25	Symmetry map of asymmetric lesion	174
7.26	Axes of asymmetric lesion	174
7.27	Color and texture symmetry MSE of 50 malignant melanoma and 50 naevi (1) . .	175
7.28	Color and texture symmetry MSE of 50 malignant melanoma and 50 naevi (2) . .	175

List of Tables

3.1	The colors of the elements	44
3.2	Important definitions related to colorimetry	46
3.3	Linear color space conversions from X, Y, Z	53
7.1	Linear classification of symmetry values for benign and malignant lesions	176
7.2	Linear classification of symmetry values for MM and naevi	176
7.3	Linear classification of melanocytic lesions using the KL transform	177

Résumé

Dans le cadre de ce travail de doctorat, quatre thèmes liés à l'analyse d'images sont traités: le filtrage, la segmentation de couleur, la détection de contours ainsi que la mesure de symétrie de texture, de couleur et de forme. Les trois premiers thèmes sont essentiels afin de détecter efficacement les objets d'intérêt présents dans l'image et de permettre leur analyse et leur classification. L'utilisation de mesures telles que le degré de symétrie est essentielle pour ce type d'applications. Bien que la combinaison de différents paramètres est généralement nécessaire à une classification efficace, nous axons nos investigations sur un paramètre unique afin de l'exploiter pleinement.

Dans le cadre du filtrage d'images numériques, l'accent est mis sur des techniques permettant de préserver la position et la forme des contours. A cet effet une étude approfondie du *filtrage isotropique non-linéaire* est proposée. Cette technique s'inspire de la diffusion de la chaleur dans la matière tout en introduisant une composante non-linéaire permettant de contrôler la diffusion en fonction du gradient local. Le réglage d'un seuil unique permet ainsi de limiter, voir même d'empêcher la diffusion au travers des transitions les plus fortes correspondantes aux contours marquants de l'image. Cette technique est particulièrement résistante au bruit de type gaussien et respecte la courbure des contours, évitant ainsi des effets d'arrondi tels que ceux rencontrés lors du filtrage gaussien.

L'utilisation de techniques de coalescence à des fins de segmentation de couleur est étudiée. En particulier, des approches utilisant la logique floue sont exploitées, avec le développement d'une nouvelle technique nommée *orientation sensitive fuzzy c-means*, dont la particularité est de tenir compte de l'orientation locale des différents groupements. Cette technique est donc particulièrement bien adaptée aux distributions de type gaussiennes et aux distributions proches de ce modèle. L'utilisation de deux composantes couleur, obtenues par transformée de Karhunen-Loève, est présentée, ainsi que le calcul de la position et du nombre de groupements par comparaison de paramètres géométriques. Ceux-ci sont calculés pour chaque maximum présent dans l'histogramme bidimensionnel calculé pour les deux composantes couleur principales.

Dans cette thèse, la détection de contours a pour but l'extraction de contours indépendamment de la topologie de l'image. Les développements sont orientés du côté des contours actifs, ou *snakes*, et de leur équivalent géométrique, les *contours actifs géodésiques*. Ces différentes techniques sont présentées et illustrées. Finalement une nouvelle approche est présentée, utilisant le filtrage isotropique non-linéaire multi-composantes appliqué à des images couleur, et l'immersion morphologique. Cette technique permet d'extraire un ensemble de courbes de niveau considérées comme contours possibles, et de sélectionner les courbes de niveau correspondant à des contours d'objets par minimisation globale ou locale d'une fonction d'énergie.

Le problème de la quantification de la symétrie est le dernier thème abordé dans cette thèse. Une attention particulière est portée à son application à des objets faiblement symétriques ou

asymétriques par une approche d'optimisation d'index de symétrie. Différentes techniques sont développées à cet effet, utilisant des approches multi-résolutions, un algorithme génétique ou encore des cartes auto-organisatrices. Les données sont supposées multi-dimensionnelles, telles que des descripteurs de couleur ou de texture. Ces deux types d'informations peuvent ainsi être intégrés indépendamment au système de mesure. Finalement, la capacité à détecter un axe de symétrie dans un environnement bruité est démontrée et illustrée.

Les différentes techniques sont appliquées au problème de l'analyse d'images dermatoscopiques. Ce type d'images est utilisé par les dermatologues pour diagnostiquer le cancer de la peau. L'efficacité des techniques de segmentation et de détection de contours, associées au filtrage isotropique non-linéaire est illustrée et démontrée. L'amélioration introduite par l'utilisation de nos techniques de mesure de symétrie dans la séparation entre lésions bénignes et malignes par rapport à des techniques plus classiques est démontrée pour une séparation linéaire. Nous cherchons à démontrer ici que l'étude approfondie des différents paramètres diagnostiques, dont la symétrie fait partie, est nécessaire à l'élaboration d'un système d'aide au diagnostic performant.

Abstract

In this thesis, four topics related to image analysis are investigated: filtering, color segmentation, contour detection, and symmetry quantification for shape, color, and texture information. The three first topics are essential to detect efficiently the objects of interest present in the image, and to allow their analysis and classification. The use of features such as the degree of symmetry are necessary to this application. Although the combination of different features is usually necessary for an efficient classification, we want to focus on a unique feature and fully exploit it.

In the framework of image filtering, we emphasize techniques which preserve the location and geometry of contours. *Nonlinear isotropic filtering* is therefore deeply investigated. This technique is inspired from the diffusion of heat in matter, with a nonlinear component that is based on the local gradient and that controls the diffusion. A unique threshold is used to limit and even stop the diffusion through the strongest transitions, which should then correspond to the object boundaries. This technique is particularly noise resistant and preserves the curvature of contours, which avoids rounding effects such as those encountered when using Gaussian low-pass filtering.

Clustering techniques for color segmentation are investigated next. In particular, techniques using fuzzy sets theory are exploited, leading to the development of a new clustering scheme called *orientation sensitive fuzzy c-means*, whose main characteristic is to take into account the cluster orientation. This technique is therefore especially adapted to Gaussian and elliptic shaped mixtures. The clustering is performed in a two-dimensional histogram, which is computed with the two principal components of the Karhunen-Loève transform of the color frames. The number of clusters and the cluster center location must be obtained as well. The selection is based on geometrical features computed for every maximum in the two-dimensional histogram.

In this thesis, the contour detection aims at the extraction of topology independent contours. Active contours, also called *snakes*, and their geometrical counterpart, *geodesic active contours*, are investigated for this purpose. These different techniques are presented and illustrated. Finally a new approach using a multi-components nonlinear isotropic diffusion applied to color images and morphological flooding is presented. This technique allows the extraction of a set of contour lines which are considered as contour candidates. The selection of valid contours is then done by global or local minimization of an energy functional.

The last topic investigated in this thesis is the quantification of symmetry. Special care is given to its application to almost symmetrical and asymmetrical objects by having an optimization approach. Different techniques are developed to this end, using multi-resolution approaches, genetic algorithms, and self-organizing maps. The data is assumed to be vector-valued, such as are color and texture descriptors. Finally, the ability to detect symmetry axes in noisy environments is proven and illustrated.

These different techniques are applied to the analysis of dermatoscopic images for diagnosis

purpose. This type of images is used by dermatologists to diagnose skin cancer. The efficiency of the segmentation and contour detection techniques combined with the nonlinear isotropic filtering is illustrated. The improved separation between benign and malignant lesions obtained with our symmetry quantification approach, when compared to more classical techniques, is demonstrated using a simple linear classifier. We want to show in this study that the different diagnostic features, like the symmetry, must be fully investigated to design an efficient computer aided diagnosis system.

Chapter 1

Introduction

1.1 Scope of the thesis

I have been very often questioned about my profession and my present activity. Spontaneously, I answered that I was electrical engineer and that I was doing research in image processing. This is at least the answer that I gave in the past because I very quickly realized that this was not intuitively appealing and that a detailed explanation was necessary. Two words present in my answer have a fundamental signification in our society: “electrical” and “image”. We live in an “electrical” society where almost every device, from the most useless to the most useful, contains an electrical component. The disappearance of electricity would mean the greatest chaos that human beings have ever experienced. For the last decade, a particular field of electrical engineering has known an incredible expansion: information technologies. Today we realize that a large part of the electric and electronic devices are information vectors, and in general communication systems. Among the different information media, the most important and powerful one is certainly the image. It stimulates an extremely developed and powerful human sense, namely the sight, and acts directly on our emotions. The eye is a very complex receptor which allows our brain to have a three-dimensional representation of the world that surrounds us. However, since the invention of photographic slides and later the television, images have been used to freeze, select, and even manipulate events that are sources of information and that have not the same spatial and temporal location than we have. We have the tendency to interpret what we see as the exact representation of the world we live in with, very often, a lack of discernment. Therefore, images are certainly the most important media and it has become essential to master the related technologies.

From a physical point of view an image is a signal. This is not true only since human created an electrical signal to transmit the image from a camera to a television receptor via a cable or an antenna, but because an image such as those that we perceive is the result of different electrical phenomena. The most important ones are those which make the world visible, namely light, and which transmit the information to the brain, namely the neural signals. Nowadays, new technologies allow us to visualize what is not intrinsically visible. An information source is transformed into an image, and then we try to understand and to interpret these images. The most impressive examples are certainly the imaging techniques developed for medical applications. Microscopy allows us to access an information level which is only visible with techniques like x-rays, ultra-sounds, magnetic resonance, and even neuro-magnetic resonance for the analysis of neural

activity. Other examples are radars, sonars, infra-red cameras, thermal cameras, etc. Images can also be created from one-dimensional signals, as it is the case in time-frequency analysis. In all these examples a particular type of information has been mapped onto a visible medium.

The notion of image is related to the phenomenon of vision and finally to the faculty to interpret this information. This sentence summarizes precisely the complexity of image processing. In numerous applications it is necessary to extract an information from a lower level of complexity than the image itself but that can be interpreted immediately (feature extraction), or to exploit the capabilities and limitations of the human visual system to reduce the redundancy in the image (compression), to insert an information which is not visible (watermarking), or to recognize more or less complex structures according to visual criteria (pattern recognition).

Finally the link between the electrical engineer and the science of images is absolutely “natural”. What renders this link less obvious to non initiated people is that everybody thinks to know exactly what an image is, because it is related to our most powerful sense.

The main application developed in this thesis is the analysis of dermatoscopic images. A non-invasive microscopy called epiluminescence microscopy (ELM) is used to produce this type of images. They are used for the imaging of skin lesions and the diagnosis of skin cancer. Even if this type of microscopy is of low complexity, it gives access to an information level that is not visible by direct observation, through a magnifying glass for example. Structures located in the lower skin layers become visible and can be interpreted to diagnose certain types of lesions. This technique implies that the user, in this case the physician, learns to detect and understand a type of information he is not used to see. For this kind of applications, image processing techniques can help in visualizing and interpreting this information.

This thesis focuses on a few important aspects of image analysis, namely filtering, color segmentation, contour detection and the extraction of a particular feature: symmetry. This feature is especially interesting because it appears very often in nature, and can be used to recognize objects as well as to reduce the amount of data necessary to store or transmit images. The title contains only the segmentation and symmetry measure parts because they represent two building blocks in a complete image analysis scheme: *object detection* and *feature extraction*. Filtering is necessary to almost every segmentation scheme and constitutes therefore a straightforward topic, while contour detection is close to segmentation in that it allows the detection of object boundaries, even if the approach is different.

1.2 Investigated problems

Figure 1.1 shows a typical image analysis scheme. The acquisition step includes the digitization in the case where photographic slides are used. As the application may be, some of the processing steps may be skipped. In this thesis we are concerned with the topics shown with a solid frame in Fig. 1.1. The investigated problems are image filtering, color science and color space transformations, color segmentation, contour detection, and symmetry analysis. The schemes developed throughout this thesis will be applied to dermatoscopic images, used for skin cancer detection.

The first investigated problem will be the image filtering. This step prepares the image for any further processing, which means that ideally the filtering should only preserve “information” that is necessary to the following processing. For segmentation or contour detection, the filtering stage must remove spurious edges, preserve the location of borders between neighboring regions,

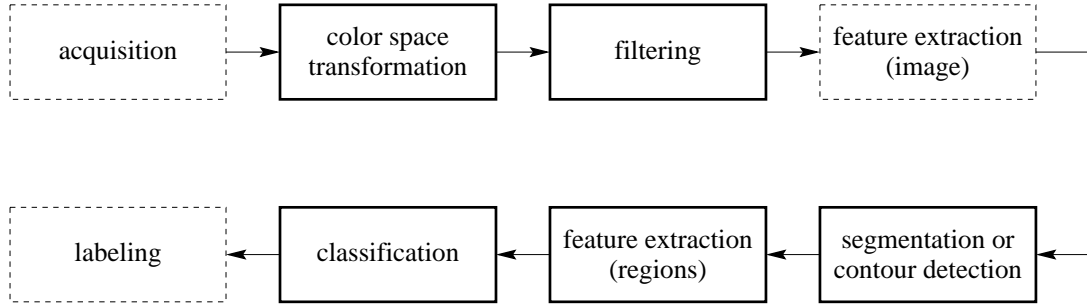


Figure 1.1: Typical image analysis scheme. The solid frames are the topics of this thesis, with application to skin cancer diagnosis.

and finally have a strong noise removal capability.

The second topic concerns color space transformations. It is presented in the second chapter, even if it occupies the second position in the image analysis scheme of Fig. 1.1. The reason is that the filtering should be done on the initial data that is used during the complete scheme while color space transformations will mainly influence processes such as image segmentation.

The segmentation and edge detection problems will be investigated separately. In the former case, we will focus on clustering techniques, and in the latter case, the detection of multiple closed contours will be investigated. The main goal is to separate one or more objects of interest from the background components.

Symmetry will be treated as a feature for object recognition. The problems of symmetry axis detection and symmetry quantification will be investigated for any kind of objects, whether they are symmetric or not. Several coefficients of symmetry must be extracted to allow the later classification and labeling of the objects. In particular, the integration of texture descriptors and color information into the analysis scheme will be essential.

The classification step will be illustrated in the application chapter for the detection of skin cancer. A simple linear classifier will be used both to evaluate our symmetry quantification scheme and to make the comparison with commonly used techniques.

1.3 Initial motivation

The initial motivation to this three years research project was the development of image processing techniques for the computer aided diagnosis of skin cancer. Medical applications require segmentation, recognition, analysis, and classification tasks, which are typical computer vision and image analysis problems. As in most of the developments in image processing (and other sciences), the prior knowledge of what has to be analyzed is exploited to obtain optimal techniques for a specific problem. In the framework of this study it was important to stay as general as possible to make my developments useful for other applications than medical diagnosis.

The development of a complete diagnosis system being a huge work, this initial study aims only at the development of image processing techniques for a limited number of problems, as explained in the previous section.

1.4 Major contributions

This thesis presents the following new contributions:

- A full analysis of diffusion processes and their behavior in noisy environment.
- A new orientation sensitive fuzzy c-means technique for data clustering.
- A complete segmentation scheme based on histogram analysis and data clustering.
- A fast method for the detection of objects with closed and smooth boundaries.
- The symmetry quantification for almost-symmetrical and asymmetrical objects.
- Multi-resolution approaches for computing a *symmetry map*.
- Heuristic search methods for symmetry quantification.
- Proposition of a computer aided diagnosis scheme for the detection of skin cancer.
- A quick and efficient scheme to detect and remove hair in dermatoscopic images.
- Validation of the segmentation and contour detection techniques for skin cancer detection.
- New results for the computer aided diagnosis of malignant melanoma.

1.5 Organization of the thesis

This thesis is organized as follows:

- Chapter 2 starts with an essential image processing field: filtering. It is a preparation phase where all unnecessary information, considered in some sense as noise, must be removed to improve the following processing steps. For example in image segmentation and edge detection applications, it is expected that the spatial and geometrical components of edges and region borders are both preserved. this constitutes an essential condition to the design of efficient filtering techniques.
- Chapter 3 presents the basics of color science and different color space transformations that might be used for later image segmentation. Linear color transformation are presented as well as more complex nonlinear transformation such as the $L^*u^*v^*$ and $L^*a^*b^*$ uniform color spaces. While linear transformations are attractive by their low complexity, the nonlinear transformations give sense to the notion of color difference and will be used later in the segmentation schemes and for the symmetry analysis.
- In Chapter 4, a color segmentation scheme by 2-D data clustering is developed. Different supervised clustering strategies are presented with a particular emphasis on methods based on fuzzy sets theory. The methods give an elegant solution to problems where human perception plays an important role and where vague problem descriptions or vague limits between neighboring solutions are unavoidable. Images showing different shades of a same color are often difficult to segment because there is no precise and unique visual answer to the problem. To take into account cluster orientation during the clustering and partitioning

phases, orientation sensitive fuzzy c-means is introduced. Finally, the valid cluster selection is also investigated. A geometrical solution is proposed to provide both the number of valid clusters in a distribution and the initial values to the cluster centers used in the clustering phase.

- Chapter 5 deals with the contour detection problem. This topic is treated separately because, in opposition to image segmentation that aims at detecting “homogeneous” regions, edge detection focuses on the detection of “strong” transitions. The main goal in this chapter is to develop an edge detector that provides closed and continuous object or region contours. Different gradient-based edge detectors are presented to illustrate the limitations of these classical approaches. The key role played by active contours in the framework of edge detection is then described, as well as a recent technique based on the curve evolution theory. Finally, a fast and reliable method based on nonlinear isotropic diffusion and morphological image flooding is presented.
- Chapter 6 defines the role of symmetry measure in the context of computer vision and gives a review of commonly used techniques. The optimization of a given symmetry measure is investigated and different methods are presented to compute a global maximum and even local maxima. Heuristic search methods based on genetic algorithms and Self-Organizing Maps, as well as multi-scale approaches in the spatial and parameter domains are proposed. The advantage of these different strategies is to allow the quantification of symmetry for almost symmetrical *and* asymmetrical objects.
- Chapter 7 concerns the application of the developed image analysis scheme: a computer aided diagnosis system for skin cancer detection. It is shown in this chapter that the segmentation and contour detection techniques developed in the previous chapters are well suited to skin cancer images. The symmetry quantification scheme developed in Chapter 6 is evaluated and compared to more classical approaches, such as the Karhunen-Loève transform.
- Finally, Chapter 8 discusses the different results, proposes a number of possible extensions, and draws some conclusions.

The different chapters are written in a tutorial manner to give a better insight to the different topics that have been investigated to design the image analysis scheme presented in this thesis. This work being a building block of a much more complex computer aided diagnosis system involving image processing, networking, database management, etc., it is essential that future researchers involved in this challenging project can understand easily and in details what has been done in the framework of this thesis.

Chapter 2

Edge preserving image filtering

2.1 Introduction

Filtering is one of the first processing step in most of the image analysis or computer vision applications and is intended, from a general point of view, to *remove unprofitable information* that may corrupt any following processing. This part of the signal can be seen as a noise component. The goal of this chapter is to open the way to efficient image segmentation and contour detection strategies. What must be achieved here is a *vector-valued image filtering that preserves the geometry of region boundaries*. The term *vector-valued* means that the pixel values are given by vectors, as for color or texture descriptors. The geometry is an important parameter when dealing with image segmentation and edge detection, since they are both strongly related to the “planar” structure in images. It is also essential that the used filter has strong denoising capabilities. Therefore both aspects are very important for the selection of an efficient filter.

At the time where filtering was performed exclusively with electric components, one had to combine them in order to obtain a filter with the expected response [1]. Since nowadays most of the signals are sampled and undergo a numerical processing in a computer or specialized hardware (DSP), the way filters are developed has significantly changed. In digital signal processing, the notion of filtering can be extended beyond the basic idea of convolving the input signal with the filter’s impulse response. The notion of filtering can be extended from the basic idea of removing frequencies to the more generic concept of black box where a linear or nonlinear process transforms the input signal. Thus any signal processing operation may be called filtering. An image segmentation scheme, for example, takes an image as input and outputs a piece-wise constant image where each region contains pixels having same characteristics (color, texture, etc). This operator can be called a *segmentation filter*.

Nonlinear filters have taken a large part in new developments. Especially the development of non-algebraic approaches, using set theory for example, has been fruitful. Median filtering [2], where every sample is replaced by the median value in a given neighborhood, or morphological filters [3], which work on the signal’s morphology are two successful examples of nonlinear filtering. Adaptive filtering [4], linear or nonlinear, was a kind of revolution in this domain. These filters are self-designing in that they rely on a recursive algorithm, which makes it possible for the filter to operate in an environment where prior knowledge of the signal is not available. These filters can be applied to a wide range of fields like communications [5], control [6], biomedical engineering [7]

and so on, with application to identification, inverse modeling, prediction or interference canceling. The adaptive filters are used mostly in 1-D signal processing, which does not minimize their capabilities in 2-D signal processing, especially for image restoration [8] and noise cancellation [9]. But the characteristics of adaptive filters can be better exploited in time-varying signals than in spatial-varying signals where discontinuities are legion.

Another recent approach to signal filtering is based on *time-varying partial differential equations* (PDE) [10, 11]. A typical example is the diffusion of pixel values in an image in the same way than heat is diffused within matter [12]. These kinds of processes are typical 2-D/3-D processes, even though they can be easily applied to 1-D problems. In this chapter we will mainly focus on this type of filters because they constitute a powerful way of conditioning images for segmentation or contour detection purpose. The applications of nonlinear isotropic diffusion are essentially noise removal [13] and edge detection [14, 15].

Generally speaking, a filtering process operates on three components of the image:

- the spatial component,
- the spectral component,
- the geometrical component.

The first one is restricted to scalar images such as gray-level images, while the second one concerns color information or, in a more general sense, vector-valued images (texture descriptors, for example). The geometrical component, which is the planar geometry of all objects present in the image, is very often neglected. A visible effect of geometrical changes due to filtering are rounded corners. Most of the classical filters do not preserve the curvature at strong edges. It is not always crucial to preserve completely a sharp corner, but simply to reduce as much as possible this rounding effect. In the following three sections this phenomenon will be illustrated with commonly used filters: Gaussian, median, and morphological. These filters are used in numerous applications and have proven to be very useful for specific tasks. Figures 2.1 and 2.2 show two test images used in this chapter to illustrate the different filtering techniques. In the first one mainly the spatial and geometrical components can be affected, as it contains geometrical figures with constant curvature (circles), with sharp edges (triangle, square), and different size (thin rectangle, large square). In Fig. 2.2 the test image contains texturized objects (plant, lobster, newspaper, etc), sharp edges (bicycle, sherry, wine, etc), thin (wheel) and large (fruits) objects. Noisy versions are obtained by adding Gaussian white noise to the original images.

In the following sections, different filtering techniques will be reviewed. The theoretical background of diffusion, the different formulations, the behavior of diffusion processes in presence of noise, and some possible extensions will be investigated.

2.2 Gaussian filter

The impulse response of a Gaussian filter is inspired by the *Gaussian* or *normal* distribution, often used in statistics. The following normalized function is used:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\} , \quad (2.1)$$

where μ is the mean or expected value of the Gaussian distribution and σ its standard deviation. This filter is used to smooth out a signal at a degree controlled by σ . Noise, small structures and

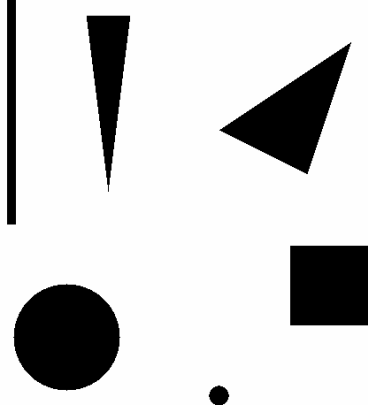


Figure 2.1: Binary test image with geometrical figures.



Figure 2.2: Gray-level test image *bicycle*.

strong edges are affected without differentiation: both the spatial and geometrical components are modified, sharp edges and angles are smoothed out. The Gaussian filter has an infinite length impulse response which must be limited in practice through windowing and μ is set so that the final filter is *causal*. The use of a square window correspond, in the frequency domain, to the convolution with a *sine cardinal* function $\text{sinc}(f) = \sin(\pi T f) / (\pi T f)$, where T is the length of the square window. The “amount of distortion” introduced by the signal windowing will depend on the window size. Note that when it tends to infinity, the sinc function tends to a *Dirac* function.

The Fourier transform of 2.1 is [16]:

$$G(w) = \exp \left\{ -\frac{w^2}{2\sigma_w^2} \right\} \exp(-jw\mu) , \quad (2.2)$$

where $\sigma_w = 1/\sigma$. The second exponential affects only the signal phase. In the Fourier domain, the signal is still Gaussian but with inverse variance: the larger the value of σ the lower the frequency cut-off. The filtered image is obtained by convolution of the image with the filter kernel, which means that the computation time can become very large for high values of σ , rendering necessary the use of fast filters. In [17], such fast filters, whose response is close to that of a Gaussian filter, are proposed. Another way to achieve Gaussian-like smoothing is to use isotropic diffusion, as described in § 2.5.2. Finally, an interesting feature of Gaussian filters is that they belong to the class of *separable filters*, which, for dimensions higher than 1, have an impulse response which is the product of d one-dimensional functions, where d is the space dimension.

The test image shown in Fig. 2.1 has been filtered using a Gaussian low-pass filter with variance $\sigma = 10$ pixels and the result is shown in Fig. 2.3. Both geometrical and spatial components are blurred: edges have a fuzzy position, corners are rounded and pixel values are interpolated from neighboring pixels. When using this filter the image is *smoothed out*. The blurring effect can be compensated by increasing the distance between observer and image, decreasing so the resolution (scale at which details are visible). The Gaussian filtering has been repeated for the real image of Fig. 2.2 with a variance $\sigma = 10$ pixels and the result is displayed in Fig. 2.4. The blurring is very

strong in this example. The resolution has been clearly modified by this filtering. This effect can be best explained from a spectral point of view: part of the frequency band has been removed, meaning that there is an over-sampling of the picture. Down-sampling the image would restore the right resolution and would correspond to reducing the image size.



Figure 2.3: Gaussian low-pass filtering of the image shown in Fig. 2.1.



Figure 2.4: Gaussian low-pass filtering of the image shown in Fig. 2.2.

Gaussian filtering is not suited for edge preserving filtering, even if its denoising capabilities are not bad, but is widely used for scale-space analysis [18, 19] and pyramidal image representation [20].

2.3 Median filter

Median filtering [21, 2] is easier to formulate literally than mathematically: every pixel is replaced by the median pixel value computed in a given neighborhood.

The median filter, which is more selective than the Gaussian filter in that relevant edges can be preserved, has also limitations from a geometrical point of view. The filter size defines how strong edges must be in order to be preserved. In some cases, the effect can be even more destructive than with a Gaussian filter. There is a morphological component in the median filter that can remove objects completely when their size is much smaller than the window size.

From a computation time point of view the window size is a very sensitive parameter. The number of values to sort is $(n \times n)/2 + 1$ from a set of size $n \times n$, where n is the square window size. Other windows may be considered, even though the square window makes the computation easier. A fast algorithm was introduced in [22], where the authors propose to store the gray level histogram obtained from the pixel values present in the window and to update it after each iteration. Another way to speed up the processing is to use *pseudo-median* filtering [23, 2], even though it only approximates a median filter. The idea behind pseudo-median filtering is to compute the median value in a cross-like window, i.e. only the values on a horizontal and vertical line from the pixel location are considered. This approach may be satisfactory in some cases, but in general it gives poor results and introduces sometimes line artifacts.

In Fig. 2.5 the test image shown in Fig. 2.1 has been filtered using a median filter with a square window of size 21×21 . There is no interpolation between neighboring pixels (only existing values can occur) and the spatial component is left unchanged while the geometry has been modified. Corners are rounded and very thin objects simply removed. However, when using a median filter for denoising purpose, the window size is much smaller and the rounding effect is almost imperceptible, compared to what is obtained with a Gaussian filter used for the same application. In Fig. 2.6 a median filtering with a square window of size 5×5 has been computed on the image of Fig. 2.2. This filter shows to have excellent denoising capabilities while preserving edges. However, there is no way, except by modifying the window size, to control the suppression of spurious edges. This filter has a *morphological* component that sets a limit to the size objects can have in order to resist to the filtering process.

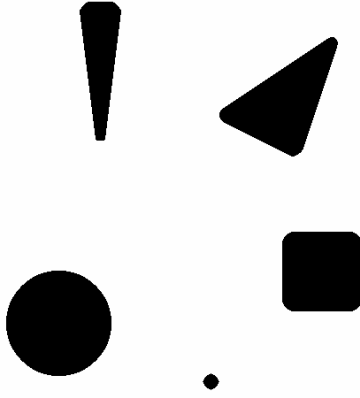


Figure 2.5: Median filtering of the image shown in Fig. 2.1.



Figure 2.6: Median filtering of the image shown in Fig. 2.2.

The next section is dedicated to morphological filters whose main characteristic is that of acting on the shape of the signals.

2.4 Morphological filters

It is worth spending some more time on the mathematical formulation of morphological operators. The filtering techniques issued from this theory are very powerful for artifact removal and will be used in later chapters. Morphological operators are based on *set theory*. A set in mathematical morphology represents the shape of an object in an image. In the case of binary images, sets are members of \mathbb{Z}^2 and each element represents the (x, y) coordinates of a black (or white, depending on the convention) pixel in the image. Gray-scale, color, time-varying components, or any vector-valued information can be included by extending the Euclidean space size. The basic morphological operators, *dilation* and *erosion*, will be presented for gray level images [3].

Let the input image be described by a function $f : \mathbb{Z}^2 \mapsto \mathbb{R}$. Gray-scale dilation is defined as:

$$(f \oplus b)(v, w) = \max\{f(v - x, w - y) + b(x, y) | (v - x, w - y) \in \mathbb{D}_f; (x, y) \in \mathbb{D}_b\},$$

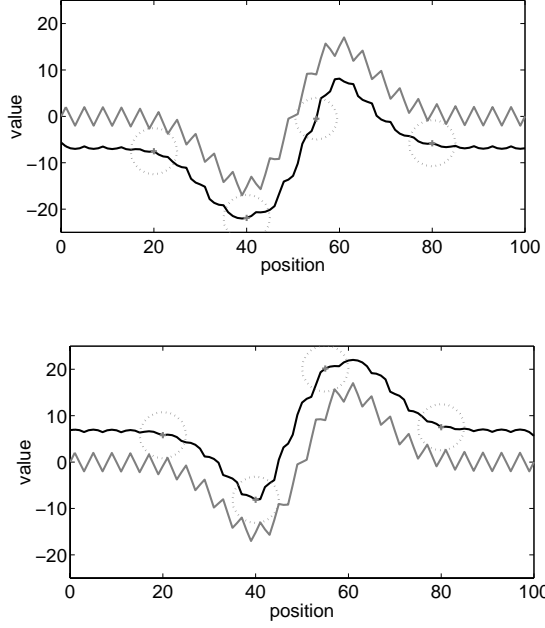


Figure 2.7: Morphological erosion (top) and dilation (bottom) operators. The structuring element has the shape of a half-circle.

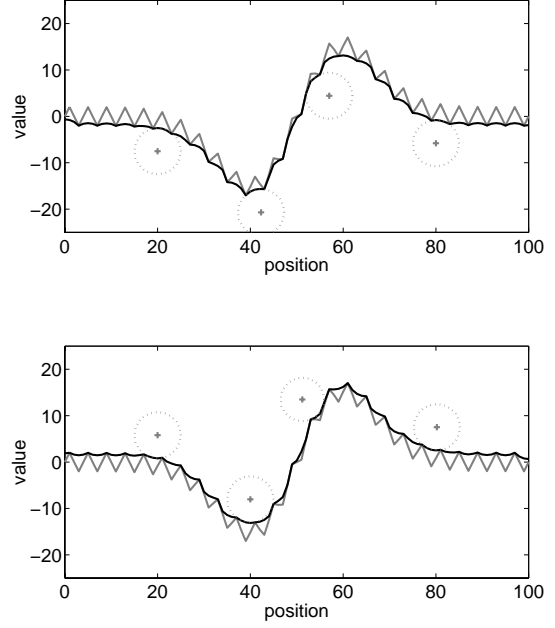


Figure 2.8: Morphological opening (top) and closing (bottom) operators. The structuring element has the shape of a half-circle.

where $b : \mathbb{Z}^2 \mapsto \mathbb{R}$ is a function called *structuring element*, \mathbb{D}_f is the domain of f and \mathbb{D}_b is the domain of b . The structuring element has a key role in this operator: it is *added morphologically* to the image at each pixel location. Figure 2.7 illustrates the gray scale dilation in the 1-D space for clarity purpose.

The contrary of dilation is erosion. This operator is defined as:

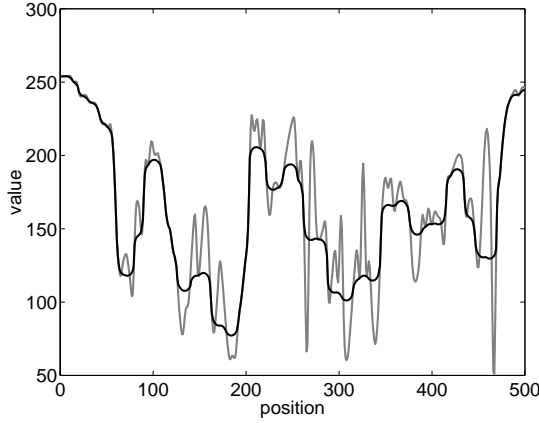
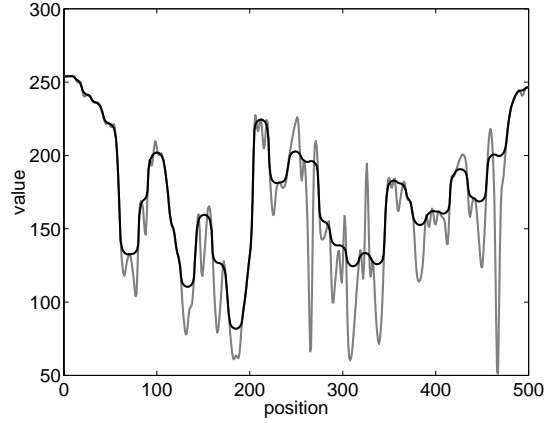
$$(f \ominus b)(v, w) = \min\{f(v + x, w + y) - b(x, y) | (v + x, w + y) \in \mathbb{D}_f; (x, y) \in \mathbb{D}_b\}.$$

In this case the structuring element is *subtracted morphologically* from the image at each pixel location. Figure 2.7 illustrates this operator in the 1-D space.

Depending on the structuring element's shape structures in the image can be suppressed or even enhanced. An immediate application can be denoising, by applying a cross-like structuring element that suppresses impulse noise. Combinations of the dilation and erosion operators provide powerful techniques for image filtering. From morphological erosion and dilation two other operators are first defined: *opening* and *closing*. They are respectively defined as:

$$\begin{aligned} f \circ b &= (f \ominus b) \oplus b, \\ f \bullet b &= (f \oplus b) \ominus b. \end{aligned}$$

The effect of opening is to *preserve holes* and *remove peaks*, while closing *preserves peaks* and *closes holes* according to the structuring element's shape. The structuring element b is *fitted from inside* in the opening case and *fitted from outside* in the closing case. Figure 2.8 illustrates both operators.

Figure 2.9: Morphological filter $(f \circ b) \bullet b$.Figure 2.10: Morphological filter $(f \bullet b) \circ b$.

A morphological filter can be defined as *any combination of morphological operators*. For example $(f \circ b) \bullet b$, opening followed by closing, or $(f \bullet b) \circ b$, closing followed by opening. It must be underlined here that these operators are neither commutative, nor associative or distributive and the filtering operators cited above are not equal. The real power of morphological filters is best illustrated with examples: Figs. 2.9 and 2.10 show a 1-D signal filtered respectively with the two operators given above. Morphological filtering allows the restoration of corrupted signals under the condition that one has prior knowledge of the original signal's morphology in order to choose the ideal structuring element. This condition can become a real drawback. An efficient way to make a morphological filter is to use one of the following operators:

$$f \odot b = \frac{1}{2}[(f \bullet b) + (f \circ b)] ,$$

or

$$f \odot b = \frac{1}{2}[(f \circ b) \bullet b + (f \bullet b) \circ b] ,$$

where the \odot symbol means *f filtered by b*. These combinations remove not only impulse noise but also small oscillations. Figures 2.11 and 2.12 illustrate both filtering operators on a 1-D signal. While the second morphological filter is the most complicated and requires a large number of iterations, the final result follows best the original signal according to the structuring element's shape. The effect on real images is shown in Figs. 2.13 and 2.14. In the latter example the result is more selective and small or thin elements are simply removed.

The real limitation of morphological operators appears in 2-D, where signals not only have an amplitude but also a geometrical component. While morphological filters can be successfully used for selective smoothing, they do not respect the natural shape of objects. Using for example a cylindrical structuring element would round any corner, while using a square structuring element would preserve only 90° corners with the same orientation. A possible way to overcome this limitation would be to use structuring elements sensitive to different angles and to keep the best matching at each pixel location. It is not necessary to show that this approach would be very time consuming.

A more elegant way to achieve a morphological filtering with better geometrical characteristics

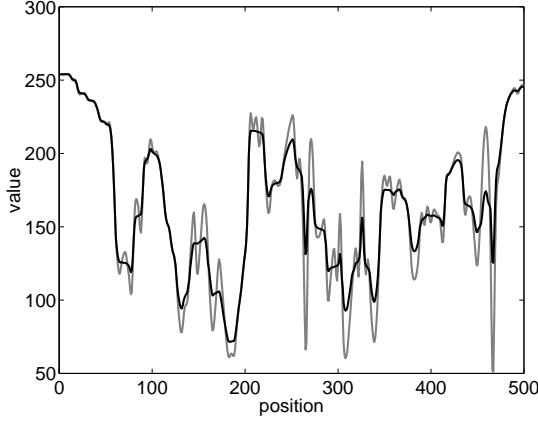


Figure 2.11: Morphological filter $[(f \bullet b) + (f \circ b)]/2$.

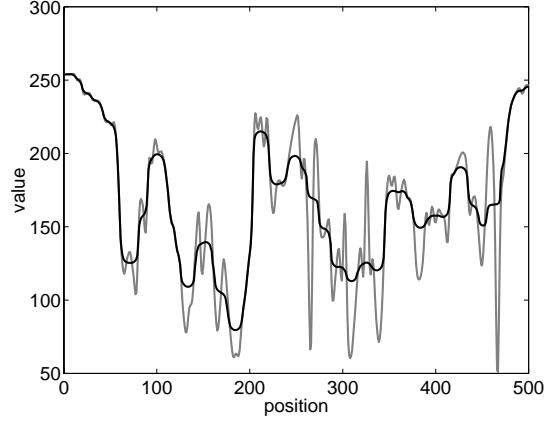


Figure 2.12: Morphological filter $[(f \circ b) \bullet b + (f \bullet b) \circ b]/2$.

is to use *geodesic reconstruction* after a morphological opening. The reconstruction process uses *geodesic dilation* which for gray-scale images is defined by:

$$(f \oplus b)^{(1)}(v, w) = \min\{(f \oplus b)(v, w), f_0(v, w)\}, (v, w) \in \mathbb{D}_f,$$

where f_0 is the reference image, usually the original image, and g is a small structuring element, usually a four (cross) or eight (square) connected element. Geodesic reconstruction is obtained by repeating n times the geodesic dilation $((f \oplus b)^{(n)})$ until idempotency is reached. The geodesic reconstruction is then written:

$$R_g = (f \oplus b)^{(i)}, \text{ with } (f \oplus b)^{(i+1)} = (f \oplus b)^{(i)}.$$

An equivalent operator can be defined for reconstruction after morphological closing which uses *geodesic erosion*. For gray-scale images it is defined as:

$$(f \ominus b)^{(1)}(v, w) = \max\{(f \ominus b)(v, w), f_0(v, w)\}, (v, w) \in \mathbb{D}_f.$$

Figure 2.15 shows an example of geodesic reconstruction after morphological opening. The result suppresses the square shape deformation introduced by the earlier process. These geodesic reconstruction operators improve significantly any filtering process for a modest additional computation time. It is however important to note that for denoising applications the reference image is the noisy image. This renders the use of reconstruction operators contradictory, but for removing small structures they perform very well.

Morphological filters are not only used for pure filtering purpose, but also for the detection of structures having a specific morphology. This characteristic will be illustrated later in Chapter 7. This kind of filters demonstrate the power of digital signal processing where even set theory can be exploited.

The next sections are devoted to diffusion processes in digital image processing. The filters that are issued from the underlying theory are very powerful for edge-preserving applications and justify the title of *edge preserving image filtering*, for denoising as well as for edge detection applications.



Figure 2.13: Morphological filter $[(f \bullet b) + (f \circ b)]/2$ applied to the image shown in Fig. 2.2 with a spherical structuring element.



Figure 2.14: Morphological filter $[(f \circ b) \bullet b + (f \bullet b) \circ b]/2$ applied to the image shown in Fig. 2.2 with a spherical structuring element.

2.5 Diffusion in the framework of image filtering

2.5.1 From physics to image processing

The most natural example of diffusion process is heat transfer inside matter. This physical phenomenon can be mathematically expressed by the following partial differential equations (PDE) [24]:

$$\begin{aligned} \mathbf{q} &= -k \nabla T, \\ c\rho \frac{\partial T}{\partial t} &= -\nabla \cdot \mathbf{q} + f, \end{aligned} \quad (2.3)$$

leading to the following second order elliptic equation:

$$c\rho \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + f.$$

To the heat transfer is associated a flux \mathbf{q} and the whole system must obey to the energy conservation law. The ∇ symbol is the *differential operator nabla*, which is defined as $\nabla = (\partial/\partial x_1, \dots, \partial/\partial x_d)$. The parameter ρ is the density of the medium, k is the thermal conductivity, c is the specific heat capacity, and f the capacity of internal heat sources. The analogy between temperature variation - intuitively we know that the temperature converges towards a smooth distribution - and value¹ variation in images is immediate. The basic formulation is obtained when the medium is supposed to be homogeneous, without sources and with constant conductivity.

In image processing applications the ideal objective is to obtain an image where only *strong* edges are preserved while noise and small structures are smoothed out. The basic idea expressed in the original paper by Perona and Malik [12] was to use diffusion as an edge preserving filtering method. The thermal conductivity is replaced by a conductivity function which adapts the diffusion to the local gradient: decreasing diffusion for increasing gradient. The above diffusion

¹The pixel *value* is what is associated with every pixel position, either a scalar or a vector.

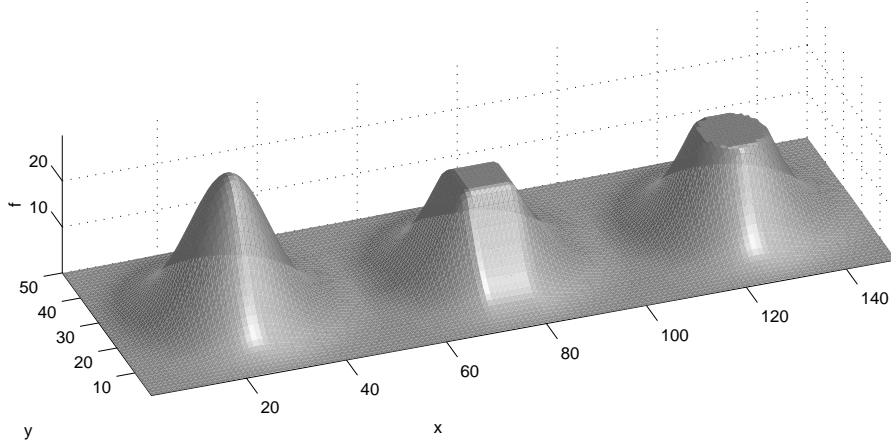


Figure 2.15: Example of geodesic reconstruction: the left shape is the original, the middle shape is obtained after morphological opening with a square structuring element, and the right shape is obtained after geodesic reconstruction using a cross-like structuring element.

equation becomes:

$$\frac{\partial v}{\partial t} = \nabla \cdot (\mathbf{D} \nabla v) ,$$

where $v(\mathbf{x}, t)$ is the signal value at time t and position \mathbf{x} , and \mathbf{D} is a conductivity matrix. The latter defines the type of diffusion:

- if \mathbf{D} reduces to a constant value k then the diffusion is *isotropic*,
- if \mathbf{D} reduces to a nonlinear function $g(\cdot)$ then the diffusion is *nonlinear isotropic*,
- if \mathbf{D} is a tensor whose elements are functions $g_{ij}(\cdot)$ then the diffusion is *anisotropic*.

In most of the literature the nonlinear isotropic diffusion is falsely called anisotropic diffusion. As it will become clear later in this chapter, in the nonlinear isotropic case the diffusion “strength” is controlled by a nonlinear function of the gradient. The direction of diffusion can only be controlled when using tensors, which then justifies the isotropic designation.

2.5.2 The diffusion filter

The real power of diffusion for signal processing applications is exploited in the nonlinear isotropic and anisotropic cases. However, the isotropic model should be discussed first since its behavior is close to that of Gaussian filtering.

Figure 2.16 shows the scale-space [18] representation of a 1-D signal (original is top row). This kind of representation fits our perception of things when the distance to objects defines how we perceive these objects. Details which remain at coarse resolutions can be seen as part of this object’s *building blocks*. It is typically the kind of characteristics one wants to extract in edge detection, where global edges should resist to filtering (scale change) while local edges should progressively disappear. A simple example is the earth in the universe: from the sun we see a dot,

from the moon we access a new level of information which is the earth structure, from an airplane we can see cities, and so on. The distance must be chosen depending what level of information is needed.

Figure 2.17 shows the zero crossings corresponding to the scale-space decomposition of Fig. 2.16. From this representation, which reveals the zero-crossings of the second derivatives at different scales, one can notice that the edge locations deviate from their original positions when the resolution decreases. Tracking valid maxima back to the original signal, especially when maxima merge, becomes a tricky problem. The maxima deviation is even stronger in 2-D signals where the scale-space analysis becomes a three-dimensional problem [19]. Nonlinear isotropic diffusion, presented in the next section, solves this problem by performing an edge preserving blurring. In some applications where classical scale-space analysis gives satisfactory results, isotropic diffusion allows for a progressive fine to coarse representation of the signal without large filter convolutions.

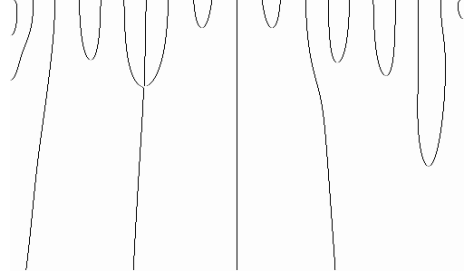


Figure 2.16: Scale-space transform of a 1-D signal.

Figure 2.17: Zero-crossings of the second derivative of scale-space transform.

Let $v(\mathbf{x}, t)$ be the pixel value at time t and position \mathbf{x} . Isotropic diffusion is then driven by the following equation:

$$\begin{aligned} \frac{\partial v}{\partial t} &= k \nabla \cdot \nabla v \\ &= k \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \end{aligned} \quad (2.4)$$

where the right hand term is a weighted Laplacian. The effect of isotropic diffusion is a “value” transfer from pixels with high value to pixels with low value and clearly smoothes out high frequencies: it is a low pass filter. We will suppose that the boundary $\partial \mathbb{D}_v$, where $\mathbb{D}_v \subset \mathbb{Z}^2$ is the image domain, is insulated and that the partial derivative in the direction normal to the image boundary is equal to zero:

$$\begin{aligned} \frac{\partial v}{\partial x} &= 0, \quad x \in \{0, L_x\}, \\ \frac{\partial v}{\partial y} &= 0, \quad y \in \{0, L_y\}, \end{aligned}$$

where the image is defined for $0 \leq x \leq L_x$ and $0 \leq y \leq L_y$. There is no risk of value collapse at the border as it happens when using zero padding, or possible overlapping of image objects close to the border with their reflected or periodical counterpart.

Equation 2.4 describes a linear system because it obeys the following equation:

$$H[k_1 f_1(x, y) + k_2 f_2(x, y)] = k_1 H[f_1(x, y)] + k_2 H[f_2(x, y)] ,$$

where $f_1(x, y)$ and $f_2(x, y)$ are two input signals, and k_1 and k_2 are two real coefficients. Such a system is then completely characterized by its impulse response, i.e. the filter response to a Dirac function:

$$\begin{aligned} \delta(x, y) &= 0 , \forall (x, y) \neq (0, 0) , \\ < \delta, f > &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x, y) f(x, y) dx dy = f(0, 0) . \end{aligned}$$

In order to analyze the filtering capabilities of isotropic diffusion it might be useful to calculate the corresponding transfer function, which is the Fourier representation of the impulse response.

Since we deal with an isotropic transformation (i.e. equal in all directions) we can analyze the diffusion equation in a unique direction. We want to find the solution to the PDE given by:

$$\frac{\partial v}{\partial t} = k \frac{\partial^2 v}{\partial x^2} ,$$

with boundary condition

$$\frac{\partial v}{\partial x} = 0 , x \in \{0, L_x\} .$$

Using Fourier's approach of variable separation, expressing $v(x, t)$ as the product of two functions of x and t respectively, the solution is given by [10]:

$$v(x, t) = \sum_{n=0}^{\infty} C_n \cos\left(\frac{n\pi x}{L_x}\right) \exp\left\{-k \left(\frac{n\pi}{L_x}\right)^2 t\right\} , t \geq 0 .$$

For $t = 0$ a Fourier cosine series representation of $v(x, 0)$ emerges, leading to the following solutions for the coefficients C_n :

$$\begin{aligned} C_0 &= \frac{1}{L_x} \int_0^{L_x} v(x, 0) dx , \\ C_n &= \frac{2}{L_x} \int_0^{L_x} v(x, 0) \cos\left(\frac{n\pi x}{L_x}\right) dx . \end{aligned}$$

With $v(x, 0) = \delta(x)$ it is straightforward that the coefficients C_n are given by:

$$C_0 = \frac{1}{L_x} , \quad C_n = \frac{2}{L_x} , \forall n \in \mathbb{N}^* ,$$

and then the impulse response is a symmetrical function that can be written as an infinite sum of modulated decreasing exponentials:

$$v(x, t) = \frac{1}{L_x} \sum_{n=-\infty}^{\infty} \cos\left(\frac{n\pi x}{L_x}\right) \exp\left\{-k \left(\frac{n\pi}{L_x}\right)^2 t\right\} .$$

A system which uses diffusion is controlled by a unique parameter t that will set the smoothing degree, as σ does in Gaussian filtering. The transfer function corresponding to the above impulse response should give a better insight into the filtering properties of isotropic diffusion. Knowing that the Fourier transform of a cosine function is given by:

$$\mathcal{F}\left[\cos\left(\frac{n\pi x}{L_x}\right)\right] = \frac{1}{2} \left(\delta\left(w_x + \frac{n\pi}{L_x}\right) + \delta\left(w_x - \frac{n\pi}{L_x}\right) \right) ,$$

then the spatial Fourier transform of $v(x, t)$ is given by:

$$\Upsilon(w_x, t) = \frac{1}{2L_x} \sum_{n=-\infty}^{\infty} \exp \left\{ -k \left(\frac{n\pi}{L_x} \right)^2 t \right\} \\ \times \left(\delta \left(w_x + \frac{n\pi}{L_x} \right) + \delta \left(w_x - \frac{n\pi}{L_x} \right) \right) ,$$

which is different from zero only for $w_x = n\pi/L_x$, $n \in \mathbb{Z}$. Finally the transfer function is given by:

$$\Upsilon(w_x, t) = \frac{1}{L_x} \exp \{ -kw_x^2 t \} \sum_{n=-\infty}^{\infty} \delta \left(w_x - \frac{n\pi}{L_x} \right) , \quad (2.5)$$

which is then the output of an *ideal sampler* with a Gaussian function as input. The comparison with Eq. 2.2 leads to the following relationship:

$$\sigma = \sqrt{2kt} .$$

The similarity with a Gaussian low-pass filter is indeed very strong, except that the transfer function of isotropic diffusion is non-zero only at equally spaced frequencies $\Delta f_x = 1/(2L_x)$. When working with discrete signals, the *discrete Fourier transform* (DFT) (with the convention that the frequency range for discrete signal is $[-1/2, 1/2]$) has N samples, which corresponds to the signal length. The interval between successive frequencies is then equal to $\Delta f_x = 1/N$. The discrete representation of the transfer function calculated above, with $L_x = N$, shows therefore a continuous Gaussian function.

The real shape of the transfer function is mainly defined by the numerical approximation to the partial derivatives. The diffusion filter is not exactly a Gaussian filter but has a similar effect. It is an interesting substitute for multi-resolution applications where successive low-pass filtering is needed. Diffusion has the advantage of having well-posed boundary conditions and a finite filter size that depends only on the “quality” of the local gradient approximation.

A basic numerical implementation of the diffusion filter uses *finite difference* approximations to the partial derivatives (see § 2.6.4). For image processing applications, the following finite length filters are used:

$$\frac{v(\mathbf{x}, t + \Delta t) - v(\mathbf{x}, t)}{\Delta t} = k \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} * v(\mathbf{x}, t) + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} * v(\mathbf{x}, t) \right\} ,$$

where $**$ is the two-dimensional convolution. This equation can be rewritten with only one filter, leading to:

$$v(\mathbf{x}, t + \Delta t) = \begin{pmatrix} 0 & k\Delta t & 0 \\ k\Delta t & 1 - 4k\Delta t & k\Delta t \\ 0 & k\Delta t & 0 \end{pmatrix} * v(\mathbf{x}, t) .$$

The effect of using discrete approximations to the partial derivatives can be observed in Figs. 2.18 and 2.19 (1-D case). In the former the theoretical transfer function of the isotropic diffusion filter is plotted. In Fig. 2.19 the evolution over time of the transfer function is plotted when using the finite difference approximation to the diffusion process. The shapes of these functions tend to be Gaussian already after a few iterations. As expected from the theoretical developments, the bandwidth decreases with time.

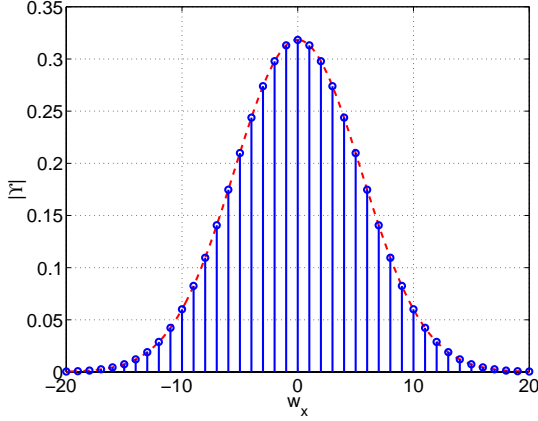


Figure 2.18: Transfer function of the continuous diffusion filter.

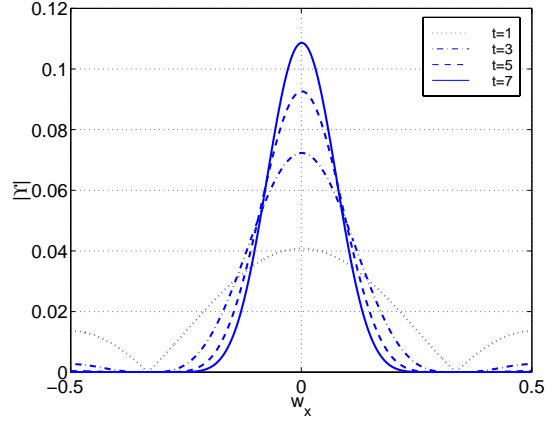


Figure 2.19: Numerical implementation of the diffusion filter using a finite difference approximation. In this example $k = 1$ and $\Delta t = 1$.

2.6 Nonlinear isotropic diffusion

2.6.1 Formulation

The main drawback of Gaussian scale-space analysis for edge detection is that the smoothing does not respect the location of natural edges. Once they have been found, a backward tracking is necessary in order to obtain an exact localization. This procedure is very complicated and long. The main goal of nonlinear isotropic filtering is to overcome this drawback [12]. In nonlinear isotropic diffusion the conductivity coefficient k is replaced by a nonlinear scalar function $g(|\nabla v|)$. The idea is to allow diffusion through edges up to a given limit and to reduce the diffusion for larger edges. The diffusion should tend to zero for very steep edges. In the ideal case this procedure should converge towards a piecewise constant image in which only relevant edges are kept and noise and small structures are smoothed out. The diffusion process is expressed by the following PDE [12]:

$$\frac{\partial v}{\partial t} = \nabla \cdot (g(|\nabla v|) \nabla v) , \quad (2.6)$$

which, defining the scalar flux² $\phi(|\nabla v|) = |\nabla v|g(|\nabla v|)$ [25], becomes:

$$\begin{aligned} \frac{\partial v}{\partial t} &= \nabla \cdot (\phi(|\nabla v|) \frac{\nabla v}{|\nabla v|}) \\ &= \phi(|\nabla v|) \nabla \cdot (\frac{\nabla v}{|\nabla v|}) + \nabla \phi(|\nabla v|) \frac{\nabla v}{|\nabla v|} . \end{aligned} \quad (2.7)$$

Before we try to define any function $g(|\nabla v|)$ it may be interesting to simplify the above equation. Let \mathbf{n} be the normalized vector parallel to the gradient, i.e. $\mathbf{n} = \nabla v / |\nabla v|$. Knowing that

²The scalar gray level flux is analogous to the norm of the heat flux given by Eq. 2.3.

$\partial v / \partial n = |\nabla v|$, the second term of the right part of the previous equation becomes:

$$\begin{aligned} \nabla \phi(|\nabla v|) \frac{\nabla v}{|\nabla v|} &= \frac{\partial}{\partial n} \phi(|\nabla v|) \\ &= \phi'(|\nabla v|) \frac{\partial}{\partial n} (|\nabla v|) \\ &= \phi'(|\nabla v|) \frac{\partial^2 v}{\partial n^2}, \end{aligned}$$

and the divergence term becomes:

$$\begin{aligned} \nabla \cdot \left(\frac{\nabla v}{|\nabla v|} \right) &= \frac{1}{|\nabla v|^3} \left\{ \frac{\partial^2 v}{\partial x^2} \left(\frac{\partial v}{\partial y} \right)^2 - 2 \frac{\partial^2 v}{\partial x \partial y} \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial^2 v}{\partial y^2} \left(\frac{\partial v}{\partial x} \right)^2 \right\} \\ &= \frac{1}{|\nabla v|} \frac{\partial^2 v}{\partial n_{\perp}^2}, \end{aligned} \tag{2.8}$$

where \mathbf{n}_{\perp} is the unit vector orthogonal to \mathbf{n} . Equation 2.7 can now be decomposed into two orthogonal components and rewritten in the following way:

$$\begin{aligned} \frac{\partial v}{\partial t} &= g(|\nabla v|) \frac{\partial^2 v}{\partial n_{\perp}^2} + \phi'(|\nabla v|) \frac{\partial^2 v}{\partial n^2} \\ &= \phi(|\nabla v|) \kappa + \phi'(|\nabla v|) \frac{\partial^2 v}{\partial n^2}. \end{aligned} \tag{2.9}$$

Equation 2.9 reveals two contributions, one depending on the second derivative in direction of the gradient and a second depending on the second derivative normal to the gradient, i.e. parallel to the edge. Note that:

$$\kappa = \frac{1}{|\nabla v|} \frac{\partial^2 v}{\partial n_{\perp}^2} = \nabla \cdot \left(\frac{\nabla v}{|\nabla v|} \right) = \text{curv}(\mathbf{x}),$$

where $\text{curv}(\mathbf{x})$ is the level sets curvature³ at position \mathbf{x} and reveals the geometrical component of the diffusion process.

Now, ideally, we know from the final equation that the function $g(|\nabla v|)$ should be defined in such a way that diffusion occurs only in a direction parallel to edges. However, diffusion should not be completely stopped through edges, but decrease with increasing edge slope. This kind of behavior is important in order to smooth out small structures that may have gradient values close to that of relevant edges. The latter belonging to larger regions, a small amount of diffusion should have a limited effect.

Depending on the sign of $g(|\nabla v|)$ one has respectively forward (smoothing) and backward (sharpening) diffusion, the latter acting like a shock filter [26] (see Section 2.8). Actually backward diffusion is an ill-posed problem⁴: additional noise may be regarded as new edges and enhanced while it is expected that noise is removed. A possible solution is the use of a gradient estimator $|\nabla_{\sigma} v| = |\nabla(G_{\sigma} ** v)|$, where G_{σ} is a Gaussian function with variance σ and $**$ the two-dimensional convolution. This solution gives weak results and poses the problem of choosing an appropriate σ , which, in turn, depends on the *amount* of noise. This kind of approach is also contradictory to the initial goal of nonlinear isotropic diffusion: overcome the weakness of Gaussian filtering. A detailed analysis leading to the formulation of conductivity functions leading to well-posed systems can be found in [25, 27]. In the next paragraph possible conductivity functions will be presented and their effect on the diffusion process illustrated.

³The level set is a curve defined by the set of points $\{\tilde{\mathbf{x}} \mid v(\tilde{\mathbf{x}}) = v(\mathbf{x})\}$.

⁴A well-posed problem is supposed to give close results for close initial conditions, i.e. there are no multiple solutions.

2.6.2 Conductivity functions

From the conductivity function decomposition given by Eq. 2.9 it becomes clear that the backward diffusion case is strongly noise sensitive. This edge enhancement will work differently for *close* images: the same image corrupted with different amounts of noise for example, will diffuse in a different way in each case.

The two following conditions can be derived to guarantee that the choice of a conductivity function gives a well-posed problem [25, 27]:

$$\lim_{|\nabla v| \rightarrow \infty} \phi(|\nabla v|) \neq 0, \quad (2.10)$$

$$\phi'(|\nabla v|) \geq 0 \text{ for all } |\nabla v| \geq 0, \quad (2.11)$$

The second condition has already been discussed in § 2.6.1: it clearly forbids backward-diffusion. The first condition on the flux $\phi(|\nabla v|)$ avoids zero gray level flux for high gradient values.

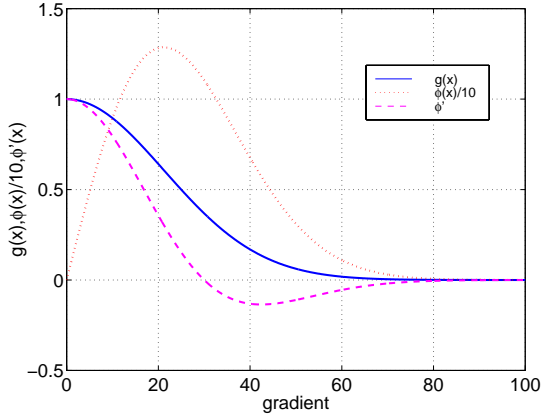


Figure 2.20: Exponentially decreasing Perona-Malik conductivity function. This figure plots the function given by Eq. 2.13, the associated flux and its first derivative.

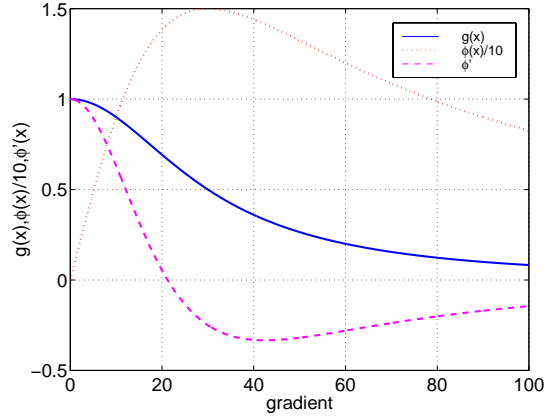


Figure 2.21: Inverse decreasing Perona-Malik conductivity function. This figure plots the function given by Eq. 2.12, the associated flux and its first derivative.

Perona and Malik proposed in their original paper the two following conductivity functions [12]:

$$g(|\nabla v|) = \exp \left[- \left(\frac{|\nabla v|}{k_0} \right)^2 \right], \quad (2.12)$$

$$\phi'(|\nabla v|) = g(|\nabla v|) \left(1 - \frac{|\nabla v|^2}{k_0^2} \right),$$

and

$$g(|\nabla v|) = \frac{k_0^2}{k_0^2 + |\nabla v|^2}, \quad (2.13)$$

$$\phi'(|\nabla v|) = g^2(|\nabla v|) \left(1 - 2 \frac{|\nabla v|^2}{k_0^2} \right),$$

where k_0 is a threshold value. Both conductivity functions proposed by Perona and Malik are ill-posed. Functions 2.12 and 2.13 give $\lim_{|\nabla v| \rightarrow \infty} \phi(|\nabla v|) = 0$ and do not satisfy 2.10. They are plotted in Figs. 2.20 and 2.21 respectively.

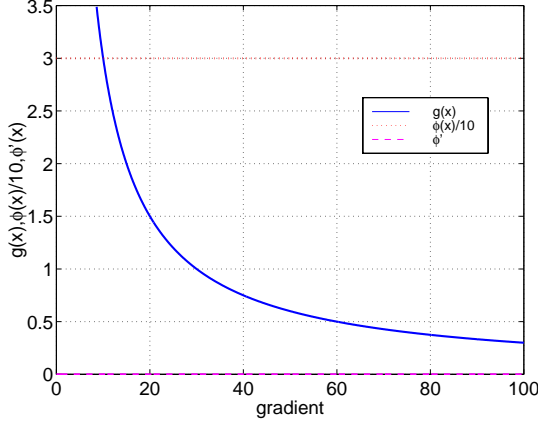


Figure 2.22: Well-posed inverse decreasing conductivity function. This figure plots the function, the associated flux and its first derivative.

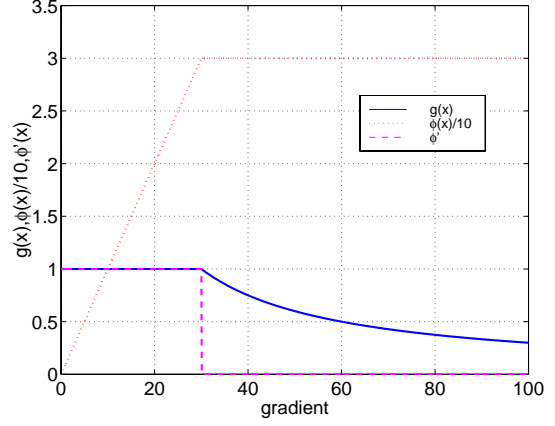


Figure 2.23: The conductivity function plotted in Fig. 2.22 has been modified to a piecewise linear function given by Eq. 2.14.

Let us focus on a third function proposed in [25]:

$$\begin{aligned} g(|\nabla v|) &= \begin{cases} 1 & \text{if } |\nabla v| \leq k_0, \\ k_0/|\nabla v| & \text{if } |\nabla v| > k_0. \end{cases}, \\ \phi(|\nabla v|) &= \begin{cases} 1 & \text{if } |\nabla v| \leq k_0, \\ 0 & \text{if } |\nabla v| > k_0. \end{cases}. \end{aligned} \quad (2.14)$$

This function satisfies both conditions 2.10 and 2.11, and is therefore well-posed. This conductivity function is analogous to the $1/|\nabla v|$ function plotted in Fig. 2.22 with a limited value for low gradients. When the gradient is close to zero, the numerical approximation of the flux may have large errors due to the limited precision, which justifies the piece-wise continuous function plotted in Fig. 2.23. Using Eq. 2.14 in Eq. 2.9 the following PDE is obtained:

$$\frac{\partial v}{\partial t} = \begin{cases} |\nabla v| \text{curv}(\mathbf{x}) + \partial^2 v / \partial n^2 & \text{if } |\nabla v| \leq k_0, \\ k_0 \text{curv}(\mathbf{x}) & \text{if } |\nabla v| > k_0. \end{cases}$$

With this conductivity function the diffusion depends only on the local curvature for high gradients, while the diffusion for small gradients tends to be isotropic. Intuitively this behavior fits exactly our vision of an edge and geometry preserving filter. Finally it must be also underlined that the computation time is much smaller for a $1/x$ function than for exponentials.

In the next paragraph the performances of these different conductivity functions will be illustrated.

Figure 2.24 shows the diffusion process after 50, 500, and 1000 iterations, and the bottom right image shows the difference between the original image and the 1000 iterations diffusion. It

appears clearly that noise and textural components have been removed first, while the *important* edges are preserved. Some strong borders are also visible in the difference image, which can be explained by the small but existing diffusion through edges with higher gradient (never null). This characteristic could be seen as a drawback, while it is a real advantage: peaks of noise, having high gradient but covering a small surface, are suppressed as well, while larger regions with high gradient are almost unchanged. Continuing the diffusion would decrease progressively the scale. This example illustrates two aspects of nonlinear isotropic diffusion: when used for denoising purposes, local or global noise estimators can be used for setting the threshold value k_0 [12] and the stopping criterion. When the diffusion is used to remove details up to a certain scale, parameter tuning becomes more tricky. Some small structures may have large gradient whose value is close to that of valid edges. What makes then the power of the diffusion is that the geometric size plays an important role (there is always some diffusion, even very small). But this ability can only be exploited with an efficient stopping criterion. Section 2.7 will give some insights into these problems.

A diffusion example with the Perona-Malik conductivity function 2.12 is shown in Fig. 2.29. The result is acceptable, even if the function is ill-posed. However, small threshold variations result in perceptible changes. This will be illustrated in the next paragraph when the image is corrupted by noise.

2.6.3 Behavior in noisy images

It is interesting to see if the same image corrupted by noise will lead to the same result as the original image, without changing the threshold value k_0 and the number of iterations. For the well-posed conductivity function the results are clear. Figure 2.26 shows the result when diffusing the image of Fig. 2.25: edges and corners are preserved, but it is clear that continuing the diffusion process would result in a deterioration of both components.

Figure 2.28 is obtained after diffusion of the image shown in Fig. 2.27. The difference with the original result is very small compared to the distortion introduced by the added noise. Figure 2.28 can be compared to the results shown in Fig. 2.24: the difference is almost imperceptible. On the contrary, Figs. 2.29 and 2.30 show the diffusion result respectively without and with noise when using the Perona-Malik exponential conductivity function. In the former case the diffusion performs well, showing a clear edge enhancement for large gradients. The result can be compared to Fig. 2.24. In the noisy case however the diffusion is strongly affected by the presence of noise: the test image of Fig. 2.27 goes through the diffusion process without perceptible improvement. The only way to force the diffusion is to increase the value of k_0 , which results in a higher blurring.

It is very important when using the Perona-Malik model to have a very efficient noise estimator in order to set the right threshold value k_0 . Figures 2.31 and 2.32 show the *peak signal-to-noise ratio* (PSNR) evolution for ill-posed and well-posed conductivity functions, and for different starting PSNR values. The original and noisy images are diffused in parallel and the PSNR measured at every iteration. The PSNR measure is defined as:

$$\text{PSNR} = 10 \log_{10} \left(\frac{(N_q - 1)^2}{\text{MSE}} \right),$$

where MSE is the *mean square error* between the original and noisy image and N_q is the number of quantification levels, usually 256 in gray level images.

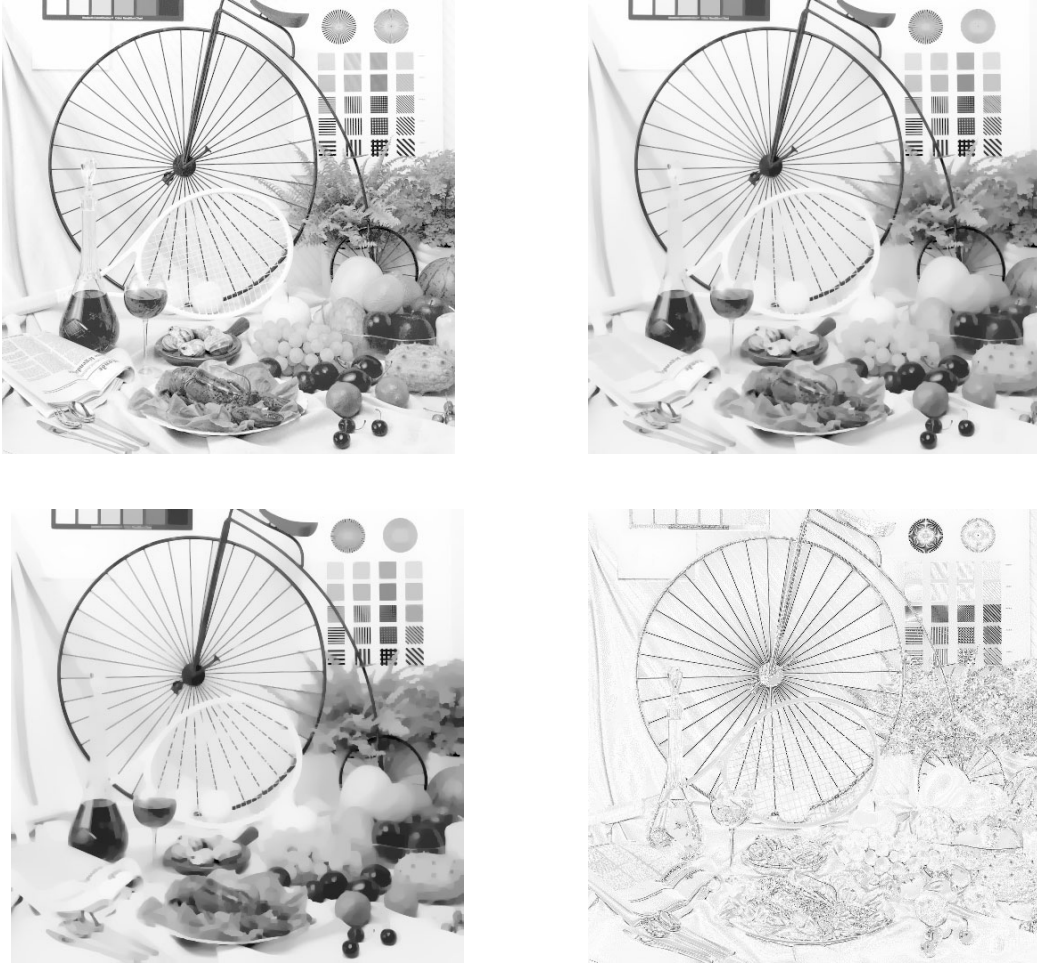


Figure 2.24: Example of well-posed nonlinear isotropic diffusion of *bicycle*. The conductivity function given by Eq. 2.14 has been used. From left to right and top to bottom, this figure shows the result after 50, 500, and 1000 iterations. The last image shows the difference between the original image and the result after 1000 iterations.

The curves shown in Figs. 2.31 and 2.32 correspond to the variation in time (iterations) of the PSNR in noisy environment. The different curves, from bottom to top, have additional Gaussian noise with zero mean and decreasing variance. In the former example, the conductivity function is the decreasing exponential proposed by Perona-Malik, while in the latter case it is given by Eq. 2.14. These two figures clearly demonstrate the lower sensitivity to noise when the conductivity function is well-posed.

2.6.4 Numerical implementation

The numerical implementation is an important aspect of nonlinear isotropic diffusion and in general for PDEs driven processes. Both the spatial and temporal sampling grids must be taken

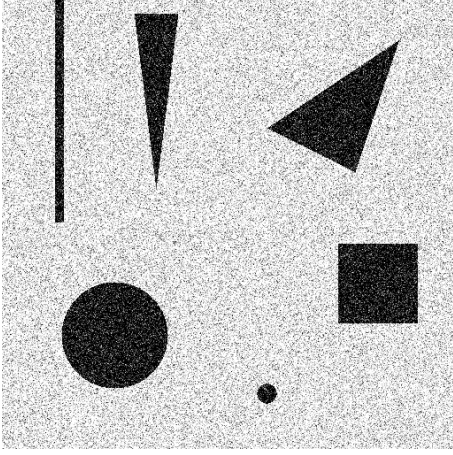


Figure 2.25: Binary test image corrupted with zero mean Gaussian noise.

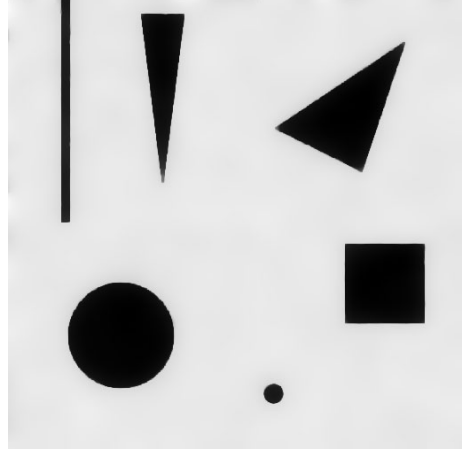


Figure 2.26: The image shown in Fig. 2.25 after 500 iterations of nonlinear isotropic diffusion. The conductivity function is given by Eq. 2.14.

into account in order to minimize the approximation error and to avoid oscillations. The finite difference approximation to derivatives is obtained by considering the two following Taylor series decompositions [28] :

$$v(x+h) = v(x) + hv'(x) + \frac{1}{2}h^2v''(x) + \frac{1}{6}v'''(x) + \dots, \quad (2.15)$$

$$v(x-h) = v(x) - hv'(x) + \frac{1}{2}h^2v''(x) - \frac{1}{6}v'''(x) + \dots. \quad (2.16)$$

The addition of these two terms gives $v(x+h) + v(x-h) = 2v(x) + h^2v''(x) + O(h^4)$, where $O(h^4)$ denotes terms containing fourth and higher powers of h and represents the approximation error. Then, one can write:

$$v''(x) \approx \frac{1}{h^2}\{v(x+h) - 2v(x) + v(x-h)\}. \quad (2.17)$$

The first derivative can be obtained by subtracting Eq. 2.16 from Eq. 2.15 and neglecting terms of order higher than h^3 :

$$v'(x) \approx \frac{1}{2h}\{v(x+h) - v(x-h)\}. \quad (2.18)$$

It is also called *central difference approximation*. Other formulations can be used, such as *forward-difference* or *backward-difference*, where respectively the $v(x-h)$ and $v(x+h)$ terms in Eq. 2.18 are replaced by $v(x)$. The 2-D extension is immediate for derivatives of the same variable. The second order partial derivative $\partial^2 v / \partial x \partial y$ is given by:

$$\frac{\partial^2 v}{\partial x \partial y} \approx \frac{1}{4h^2}\{v(x+h, y+h) - v(x-h, y+h) - v(x+h, y-h) + v(x-h, y-h)\}. \quad (2.19)$$

This very simple formulation for approximating partial derivatives has the advantage of being very fast to process, but it may sometimes lead to large errors. In the case of nonlinear isotropic diffusion, simulations showed that this approximation gives satisfactory results in a well-posed

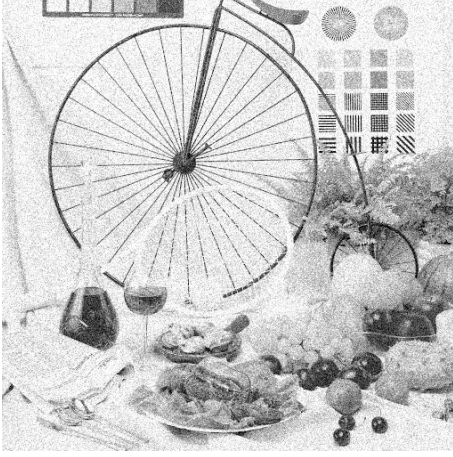


Figure 2.27: *Bicycle* corrupted with zero mean Gaussian noise.



Figure 2.28: The image shown in Fig. 2.27 after 1000 iterations of nonlinear isotropic diffusion. The conductivity function is given by Eq. 2.14.

forward (and slow) diffusion case. When backward-diffusion occurs, or in the presence of any edge enhancing term, more sophisticated models must be used.

The next numerical approximation is based on hyperbolic conservation laws and the theory of viscosity solutions [29]. The numerical approximation of Eq. 2.6 for a sampling grid (i, j) is given by:

$$\begin{aligned} v_{ij}(n+1) &= v_{ij}(n) + \frac{\Delta t}{h} \left(\Delta_-^x \{g(|\nabla^x v_{ij}(n)|)\} \Delta_+^x v_{ij}(n) + \Delta_-^y \{g(|\nabla^y v_{ij}(n)|)\} \Delta_+^y v_{ij}(n) \right), \\ |\nabla^x v_{ij}(n)| &= \sqrt{(\Delta_+^x v_{ij}(n))^2 + \text{minmod}^2(\Delta_+^y v_{ij}(n), \Delta_-^y v_{ij}(n))}, \\ |\nabla^y v_{ij}(n)| &= \sqrt{(\Delta_+^y v_{ij}(n))^2 + \text{minmod}^2(\Delta_+^x v_{ij}(n), \Delta_-^x v_{ij}(n))}, \end{aligned}$$

where Δt and h are the time step size and space grid size respectively. The $\Delta_+^x v_{ij}$, $\Delta_-^x v_{ij}$, $\Delta_+^y v_{ij}$, and $\Delta_-^y v_{ij}$ differences are the different backward $(-)$ and forward $(+)$ differences for x and y . The minmod operator is defined as:

$$\text{minmod}(a, b) = \begin{cases} \min(a, b) & \text{if } a > 0 \text{ and } b > 0, \\ \max(a, b) & \text{if } a < 0 \text{ and } b < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2.20)$$

and the step size restriction for stability is given by $\Delta t/h^2 \leq 0.25$.

The equations given above constitute a *total variation preserving approximation* (TVP) to the diffusion PDE. The proper norm for images is the TV norm (and not the L_2 norm) which is essentially a L_1 norm of derivatives [29]. Intuitively the numerical equations given above for the evaluation of derivatives include a lot of tests in order to avoid sudden sign changes that would result in local oscillations. See Section 5.5 and [30] for additional details.



Figure 2.29: Example of nonlinear isotropic diffusion using the conductivity function given by Eq. 2.12.

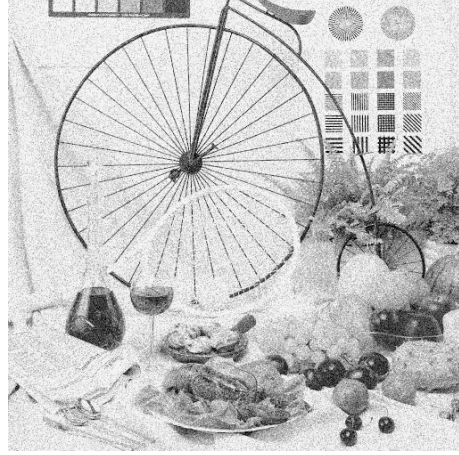


Figure 2.30: Example of nonlinear isotropic diffusion in a noisy environment using the conductivity function given by Eq. 2.12.

2.7 Extended diffusion schemes

2.7.1 Introduction

The diffusion processes presented so far are basic formulations. A number of extensions and improvements can be proposed, which respond to specific requirements. Tensor driven anisotropic diffusion is very interesting to illustrate the directionality of anisotropic diffusion. An automatic setting of k_0 may be needed for some applications. Time-varying conductivity functions would allow a combined smoothing and edge enhancement. The *pilot* (a down-sampled version of the image) for scale dependent diffusion is very useful for texturized images where scale controls completely the filtering process. All these different aspects of anisotropic diffusion are presented in this section.

2.7.2 Anisotropic diffusion

Up to now two diffusion schemes have been presented, namely the isotropic and the nonlinear isotropic ones. In the former diffusion occurs equally in all directions, whereas in the latter the diffusion “strength” is controlled by the conductivity function while the direction of diffusion remains isotropic. To be complete the anisotropic diffusion which reaches a higher level of complexity and allows for a directional diffusion will be presented in this paragraph.

In anisotropic diffusion the conductivity function becomes a tensor whose elements are functions of the gradient. The diffusion equation is:

$$\frac{\partial v}{\partial t} = \nabla \cdot (\mathbf{D}(\nabla v) \nabla v) .$$

The separation into two orthogonal contributions is not straightforward. However, the tensor \mathbf{D} is intended to control the direction and speed (or amount) of diffusion. This tensor should

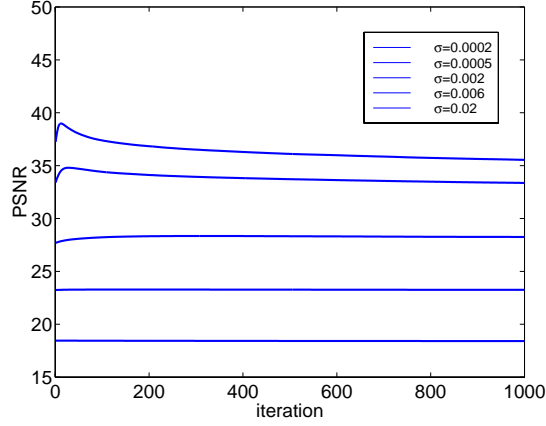


Figure 2.31: Variation in time (iterations) of the PSNR in noisy environment. The conductivity function is the decreasing exponential proposed by Perona-Malik.

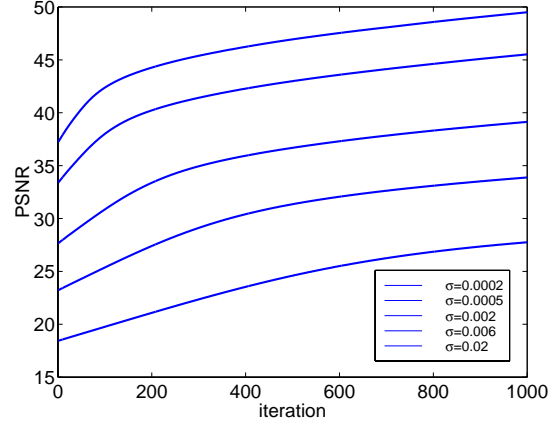


Figure 2.32: In this example the (well-posed) proposed conductivity function given by - Eq. 2.14 is used.

reflect local structure. Let us define the *structure tensor* by the following equation [27]:

$$\begin{aligned} \mathbf{S}(\nabla v) &= \mathbf{S}(\nabla(G_\sigma * v)) \\ &= \nabla_\sigma v \otimes \nabla_\sigma v, \end{aligned}$$

where \otimes denotes the tensor product. $\mathbf{D}(\nabla v)$ is a 2×2 symmetrical and semi-positive definite matrix. Therefore its eigenvectors are orthonormal and the eigenvalues positive. When σ tends to zero, the two eigenvectors tend to be parallel and orthogonal to the gradient respectively. The corresponding eigenvalues are the gradient amplitude and zero. In general, the two eigenvalues (let us call them $\mu_1 \geq \mu_2$) describe the local structure. Flat areas give $\mu_1 = \mu_2 = 0$, straight edges give $\mu_1 \gg \mu_2 = 0$, and corners give $\mu_1 \geq \mu_2 \gg 0$.

In order to control the diffusion, \mathbf{S} will not be used directly with the diffusion equation. It is desirable that the tensor \mathbf{D} has the same eigenvectors as \mathbf{S} , but different eigenvalues, thus controlling the diffusion in both directions. Let the two eigenvalues of tensor \mathbf{D} be given by:

$$\begin{aligned} \lambda_1 &= g(\mu_1) \quad \lambda_1 \in [0, 1], \\ \lambda_2 &= \sqrt{1 - \lambda_1^2}. \end{aligned}$$

The diffusion in direction of eigenvector \mathbf{v}_1 is controlled by a conductivity function in the same manner than for the anisotropic diffusion of Section 2.6. The product $\mathbf{D}(\nabla v)\nabla v$ is obtained from:

$$\mathbf{D}(\nabla v)\nabla v = \lambda_1(\nabla v \cdot \mathbf{v}_1)\mathbf{v}_1 + \lambda_2(\nabla v \cdot \mathbf{v}_2)\mathbf{v}_2.$$

Figures 2.33 and 2.34 show the diffusion result when decomposed into its normal and parallel components respectively. In the former case the *creamy* aspect results from the gray-level flow along the level-sets. In the parallel case, the high value of k used to obtain this result implies a stronger smoothing.



Figure 2.33: Result obtained when the diffusion is only allowed in a direction normal to the gradient.



Figure 2.34: Result obtained when the diffusion is only allowed in a direction parallel to the gradient .

The real improvement brought by tensor driven diffusion is the precise control of the diffusion direction. The use of a local gradient estimator with variable size reflects the local orientation instead of the punctual one. However This formulation proved to be impractical due to its complexity. The process is slowed down by the tensor computation and the eigenvector/eigenvalue decomposition. For an edge preserving filtering the less complicated diffusion of Section 2.6 should be used, including the suggested extensions described in the following paragraphs.

2.7.3 Automatic setting of k_0 and stopping criterion

In this paragraph the role of the threshold value k_0 used in most of the conductivity functions and the definition of a stopping criterion will be shortly discussed. Experiments have shown that in most applications a threshold value can be found empirically that will work well, especially when a uniform space is used (see § 2.7.6). In this case, setting a numerical limit between smooth and strong edges has a visual meaning, even if there is always a part of subjectivity in the definition of uniform spaces.

More sophisticated approaches can be used, based on a noise estimation (see for example [31]). These definitions are restricted to denoising applications while they are not adapted for scale-dependent smoothing, where structures may have high internal gradients. In [12] the authors propose to use the noise estimator given in Canny's paper on edge detection [32] (see Section 5.3). A histogram of the gradient absolute values is computed for the whole image and k_0 is set to be equal to 90 % of its integral. The value of k_0 is recomputed at every iteration. With this approach the author assumes that the gradient corresponding to noise has a Gaussian distribution while the gradient corresponding to step edges is large and occurs very infrequently. Therefore the low percentile of the histogram should determine mainly the noise energy. However when the noise component correspond to small structures, this model does not fit.

Defining the *mean squared diffusion* (MSD) as:

$$\text{MSD} = \frac{1}{L_x L_y} \int_0^{L_x} \int_0^{L_y} \left(\frac{\partial v}{\partial t} \right)^2 dx dy ,$$

a threshold can be set on this measure in order to have a stopping criterion. This approach is again best implemented when using a perceptually-uniform space, such as $L^*u^*v^*$, leading to a measure of total color difference between successive iterations.

For scale-dependent image filtering, where only details at a given scale must be smoothed out, the use of a *diffusion pilot* to control the conductivity function is introduced in the next paragraph.

2.7.4 Pilot for scale dependent diffusion

Beside denoising, anisotropic diffusion is essentially interesting for *scale-dependent* smoothing: the diffusion is not only controlled by the gradient at full size but also by the gradient at a smaller scale. The idea is to inhibit the standard conductivity function for regions being homogeneous at a given scale, even if at full scale they contain texture information. The drawback of using anisotropic diffusion on high PSNR images is that it is difficult to smooth out small structures while keeping region borders unchanged. Actually the gradient within small structures is often as high or even higher than that of large and important regions.

Let us define the *diffusion pilot* as a down-scaled version of the original image. The gradient value in this image will define if a pixel is located in a sensitive region (close to a border) or in an homogeneous region. Equation 2.6 becomes:

$$\frac{\partial v}{\partial t} = \nabla \cdot (g_s(|\nabla v^{(1)}|, |\nabla v^{(i)}|) \nabla v) ,$$

where $v^{(i)}$ is the image at scale i , where the image is supposed to be down-sampled by a factor 2^i . The following conductivity function is proposed: $g_s(|\nabla v^{(1)}|, |\nabla v^{(i)}|)$:

$$g_s(|\nabla v^{(1)}|, |\nabla v^{(i)}|) = g((1 - \exp\{-\beta |\nabla v^{(i)}|^m\}) |\nabla v^{(1)}|).$$

Both parameters β and m are used to set the limit between high and low gradients, and the transition strength. Once again the use of uniform spaces is very useful to give these values a visual meaning.

Figure 2.35 illustrates the use of a down-sampled image to control the diffusion. The *pilot* image (bottom right corner) sets the scale limit at which details should remain. The dark gradient values show regions labelled as being sensitive and where diffusion should occur as a function of the real gradient. Light gradient values show homogeneous regions (at the given scale) where the conductivity function is inhibited and the diffusion tends to be high.

The diffusion is only performed on the original image because the down-scaled version reveals only objects and boundaries that must be preserved but with an imprecise edge location. The edges will be revealed by the diffusion process.

2.7.5 Time varying conductivity function

In § 2.6.2 the effect of backward diffusion has been discussed. While it may clearly degrade the diffusion process and make the image converge to a local steady state, it would be very interesting to take advantage of the enhancement capability of backward diffusion. The reason is that when



Figure 2.35: Use of a down-sampled image to control the diffusion: the *pilot* image (bottom right corner) sets the scale at which details should be preserved.

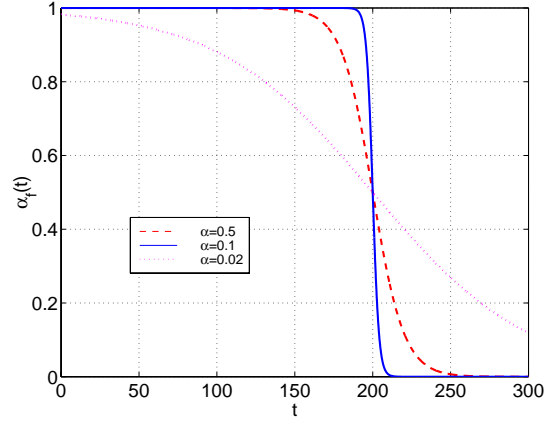


Figure 2.36: Sigmoid function given by Eq. 2.21 for different parameters: T is the transition value (at $x = 0.5$) and α controls the transition speed.

small edges and spurious details have been removed, strong edges should be enhanced in order to obtain an almost piecewise constant image. This would meet the ideal objective of anisotropic diffusion. A simple way to achieve this result is to apply, after a number of iterations with a well-posed conductivity function, a *shock filter* (see Section 2.8).

A natural choice for combining a well-posed diffusion with an edge enhancing component would be to use weighted diffusion, where one smoothing and one enhancing component are combined. The diffusion equation changes to:

$$\begin{aligned} \frac{\partial v}{\partial t} &= \alpha_f(t) \nabla \cdot (g_1(|\nabla v|) \nabla v) + \alpha_b(t) \nabla \cdot (g_2(|\nabla v|) \nabla v) \\ &= \nabla \cdot (\alpha_f(t) g_1(|\nabla v|) + \alpha_b(t) g_2(|\nabla v|)) , \end{aligned}$$

where $g_1(|\nabla v|)$ allows only forward diffusion and $g_2(|\nabla v|)$ backward diffusion for high gradient values. The weights $\alpha_f(t)$ and $\alpha_b(t)$ control the contribution of both diffusion components, with the constraint that backward diffusion is restricted to a short period at the end of the diffusion process. In order to have normalized values and correlated weights, one should have $\alpha_{f,b} \in [0, 1]$ and $\alpha_b(t) = 1 - \alpha_f(t)$ (or more general $\alpha_b(t) = (1 - \alpha_f(t))^m$). A possible choice for $\alpha_f(t)$ is to use a *sigmoid* function :

$$\alpha_f(t) = \left(1 - \frac{1}{1 + \exp\{-\alpha(t - T)\}} \right) , \quad (2.21)$$

where T controls the transition time from forward (smoothing) to backward (sharpening) diffusion. The scalar α controls the transition strength (smooth or sharp). Figure 2.36 shows different plots of Eq. 2.21 for different values of α : high values correspond to hard transitions. This progressive smoothing-to-sharpening diffusion is used when the number of iterations is known in advance, usually for applications where it can be set empirically.

For applications where the stopping criterion is evaluated during the successive iterations, the sharpening stage should be replaced by the use of a shock filter after the anisotropic diffusion

(see Section 2.8). However, one should be prudent when using this filter, because the sharpening process introduces step edges and suppresses the quantitative information about the strength of the transitions. Thus it should be used only if a segmented image does not undergo further processing. Energy minimization approaches to edge detection (see Chapter 5) are sensitive to the gradient values at strong edges and using a shock filter in the pre-processing stage would have a negative effect.

2.7.6 Application to vector-valued images

All the mathematical developments presented in the previous sections are applied to scalar images. The logical following step is the application to vector-valued images $\mathbf{v} : \mathbb{D}_v \mapsto \mathbb{R}^d$, $\mathbb{D}_v \subset \mathbb{Z}^2$. A typical example of vector-value is color. In [25] the authors give a possible insight to the problem. The different components are linked by the conductivity function that becomes a function of all components gradients:

$$\frac{\partial v_i}{\partial t} = \nabla \cdot (g(|\nabla v_1|, \dots, |\nabla v_d|) \nabla v_i) ,$$

while the diffusion itself is done separately for every component. It seems reasonable to control the diffusion according to the different gradients obtained from every image component. For color images, for example, the blue component may have no large gradient at a specific location, while the color image shows clearly a transition. To make this edge visible, all three components must be taken into account in order to decide if the local structure is flat or not.

This problem is dealt with in detail in [33]. The analysis is based on the minimization of the *first fundamental form*. This quadratic form is given by:

$$d\mathbf{v}^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{\partial \mathbf{v}}{\partial x_i} \frac{\partial \mathbf{v}}{\partial x_j} dx_i dx_j ,$$

where the (x, y) coordinates have been replaced by (x_1, x_2) for legibility purpose. The eigenvectors and eigenvalues are computed for the 2×2 matrix whose elements are $g_{ij} = (\partial \mathbf{v} / \partial x_i)^T (\partial \mathbf{v} / \partial x_j)$. The eigenvectors give the directions of maximal and minimal change, and the eigenvalues λ_+ and λ_- the rates of these changes. The edge strength is then measured by a function $f = f(\lambda_+, \lambda_-)$. In the 1-D case, the eigenvalues are given by $\lambda_+ = |\nabla v|^2$ and $\lambda_- = 0$.

One can argue that the visual contribution of the different components is not always clear, especially for texture descriptors. In the case of color images, the use of uniform color spaces [34] can provide a multi-dimensional gradient that fits the perceived color changes. Results on gradient estimation based on perceptual attributes for color images [35] may be integrated to the diffusion process. The gradient value, however, must be recomputed at every iteration, which renders this approach less interesting. In the case of uniform color spaces, the sum of gradients can be taken as argument to the conductivity function which can then be written $g(\sqrt{\sum_i |\nabla v_i|^2})$. This avoids the long processing proposed in [33]. The color-space change being computed once at the beginning, there is no additional time-consuming processing. The reader can find additional information about uniform color spaces in Chapter 3.

2.8 An alternative to backward-diffusion: shock filters

Other image processing techniques than diffusion can be driven by partial differential equations. In this section an edge enhancement technique called *shock-filter* [26] is described. This filter can

be compared to backward diffusion and is intended to restore smooth edges.

In 2-D the following equation is considered [26]:

$$\frac{\partial v}{\partial t} = -|\nabla v|F(\mathcal{L}(v)) ,$$

where F is a Lipschitz continuous function that satisfies $F(0) = 0$ and $X(u)F(u) > 0$, for $u \neq 0$ and $X(u) = 1$ if $u > 0$, $X(u) = -1$ if $u < 0$ and $X(0) = 0$. The *edge detector* $\mathcal{L}(v)$ is a second-order, nonlinear elliptic operator.

The basic idea behind this equation is to lower the pixel value where the second-derivative in gradient direction is positive (starting edge) and to increase this value in the opposite case (ending edge). The amount of variation is controlled by the local gradient $|\nabla v|$. Two possible functions F and \mathcal{L} are given by [26]:

$$\begin{aligned} F(\mathcal{L}) &= \text{sign}(\mathcal{L}) , \\ \mathcal{L}(v) &= \frac{\partial^2 v}{\partial n^2} , \end{aligned}$$

where $\mathbf{n} = \nabla v / |\nabla v|$.

Behind this simple formulation there is an exhaustive analysis devoted to the numerical implementation of such a process. It is clear that, depending on the approximation used for the derivatives, the system may oscillate, as illustrated in Figs. 2.37 and 2.38. A simple finite difference [28] approximation, for example, would make the process fail.

The different operators and the final iterative equation for a sampling grid (i, j) are given by:

$$\begin{aligned} v_{ij}(n+1) &= \begin{cases} v_{ij}(n) - \Delta t |\nabla_+ v_{ij}(n)| / h & \text{for } F_{ij} \leq 0, \\ v_{ij}(n) + \Delta t |\nabla_- v_{ij}(n)| / h & \text{for } F_{ij} > 0, \end{cases} \\ |\nabla_+ v_{ij}(n)| &= \sqrt{((\Delta_+^x v_{ij}(n))^+)^2 + ((\Delta_-^x v_{ij}(n))^-)^2 + ((\Delta_+^y v_{ij}(n))^+)^2 + ((\Delta_-^y v_{ij}(n))^-)^2} , \\ |\nabla_- v_{ij}(n)| &= \sqrt{((\Delta_+^x v_{ij}(n))^-)^2 + ((\Delta_-^x v_{ij}(n))^+)^2 + ((\Delta_+^y v_{ij}(n))^-)^2 + ((\Delta_-^y v_{ij}(n))^+)^2} , \end{aligned}$$

where the $\Delta_+^x v_{ij}$, $\Delta_-^x v_{ij}$, $\Delta_+^y v_{ij}$, and $\Delta_-^y v_{ij}$ differences are the different backward (−) and forward (+) differences for x and y , as described in § 2.6.4. Two other operators are introduced here: $(\cdot)^+ = \max(\cdot, 0)$ and $(\cdot)^- = \min(\cdot, 0)$. For the computation of $\mathcal{L}_{ij}(v(n))$ the second order derivatives are approximated by Eq. 2.17 and $\partial^2 v / \partial x \partial y$ by symmetric differencing, i.e. $\partial^2 v / \partial x \partial y = (1/2h^2)(\Delta_-^x \Delta_-^y v + \Delta_+^x \Delta_+^y v)$. The first derivatives, if needed, are approximated using the minmod operator given by Eq. 2.20.

This technique performs quite well for restoring blurred images. Figures 2.39 and 2.40 show the anisotropic diffusion results of Figs. 2.24 and 2.28 after edge enhancement with a shock filter. The result improves the transitions between neighboring regions but not without drawback: a stair-casing effect appears in both images. The difference between these results is also revealed by this edge enhancement method because the presence of noise does not only affect the visual aspect of the image but it changes the strength of the transitions after anisotropic diffusion.

2.9 Other developments

It would be unfair to end this chapter without saying anything about other relevant work on anisotropic diffusion that has been published, even if these contributions increase most of the time

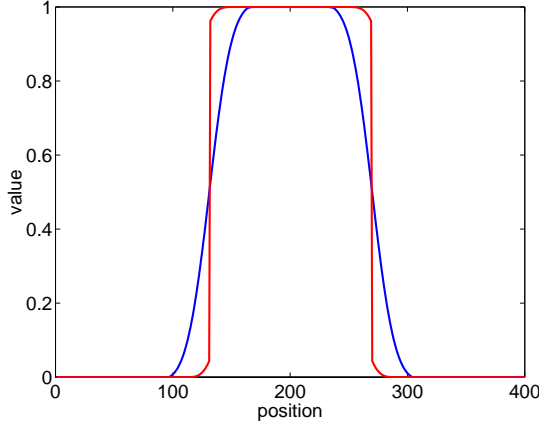


Figure 2.37: 1-D cut of a 2-D image containing a binary square blurred by a Gaussian low-pass filter. A well implemented shock filter is used to restore the signal.

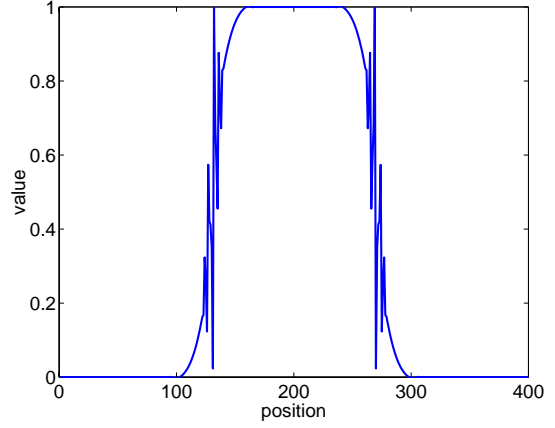


Figure 2.38: The signal of Fig. 2.37 has been *shocked* using a simple difference approximation to the different partial derivatives. After a few iterations the signal starts to oscillate (Gibbs phenomenon).

the complexity and processing time without perceptible improvement for most applications. In [31] the authors give a robust statistical interpretation of nonlinear isotropic diffusion and propose a new conductivity function based on *Tukey's biweight*, and finally derive the relationship between nonlinear isotropic diffusion and line processes. The application to vector-valued images is tackled: possible options are proposed, but no solution is investigated and illustrated. This paper is very interesting in its approach but did not show significant improvement compared to the techniques proposed in this chapter.

In [36] a new approach for image recovery is presented: the conductivity function given by 2.13 is computed on a filtered version of the input image. The filter kernel used is a two-dimensional symmetric exponential, which is an optimal linear operator for edge detection and noise removal. It outperforms the basic Perona-Malik model in presence of noise with large variance and has a good edge enhancement capability. But the use of an exponential filter kernel has a negative effect on the image geometry.

Multi-grid nonlinear isotropic diffusion has been presented in [37]. This implementation facilitates the application of nonlinear isotropic diffusion to time-critical processes. The diffusion equations are successively transferred to coarser grids and used in a coarse-to-fine error correction scheme. The author develops the theory of multi-grid nonlinear isotropic diffusion and discusses the inter-grid transfer functions, the relaxation techniques, the conductivity functions and the boundary conditions.

Additional reading on the subject can be found in [38, 39, 40, 13, 41].



Figure 2.39: In this example the image of Fig. 2.24 after 1000 iterations of anisotropic diffusion has been enhanced with a shock filter.



Figure 2.40: The same experiment than that shown in Fig. 2.39 has been repeated with the noisy image of Fig. 2.27.

2.10 Summary

In this chapter image filtering for high level image processing, like segmentation, image analysis, or feature extraction, has been presented. Classical filtering techniques such as the Gaussian, median and morphological filters, have been presented and illustrated, and finally the largest part of the chapter has been devoted to a powerful edge preserving filtering technique: nonlinear isotropic diffusion.

A first section has developed the isotropic diffusion model and has shown its connection to Gaussian low-pass filtering. This filter is implemented by convolving the filter kernel with the signal, which means that the infinite impulse response must be limited to a finite number of samples. The distortion introduced is usually negligible, provided that the number of samples is high. For strong smoothing the computation time may become significant. Isotropic diffusion has very well-defined boundary conditions, does not introduce distortion due to windowing, and has very low computation time. Successive resolutions can be obtained with constant computation time between successive iterations.

The two following sections have exposed the basics of nonlinear isotropic diffusion and possible extensions. Different conductivity functions have been proposed and tested with binary images and scenes, with and without additive noise. In the latter case the importance of choosing the right conductivity function has been clearly demonstrated. The use of tensors to control diffusion, the choice of a proper threshold value k_0 , and a scale-dependent nonlinear isotropic diffusion have been presented in Section 2.7. Section 2.8 has been dedicated to Shock filters, which have a backward diffusion behavior and are a powerful edge enhancement technique. A typical application of shock filters in this study is the edge enhancement after diffusion: when all unnecessary details have been removed, object borders can be enhanced.

When diffusion is used to prepare the image for the next processing step, the formulation given by Eq. 2.6, driven by the conductivity function 2.14 and used with 100-200 iterations has

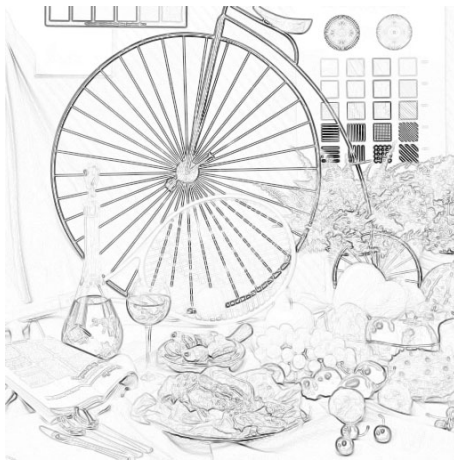


Figure 2.41: Gradient of the image shown in Fig. 2.2.



Figure 2.42: Gradient obtained after anisotropic diffusion of the image shown in Fig. 2.2.

been revealed to be highly efficient. This will be illustrated later in Chapter 5. Finally, the main purpose of developing diffusion theory in this chapter was to introduce the segmentation and contour detection problems that will be investigated respectively in Chapter 4 and Chapter 5. For both applications the precision will depend on how the image has been prepared for these operations: unprofitable information must be removed and transitions between regions enhanced. Figures 2.41 and 2.42 show the image gradient before and after anisotropic diffusion respectively. The filtering suppresses small spurious details while preserving region borders. The filtered version can be used for segmentation or contour detection.

The reader will find the most relevant and complete literature related to nonlinear isotropic diffusion in [12, 25, 27, 42].

Bibliography

- [1] A. S. Sedra and K. C. Smith. *Microelectronic circuits*. Saunders College Publishing, third edition, 1991.
- [2] W. K. Pratt. *Digital Image Processing*. John Wiley & Sons, second edition, 1991.
- [3] J. Serra. *Image analysis and mathematical morphology*. Academic Press, 1982.
- [4] S. Haykin. *Adaptive filter theory*. Prentice Hall, Englewood Cliffs, New Jersey, second edition, 1991.
- [5] C. R. Johnson. Interplay between recursive system identification and adaptive filtering in communication systems. In *System Identification (SYSID'94)*, volume 1, pages 13–24, 1994.
- [6] J. R. Cloutier, J. H. Evers, and J. J. Feeley. Assessment of air-to-air missile guidance and control technology. *IEEE Control Systems Magazine*, 9(6):27–34, October 1989.
- [7] M. Karrakchou, K. Vibe-Rheymer, J.-M. Vesin, E. Pruvot, and M. Kunt. Improving cardiovascular monitoring through modern techniques. *IEEE Engineering in Medicine and Biology*, 15(5):68–78, September 1996.
- [8] K. Erler and E. Jernigan. Adaptive image restoration using recursive image filters. *IEEE Transactions on Signal Processing*, 42(7):1877–1881, July 1994.
- [9] T. H. Farquhar, G. Chinn, C. K. Hoh, S. C. Huang, and E. J. Hoffman. A nonlinear, image domain filtering method for cardiac pet images. *IEEE Transactions on Nuclear Science*, 45(4):2073–2079, August 1998.
- [10] H. Levine. *Partial Differential Equations*. Studies in Advanced Mathematics. American Mathematical Society and International Press, 1997.
- [11] E. M. Landis. *Second order equations of elliptic and parabolic type*. American Mathematical Society, 1998.
- [12] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.
- [13] S. Biswas, N. R. Pal, and S. K. Pal. Smoothing of digital images using the concept of diffusion process. *Pattern Recognition*, 29(3):497–510, 1996.
- [14] J. Maeda, T. Iizawa, T. Ishizaka, C. Ishikawa, and Y. Suzuki. Segmentation of natural images using anisotropic diffusion and linking of boundary edges. *Pattern Recognition*, 31(12):1993–1999, December 1998.

- [15] F. L. Fontaine and S. Basu. Wavelet-based solution to anisotropic diffusion equation for edge detection. *International Journal of Imaging Systems and Technology*, 9(5):356–368, 1998.
- [16] S. Mallat. *A Wavelet tour of signal processing*. Academic Press, 1998.
- [17] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16(1):20–51, May 1981.
- [18] J. Sporring, M. Nielsen, L. Florack, and P. Johansen (Eds.). *Gaussian Scale-Space Theory*. Computational Imaging and Vision. Kluwer Academic, 1997.
- [19] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [20] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.
- [21] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, second edition, 1993.
- [22] T. S. Huang, G. J. Yang, and G. Y. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, February 1979.
- [23] M. A. Schulze and J. A. Pearce. Some properties of the two-dimensional pseudomedian filter. In *Proceedings of the SPIE*, volume 1451, pages 48–57, 1991.
- [24] A. A. Samarskii and P. N. Vabishchevich. *Computational Heat Transfer*, volume 1. Wiley & Sons, 1995.
- [25] Y.-L. You, W. Xu, A. Tannenbaum, and M. Kaveh. Behavioral analysis of anisotropic diffusion in image processing. *IEEE Transactions on Image Processing*, 5(11):1539–1553, November 1996.
- [26] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, August 1990.
- [27] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, Universität Kaiserslautern, January 1996.
- [28] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, Oxford, third edition, 1985.
- [29] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [30] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [31] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.
- [32] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 8(6):679–698, November 1986.

- [33] G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 5(11):1582–1586, November 1996.
- [34] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. Wiley, New-York, 1982.
- [35] Ph. Pujas and M.-J. Aldon. Estimation of the colour image gradient with perceptual attributes. In *Proceedings of the 9th International Conference on Image Analysis and Processing*, pages 103–110. Springer Verlag, 17-19 September 1997.
- [36] F. Torkamani-Azar and K. E. Tait. Image recovery using the anisotropic diffusion equation. *IEEE Transactions on Image Processing*, 5(11):1573–1578, November 1996.
- [37] S. T. Acton. Multigrid anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):280–291, March 1998.
- [38] B. B. Kimia and K. Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. *Computer Vision and Image Understanding*, 64(3):305–322, November 1996.
- [39] B. Fischl and E. L. Schwartz. Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):342–352, April 1997.
- [40] G. H. Cottet and M. El Ayyadi. A volterra type model for image processing. *IEEE Transactions on Image Processing*, 7(3):292–303, March 1998.
- [41] Ph. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529, June 1991.
- [42] J. Weickert. A review of nonlinear diffusion filtering. In B. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Scale-Space Theory in Computer Vision*, volume 1252 of *Lecture Notes in Computer Science*, pages 3–28. Springer, July 1997.

Chapter 3

Colorimetry and color spaces

3.1 Introduction

This chapter is dedicated to some basic concepts of color theory. While most of the image processing techniques have been developed for single-valued or gray-level images, the image signal is usually much more complex. Color is one of most natural and straightforward high level image representations. This information is strongly related to the *human visual system* (HVS) because it defines the way we perceive, differentiate, and analyze colors. In numerous image analysis applications, where human decision must be reproduced, the human visual criteria must be well understood. Therefore, prior to the clustering techniques for image segmentation presented in the next chapter, it is essential to understand what is color, how it is encoded, and how it can be represented.

Basic colorimetric concepts and some color space transformation which are intended to facilitate the color analysis for specific applications will be respectively presented in Sections 3.2 and 3.3. In particular, the use of uniform color spaces is interesting when the *amount of visual difference* between colors plays an important role, as it does for example in segmentation processes: there is often no unique result and it is up to the human color perception to define the (fuzzy) location of region borders.

3.2 Colorimetry

3.2.1 Principles and definitions

When including color information in a numerical system it is necessary to understand the origin and meaning of color. This is a very subjective issue when related to the *human visual system* (HVS), while the problem is absolutely objective when color is seen as light with a specific wavelength. When computer vision or image analysis techniques must follow human perception of color, they have an important probabilistic side, which means that the model or standard used is established according to the mean perception of a number of criteria. Let us start with the origin of color models before talking about more complex aspects.

The earliest color systems were based on astronomy. Pythagoras designed solids to portray the four elements (fire, water, air, and earth) and is said to be the first to have associated colors

	Earth	Water	Fire	Air	Wood	Metal
Hindu	black	white	red			
Greek	blue	green	red	yellow		
Chinese	yellow	black	red		green	white
Leonardo da Vinci	yellow	green	red	blue		

Table 3.1: The colors of the elements.

to them. Aristotle's doctrines and ideas affected the scientific world for almost eighteen centuries but unfortunately those concerning color were highly misled. He stated that *simple colors are the proper colors of the elements, i.e. of fire, air, water, and earth. Air and water when pure are by nature white, fire yellow, and the earth naturally white.* Through some questionable logic, Aristotle concluded that all colors could be obtained by mixing black and white. His work was taken by many as the definitive and this absurd notion persisted for centuries.

Leonardo da Vinci is credited for the first sensible interpretation of color theory. He stated that *the simple colors are six, of which the first is white, yellow is the second, green is third, blue is fourth, red is fifth, and black is the sixth.* The logic of the Greeks had separated color from human experience, and da Vinci proposed a more perceptual interpretation. Table 3.1 give the color perception of different natural elements for different civilizations and scientists.

In 1666 Newton performed his famous experiment with light and prisms: when a continuous beam of pure white light crossed a prism, it emerged as a rainbow on the other side. As we know today, visible light is part of the electromagnetic spectrum. Different wavelengths through this spectrum produce different hues, and because of the difference in wavelength they are refracted at different angles, splitting them apart. Newton's original findings clearly marked the beginning of modern color representation, and led to the standard color theory of physics based on radiation measurements, known as colorimetry.

One of the most significant color systems has been created by Albert Munsell who published his work in 1905. His system was based on human perception, and colors were chosen so as to be equally spaced one from another. He assigned three values to every color he measured: *hue*, *value* or brightness, and *chroma*, also known as *saturation* or *purity* of color. In his model the three components were represented in a cylindrical coordinate system, with a scale from 0 (black) to 10 (white) running vertically for the value. He then assigned 10 hues, with 10 steps in between, and the chroma ran out from the center in even steps. It must be underlined here that some colors have the potential for higher saturation and that the cylinder turns into an irregular shape. Precise measurements have verified that the visible color space is indeed not symmetrical. Slight variations of Munsell's original system are still used today in image editing, software, color selection books, etc.

Since these historical works, a large number of colorimetric standards has been defined by the *Commission Internationale de l'Eclairage* (CIE), an organization devoted to international cooperation and exchange of information among its member countries on all matters relating to the science and art of lighting. The CIE is an autonomous organization which was not appointed by any other organization, political or otherwise, but has grown out of the interests of individuals working in illumination. Since its inception, the CIE has been accepted as representing the best authority on the subject and as such is recognized by the ISO as an international standardization body. The following text refers mainly to the CIE standards.

In order to extend image processing techniques to color images it is important to understand how color information is integrated in digital images. Colorimetry is the branch of color science concerned with specifying numerically the color of a physically defined visual stimulus. A number of important psychophysical terms used in the following paragraphs are defined in Table 3.2. These definitions are taken from the International Lighting Vocabulary (CIE) [1]. They are psychophysical terms of color that refer to the color matching of one visual half-field with another, and to the judgments of similarities and degree of difference between two such half-fields. Psychophysical terms of color are distinguished from psychological terms which refer to color perception. Psychological terms of color apply to visual concepts that enable the individual observer to describe color perception.

The notion of color is strongly correlated with the human visual system. It is how human perceive color that makes color science. In order to be complete, one should study the light physics, light sources, optics, reflecting properties of matter, the eye structure, how it works and how the information is perceived and analyzed by the human brain. Since the goal here is to use color information to improve digital image processing techniques, a more quantitative approach is adopted. The reader should refer to [2] for additional information.

In modern colorimetry all colors are obtained through additive or subtractive mixture of three primary color *stimuli* (trichromatic model). By additive mixture we mean a color stimulus for which the radiant power in any wavelength interval, small or large, and in any part of the spectrum, is equal to the sum of the powers in the same interval of the constituents of the mixture. They are assumed to be *optically incoherent* (visually independent). These constituents are red ($\lambda_R = 700$ nm), green ($\lambda_G = 546.1$ nm), and blue ($\lambda_B = 435.8$ nm) and this model is mainly used in television applications where the signal is projected onto a surface with “ideal” diffuse reflection properties. Subtractive color systems have different constituents which are usually the secondary colors cyan (green and blue), magenta (red and blue), and yellow (red and green). Dye filters used in photography and color printing use these three primary color stimuli because they permit a greater transmission of light energy than the three narrow-band red, green, and blue. A fourth filter of variable gray level density is usually used in order to achieve higher contrast in reproduction. For our applications the additive model is used and all further developments are based on the three primary stimuli red, green, and blue.

In color matching experiments, where the observer attempts to match a given color stimulus by superimposing appropriate amounts of the three primary colors, the observational conditions are very important. Two stimuli are normally presented as contiguous light patches of similar shape and area. However, a color match of two stimuli of different spectral radiant power distributions, valid for a given observer when looking directly at the center of the matching field, will not generally remain valid if the observer looks to the side; and changing the area of the matching field may also upset the match. These viewing conditions must be fixed if critical color measurements are to be made. In practical colorimetry, direct viewing is of principal interest and can be assumed unless it is indicated otherwise. The angular size of the matching fields used varies widely and this factor has been taken into account in the CIE recommendations, as explained in § 3.2.3.

In the following paragraph the real and imaginary primary stimuli are presented in more details.

<i>Color</i>	Color (in the psychophysical sense) is that characteristic of a visible radiant power by which an observer may distinguish differences between two structure-free fields of view of the same size and shape, such as may be caused by differences in the spectral composition of the radiant power concerned in the observation. Psychophysical color is specified by the tristimulus values of the radiant power (color stimulus) entering the eye.
<i>Color stimulus</i>	A color stimuli is radiant power of given magnitude and spectral composition, entering the eye and producing a sensation of color.
<i>Monochromatic stimulus</i>	A monochromatic stimulus is monochromatic radiant power of <i>given power and wavelength</i> , entering the eye and producing a sensation of color.
<i>Achromatic stimulus</i>	An achromatic stimulus is the color stimulus chosen because it usually yields a color perception which is devoid of hue under the desired observing conditions.
<i>Primary color stimuli</i>	Primary color stimuli are color stimuli by whose <i>additive mixture</i> (linear combination with positive coefficients) nearly all other color stimuli may be completely matched in color.
<i>Tristimulus values</i>	The tristimulus values of a color stimulus are the amount of three primary color stimuli required to give by additive mixture a color match with the color stimulus considered.
<i>Color matching functions</i>	The color matching functions are the tristimulus values, with respect to three given primary color stimuli, of monochromatic stimuli of equal radiance, regarded as functions of the wavelength.
<i>Chromatic coordinates</i>	The chromatic coordinates of a color stimulus are the ratios of each tristimulus value of the color stimulus to their sum. <i>Note:</i> A diagram in which any one of the three chromaticity coordinates is plotted against any other is called <i>chromaticity diagram</i> . In this diagram, the chromaticity of a color stimulus plots as a point, the <i>chromaticity point</i> .

Table 3.2: Some basic colorimetric terms and their definitions.

3.2.2 Real and imaginary primary stimuli

In the *tristimulus space* an arbitrary color stimulus \mathbf{Q} , assumed to be produced by the imaging on the retina of a surface in the external field uniformly emitting radiant power¹, matches an additive mixture of the three fixed primary stimuli $\mathbf{R}, \mathbf{G}, \mathbf{B}$:

$$\mathbf{Q} = R\mathbf{R} + G\mathbf{G} + B\mathbf{B} ,$$

where R, G, B are the *tristimulus values*. In the first instance the given equation applies only if \mathbf{Q} can be matched with positive or nil tristimulus values. In the general case, \mathbf{Q} may have to be mixed with one or more primary stimuli in order to be “desaturated” before a color match can be made with the remaining primaries. In that case one or more tristimulus values may have a negative value.

The representation of color stimuli in tristimulus space as vectors is informative, but usually not convenient in colorimetric practice. A two-dimensional representation is commonly preferred, though it is recognized that this necessarily reduces the information contained in the tristimulus vector of the stimulus: the location in the two-dimensional space uniquely determines the direction of \mathbf{Q} but the length information that determines the “intensity” of \mathbf{Q} is lost. The *chromaticity* coordinates are given by the following equations:

$$\begin{aligned} r &= \frac{R}{R + G + B} , \\ g &= \frac{G}{R + G + B} , \\ b &= \frac{B}{R + G + B} , \end{aligned}$$

with $r + g + b = 1$. The chromaticity diagram shown in Fig. 3.1 is in the shape of an equilateral triangle, used in the early days of colorimetric practice, and sometimes referred to as the *Maxwell color triangle*. A more convenient version is a triangle in which r and g coordinate axes are perpendicular to one another, as shown in Fig. 3.2 for the imaginary chromaticity diagram. Such a right-angled triangle is now almost exclusively used when plotting chromaticity points.

The quantity $P_\lambda d\lambda$ represents the radiant power in the wavelength interval of width $d\lambda$ centered at the wavelength λ and defines a *monochromatic stimulus* of wavelength λ denoted by \mathbf{Q}_λ . The corresponding tristimulus values are called *spectral tristimulus values*. A particularly important set of these values is obtained when all the monochromatic stimuli contained in a color stimulus have unit radiant power at every wavelength within the visible spectrum. Such a stimulus is called *equal energy stimulus* and denoted by \mathbf{E} . The monochromatic constituents \mathbf{E}_λ obey the following equation:

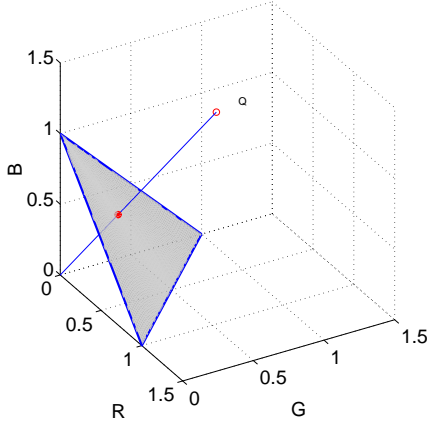
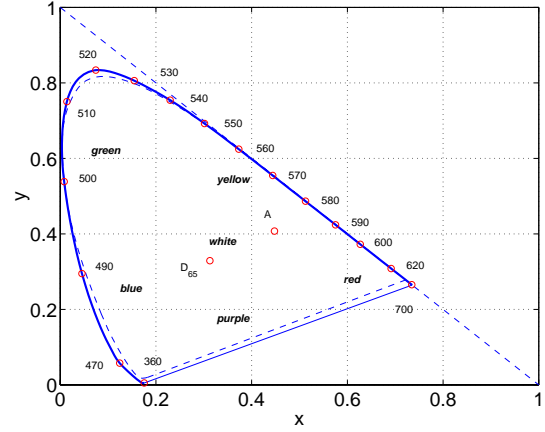
$$\mathbf{E}_\lambda = \bar{r}(\lambda)\mathbf{R} + \bar{g}(\lambda)\mathbf{G} + \bar{b}(\lambda)\mathbf{B} ,$$

where $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ are the tristimulus values of \mathbf{E}_λ .

Figures 3.3 and 3.4 show the color matching functions for two different systems based on the 1931 and 1964 CIE standard colorimetric observers described in § 3.2.3. In the former case the units of the primary stimuli have been fixed so that the chromaticity point of the equal energy stimulus is given by $r = g = 1/3$, while in the latter model the units are of unit radiant power.

The values of the real primary stimuli, especially red, are sometimes negative as can be seen in Figs. 3.3 and 3.4. In order to have all tristimulus values positive (negative values are not realistic),

¹The stimulus is then uniquely-defined by his absolute spectral radiant power distribution $\{P_\lambda d\lambda\}$.

Figure 3.1: The *rgb* chromaticity diagram.Figure 3.2: Chromaticity diagram of CIE 1931 standard colorimetric observer in the system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$.

the CIE has developed a standard artificial coordinate system. The XYZ system has been chosen so that the Y tristimulus value *is equivalent to the luminance of the color that must be matched*. The $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ primary stimuli are imaginary, which means that they cannot be realized by actual color stimuli.

The relationship between the tristimulus values R, G, B and X, Y, Z is given by the following linear transformation:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.7690 & 1.7518 & 1.1300 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Figures 3.5 and 3.6 plot the chromaticity functions for both the 1931 and 1964 colorimetric systems. These two systems constitute the official CIE 1931 and CIE 1964 system for practical colorimetry involving color-matching fields of angular size between one and four degrees, and greater than four degrees respectively. The units of these imaginary stimuli have been fixed so that the chromaticity point of the equal energy stimulus is given by $x = y = 1/3$.

In Fig. 3.2, the chromaticity diagram corresponding to the 1931 system is drawn. The dashed line shows the correction introduced by the 1964 system. Different standard illuminants are shown as well: D_{65} (natural or *warm* light) is close to the blue and A (artificial or *cold* light) is close to yellow.

Beside the two main reasons why the color-matching functions \bar{x} , \bar{y} , and \bar{z} have been chosen (they have no negative values and \bar{y} is an approximation to the brightness) there are some irritating aspects as well. First of all there is no set of physically realizable primary stimuli that may yield these functions by direct measurement. Such stimuli should have negative energy at some wavelength and can therefore not be instrumented. The second main drawback is that these early estimates are inaccurate in the short-wavelength region and have been improved upon. However the modified set of functions, used by some scientists, have not displaced the industrial standards.

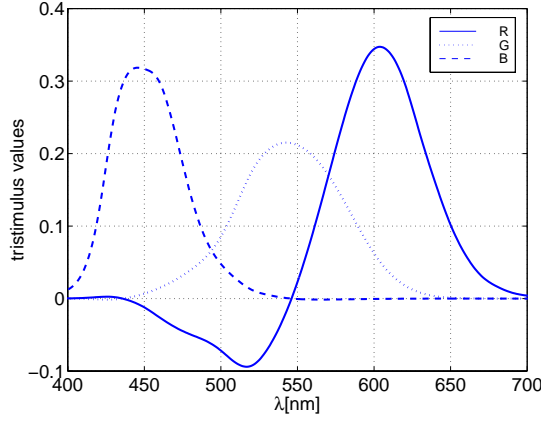


Figure 3.3: Color-matching function of CIE 1931 standard colorimetric observer in the system of real primary stimuli $\mathbf{R}, \mathbf{G}, \mathbf{B}$.

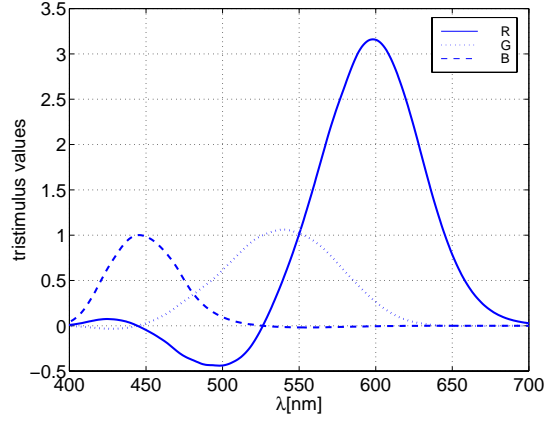


Figure 3.4: Color-matching function of CIE 1964 supplementary standard colorimetric observer in the system of real primary stimuli $\mathbf{R}, \mathbf{G}, \mathbf{B}$.

Two types of observers defined by the CIE and used for color-matching experiments, are described in the next paragraph.

3.2.3 Standard colorimetric observers

Not all observers need three independent primary stimuli to make additive color matches with an arbitrary color stimulus:

- Two primary stimuli are sufficient for a small but significant proportion, about 2.6 % males and less than 0.05 % females. They are called *dichromats* and are classed as color defective.
- Some observers have no color discrimination and their matching limits to brightness matching. They are called *monochromats* and represent a much smaller proportion, about 0.003 %.

Some observers show to have pronounced deviations from those of the main group, deviations usually accompanied by a lowered sensitivity to color differences, and sometimes by other differences in visual response. They are called *anomalous trichromats* and represent about 5.5 % males and 0.4 % females. The distinction with *normal trichromats* is not clearcut because the fixing of a point where the deviation from average constitutes an anomaly is somehow arbitrary.

Figure 3.7 describes the color matching experiment: the observer views a bipartite field and adjusts the three primary stimuli \mathbf{R}, \mathbf{G} , and \mathbf{B} to match an arbitrary color stimulus \mathbf{Q} . The stimuli are surrounded by white light and their sizes are given by their angular sizes. This experiment has been used to establish the different color matching functions.

An ideal observer was established 1931 by the CIE under the name *CIE 1931 Standard Colorimetric Observer*. The color-matching properties of this observer are defined by the color-matching functions \bar{r} , \bar{g} , and \bar{b} (see § 3.2.2) which were defined in the range 380-780 nm at intervals of 5 nm. The experimental data for actual observers that provided the values for the color-matching functions were obtained with matching fields of two degrees angular size directly viewed so that any

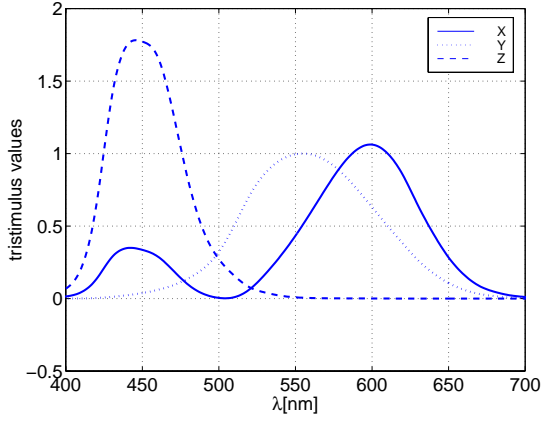


Figure 3.5: Color-matching function of CIE 1931 standard colorimetric observer in the transformed system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$.

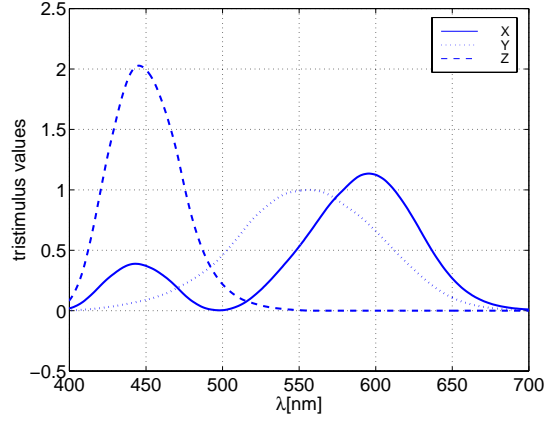


Figure 3.6: Color-matching function of CIE 1964 supplementary standard colorimetric observer in the transformed system of imaginary primary stimuli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$.

participation of rod vision² in the measurements was avoided. The CIE 1931 standard observer is suitable for application to matching fields of between one and four degrees angular size.

As a supplement to the 1931 standard observer the CIE defined the *CIE 1964 Supplementary Standard Colorimetric Observer*. An alternative set of standard color-matching functions has been established, that should be used whenever more accurate correlation with visual color-matching in fields of angular size of more than four degrees is desired. This standard is based on matching data that has been obtained for centrally viewed matching fields of ten degrees angular size.

The conversion formulas between the ten degrees color matching functions based on the $\mathbf{R}, \mathbf{G}, \mathbf{B}$ and $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ primary stimuli are given by:

$$\begin{aligned}\bar{x}_{10}(\lambda) &= 0.341080\bar{r}_{10}(\lambda) + 0.189145\bar{g}_{10}(\lambda) + 0.387529\bar{b}_{10}(\lambda) , \\ \bar{y}_{10}(\lambda) &= 0.139058\bar{r}_{10}(\lambda) + 0.837460\bar{g}_{10}(\lambda) + 0.073316\bar{b}_{10}(\lambda) , \\ \bar{z}_{10}(\lambda) &= 0.039553\bar{g}_{10}(\lambda) + 2.026200\bar{b}_{10}(\lambda) ,\end{aligned}$$

which gives for the tristimulus values the following matrix representation:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.341080 & 0.189145 & 0.387529 \\ 0.139058 & 0.837460 & 0.073316 \\ 0.000000 & 0.039553 & 2.026200 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} .$$

The color perception is strongly affected by the illumination. The color of objects is reflected differently into the eye depending on the light temperature incident on the object surface. The use of standard illuminants gives a way of integrating this parameter in color measurements and is exposed in the next paragraph.

²The book by Wandell [3] is an excellent reference for a complete description of the human visual system.

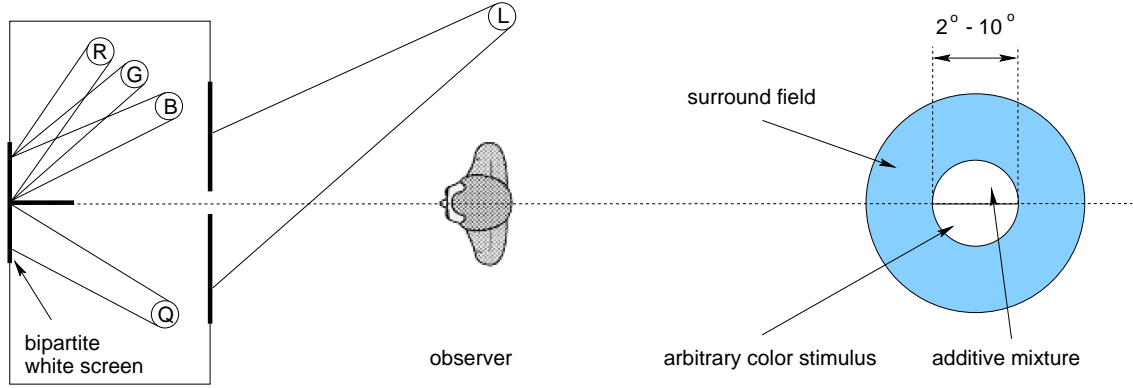


Figure 3.7: Color matching experiment: the observer views a bipartite field and adjusts the intensities of the three primary stimuli to match an arbitrary color stimulus. The experimental apparatus is shown left and the appearance of the stimuli to the observer is shown right.

3.2.4 CIE standard illuminants

In colorimetry the main concern is with the measurement of the color of objects. Light emitted from a source of radiant power (sun, lamp, etc.) strikes a given object and makes it visible to the observer. The color stimulus Q emerging from the object and entering the eye produces the sensation of color that is identified as the color of the object. The spectral radiant power distribution of the object color stimulus is given by:

$$\{P_\lambda d\lambda\} = \{\beta(\lambda)S_\lambda d\lambda\},$$

and holds only for non-fluorescent objects. The spectral concentration of the radiant power incident on the object is defined by S_λ and the spectral reflectance properties of the object in accordance with the geometrical conditions of illumination and viewing is defined by $\beta(\lambda)$.

Whenever object-color stimuli are of main concern, it is desirable to restrict colorimetric measurements and calculations to a few specific and well-defined spectral distributions $\{S_\lambda d\lambda\}$ of radiant power incident on the objects under study. The CIE recommends a set of such spectral radiant power distributions called *CIE standard illuminants*.

CIE standard illuminant A represents light from the *full radiator*³ at absolute temperature 2856 K according to *The International Practical Temperature Scale*, 1968. Its relative power distribution has been calculated by means of Planck's radiation formula [2].

The standard illuminant D_{65} represents a phase of natural daylight with a correlated color temperature of approximately 6504 K . Illuminant B is intended to represent direct sunlight with a correlated color temperature of approximately 4874 K and illuminant C represents average daylight with a color temperature of approximately 6774 K . Both illuminants B and C are considered inadequate in representing the intended phases of natural daylight and have been

³Also called Planckian or blackbody radiator: the radiant energy of this hot body enclosed in a cavity whose walls are maintained at a uniform temperature T depends on the density and spectral distribution of the wall temperature. A small aperture constitutes the ideal blackbody radiator and the emitted radiant power depends solely on the wall temperature.

dropped from the list of recommended illuminants, while illuminant D_{65} is widely used as the representative of average daylight for colorimetry.

In the case of illuminant A (“cold” white of tungsten bulb), most of the spectral radiant power is at large wavelength (blue). Illuminant D_{65} has a more regular distribution and is more accurate than illuminant C , especially at narrow wavelength (ultraviolet region). Figure 3.2 shows the chromaticity diagram of CIE 1931 standard colorimetric observer in the system of imaginary primary stimuli as well as the two illuminants A and D_{65} . The former is close to yellow while D_{65} is close to blue. In photographic applications this proximity with the yellow and blue colors can be observed when taking inside pictures with artificial light (no flash) and with a film specified for outside pictures and inversely: the resulting pictures have respectively a yellow and blue dominant.

The CIE also provided methods for calculating the relative spectral radiant power distribution of any daylight illuminant D_T (the convention is that the temperature is written as $T \cdot 10^2$) with correlated color temperatures ranging from 4000 to 25000 K . The temperature is related to the chromaticity coordinates (x_D, y_D) of that D illuminant by the following equations [2]:

- for $4000 \leq T \leq 7000$:

$$x_D = -4.607 \frac{10^9}{T^3} + 2.9678 \frac{10^6}{T^2} + 0.09911 \frac{10^3}{T} + 0.244063 ,$$

- for $7000 < T \leq 25000$:

$$x_D = -2.0064 \frac{10^9}{T^3} + 1.9018 \frac{10^6}{T^2} + 0.24748 \frac{10^3}{T} + 0.237040 ,$$

and $y_D = -3x_D^2 + 2.87x_D - 0.275$.

While these equations are the most interesting and important, the CIE has also defined *standard sources* and *illuminating and viewing conditions* that complete the definitions given above. The reader should refer to [2] for a complete description.

Now that we have defined a way of computing a reference white based on the light temperature used to illuminate the observed scene or objects, we can proceed with the definition of different color spaces obtained from the *RGB* tristimulus values. Linear transformations are described in the next section and nonlinear transformations are described in § 3.3.3. The reason why some color space may be more appropriate to a specific application is that the color content is better distributed in order to facilitate, for example, the detection of clusters. An important feature of uniform color spaces is the existence of *color difference* which can be very useful to approximately quantify the visual difference between colors. In § 3.3.2 the *principal component* decomposition is described. A linear transformation is computed from and for an input image that orients the base vectors in directions of maximal variance. This transformation is very useful when a reduction of the color space dimension is needed to reduce the complexity of a following processing step.

3.3 Color spaces

3.3.1 Linear colorimetric transformations

For most applications that deal with color images the *RGB* representation leads to sub-optimal results. The reason is usually the non-uniformity of this color space and the non-existent separation between luminance and chrominance. Depending on the type of application it is necessary to use

	Conversion formula
$R_N G_N B_N$	$\begin{pmatrix} R_N \\ G_N \\ B_N \end{pmatrix} = \begin{pmatrix} 1.910 & -0.532 & -0.288 \\ -0.985 & 2.000 & -0.028 \\ 0.058 & -0.118 & 0.898 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$
YIQ	$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.000 & 1.000 & 0.000 \\ 1.407 & -0.842 & -0.451 \\ 0.932 & 1.189 & 0.233 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$
YUV	$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.000 & 1.000 & 0.000 \\ 1.688 & -0.058 & -0.251 \\ 0.015 & 1.456 & 0.441 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$
UVW	$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 0.667 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \\ -0.500 & 1.500 & 0.500 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$

Table 3.3: Linear color space conversions from the tristimulus values X, Y, Z .

different color representations. The easiest color transformations are linear and some of them are presented in this paragraph, while nonlinear transformations are described in § 3.3.3. Table 3.3 gives the conversion formulas from RGB to the different systems presented in this paragraph.

The CIE RGB system is sometimes referred to as the $R_C G_C B_C$ system in order to be differentiated from other primary systems [4], for example the American *National Television Systems Committee* (NTSC) $R_N G_N B_N$ *Receiver Primary Color Coordinate System*. It is based on the NTSC phosphor standard⁴ and the gamut of producible colors is smaller than that available from the spectral primaries.

In order to separate color luminance and chrominance information, allowing the use of color television signals by monochrome receivers, the NTSC formulated the YIQ *Transmission Color Coordinate System*. The Y tristimulus value is the luminance, and the I and Q values jointly describe the hue and saturation attributes. It was found possible to limit the spatial bandwidth of the two latter signals without noticeable image degradation, and a clever analog modulation scheme was developed such that the bandwidth of a color television carrier could be restricted to the same bandwidth as a monochrome carrier [4].

The YUV *Color Difference Color Coordinate System* was initially proposed by the NTSC but it was found that the I and Q signals could be reduced in bandwidth to a greater amount than the U and V signals for an equal visual quality. The Y component is the same than that of the YIQ system, and the U and V components are related to I and Q by the following relationship:

$$\begin{aligned} I &= -U \sin(33^\circ) + V \cos(33^\circ) , \\ Q &= U \cos(33^\circ) + V \sin(33^\circ) , \end{aligned}$$

which is simply a rotation of 33° . The YUV coordinates are used in the PAL and SECAM color television systems in many countries. It is also widely used in the image compression community.

⁴Commercial television receivers employ a cathode ray tube with three phosphors that glow in the red, green, and blue regions of the visible spectrum but are not pure monochromatic sources of radiation.

Two interesting color spaces are the UVW and $U^*V^*W^*$ color spaces. The former has been defined in a first attempt to create a uniform color space and called the *Uniform Chromaticity Scale Color Coordinate System* (UCS), because equal changes in the chromaticity coordinates result, to a good approximation, in equal changes in the perceived hue and saturation of a color. Note that V is the luminance component. The u, v chromaticity coordinates are given by:

$$\begin{aligned} u &= \frac{4x}{-2x + 12y + 3}, \\ v &= \frac{6y}{-2x + 12y + 3}. \end{aligned}$$

In the $U^*V^*W^*$ color space uniform shifts in luminance and chrominance are, to a good approximation, uniformly perceptible. However, the introduction of the $L^*a^*b^*$ and $L^*u^*v^*$ uniform color spaces (see § 3.3.3) rendered the UVW and $U^*V^*W^*$ color spaces obsolete.

These different color spaces illustrate how the initial RGB tristimulus values may be transformed depending on the kind of application and/or information that must be accessed. In some cases it is necessary to compute a luminance correlate from a color image or to project the image onto a preferential axis in the color space. Generally the color space that must be used is defined by the application. In the case of color clustering, the use of uniform color spaces may be best adapted in order to find the limit between neighboring clusters. Next section presents an important linear transformation that can be combined with the use of uniform color spaces as well, as demonstrated in § 3.3.3.

3.3.2 The principal component decomposition

One can think of an adaptive color space transformation based on a predefined criterion. Such a transformation is the *principal component decomposition*, also called *Karhunen-Loève* or *Hotelling* transform. The basic idea is to find orientations in the data distribution along which the variance of the projected data is maximum. The complete description is presented in Chapter 6 and only the formulation is given here. This decomposition uses the eigenvectors of the data covariance matrix, computed using the following equations [5]:

$$\begin{aligned} \mathbf{m}_x &= \frac{1}{M} \sum_{k=1}^M \mathbf{x}_k, \\ \mathbf{C}_x &= \frac{1}{M} \sum_{k=1}^M \mathbf{x}_k \mathbf{x}_k^T - \mathbf{m}_x \mathbf{m}_x^T, \end{aligned} \tag{3.1}$$

where M is the number of samples (or pixels in our case) and \mathbf{m}_x is the mean sample vector. Defining a matrix \mathbf{A} whose lines are the eigenvectors ordered with decreasing eigenvalue, the Karhunen-Loève transform is defined by [5]:

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x). \tag{3.2}$$

The eigenvalues are the variance along the corresponding eigenvectors, which form an orthonormal basis. This decomposition re-orientates the color system in directions of maximal variance. If variance is taken as a measure of contrast, then this transformation projects most of the information content on the first or two first frames.

The pixel distribution is rarely linear and usually build of multiple clusters oriented in different directions. A nonlinear projection may be better suited, but the computation time may also be

disproportionate to the desired effect. A piece-wise constant projection like that proposed in [6] is a good alternative but requires prior knowledge of the number of clusters. In our case it would mean to use the solution to solve the problem.

Simulations showed that the last component almost always contains most of the image noise and that discarding it may have rather a positive effect on the segmentation. It also reduces significantly the processing time.

3.3.3 Nonlinear transformations and uniform color spaces

Before talking about uniform color spaces, which are of prime interest in this study, it is worth to cite a nonlinear transformation from the *RGB* color space to an intensity, hue, and saturation representation. It is defined by the following relations [4]:

$$\begin{aligned} \begin{pmatrix} I \\ V_1 \\ V_2 \end{pmatrix} &= \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1/\sqrt{6} & -1/\sqrt{6} & 2/\sqrt{6} \\ 1/\sqrt{6} & -1/\sqrt{6} & 0 \end{pmatrix} \begin{pmatrix} R_N \\ G_N \\ B_N \end{pmatrix}, \\ H &= \tan^{-1}(V_2/V_1), \\ S &= (V_1^2 + V_2^2)^{1/2}, \end{aligned}$$

and is called *IHS color coordinate system*. From this definition the zero reference for hue is the color blue.

This set of equations is not unique and other formulations than those given above can be found [5]. It illustrates at least that the notions of *intensity*, *luminance*, *hue*, and *saturation* are not precisely and uniquely-defined and that a strong relationship to the underlying physical phenomenon is not straightforward. These equations provide at least a good approximation and may be very useful when a specific component is needed. For example, a “pure color” segmentation can be done using only the hue and saturation components.

The CIE has proposed some uniform color spaces, such as $L^*a^*b^*$ or $L^*u^*v^*$, established using psycho-visual experiments [2]. In these color spaces, color difference can be expressed using the Euclidean distance. The $L^*u^*v^*$ quantities are defined by:

$$\begin{aligned} L^* &= \begin{cases} 116 (Y/Y_n)^{1/3} - 16 & \text{if } Y/Y_n > \gamma, \\ \beta Y/Y_n & \text{if } Y/Y_n \leq \gamma, \end{cases} \\ u^* &= 13L^*(u' - u'_n), \\ v^* &= 13L^*(v' - v'_n), \end{aligned} \tag{3.3}$$

where the u' and v' quantities are calculated from:

$$\begin{aligned} u' &= \frac{4X}{X + 15Y + 3Z}, \\ v' &= \frac{9Y}{X + 15Y + 3Z}, \end{aligned}$$

and the Y_n , u'_n , and v'_n quantities are obtained for the nominally white object color stimulus. X_n , Y_n , and Z_n are usually equal to the tristimulus values of the standard illuminant used with $Y_n = 255$ (or the maximum value).

The luminance function is defined in order to have zero luminance for $Y = 0$. The linear part is then given by the tangent to the curve that crosses the origin:

$$\frac{L^*}{Y/Y_n} = (L^*)' ,$$

which leads to:

$$\begin{aligned} \gamma &= \left(\frac{6}{29}\right)^3 \approx 0.008856 , \\ \beta &= \frac{116}{3} \left(\frac{29}{6}\right)^2 \approx 903.3 . \end{aligned}$$

The second uniform color space is defined by the $L^*a^*b^*$ quantities. L^* is given by Eq. 3.3, a^* and b^* are given by:

$$\begin{aligned} a^* &= 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] , \\ b^* &= 500 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] , \end{aligned}$$

where

$$f(x) = \begin{cases} x^{1/3} & \text{if } x > \gamma, \\ 841x/108 + 4/29 & \text{if } x \leq \gamma . \end{cases}$$

The two factors 841/108 and 4/29 are computed so that the relationship becomes linear and follows the first derivative at position γ .

It is surprising that the color components a^* and b^* do not depend on the luminance, knowing that color perception is strongly influenced by the luminance. In the $L^*a^*b^*$ space different chrominance (a^* and b^* components) values are possible for zero luminance, which is contrary to our intuition. Therefore we will use the $L^*u^*v^*$ transformation for our image segmentation application.

A number of correlates can be derived from these two color spaces:

- L^* serves as the correlate of *lightness*,
- the quantity $C_{uv}^* = [(u^*)^2 + (v^*)^2]^{1/2}$ serves as correlate of *chroma*,
- $s_{uv}^* = C_{uv}^*/L^*$ can be used as a correlate for *saturation*,
- and the quantity $h_{uv} = \arctan(v^*/u^*)$ defines the *hue angle*.

A color difference formula is associated with both uniform color spaces and allows one to compute the total color difference between two color stimuli:

$$\Delta E_{uv}^* = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{1/2} . \quad (3.4)$$

These different correlates and the color difference formula work also in the $L^*a^*b^*$ space.

It is important for our application to note that a principal component decomposition applied to a uniform color space does not change its uniformity. If we use Eq. 3.2 in a uniform color space with two tristimulus vectors \mathbf{q}_1 and \mathbf{q}_2 , the color distance becomes:

$$\Delta E^* = \left[(\mathbf{q}_1 - \mathbf{q}_2)^T \mathbf{A}^T \mathbf{A} (\mathbf{q}_1 - \mathbf{q}_2) \right]^{1/2} .$$

The uniformity is not altered by this linear projection only if $\mathbf{A}^T \mathbf{A} \sim \mathbf{I}_3$, where \mathbf{I}_3 is the 3×3 identity matrix. This condition is fulfilled since the rows of matrix \mathbf{A} form an orthonormal basis (eigenvectors of the covariance matrix \mathbf{C}_q computed with all image pixels). Therefore $\mathbf{A}^T \mathbf{A} = \mathbf{I}_3$, which means that both the uniformity and the scale are preserved. This feature is very interesting if one wants to reduce the data dimension and use color distance at the same time. It is exploited in the segmentation cluster detection and segmentation scheme presented in Chapter 4.

3.4 Summary

The different and most important aspects of colorimetry have been presented in this chapter and led to the definition of nonlinear color space transformations with an associated color difference formula. Color being the information associated with every image pixel and used in the segmentation scheme presented in the next chapter, Sections 3.2 and 3.3 have given the necessary background to colorimetry and color spaces respectively.

It is important to understand the basic concepts of color science and the different ways color information can be represented. The choices we will make later will dramatically influence the segmentation result and it is therefore important to adapt the color representation to the underlying problem. In image segmentation applications the main step is the detection of color clusters and the partition of the data. Once this processing step has been performed, the image can be segmented by labeling every image pixel according to its corresponding color class. The next chapter investigates different aspects of data clustering for image segmentation and presents an efficient cluster center computation and data partitioning method called *Orientation-Sensitive Fuzzy c-means*.

Bibliography

- [1] Commission Internationale de L'Eclairage (CIE) and International Electrotechnical Commission (IEC). International lighting vocabulary, 1987.
- [2] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. Wiley, New-York, 1982.
- [3] B. A. Wandell. *Foundations of Vision*. Sinauer Associates, 1995.
- [4] W. K. Pratt. *Digital Image Processing*. Wiley, New York, second edition, 1991.
- [5] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [6] G. Fernández i Ubiergo. Lossless region-based multispectral image compression. In *Proceedings of the Sixth International Conference on Image Processing and its Applications*, volume 1, pages 64–68, 1997.

Chapter 4

Image segmentation by color clustering

4.1 Introduction

Image segmentation was, is and will be a major research topic for many image processing researchers. The reasons are obvious and the applications endless: most computer vision and image analysis problems require a segmentation stage in order to detect objects or divide the image into regions which can be considered homogeneous according to a given criterion, such as color, motion, texture, etc. The goal is to obtain a piece-wise constant image, where each region contains image pixels sharing the same punctual or local characteristics. In this thesis we will differentiate *image segmentation* and *contour detection*. An image being a set \mathbb{I} of vectors (position and value associated with every image pixel), image segmentation aims at the creation of subsets \mathbb{I}_i whose elements share the same characteristics (spatial, spectral, geometrical, etc.) and where $\mathbb{I}_i \cap \mathbb{I}_j = \emptyset$ when $i \neq j$. Segmentation is achieved only when the following condition is fulfilled:

$$\mathbb{I} = \bigcup_i \mathbb{I}_i . \quad (4.1)$$

Edge detection is the construction of subset \mathbb{I}_{edge} of all image pixels that belong to edges. Edges physically isolate regions but do not give any information about these regions. In addition some of them can contain edges that do not necessarily correspond to a region border. This chapter is dedicated to image segmentation, while edge detection is treated separately in Chapter 5.

Image segmentation is usually the combination of different (but not all necessary) processing steps: filtering, coarse segmentation, fine segmentation, and post-processing. The first step has been discussed and investigated in the previous chapter. The coarse processing is intended to give quickly an approximate result and be of low complexity (compared to the complete processing scheme). It will most of the time lead to an over-segmentation, which means that the *true* regions are split into smaller parts or contain some spurious regions. The fine segmentation is then intended to merge neighboring regions according to some predefined criteria. This high level processing is usually time consuming and the coarse segmentation is then necessary in order to limit the processing time. The better the approximation provided by the coarse segmentation, the faster the fine segmentation. Finally the post-processing may be a high level processing where the different regions are identified and labelled.

In this chapter the segmentation of color images by 2-D color clustering will be investigated. This kind of approach is especially interesting for images where objects have a colorimetric signature, i.e. they are completely defined by their color and show smooth transitions between neighboring regions. The distribution of pixel values is usually made of several clusters that can be detected and compared one with another in order to evaluate their respective importance. The clustering operation will lead to a segmented image where every region correspond to a cluster or group of pixels. Image segmentation by color clustering is very efficient when there is a strong correlation between spatial and spectral information.

Clustering is a way to achieve data classification by detecting different groups of data. The data of a given group or cluster share common characteristics and it is expected in image processing applications that these are visual characteristics. High dimensional data is best adapted to clustering techniques but slows down the computation. It is therefore important to carefully choose the data representation by appropriate computation of descriptors or color space transformations.

Different approaches to image segmentation and basic clustering techniques will be presented respectively in Sections 4.2 and 4.3. *Fuzzy partitions* and the *fuzzy c-means clustering technique*, with a significant improvement for oriented clusters, will be introduced in Section 4.4. The problem of finding the number of clusters is part of every clustering problem. It will be investigated in Section 4.5. Finally, the pre and post-processing tasks will be developed in Section 4.6.

4.2 Image segmentation strategies

The literature is full of segmentation techniques and the newcomer has the difficult task of taking the (right) decisions that will orient his investigations. Ideally the researcher wants to find the *universal technique* that succeeds for any type of images. If it is not impossible, it is at least a life's work. In general the researcher must focus on a limited number of applications that share some common characteristics. It often happens that a segmentation technique works best for one application, which explains why this research domain has not been fully investigated yet. There are different approaches to image segmentation which perform well in most of the cases, but they must be adapted to any specific problem in order to give the best possible performances. An important aspect of image segmentation is the integration of color information. A lot of methods have been proposed in the literature for gray-level images but the extension to color images is usually not straightforward.

The fastest and least complex way to segment an image is certainly *histogram thresholding* [1]. The pixel position is discarded and only the “spectral” information is used. A histogram is first constructed with the pixel values. In the simplest case histogram thresholding is the detection of minima which are supposed to correspond to the limit between neighboring (in the histogram) groups of pixels. It is also assumed that the histogram has been previously smoothed out in order to suppress small local maxima. This approach becomes much more tricky when one tries to define robust strategies for the threshold selection, using scale-space analysis [2], fuzzy sets [3] or even neural networks [4]. A comparison between different strategies can be found in [5].

The main drawback of histogram thresholding is that spatial information is not taken into account. In order to overcome this limitation in the process, several methods based on a combination of *split* and *merge* steps have been proposed. The most natural way of doing merging, called *region growing*, is to start at the pixel level and merge progressively neighboring pixels sharing

the same characteristics. Another more practical way of doing merging is *region merging*, where an over-segmented image, obtained with an histogram thresholding method for example, must be refined. Finally, an approach in between would be to cut the image into fixed size square regions and to progressively merge neighboring windows that belong to a common region. The drawback of this last method is that the square size defines the highest resolution and that initial regions may contain pixels that belong to different regions. In the split approach the whole image is recursively split into smaller parts until all regions have been extracted.

The most widely used approach is called *split-and-merge* and combines both strategies. It often uses an adjacency graph [6, 7] to decide if a region must be split into smaller parts or merge with a neighboring region. An alternative to the classical approach is presented in [8] and uses a piecewise least-square approximation of image intensity functions to adapt the algorithm to the image semantics. Actually the split-and-merge approach is often combined with a segmentation technique that leads to an over-segmented image. Morphological approaches, like the *watershed algorithm* [9], are very efficient for image segmentation and can be used to produce an approximate result. The different regions are then treated using a split-and-merge approach [7]. Finally, statistical approaches like *Markov random fields* are used for image segmentation and often integrate split-and-merge phases [10].

Recent research in the image segmentation domain has shown increasing interest for color images. However most of the published segmentation techniques work with gray-level images. A simple approach consists of putting together the results obtained separately for the different color frames. To improve the process it is necessary to take into account the correlation between the different components. In particular, special care should be taken in the choice of the color space in order to better separate components like *intensity*, *hue*, or *saturation*, and be able to compare colors uniformly in the whole color space.

An efficient, general purpose, color segmentation scheme was proposed in [11]. This method uses the histogram information of the three color components to compute the number of valid classes, which corresponds to the number of valid maxima found in the histograms. The image is then segmented into the number of classes previously extracted using the *fuzzy c-means* (FCM) clustering technique.

The proposed method suffers from the following limitations:

- The three histograms are computed separately. This projection of the color histogram on the three components may hide some relevant maximum.
- The parameters are supposed to be tuned once for a given application. However simulations showed that different type of images require different parameter values, rendering the process tedious.
- A coarse segmentation assigns all pixels within a neighborhood of a maxima center to the corresponding class. If the FCM technique is used to recompute the cluster centers, all the pixels must be considered, even those which have already been clustered. If, on the other hand, these pixels are considered to be definitely labelled, using the FCM has no sense (see § 4.4.2).

This method constituted an interesting starting point for the segmentation of color images through multi-dimensional clustering. However the choice of a particular color space, the histogram

analysis and the clustering phase can be better adapted to efficiently solve the segmentation problem under specific conditions.

4.3 Data clustering

4.3.1 Introduction

Clustering is a way of doing *pattern recognition* (PR), that can be characterized as an *information reduction, information mapping, or information labeling process* [12]. A complete PR system usually consists of a sensor that provides a quantitative representation of some measure, a feature extraction mechanism and a classification or description algorithm. In PR problems it is sometimes assumed that some data, called *training set*, is already available in order to train the classification system. There are different approaches to PR, the most commonly used being *statistical*, *syntactic* (structural analysis) and *neural* (non-algorithmic strategy which is trainable) [12]. Statistical PR attempts to classify patterns based on a set of extracted features and an underlying statistical model for the generation of these patterns, and includes clustering techniques. It assumes that the underlying PR problem can be approached by using a single straightforward procedure that starts with the feature extraction, then trains the classification system and finally classifies the data.

Clustering is *the search for distinct groups in the feature space*. It is expected that these groups have different structures and that they can be clearly differentiated. The clustering task separates the data into a number of *partitions*, which are volumes in the d-dimensional data space. These partitions define the hard limit between the different groups and depend on the functions used to model the data distribution. The success depends on different factors such as the data representation and choice of color space. For texture segmentation the descriptors (vectors that, for each position, describe the local structure) are critical in deciding how well textures can be differentiated. In color segmentation, different color spaces can be used leading all to different structures in the data distribution. Depending on the application and the decision rule one specific color space may be best adapted.

Color segmentation produces a spatial partition (pixel position) of the data according to Eq 4.1, while color clustering produces a “spectral” partition (pixel value). When using clustering for color segmentation applications it is assumed that there is a direct link between spatial and spectral partitions. But the correspondence is not necessarily unique. Suppose that a particular image is defined as a function $f : \mathbb{Z}^2 \mapsto \mathbb{R}^d$, then the segmentation by clustering is a function $g : \mathbb{R}^d \mapsto \mathbb{Z}^2$ and both spectral and spatial partitions obey the following relationships:

$$\begin{aligned} \mathbb{I}^{(\mathbb{Z}^2)} &= \bigcup_i \mathbb{I}_i^{(\mathbb{Z}^2)}, \quad i = 1, \dots, c^{(\mathbb{Z}^2)}, \\ \mathbb{I}^{(\mathbb{R}^d)} &= \bigcup_i \mathbb{I}_i^{(\mathbb{R}^d)}, \quad i = 1, \dots, c^{(\mathbb{R}^d)}, \end{aligned}$$

where the different subsets in both spatial and spectral partitions are disjoint. Then it often happens that for a particular $I_i^{(\mathbb{R}^d)}$ the following relationship holds:

$$I_i^{(\mathbb{R}^d)} \mapsto I_{i_1}^{(\mathbb{Z}^2)} \cup \dots \cup I_{i_n}^{(\mathbb{Z}^2)}.$$

The inverse problem, when a particular partition in the spatial domain results in more than one partition in the spectral domain, is not desirable. The image obtained by color clustering is then over-segmented.

The main power of data clustering in segmentation applications is the use of feature vectors to describe data. For a unique clustering technique there are infinite ways to represent the data used as input, leading generally to different results. The feature extraction is therefore an important step in the whole process. For color segmentation applications it will be the choice of an appropriate color space. Finally, a clustering technique usually makes no assumption about the kind of data that must be processed and can be applied to a large number of problems. Clustering is used in numerous applications, such as astronomy [13], statistical data analysis, color and texture segmentation [14], patient discrimination for medical applications, etc.

In the next paragraphs different phases and strategies used for data clustering will be reviewed. Different approaches to *probability density estimation* are presented in § 4.3.2 and a short review of some important clustering strategies is presented in § 4.3.3.

4.3.2 Probability density estimation

Statistical data clustering is based on a statistical model of the data distribution. It is therefore important to choose a model that fits precisely the real distribution in order to obtain a partition that is statistically coherent. This task is called *probability density estimation* and has different applications, such as the *posterior probability estimation*¹ used to build classifier systems. In the context of neural networks, it can be applied to the input data distribution as part of the training process or provide a method for validating the output of a trained neural network. In the framework of our study we want to fit an assumed model to the data that must be classified. There are different approaches, the most complex being the *non-parametric* approach which does not assume any particular functional form. The number of parameters will grow with the size of the data set, so that the models can quickly become unwieldy. The opposite approach is the *parametric* approach where a specific functional form is assumed, the Gaussian function being the most widely used. The simplicity of this approach often result in an incomplete representation of the real density.

An example of non-parametric approach is the histogram representation, which is a particular d-dimensional partition of the input data. A histogram is discrete by nature, which means that there is a limited number of possible values. Every histogram position is called a *bin* and counts the number of samples that fall into a given interval centered at the bin position. The histogram dimension does not necessarily correspond to the data dimension. When dealing with high dimensional data like multi-spectral images, texture descriptors or even 3-D color vectors, a dimension reduction is usually necessary. This reduction is obtained by *projection of the input data onto a basis of lower dimension*. Linear projections are the easiest to implement and allow for a quick processing. However, the best results are obtained by nonlinear processes. An efficient method called *self-organizing maps* (see Chapter 6) produces a mapping of d-dimensional data to a lower dimension (often 2-D) which is then equivalent to a two-dimensional estimation of a d-dimensional probability density function. However, this kind of approach is based on competition and random selection during the learning phase, which means that the solution is only approximated and the result can not be reproduced precisely. For our application the use of a principal component decomposition (see § 3.3.2) give satisfactory results, with the advantage of a lower computational complexity. For color images whose data is distributed in a finite volume of the color space this decomposition often projects most of the image content on the two principal components. This

¹The probability that a *measurement* \mathbf{x} belongs to a particular class, as used in *Bayes' theorem* [15].

will be illustrated later in Chapter 7.

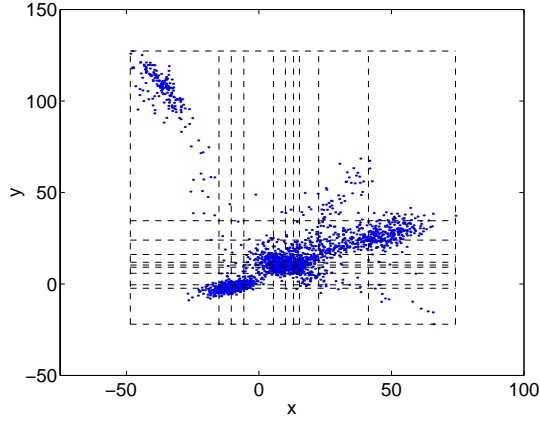


Figure 4.1: 2-D data distribution and the superimposed partition into Parzen windows.

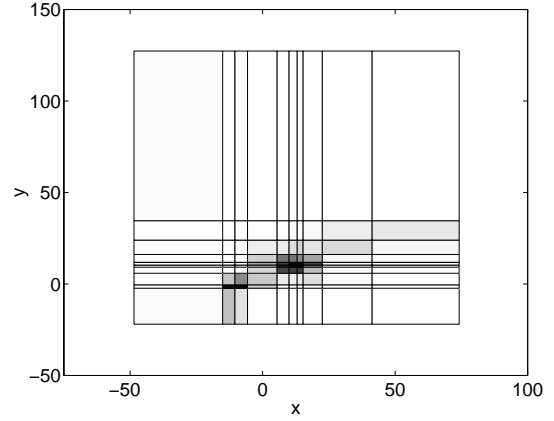


Figure 4.2: Density plot corresponding to the Parzen windows of Fig. 4.1.

Every histogram bin being a local density approximation, it is important to choose appropriate bin centers and bin widths in order to obtain a fine resolution where the data is dense. When using equally spaced bins, the resolution is too fine in sparse regions and too coarse in dense regions. A way of computing the bin centers and widths is to compute the Parzen windows [12] for the data projected onto each component. Every bin counts the same number of samples and the density is then defined by the bin width. Figures 4.1 and 4.2 illustrate this principle. In the former dashed lines are drawn so that they separate the data projection onto both axes into equal number of items. The latter figure shows the corresponding densities (black for high density) and the different regions are called Parzen windows. This approach is especially interesting when the data form clusters distant from each other. The width of every histogram bin depends on the local density, which ensures that (square) regions of high density are sampled with higher precision. The drawback, however, is that the projection onto the different components is not optimal.

Probability density modeling often uses Gaussian functions to represent the distribution of the different data classes. This is an important approach because “natural” data often show to have a normal or close-to-normal distribution. A Gaussian mixture is the sum of two or more Gaussian functions and can be expressed by $\mathcal{G}(\mathbf{x}) = \sum_i \mathcal{G}_i(\mathbf{x})$. In the d-dimensional case a Gaussian

function is given by the following equations:

$$\begin{aligned}\mathcal{G}_i(\mathbf{x}) &= g_i \exp \left\{ -\frac{(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathcal{R}_i^T \boldsymbol{\Sigma}_i^2 \mathcal{R}_i (\mathbf{x} - \boldsymbol{\mu}_i)}{2} \right\}, \\ \mathcal{R}_i &= \begin{pmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_d^T \end{pmatrix}, \\ \boldsymbol{\Sigma}_i &= \begin{pmatrix} \sigma_{i,1}^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{i,d}^{-1} \end{pmatrix},\end{aligned}\tag{4.2}$$

where $\boldsymbol{\mu}_i$ is the center of the function, or mean value, $\sigma_{i,1}, \dots, \sigma_{i,d}$ the variance along every axis (orientation), and $\mathbf{a}_1, \dots, \mathbf{a}_d$ unit vectors in direction of the different axes. In the 2-D case, \mathcal{R}_i reduces to:

$$\mathcal{R}_i = \begin{pmatrix} \cos(\phi_i) & \sin(\phi_i) \\ -\sin(\phi_i) & \cos(\phi_i) \end{pmatrix}.$$

This *orientation matrix* becomes very complex for higher dimensions. For the first axis one need $d - 1$ angles, for the second $d - 2$, etc., which means that the total number of angles needed in the d -dimensional case to define matrix \mathcal{R}_i is $\sum_{j=1}^d (d - j)$. In 2-D there is 1 angle, 3 in 3-D, 6 in 4-D, etc. It is therefore essential to limit the data dimension when using Gaussian mixture models. If a unique Gaussian function must be fitted to given data, the matrix product $\mathcal{R}_i^T \boldsymbol{\Sigma}_i^2 \mathcal{R}_i$ is approximated by the inverse covariance matrix of the data [15]. In the case of mixtures, iterative approaches must be used, as explained below.

For Gaussian densities the functions \mathcal{G}_i are normalized and the coefficients g_i are given by [12]:

$$g_i = \frac{1}{\sqrt{(2\pi)^d |\mathcal{R}_i^T (\boldsymbol{\Sigma}_i^{-1})^2 \mathcal{R}_i|}},$$

where $|\cdot|$ is the matrix determinant. In the case of a mixture density, i.e. a probability density function (PDF) built of several Gaussian distributions, the g_i 's are normalized so that the integral over the whole domain is equal to one.

In the parametric approach the density function is represented as a linear superposition of kernel functions, typically Gaussian functions. This type of densities are not only used in the parametric approach to probability density estimation but also in the *semi-parametric* approach, where a very general class of functional forms is allowed but where the number of parameters can be varied independently from the data size. Now we shall consider models in which the density function is again formed from a linear combination of basis functions, but where the number of functions is treated as a parameter of the model. For the density this model can then be written in the form [15]:

$$p(\mathbf{x}) = \sum_{i=1}^c p(\mathbf{x}|i) P(i),\tag{4.3}$$

where $p(\mathbf{x}|i)$ is the PDF associated with component i of the mixture and $P(i)$, called *mixing parameter*, is the *prior* probability that a data point has been generated from component i . These different probabilities are defined in the interval $[0, 1]$ and have normalized values so that

$\sum_{i=1}^c P(i) = 1$ and $\int p(\mathbf{x}|i) d\mathbf{x} = 1$. Using Bayes' theorem the corresponding *posterior* probability is given by [15]:

$$P(i|\mathbf{x}) = \frac{p(\mathbf{x}|i)P(i)}{p(\mathbf{x})},$$

where $p(\mathbf{x})$ is given by Eq. 4.3 and with $\sum_{i=1}^c P(i|\mathbf{x}) = 1$. This posterior probability represents the probability that a particular \mathbf{x} has been generated by component i .

Different training methods have been developed for mixture models, especially when the kernel function is Gaussian. The approach described next is based on maximizing the likelihood of the parameters for the given input data [16]. We will assume that the $p(\mathbf{x}|i)$ densities are given by Eq. 4.2 with an orientation matrix equal to the identity matrix of same size and with the same variance σ_i for all dimensions. This simplification improves the legibility while the results can be easily extended to general Gaussian functions.

Defining the *negative log-likelihood* for the input data set of size N as:

$$\begin{aligned} E &= -\log \mathcal{LH} \\ &= -\sum_{n=1}^N \log p(\mathbf{x}_n) \\ &= -\sum_{n=1}^N \log \left\{ \sum_{i=1}^c p(\mathbf{x}_n|i)P(i) \right\}, \end{aligned}$$

maximizing the likelihood $\mathcal{LH} = \prod_{n=1}^N p(\mathbf{x}_n)$ is then equivalent to minimizing E , which can be regarded as an error function. The minimization of E leads to the following solutions [15]:

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{n=1}^N P(i|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^N P(i|\mathbf{x}_n)}, \quad (4.4)$$

$$\hat{\sigma}_i = \frac{1}{d} \frac{\sum_{n=1}^N P(i|\mathbf{x}_n) \|\mathbf{x}_n - \hat{\boldsymbol{\mu}}_i\|^2}{\sum_{n=1}^N P(i|\mathbf{x}_n)}, \quad (4.5)$$

$$\hat{P}(i) = \frac{1}{N} \sum_{n=1}^N P(i|\mathbf{x}_n), \quad (4.6)$$

where $\hat{P}(i)$ is the *amplitude* of component i . As it often occurs in probability density estimation or clustering, the number of components or classes is supposed to be known. One must then find a way to extract automatically this information if an unsupervised processing is wanted.

Equations 4.4, 4.5, and 4.6 represent nonlinear coupled equations and do not provide a direct method for calculating the parameters. By their nonlinear nature they suggest the use of an iterative scheme to find the minima of E . The *expectation-maximization* (EM) algorithm [17] guarantees to decrease the error function at each iteration, provided some care is taken over the way in which the updates are performed. At every iteration Eqs. 4.4, 4.5, and 4.6 are used to compute the new values of $\hat{\boldsymbol{\mu}}_i$, $\hat{\sigma}_i$, and $\hat{P}(i)$ with the previously computed probabilities. Once the new parameters have been obtained, the new probabilities can be computed for the next iteration. The estimation of $\hat{\sigma}_i$ with Eq. 4.5 is done using the new value of $\hat{\boldsymbol{\mu}}_i$.

This example of density estimation is very interesting for data classification as performed in clustering approaches. The main drawback of this technique is the amount of operations needed to compute the different parameters for several iterations and the fact that the convergence is guaranteed for a *local minimum* of E . Also, the result depends on how well the data fits the

assumed distribution. The EM technique can be compared with the *orientation-sensitive fuzzy c-means* introduced in § 4.4.3. The advantage of the latter is the reduced computation time and that it uses only the local orientation to model the distribution.

Clustering is often performed directly with the input data, without prior density estimation. However, the histogram representation can be used to determine the number of clusters and an approximate cluster center location. In order to facilitate its analysis, the histogram is first smoothed out using a diffusion filter or a Gaussian filter kernel. The filter size defines the smoothing degree and therefore the smallest acceptable cluster size. All maxima remaining in the histogram correspond to potential class candidates which must be analyzed and compared to each other in order to make the final class selection. This approach is discussed in Section 4.5.

In the next paragraph, different classical clustering techniques are presented. They do not directly use probability density estimation and require the prior knowledge of the number of clusters, also called *classes*.

4.3.3 Classical clustering techniques

In general the detection of classes and the classification of data must be done without any prior information about the type and distribution of data. A strict definition of *unsupervised clustering* is the *partition of a data set into subsets without any knowledge about the type of data and about the underlying problem*. A less restrictive definition would be to say that *the clustering is unsupervised if the system works without any parameter setting or tuning*. The difference between these two definitions is that in the second case the nature of data is known and that this knowledge can be used for the design of an efficient clustering technique. In our case image pixels constitute the input data and have been previously processed, through filtering or extraction of descriptors, in order to form distinct classes in the data space. Clustering should provide both the cluster centers (centroids) and the classification (partition) of the data in an unsupervised way. We will however assume, up to Section 4.5, that the number of classes or clusters c is known and that the clustering is *supervised*.

The first example of clustering technique is the *c-means* or “*basic*” *ISODATA* algorithm, given by the following iterative procedure [12]:

1. Choose the number of classes c .
2. Choose the cluster centers $\mathbf{v}_i, i = 1, \dots, c$.
3. Classify each sample $\mathbf{x}_j, j = 1, \dots, n$.
4. Recompute the estimates for the cluster centers using the results of 3.
5. If the cluster centers are consistent, stop the process; otherwise go to step 3.

This algorithm has a number of degrees of freedom: the number of classes, the initial cluster centers, the classification criteria, the cluster center computation, and finally the stopping criterion. The choices will depend on the type of problem. Ideally, the initialization should have no influence on the result (which is usually hard to obtain), the classification may be based on a *nearest neighbor* criterion or *Bayesian formulations* [12] for example, the cluster update can be a simple *center of gravity* or a more complex formulation, and the stopping criterion, which is maybe the less critical point here, may be based on a *total variation* measure with threshold. This algorithm converges

to a (local) minimum of the *within-group sum of squares* (WGSS) objective functional $\mathbf{J}_m(\mathbf{U}, \mathbf{v})$ given by:

$$\mathbf{J}_m(\mathbf{U}, \mathbf{v}) = \sum_{k=1}^n \sum_{i=1}^c u_{ik} (d_{ik})^2 ,$$

where $(d_{ik})^2 = \|\mathbf{x}_k - \mathbf{v}_i\|^2$ and the u_{ik} define a hard partition: $u_{ik} = 1$ if \mathbf{x}_k belongs to class i and $u_{ik} = 0$ otherwise. The c-means is an example of *objective function method* to cluster analysis. This kind of approaches allow the most precise but not necessarily more valid formulation of the clustering criterion.

Another strategy known as *Learning Vector Quantization* (LVQ) quantizes input data into codebook values and uses them for classification purpose. LVQ assumes that a set of reference vectors $\{\boldsymbol{\mu}_i \mid i = 1, \dots, c\}$ is available as well as a set of *labeled samples* (input data) $\mathbb{X} = \{\mathbf{x}_j \mid j = 1, \dots, n\}$ that will be used for further refinement through a simple similarity measure like the Euclidean distance. The following iterative procedure is used for LVQ [12]:

1. For every sample in \mathbb{X} find the closest reference vector $\boldsymbol{\mu}_i$ which is then noted $\boldsymbol{\mu}_c$.
2. Update $\boldsymbol{\mu}_c$ at iteration t as follows:

$$\boldsymbol{\mu}_c(t+1) = \boldsymbol{\mu}_c(t) \pm \alpha(t)(\mathbf{x}_j - \boldsymbol{\mu}_c(t)) ,$$

where the second right term is added if the sample \mathbf{x}_j is labeled with the class corresponding to $\boldsymbol{\mu}_c$, and subtracted otherwise. The other reference vectors are left unchanged.

3. Continue with the next element of \mathbb{X} .

The $\alpha(t)$ parameter is used to control the convergence and should decrease monotonically. For stability $0 < \alpha(t) < 1$. This technique is intuitively appealing: when the closest reference vector is of the same class, then it is moved towards \mathbf{x}_j . If it is not the case, it is moved in the opposite direction. This kind of algorithm is based on *competitive concepts*, like the *Self-Organizing Maps* described in Chapter 6. It is important to note that this approach will not converge to the same result when the data samples are taken in a different order. The solution is therefore only approximated.

An interesting method for data clustering which is very similar to LVQ was proposed in [18] for application to color image segmentation. Like for vector quantization a competitive approach is proposed for the generation of reference vectors. The number of vectors must not be known in advance, which constitutes the main feature of the proposed algorithm. However, it is not clear from this paper how the number of classes converges to the exact number of clusters. The algorithm proposed in [18] is the following (using the same notation as before):

1. The algorithm starts with a unique centroid (called *unit*) $\mathbf{v}_1 = \boldsymbol{\mu}$:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k ,$$

and a variable *wincount* initialized to zero that counts how many times each unit wins. A second variable N_R is used to count the number of iterations and is initialized to zero.

2. A vector $\mathbf{x} \in \mathbb{X}$ is randomly selected. The closest unit, based on the Euclidean distance, is taken as winner and written \mathbf{v}_w . If there are two or more winners, choose a unit randomly.

3. Vector \mathbf{v}_w is updated as follows:

$$\mathbf{v}_w(t+1) = \mathbf{v}_w(t) + \gamma(\mathbf{x} - \mathbf{v}_w(t)) ,$$

and the corresponding *wincount* is incremented. N_R is also incremented.

4. If the winner's *wincount* variable has reached a given threshold θ_t and the number of units is smaller than c , then a new unit is generated with same centroid and both *wincount* variables are set to zero.
5. If N_R is equal to a threshold N_{max} then stop. Otherwise go to step 2.

The authors propose to set N_R based on the slowest and fastest cases for generating exactly $c - 1$ classes (one class exists from beginning):

$$((c - 1) - 1)\theta_t \leq N_R < (2(c - 1) - 1)\theta_t ,$$

which means that one has exactly c classes when $N_R \geq (2c - 3)\theta_t$.

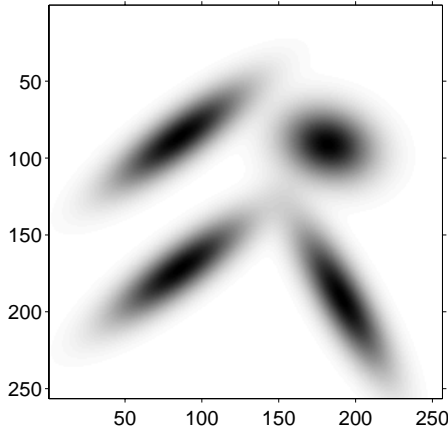


Figure 4.3: Mixture of four well differentiated Gaussian distributions.

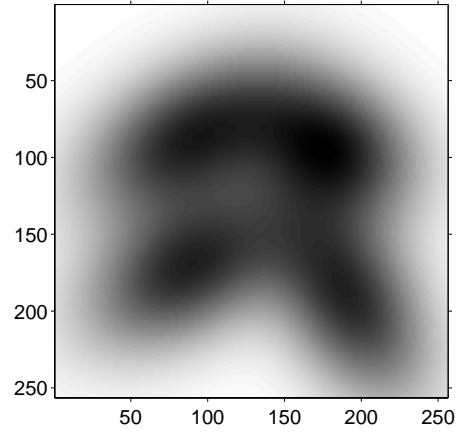


Figure 4.4: Mixture of four Gaussian distributions with large variance parameters.

This algorithm has the advantage of avoiding that several centroids converge to a region of high density. But on the other hand it generates clusters that do not exist but are necessary to split dense regions into smaller ones.

All the clustering algorithms presented in this paragraph share a common characteristic: they are iterative processes. While the c-means technique attempts to minimize an objective functional, the competitive approaches produce a kind of probability density estimation with a small number of reference vectors (centroids). Dense regions will have the tendency to attract the different centroids. On the other hand, competitive approaches have a non-negligible advantage in terms of computation time. They are usually very fast and may provide initial centroids to a more complex and objective approach like the c-means or better the fuzzy c-means presented in the next section.

Figures 4.3 and 4.4 show two Gaussian mixtures used to illustrate the clustering techniques described in this and the following paragraphs. The former shows well separated clusters while

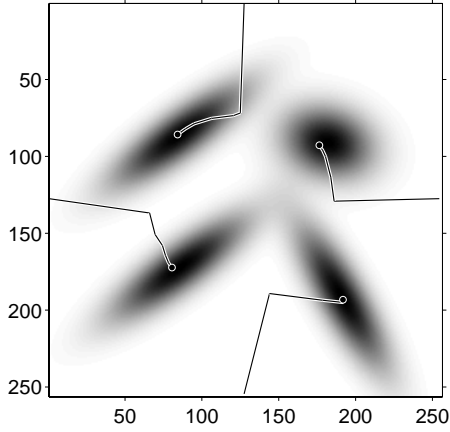


Figure 4.5: C-means clustering of sparse Gaussian mixture.

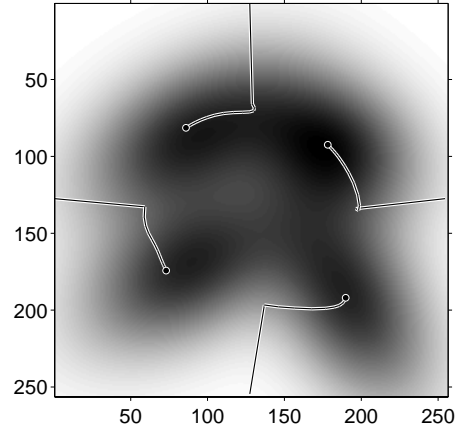


Figure 4.6: C-means clustering of dense Gaussian mixture.

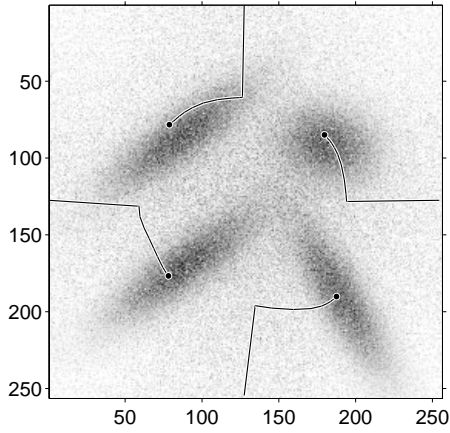


Figure 4.7: C-means clustering of noisy Gaussian mixture.

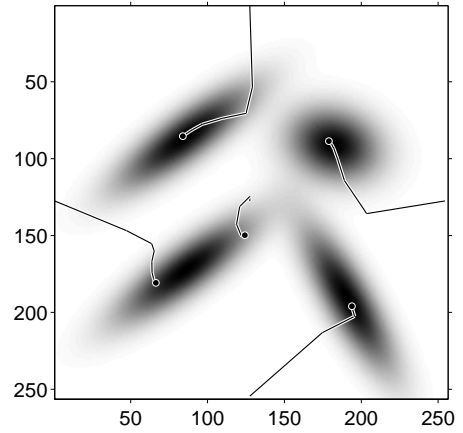


Figure 4.8: C-means clustering of Gaussian mixture initialized with more centers.

in the second mixture the different clusters overlap each other. Figures 4.5 to 4.8 illustrate the c-means clustering technique for later comparison with the fuzzy c-means. In Fig. 4.5 the sparse Gaussian mixture is successfully clustered. The dark lines show the cluster center (centroids) trajectories, the initial positions have been chosen outside any cluster and the classification is based on a *nearest-neighbor* criterion. The cluster centers are the centers of gravity obtained for each partition. In this ideal case the algorithm shows acceptable convergence. For the dense Gaussian mixture the result is less accurate, as it can be seen in Fig. 4.6, which is certainly due to the simple functional used in this algorithm. Figure 4.7 shows the result obtained with a more realistic data distribution, obtained from the distribution of Fig. 4.3 with added Gaussian noise. Even if the result is not precise enough to be acceptable, the c-means converge towards a solution

which is surprisingly close to the real cluster centers despite the unfavorable initial cluster centers. Finally Fig. 4.8 shows the result obtained when more centers than clusters are used.

4.4 Fuzzy c-means clustering technique

4.4.1 Basic algebra of fuzzy sets and fuzzy partition space

Uncertainty is the source of paradoxes in modern science where the amount and effects of uncertainty in mathematical models must be captured precisely. One can identify three main causes of uncertainty [19]:

- inaccurate measurements,
- random occurrences,
- vague description.

These three categories provide an adequate way to describe the manifestation of uncertainty in deterministic, probabilistic, and fuzzy models respectively. A process is *deterministic* if its outcome can be predicted with absolute certainty, upon replication of the circumstances defining it. *Stochastic* or *probabilistic* models exhibit uncertainty that arises when the outcome of a physical process is believed to be random. In this case there is a probability or chance concerning a particular evolution of the process which is unaffected by environmental imprecision.

Let us consider the question “is this color nearly brown?”. There is a little doubt concerning the answer, yes or no. In addition to the abovementioned ones, there is an element of uncertainty which is not caused by measurement error, nor by random occurrence. The adjective *nearly* causes uncertainty as to how one determines an appropriate response, which means that we must devise first a model of “nearly brown”, and then elicit from it enough information to infer a decision. This physical situation manifests a source of non-statistical uncertainty or *fuzziness*.

Let us define \mathbb{S} as a set of n monochromatic color stimuli. In a conventional approach, one would associate with the color brown a central wavelength λ and an interval of width $\Delta\lambda$. Two possible sets of solutions to our previous question would be:

$$\mathbb{A}_1 = \{q \in \mathbb{S} \mid c(q) = \lambda\} ,$$

and

$$\mathbb{A}_2 = \{q \in \mathbb{S} \mid c(q) = \lambda \pm \Delta\lambda/2\} ,$$

where $c(q)$ is the wavelength of color stimulus q . These two subsets are hard sets because the underlying models have hard decision limits. If one assigns to every stimulus a probability $P(q)$ of being in \mathbb{A}_2 this would measure the chance that a specific stimulus is in the interval $[\lambda - \Delta\lambda/2, \lambda + \Delta\lambda/2]$. But it would not preclude the possibility that the wavelength correspond to a monochromatic blue, for example.

Another type of model might be even more appropriate for this kind of problems. Zadeh suggested the following approach [20]: since set membership is the key to our decision, let us alter our notion of sets when the process suggests it, and proceed accordingly. Let us define:

$$\mathbb{A}_3 = \{q \in \mathbb{S} \mid q \text{ is nearly brown}\} .$$

This is not a conventional hard set and there is no set-theoretic realization for it. We can, however, think of a function-theoretic representation by a function $u_3 : \mathbb{S} \mapsto [0, 1]$, whose values give the *grade of membership* of q in the fuzzy set u_3 . A possible definition of this set could be:

$$u_3(q) = \begin{cases} 1 & \text{if } \lambda_i \leq c(q) \leq \lambda_{i+1} , \\ 0.95 & \text{if } \lambda_{i-1} \leq c(q) \leq \lambda_i \text{ or } \lambda_{i+1} \leq c(q) \leq \lambda_{i+2} . \\ \vdots & \vdots \end{cases}$$

If a value of $u_3(q)$ is given we have in this model the capacity for rendering qualitative judgments about relative colors.

Following Zadeh's original idea and using the notation of § 4.3.3, a *fuzzy subset* of \mathbb{X} can be seen as a function $u : \mathbb{X} \mapsto [0, 1]$. We assume that data set \mathbb{X} is a subset of a d -dimensional vector space of finite length n : $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$. Every \mathbf{x}_k is called a *feature* or *pattern vector*.

The *hard c-partition space* for \mathbb{X} is the set:

$$\mathbb{M}_c = \left\{ \mathbf{U} \in \mathbb{V}_{cn} \mid u_{ik} \in \{0, 1\} \quad \forall i, k ; \sum_{i=1}^c u_{ik} = 1 \quad \forall k ; 0 < \sum_{k=1}^n u_{ik} < n \quad \forall i \right\} ,$$

where c is the number of classes ($c > 2$) and \mathbb{V}_{cn} is the set of real $c \times n$ matrices. Similarly, the *fuzzy c-partition space* is the set:

$$\mathbb{M}_{fc} = \left\{ \mathbf{U} \in \mathbb{V}_{cn} \mid u_{ik} \in [0, 1] \quad \forall i, k ; \sum_{i=1}^c u_{ik} = 1 \quad \forall k ; 0 < \sum_{k=1}^n u_{ik} < n \quad \forall i \right\} .$$

The hard c -partition space is the set of all possible hard partitions for a data set \mathbb{X} in the case of hard partitions, as used in the c -means clustering algorithm. \mathbb{M}_{fc} is the equivalent in the fuzzy sets theory. For a complete description of fuzzy sets properties the reader should refer to [19].

The solution to our clustering problem is in \mathbb{M}_{fc} and we need now to define a “selection” criterion. Let us define the *fuzzy c-means functionals* $\mathbf{J}_m : \mathbb{M}_{fc} \times \mathbb{R}^{c \cdot d} \mapsto \mathbb{R}^+$:

$$\mathbf{J}_m(\mathbf{U}, \mathbf{v}) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2 , \quad (4.7)$$

where $\mathbf{U} \in \mathbb{M}_{fc}$ is a fuzzy c -partition of \mathbb{X} and $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_c) \in \mathbb{R}^{c \cdot d}$ contains the cluster centers of u_i , $1 \leq i \leq c$. The distance measure is $(d_{ik})^2 = \|\mathbf{x}_k - \mathbf{v}_i\|^2$ and $\|\cdot\|$ is any inner product induced norm on \mathbb{R}^d , and $m \in [1, \infty)$ is a weighting exponent. \mathbf{J}_m is a squared error clustering criterion and solutions of

$$\min_{\mathbb{M}_{fc} \times \mathbb{R}^{c \cdot d}} \{\mathbf{J}_m(\mathbf{U}, \mathbf{v})\} , \quad (4.8)$$

are least-squared error stationary points of \mathbf{J}_m . An infinite family of fuzzy clustering algorithms (one for each m) is obtained via necessary conditions for solutions of Eq. 4.8. The basic theorem that follows [19] defines the fuzzy c -means clustering technique. This clustering algorithm is described in the next section.

4.4.2 Fuzzy c-means

For the fuzzy c -means functionals defined by Eq. 4.7 a global minimum can be established under specific conditions on u_{ik} [19]. The fuzzy c -means clustering algorithm approximates a solution

of the minimization problem given by Eq 4.8 and uses the two following recurrent equations in a similar way as described in § 4.3.3 for the ISODATA algorithm:

$$u_{ik} = \left(1 + \|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{2}{m-1}} \sum_{\substack{j=1 \\ j \neq i}}^c \frac{1}{\|\mathbf{x}_k - \mathbf{v}_j\|^{\frac{2}{m-1}}} \right)^{-1}, \quad (4.9)$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^n (u_{ik})^m \mathbf{x}_k}{\sum_{k=1}^n (u_{ik})^m}, \quad (4.10)$$

where \mathbf{x}_k is the k^{th} sample vector, n the number of sample vectors, c the number of classes, \mathbf{v}_i the i^{th} class center, u_{ik} the fuzzy membership of \mathbf{x}_k to class i , and $m \in [1, \infty)$ is the weighting exponent. $\|\cdot\|$ is any inner product norm of the form $\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{A} \mathbf{x}$, \mathbf{A} being a positive definite matrix, i.e. \mathbf{A} is symmetrical and its eigenvalues are all non-negative. The weighting exponent m defines the fuzziness of the membership values. Bezdek demonstrated that this algorithm always reaches (in theory) a strict *local minimum* for different initializations of \mathbf{v}_i [21].

When exponent m tends to infinity, the fuzzy membership becomes constant and equal to $1/c$, except if the sample vector \mathbf{x}_k is equal to a cluster center. In that case, the fuzzy membership is equal to 1. Therefore large values of m should be avoided, since this increases the fuzziness of the system. Simulations have shown to give very satisfying results with $m = 2$. One should finally notice that a linear transformation of type $a\mathbf{x} + b$ applied to the data does not modify the fuzzy membership, which means that a scaling or translation of the data distribution has no effect on the clustering and on the fuzzy membership values.

The fuzzy membership of a vector to a class depends on the distance from its center to the other classes. Sample vectors located at a given distance from an isolated cluster center will have a larger fuzzy membership than vectors located at the same distance from another cluster center, but surrounded by several other clusters. Equation 4.9 has been written in that form in order to reveal a number of characteristics:

- the summing term being always positive or nil (no negative distances) $u_{ik} \in [0, 1]$,
- the higher the distance to a given cluster center, the lower the fuzzy membership. However, the effect is weighted by the inverse sum of distances to the other cluster centers.

In order to be statistically coherent, it is not only necessary that $u_{ik} \in [0, 1]$ but also that for every $\mathbf{x}_k \in \mathbb{X}$ the sum of fuzzy membership to the different classes is equal to one, i.e. the probability that \mathbf{x}_k belongs to one of the existing clusters is necessarily 1. This was one of the conditions that led to the above system [19] and can be verified easily. From Eq. 4.9 this sum is given by:

$$\begin{aligned} \sum_{i=1}^c u_{ik} &= \sum_{i=1}^c \left(\|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{2}{m-1}} \sum_{j=1}^c \frac{1}{\|\mathbf{x}_k - \mathbf{v}_j\|^{\frac{2}{m-1}}} \right)^{-1} \\ &= \sum_{i=1}^c \|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{m-1}{2}} \left(\sum_{j=1}^c \|\mathbf{x}_k - \mathbf{v}_j\|^{\frac{m-1}{2}} \right)^{-1} \\ &= \frac{\sum_{i=1}^c \|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{m-1}{2}}}{\sum_{j=1}^c \|\mathbf{x}_k - \mathbf{v}_j\|^{\frac{m-1}{2}}} \\ &= 1. \end{aligned}$$

It must be outlined that using the FCM is meaningful only when using both Eqs. 4.9 and 4.10. The FCM algorithm must not be used only to partition a set of vectors into different classes with known centers. In that case, using Eq. 4.9 would lead to the same result than simply minimizing the distance $\|\cdot\|$. This can be shown in the following way: the cluster center \mathbf{v}_i that maximizes Eq. 4.9 minimizes the following equation:

$$\begin{aligned} u_{ik}^{-1} &= \sum_{j=1}^c \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|^2}{\|\mathbf{x}_k - \mathbf{v}_j\|^2} \right)^{\frac{1}{m-1}} \\ &= \|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{2}{m-1}} \sum_{j=1}^c \left(\frac{1}{\|\mathbf{x}_k - \mathbf{v}_j\|^2} \right)^{\frac{1}{m-1}}, \end{aligned}$$

and then

$$\begin{aligned} u_{ik}^{-1} &= \alpha u_{ik}^{-1} \\ &= \alpha \|\mathbf{x}_k - \mathbf{v}_i\|^{\frac{2}{m-1}} \sum_{j=1}^c \left(\frac{1}{\|\mathbf{x}_k - \mathbf{v}_j\|^2} \right)^{\frac{1}{m-1}}, \end{aligned}$$

for $i \neq \hat{i}$ and with $\alpha \geq 1$. The summing term being constant for a given k (sample), α is equal to:

$$\alpha = \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|^2}{\|\mathbf{x}_k - \mathbf{v}_{\hat{i}}\|^2} \right)^{\frac{1}{m-1}},$$

which means that $\|\mathbf{x}_k - \mathbf{v}_{\hat{i}}\| \leq \|\mathbf{x}_k - \mathbf{v}_i\|$, $\forall i = 1, \dots, c$. This last relationship would lead to the same classification obtained using the u_{ik} .

This allows one to outline another limitation of the segmentation algorithm by histogram thresholding and fuzzy c-means clustering proposed in [11]: the separation into a coarse and a fine segmentation. The coarse segmentation definitely labels all the pixels lying within a given neighborhood of each maximum location. Since most of the pixels have been labelled before using the FCM, recomputing the cluster centers would make no sense if only the non-labelled pixels were used. If the cluster centers found during the maxima selection are supposed to be precise, and do not need to be recomputed, then the Euclidean distance can be used instead of Eq. (4.9).

The distance function in Eq. 4.9 corresponds to a linear projection of the input data. In the case where color information is used, it is interesting to note that performing a color space change $\tilde{\mathbf{x}} = \mathbf{C}\mathbf{x}$ is equivalent to changing from Euclidean distance ($\mathbf{A} = \mathbf{I}_n$) to $\mathbf{A} = \mathbf{C}^T \mathbf{I}_n \mathbf{C} = \mathbf{C}^T \mathbf{C}$. It seems to be more convenient for this specific application to keep the Euclidean distance for the FCM and to compute the color space transformation before according to the formulas of § 3.3.3, which have a precise and intuitive signification. In addition, nonlinear projections can be obtained by using nonlinear transformations and uniform color spaces.

It has been proven in [21] that the FCM algorithm converges. However, it may stop at a local minimum and even at a saddle point [22]. Solutions to the global minimization can be developed by using methods like *simulated annealing* or *genetic algorithms* (see Section 6.5 for details). Another approach is to find a close solution to initialize the FCM algorithm and is addressed in Section 4.5.

A number of examples are given in this paragraph in order to illustrate the FCM clustering technique. The two Gaussian mixtures of Figs. 4.3 and 4.4 are used to represent the data distribution. In Fig. 4.9 the cluster center trajectory is plotted onto the sparse mixture. The convergence

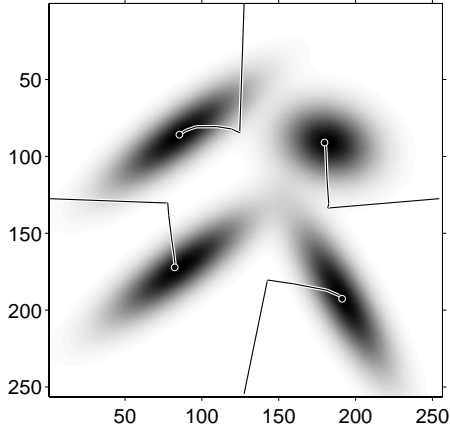


Figure 4.9: Fuzzy c-means clustering of the Gaussian mixture shown in Fig. 4.3.

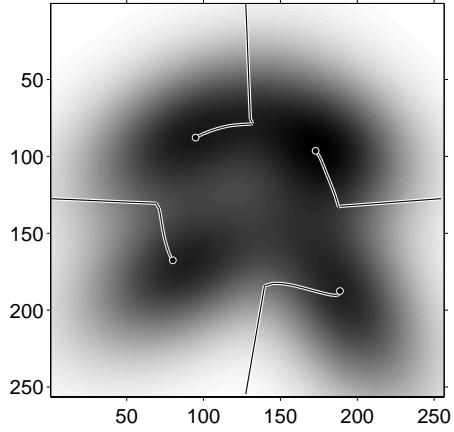


Figure 4.10: Fuzzy c-means clustering of the Gaussian mixture shown in Fig. 4.4.

is quick and precise, as expected for such a distribution. A similar result is obtained for a dense mixture, as shown in Fig. 4.10, with a perceptible improvement compared to the c-means technique. When the distribution is corrupted with noise, the convergence is slow and less precise. A way to control the final cluster center location is to increase the exponent m . Figure 4.11 and 4.12 illustrate the FCM convergence for $m = 2$ and $m = 2.5$ respectively. It must be outlined that this noisy distribution is an unfavorable model where noise is introduced in significant amount outside any cluster. However, the fact that the algorithm converges to a close solution is a clear demonstration of its powerful capabilities.

Figures 4.13 and 4.14 show what happens when the fuzzy c-means algorithm is initialized with respectively less or more classes (than actually present). The case where less centers are used can be better detected because some centers are located outside any dense region. In the example of Fig. 4.13 the left center is in competition with two clusters and finally ends in between. It is also important to note that in that case the system do not converge towards a unique solution and that the final result depends on the initial cluster center positions. When the number of centers is larger than the number of clusters, the additional centers will share a cluster with one or more other centers, depending on the cluster density. This is illustrated in Fig. 4.14.

The main drawback of the FCM technique lies in the poor data partition. Figures 4.15 and 4.16 show respectively the maximum fuzzy membership at every histogram location and the final partition. The former figure makes visible the cluster center and the border between neighboring clusters. The use of a nearest-neighbor criterion is sub-optimal and one loses all geometrical information about the different clusters: the “natural limits” are not respected. The limitation of the FCM technique, even if other distances than the Euclidean distance can be used, appears clearly in Fig. 4.16: oriented clusters with $\sigma_1 \gg \sigma_2$ are cut and part of their data is assigned to a neighboring cluster. The current FCM formulation is a powerful mean of computing cluster centers but performs poorly for data partition.

Since the first fuzzy generalization of the ISODATA clustering technique [23], several papers have addressed the convergence problem [21, 22], efficient implementations [24, 25], proposed

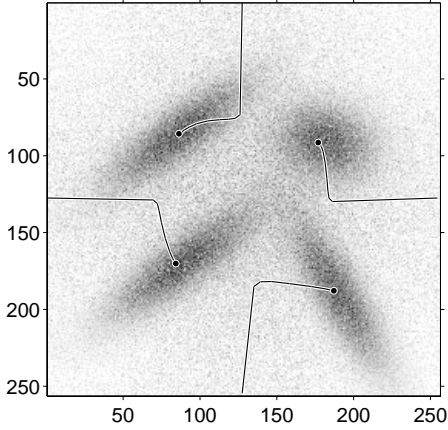


Figure 4.11: Fuzzy c-mean clustering of noisy Gaussian mixture ($m = 2$).

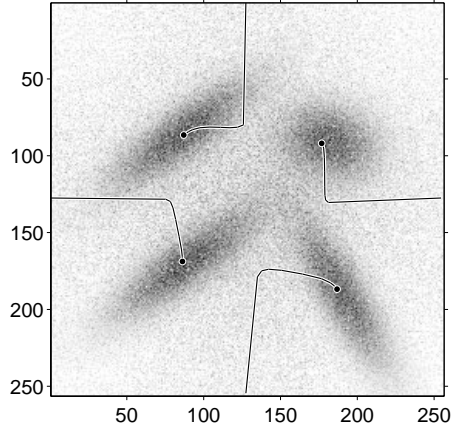


Figure 4.12: Fuzzy c-mean clustering of noisy Gaussian mixture ($m = 2.5$).

different extensions and improvements [25, 26, 27], as well as a large number of application fields [28, 29, 30]. The way the distance d_{ik} between data and the different cluster centers is computed furnishes a large number of alternatives to the original FCM formulation. Knowing that Euclidean distance, which is often used, is a particular case of the Minkowski distance $d_{ik} = \left(\sum_{l=1}^d (x_{kl} - v_{il})^p \right)^{1/p}$ with $p = 2$, then different formulations can be used for different values of parameter p . In [26] the L_1 -norm based fuzzy clustering is derived and corresponds to the case where $p = 1$. The objective function and the formula for centroids in the optimal solution are derived. However, this kind of propositions is interesting from a theoretical point of view but often do not lead to a relevant improvement.

In [27] the author proposes to use *genetic algorithms* (see Chapter 6) to optimize the performances of the FCM algorithm in terms of speed and error reduction for color segmentation applications. This thesis describes mainly an interesting way to establish the number of clusters without prior knowledge. The author uses also a histogram representation of data to access a number of parameters, namely the approximate number of clusters and the initial centroids.

A significant limitation of the previous formulations is the lack of geometrical parameters that take into account the spectral orientation of data clusters. In the next paragraph an improved FCM algorithm is proposed that takes into account the local orientation during the iterative process.

4.4.3 Orientation sensitive FCM

The advantage of using the FCM technique is its ability to compute precisely the cluster centers. Once they have been obtained, the input must be labelled in order to generate the segmented image. The limit between two classes is located at half-distance from both centers, as explained in the previous section. When the data distribution contains oriented clusters, then the limit between the different clusters should depend not only on their center but also on their orientation.

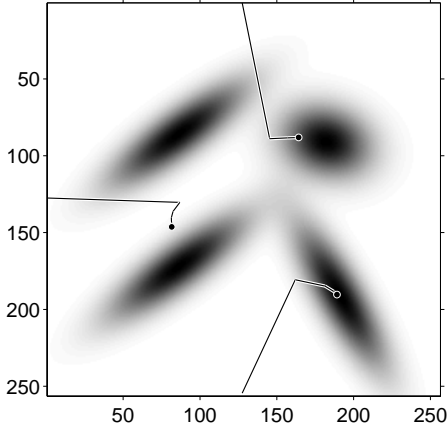


Figure 4.13: Fuzzy c-means clustering initialized with less centroids than existing clusters.

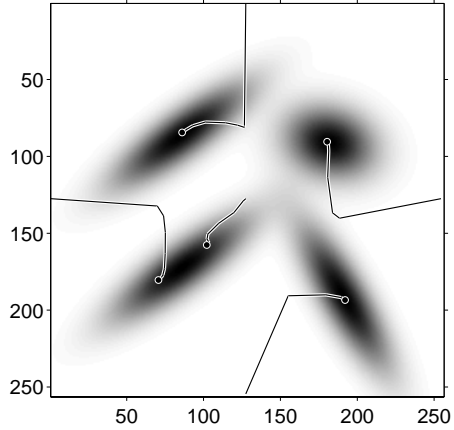


Figure 4.14: Fuzzy c-means clustering initialized with more centroids than existing clusters.

Let us rewrite Eq. 4.9 in the following way:

$$u_{ik} = \left(1 + \sum_{\substack{j=1 \\ j \neq i}}^c \left(\frac{(\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i)}{(\mathbf{x}_k - \mathbf{v}_j)^T \mathbf{A}_j (\mathbf{x}_k - \mathbf{v}_j)} \right)^{\frac{1}{m-1}} \right)^{-1}, \quad (4.11)$$

where the matrices \mathbf{A}_i used for the distance computation take into account the local orientation. Let us introduce the *fuzzy covariance matrix* \mathbf{C}_i , defined as:

$$\mathbf{C}_i = \frac{\sum_{k=1}^n (u_{ik})^m (\mathbf{x}_k \mathbf{x}_k^T - \mathbf{v}_i \mathbf{v}_i^T)}{\sum_{k=1}^n (u_{ik})^m},$$

for each cluster i . Let \mathbf{L}_i be a diagonal matrix containing the inverse of the eigenvalues of \mathbf{C}_i and \mathbf{V}_i the matrix whose lines are the corresponding eigenvectors. We can then write:

$$\mathbf{A}_i = \mathbf{V}_i^T \mathbf{L}_i \mathbf{V}_i. \quad (4.12)$$

The iterative process using Eqs. 4.11, 4.12, and 4.10 is called the *orientation-sensitive fuzzy c-means* (OS-FCM). This modification of the initial FCM algorithm shows a significant improvement in both the final cluster center location and the fuzzy partition for oriented clusters. The distance matrix given by Eq. 4.12 projects the vectors going from the cluster centers to the data onto the local orientation axes. These axes are weighted by the inverse of the corresponding data variance in order to take into account the shape of the cluster. One can see the clusters as ellipses with axes of length equal to the two eigenvalues of matrix \mathbf{C}_i . All samples located on an ellipse whose axes are multiple of both eigenvalues are then at an equal distance to the center.

Using the relationship between the eigenvalues $\lambda_{i,j}$ and the corresponding eigenvectors $\mathbf{p}_{i,j}$ of

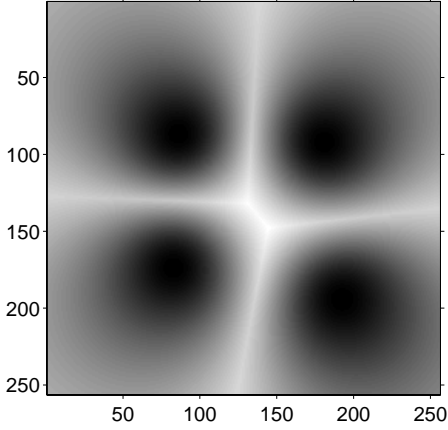


Figure 4.15: Maximum fuzzy membership for every bin of the 2-D histogram.

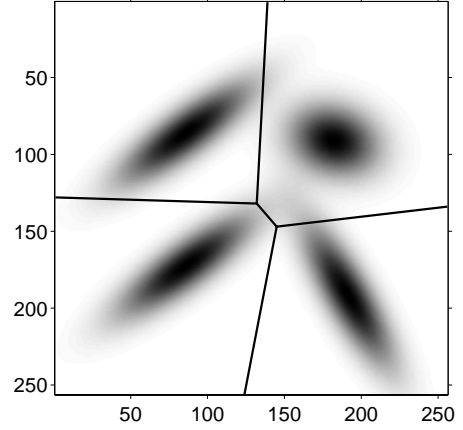


Figure 4.16: Hard partition obtained from the fuzzy memberships displayed in Fig. 4.15.

matrix \mathbf{C}_i , that is $\mathbf{C}_i \mathbf{p}_{i,j} = \lambda_{i,j}$, the following equation can be derived:

$$\mathbf{C}_i \begin{pmatrix} \mathbf{p}_{i,1} & \cdots & \mathbf{p}_{i,d} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{i,1} & \cdots & \mathbf{p}_{i,d} \end{pmatrix} \begin{pmatrix} \lambda_{i,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{i,d} \end{pmatrix},$$

where d is the data dimension. Using the notation of Eq. 4.12 the above equation can be rewritten as:

$$\mathbf{C}_i \mathbf{V}_i^T = \mathbf{V}_i^T \mathbf{L}_i^{-1}.$$

Since \mathbf{C}_i is a symmetrical matrix its eigenvectors form an orthonormal basis and matrix \mathbf{V}_i is therefore orthogonal. The consequence is that $\mathbf{V}_i^{-1} = \mathbf{V}_i^T$, and finally:

$$\begin{aligned} \mathbf{C}_i &= \mathbf{V}_i^T \mathbf{L}_i^{-1} \mathbf{V}_i, \\ \mathbf{C}_i^{-1} &= \mathbf{V}_i^T \mathbf{L}_i \mathbf{V}_i. \end{aligned}$$

The comparison with Eq. 4.12 leads to the following relationship:

$$\mathbf{A}_i = \mathbf{C}_i^{-1}. \quad (4.13)$$

The OS-FCM need the inverse fuzzy covariance matrix to compute the distance to every cluster center. The algorithm is initialized with identity matrices of same size so that the first fuzzy membership computation uses Euclidean distance.

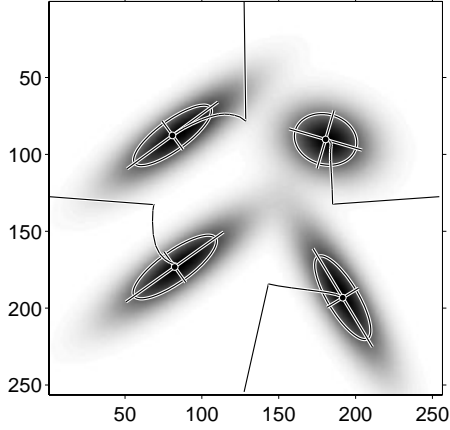


Figure 4.17: OS-FCM clustering of the Gaussian mixture shown in Fig. 4.3.

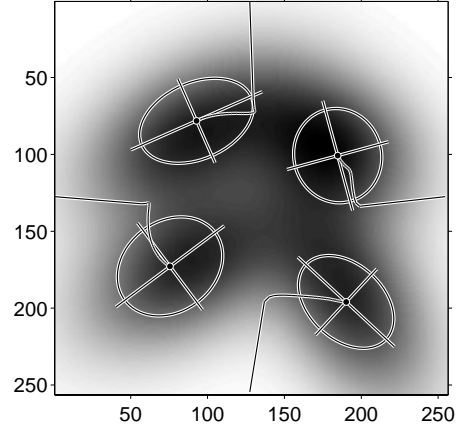


Figure 4.18: OS-FCM clustering of the Gaussian mixture shown in Fig. 4.4.

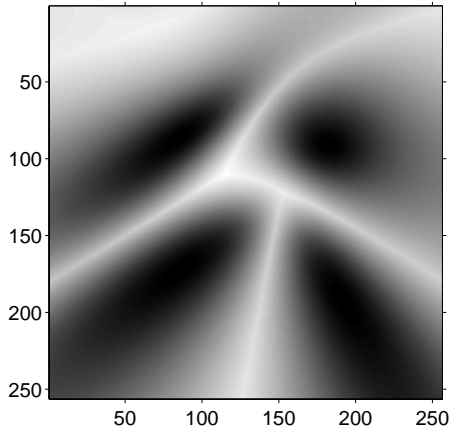


Figure 4.19: Maximum fuzzy membership of the OS-FCM computed for every bin.

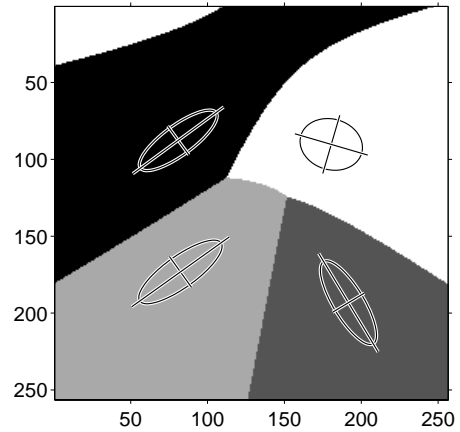


Figure 4.20: Hard partition obtained from the fuzzy memberships displayed in Fig. 4.19.

Figures 4.17 and 4.18 show the cluster center evolution for the sparse and dense mixtures, respectively. In the former case the algorithm converges very well and even shows a slight improvement in the cluster center computation compared to the classical FCM algorithm. The dense mixture of Fig. 4.18 is difficult to cluster because the different centroids are strongly influenced by neighboring clusters. Especially the fuzzy covariance matrices reflect this influence. The consequence is an increased variance and a modification of the orientation. The real advantage of the OS-FCM becomes visible when computing the data partition. Figures 4.19 and 4.20 show respectively the maximum fuzzy membership for every histogram location and the partition result. The border follows the “natural” separation between neighboring clusters and give a more realistic solution to the partition problem. For performance comparison, the clustering result obtained

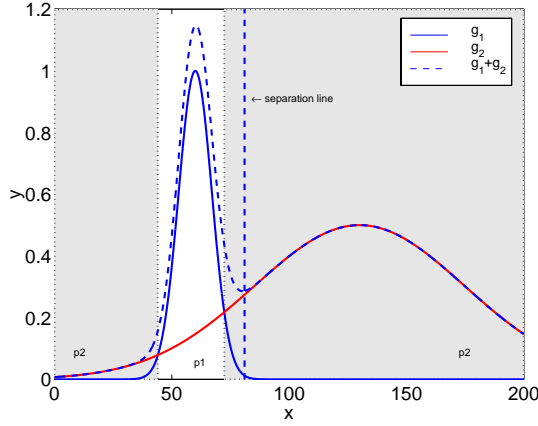


Figure 4.21: 1-D example of hard partition using probability density estimates.

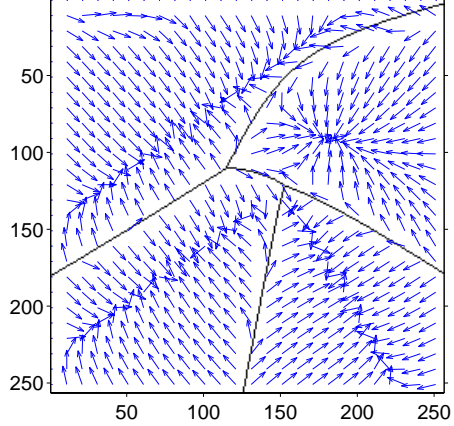


Figure 4.22: Quiver plot of the Gaussian mixture gradient with the superimposed partition.

for the mixture of Fig. 4.3 gives 5.3 % of misclassified data when using the c-means technique. For the FCM the rate is 4.7 % and it finally reduces to less than 0.8 % for the OS-FCM. At the top of Fig. 4.20 a false region appears, due to the weighted distance computation. This effect is illustrated in the one-dimensional case in Fig. 4.21: the data distribution forms two clusters, one large and one narrow. The “natural” limit between clusters is located where both classes have equal probabilities to occur. From a probabilistic point of view this is absolutely correct, but it does not fit our intuition. In the one-dimensional case, the solution would be simple: data between two clusters is classified according to the distributions, data which is not located between two clusters is attributed to the nearest cluster. In dimensions higher than one, it is suggested to use the following rule: every partition that does not contain a centroid is labeled according to the nearest partition, as it is the case in Fig. 4.20. In Fig. 4.22 a quiver plot of the Gaussian mixture gradient is shown, with the superimposed partition. The result fits precisely the “natural” border between the different clusters.

In Fig. 4.23 the OS-FCM are used to cluster a noisy Gaussian mixture. Different values of exponent m have been used in order to reduce the influence of noise but without success. The presence of noise corrupts the OS-FCM, especially in this example which contains much more noise data than in practical applications where an efficient filtering can usually remove most of the noisy pixels. This will be later illustrated in Chapter 7.

The integration of cluster orientation or any other geometrical parameter has been investigated in previous studies [31, 32, 33, 34]. The authors of [31] define the *fuzzy k-ellipses algorithm* for the detection and parameter estimation of elliptic ring-shaped clusters. This algorithm is very similar to the OS-FCM and has proved to be very efficient for the partitioning of disjoint and intersecting compact clusters with elliptic shapes. In [33] an elegant approach to the detection of clusters of distinct geometrical categories is proposed. The idea is to modify the objective function that must be minimized in order to include a number of structures of different order: linear, spherical, quadratic, etc. Each cluster is then considered to be a linear combination of the different prototypes.

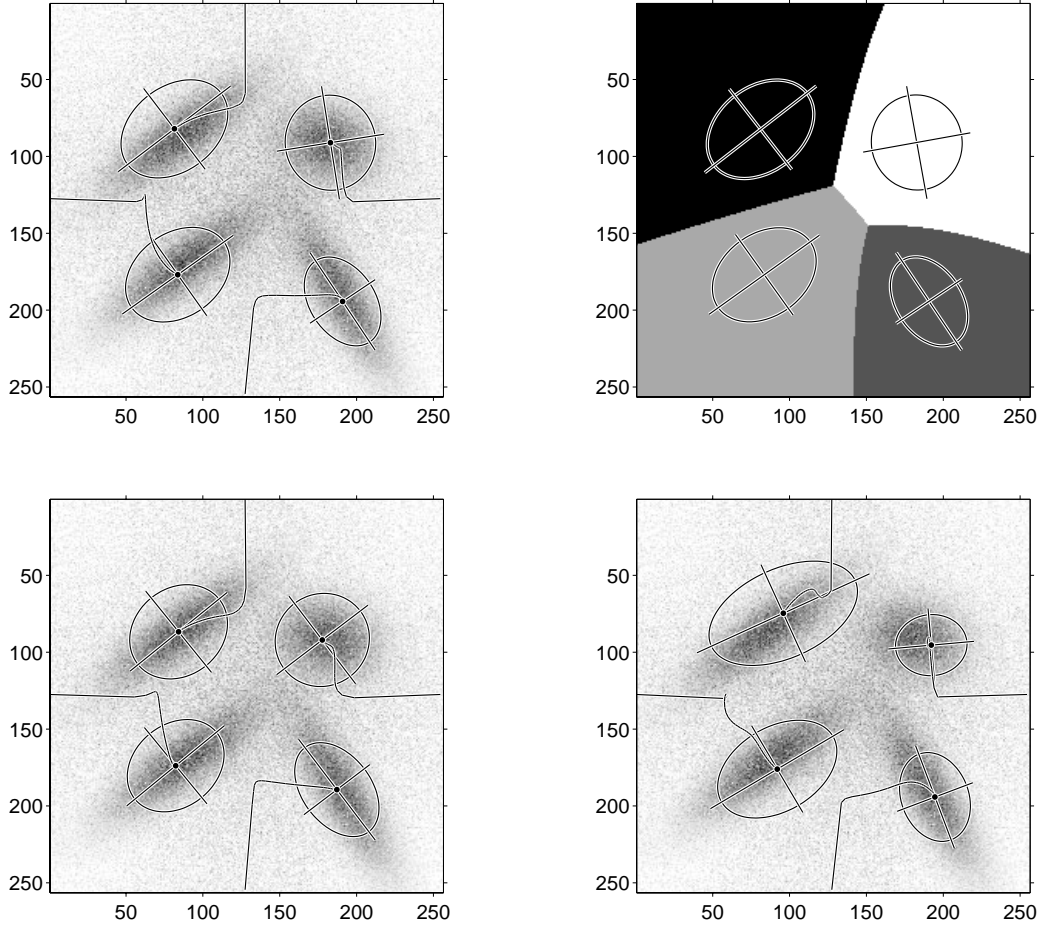


Figure 4.23: OS-FCM clustering of noisy Gaussian mixture with $m = 2$ (top left), $m = 3$ (bottom left), and $m = 1.5$ (bottom right). The top right image shows the partition obtained for $m = 2$.

Now that we have a powerful clustering tool, we need to solve an essential problem: the number of clusters needed to initialize the OS-FCM algorithm. This subject is investigated in Section 4.5. Before, an extension to the FCM technique that reduces the effect of noise (seen as data located outside any valid cluster) is presented and can be integrated, if necessary, to the FCM or OS-FCM algorithms.

4.4.4 Noise reduction

One way to reduce the influence of noise, usually located in the histogram at a large distance from any cluster center, is the use of a virtual noise-class, as proposed in [35]. This class has a floating

center and is used to attract “outliers”. Equation 4.9 must be replaced by [35]:

$$u_{ik} = \left(1 + \left(\frac{\| \mathbf{x}_k - \mathbf{v}_i \|^2}{\delta^2} \right)^{\frac{1}{m-1}} + \sum_{\substack{j=1 \\ j \neq i}}^c \left(\frac{\| \mathbf{x}_k - \mathbf{v}_j \|^2}{\| \mathbf{x}_k - \mathbf{v}_i \|^2} \right)^{\frac{1}{m-1}} \right)^{-1}.$$

The added term is negligible when $\| \mathbf{x}_k - \mathbf{v}_i \|^2$ is small compared to δ^2 . The fuzzy membership is reduced when these two distances are close and even more if $\| \mathbf{x}_k - \mathbf{v}_i \|^2$ is large compared to δ^2 . Decreasing the fuzzy membership of outliers reduces their influence on the class center computation. δ^2 may be computed using the following expression [35]:

$$\delta^2 = \lambda \left(\frac{\sum_{i=1}^c \sum_{j=1}^n \| \mathbf{x}_k - \mathbf{v}_i \|^2}{nc} \right),$$

with n the number of samples and c the number of clusters. The parameter λ is chosen empirically.

4.5 Cluster detection

4.5.1 Introduction

The different clustering techniques described up to now assume that the number of clusters is known. In supervised applications this information can be given by the user. For color segmentation the user can make a visual estimation of the number of color classes but it does not necessarily correspond to different clusters in the data distribution. Considering the data plotted in a d-dimensional histogram is only visually possible for dimensions less or equal to three, and the data dispersion renders the estimation very difficult. A more visible solution consists of using an objective cluster validity measure that ideally allows for the detection or rejection of the different cluster candidates. But this is not sufficient to solve this complex problem. The number of possible partitions of n vectors into c non-empty subsets is [12]:

$$\frac{1}{c!} \sum_{k=1}^c \binom{c}{k} (-1)^{c-k} k^n \approx \frac{c^n}{c!}.$$

For example, a 256×256 pixels image and $c = 4$ sets would have approximately 4×10^{498} possible partitions. Even if most solutions can be eliminated because, for example, the data partition is clearly unfair from the class size point of view, the candidates are still numerous and efficient selection methods must be developed to solve this problem.

The different approaches proposed in the literature are often based on *artificial neural networks* (ANN). In [36] an approach using dynamic competitive learning is presented. One example of competitive learning has been presented in § 4.3.3, namely the c-means clustering technique. More sophisticated strategies were proposed for neural²competition, like Kohonen’s Self-Organizing Maps described in Chapter 6, to avoid the deficiencies of the simple *Winner-Take-All* (WTA) approach used in the c-means algorithm. These deficiencies are mainly due to the fact that only reference vectors close to the input vectors are updated while the others are underutilized. The system is therefore very sensitive to initial reference vector values. The second deficiency is that despite the calculation of the distance for all the reference vectors much information is thrown away by

²The reference vectors used in competitive learning are called *neurons*, in analogy to neural competition that inspired these techniques.

shutting down the activity of all the losing vectors. And finally, competition systems use random selection of input vectors to train the neurons, which means that results can not be reproduced precisely. The two first deficiencies are addressed in [36].

Parametric and non-parametric approaches to probability density estimation have been presented in § 4.3.2. The unsupervised cluster analysis proposed in [37] aims at comparing both approaches for cluster detection applications. The parametric method assumes Gaussian clusters and the non-parametric method uses scale-space filtering. The author shows that the non-parametric approach is more robust when the cluster shape is not Gaussian.

The geometrical approach to cluster selection presented in the next paragraph is a *hierarchical* approach. It is assumed that the cluster geometry is a sign of importance and that geometrical measures can be used to compute the “importance” of each cluster candidate. A graph-based approach is then used to select the valid clusters. This non-parametric approach combined to the OS-FCM constitutes the core of the segmentation scheme developed in this chapter.

4.5.2 Geometrical approach

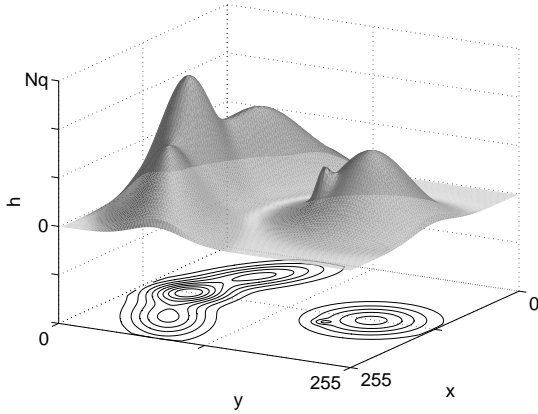


Figure 4.24: Example of two-dimensional histogram with some of the contour lines.

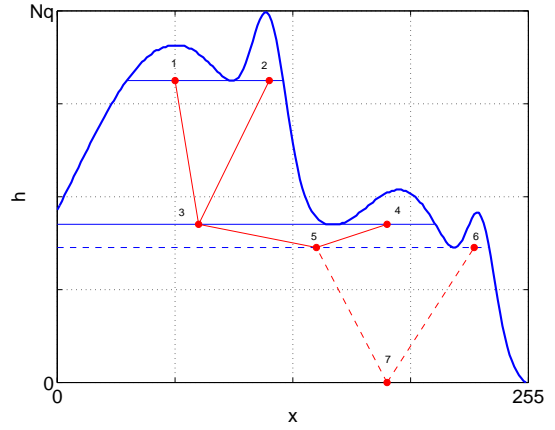


Figure 4.25: Tree representation used to analyze the different histogram contour lines.

From the principal component decomposition given in § 3.3.2 only the first two are used to compute a two dimensional histogram. It is filtered with a Gaussian low-pass filter (a morphological filter may also be used), in order to suppress some of the irregularities and to reduce the number of local maxima, and then linearly quantized on 256 levels in order to store each bin value in a byte type variable. The algorithm extracts the contour lines at each level and the following geometrical features: the number of pixels enclosed by the contour line, its area, position and level, the distance to the top of the maxima it belongs to, as well as its center of gravity, according to the pixel distribution within the region. Figure 4.24 shows an example of 2-D histogram contour lines. A tree representation is used to store the different features, where each single connected node corresponds to a local maximum in the histogram. Every two-connected node is suppressed, since it connects contour lines of a same cluster. When two maxima merge together, they form a new maximum and a node is created, connected to the maxima that merged. This

corresponds to a scale change and constitutes an elegant alternative to the *scale-space* analysis proposed in [11] which requires a significant processing time in 2-D. The tree representation is illustrated in Fig. 4.25.

For every maximum candidate a *strength* value is computed:

$$strength_i = p_i \frac{h_i}{s_i}, i = 1, \dots, l,$$

where p_i is the number of pixels enclosed by the base of local maximum i , s_i is the number of bins enclosed, h_i is the height of the maximum, and l is the number of local maxima. This strength measure is normalized so that the maximum corresponding to the whole histogram has a strength equal to 1. The tree is then analyzed from top (high-level nodes) to bottom (low-level nodes) and the following strategy is used to select active nodes, which correspond to valid clusters:

1. group end nodes connected to a common underlying node,
2. for every set of nodes, suppress those whose ratio to the largest strength value within the set is smaller than a given threshold,
3. sets which end with a unique node are completely removed,
4. if at least one set has been completely removed and if the number of end nodes is larger than two, then return to step 1. Otherwise stop the selection.

This procedure avoids the selection of geometrically insignificant clusters. The threshold used in this approach is meaningful if the data representation uses uniform spaces, in order to allow such a comparison between clusters.

This approach has the following advantages:

- the selected maxima provide a very precise initial value for the cluster centers which speeds up the OS-FCM computation,
- the use of geometrical features to select valid clusters is intuitively meaningful, provided that the data space is nearly uniform,
- this technique can be used in a supervised manner as well, and the user can then modify the cluster selection without additional computation.

Knowing that the FCM and any derived algorithm do not necessary converge towards the global minimum of $J_m(\mathbf{U}, \mathbf{v})$ but towards a local minimum, initial cluster centers close to the final solution guarantee an improved convergence.

A second approach to valid cluster selection can be used, which uses the previous tree representation, but different criteria. The different maxima can be seen as smooth approximations to real clusters, whose strength must be evaluated. Following the fuzzy membership computation, small clusters located far from any other cluster have a higher degree of validity than a cluster of similar shape surrounded by several larger cluster. Therefore the following cluster validity procedure is proposed:

1. Compute the fuzzy membership of all \mathbf{x}_j , $j = 1, \dots, n$, for every end-nodes (one-connected nodes). The center of gravity associated with each node is taken as cluster center
2. For every end-node, compute the sum of fuzzy memberships $s_i = \sum_{k=1}^n (u_{ik})^m$.

3. Suppress the nodes for which $s_i/s_j < threshold$, where s_j is the largest sum of membership. When two end nodes are connected to the same node and at least one is not valid, then both are suppressed. If the suppression of nodes do not result in new end nodes or if all nodes are valid, then stop. Otherwise go to 1.

For example, in the case of Fig. 4.25 nodes 1, 2, 4, and 6 are evaluated first. If only 6 is considered as not valid, then the selection stops and the maxima corresponding to nodes 1, 2, and 4 are used to initialize the FCM clustering. If in the same example nodes 6 and 2 are not valid, then node 1 is also suppressed and both nodes 1 and 2 are replaced by node 3. This new cluster candidate must be evaluated and the selection procedure is repeated.

In the following section the image manipulations necessary to prepare the image for the segmentation process and to clean the result are shortly discussed.

4.6 Preparation and cleaning phases

4.6.1 Pre-processing

The image manipulations that precede the segmentation are very important and influence strongly the final result. It is recommended to filter the image in order to improve the region separation in the spatial domain. The spectral domain will be influenced by the color space used to represent the pixel values. How important both aspects are will be illustrated in Chapter 7.

Different filtering techniques have been presented in Chapter 2 with a particular emphasis on diffusion processes. While these techniques are very well suited for edge detection, they may be time consuming for our segmentation application. Median filtering has very good denoising capabilities which may be sufficient in most of the cases. If, however, the image contains regions with textures that may influence the cluster geometry, anisotropic diffusion is an optimal way to reduce this effect. As said in the previous chapter, denoising must be differentiated from selective smoothing, where not only noise is removed but also “natural” structures that do not carry any useful information for the following processing steps.

4.6.2 Region selection and filling

It may happen that even if the image has been filtered before the clustering step, several small regions, usually not larger than a few pixels, appear in the final result. The selection of such regions can be based on a geometrical criterion (size) or a spectral criterion. In the latter case neighboring regions are compared in terms of *mean colorimetric difference*, a measure that is especially well suited when using uniform color spaces. A simple threshold is used to decide if a region shows sufficient difference in luminance and chrominance when compared to its neighbors. This threshold, since we work with color difference formulas in a uniform color space, can be set empirically.

Two options are possible concerning the regions to remove: reassign every pixel independently to a neighboring class, with the risk that smaller regions are created, or use a spatial approach. In the latter case the pixels belonging to a region can be considered to be definitely bounded, which means that the whole region must merge with one of the neighbors. The approach we propose here is rather to extract the spurious regions and to use mathematical morphology to fill in the “holes” left in the image. Let us quickly review the necessary morphological operators.

It is first necessary to label the different regions present in the segmentation result, since more than one region can share the same class. For that purpose we will use *binary region filling* [1], for every class and until every region has been labeled. Suppose that \mathbb{X}_i is the set of all pixels which belong to class i , with $i = 1, \dots, c$, and that $\mathbb{A}_0 = \{\mathbf{p}\} \subset \mathbb{Z}^2$ is a set containing a unique and non-labeled pixel of class i , then the following operator is used to build the set of all pixels belonging to the region of \mathbf{p} :

$$\mathbb{A}_k = (\mathbb{A}_{k-1} \oplus \mathbb{B}) \cap \mathbb{X}_i, \quad k \in \mathbb{N}^*,$$

where \mathbb{B} is a structuring element, usually a cross (4-connected) or a 3×3 square (8-connected). The algorithm has converged when $\mathbb{A}_k = \mathbb{A}_{k-1}$ and we let then $\mathbb{Y}_l = \mathbb{A}_k$ be a set of pixels which forms a region with label l . In practice this algorithm is implemented in a recursive fashion with a pixel of the segmented image as input. It is compared to the surrounding pixels which are the given, if they are the same, as input to a recursive call to the function if they have the same value.

When all regions have been extracted, which mean that they have a unique label, the selection phase can begin. When spurious regions are detected, their content is removed from the labeled image. Successive gray-scale dilation, which has been defined in Chapter 2, is used to fill in the regions. The structuring element used for this last operation has a cross or square shape. The advantage of taking one of these elementary shapes is that the filling operation is done with a one pixel width at every iteration, which results in a smooth interpolation of region borders that pass through the spurious regions.

4.7 Summary

In this chapter all the necessary phases for an efficient color segmentation scheme based on color clustering have been described. In Section 4.2 a number of image segmentation strategies with a particular emphasis on histogram-based approaches have been presented, as an introduction to the scheme that have been developed throughout this chapter.

Data clustering has been discussed in Section 4.3 to introduce the use of fuzzy sets theory that has been presented in Section 4.4. The use of local orientation in the clustering process has been introduced in § 4.4.3 in order to improve the partition in presence of clusters with a near-Gaussian or elliptic shape. A geometrical approach to the valid cluster selection has been proposed in Section 4.5. It provides both the number of clusters and the initial values for the cluster centroids. Finally the importance of image filtering for denoising purpose and to optimally “prepare” the image for the segmentation process has been discussed in Section 4.6. The suppression of spurious regions has been tackled in the same section and a morphological approach has been proposed for the labeling of regions and the smooth filling of the non-valid regions. The image segmentation scheme that has been presented in this chapter will be later evaluated and illustrated in Chapter 7.

The segmentation scheme can be summarized as follows:

1. Image filtering: the denoising phase is essential for an efficient data clustering. Median filtering should be used when the presence of structures within regions do not influence the segmentation, i.e. structures show to have lower contrast and color variations. Anisotropic diffusion should be used otherwise, but in that case after step 2 if a uniform color space is used.

2. Color space transform: the *RGB* color representation is not adapted to color segmentation applications, even if it is not impossible to obtain satisfactory results with this color representation. The use of a uniform color space like $L^*u^*v^*$ is suggested in order to give a sense to the distance used in the clustering phase and the cluster selection based on geometrical characteristics.
3. Histogram computation and analysis: a 2-D histogram is computed for the two principal components of the $L^*u^*v^*$ data. The histogram is then analyzed following the scheme given in § 4.5.2.
4. Data clustering: the orientation-sensitive fuzzy c-means algorithm has been proposed to integrate cluster orientation in both the cluster center computation and the data partition. This method should be used to avoid significant data misclassification.
5. Region selection: in many applications small spurious regions must be removed.

The different steps, and especially the choice of an appropriate color space and filtering technique, can be adapted to any specific problem, rendering this scheme very flexible.

Bibliography

- [1] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [2] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [3] C. A. Murty and S. K. Pal. Histogram thresholding by minimizing graylevel fuzziness. *Information Sciences*, 60(1-2):107–135, March 1992.
- [4] Chun-Hung-Li and P. K. S. Tam. Modular expert network approach to histogram thresholding. *Journal of Electronic Imaging*, 6(3):286–293, July 1997.
- [5] C. A. Glasbey. An analysis of histogram-based thresholding algorithms. *Graphical Models and Image Processing*, 55(6):532–537, November 1993.
- [6] K. Saarinen. Color image segmentation by a watershed algorithm and region adjacency graph processing. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 3, pages 1021–1025, 1994.
- [7] K. Haris, S. N. Efstratiadis, N. Maglaveras, and C. Pappas. Hybrid image segmentation using watersheds. In *Proceedings of the SPIE*, volume 2727, pages 1140–1151, 1996.
- [8] X. Wu. Adaptive split-and-merge segmentation based on piecewise least-square approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):808–815, August 1993.
- [9] S. Beucher. Watersheds of functions and picture segmentation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1921–1931, 1982.
- [10] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textures color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):939–954, October 1995.
- [11] Y. W. Lim and S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9):935–952, 1990.
- [12] R. Schalkoff. *Pattern recognition: statistical, structural and neural approaches*. Wiley, 1992.
- [13] C. McGill and H. M. P. Couchman. Detecting clusters in galaxy catalogs. *Astrophysical Journal*, 364:426–432, December 1990.

- [14] Ph. Schroeter and J. Bigün. Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement. *Pattern Recognition*, 28(5):695–709, 1995.
- [15] Ch. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1997.
- [16] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):295–239, 1984.
- [17] A. P. Dempster, N. M. Laird, and D. B. Rudin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39(1):1–38, 1977.
- [18] T. Uchiyama and M. A. Arbib. Color image segmentation using competitive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1197–1206, December 1994.
- [19] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
- [20] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338–353, 1965.
- [21] J. C. Bezdek. A convergence theorem for the fuzzy ISODATA clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(1):1–8, January 1980.
- [22] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker. Convergence theory for fuzzy c-means: counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5):873–877, September 1987.
- [23] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.
- [24] R. L. Cannon, J. V. Dave, and J. C. Bezdek. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):248–255, March 1986.
- [25] M. S. Kamel and S. Z. Selim. New algorithms for solving the fuzzy clustering problem. *Pattern Recognition*, 27(3):421–428, 1994.
- [26] K. Jajuga. L_1 -norm based fuzzy clustering. *Fuzzy Sets and Systems*, 39:43–50, 1991.
- [27] K. S. S. Kumar. *Genetically Optimized Improved Fuzzy Clustering for Color Image Segmentation*. PhD thesis, University of Bremen, Germany, 1996.
- [28] L. H. Chen, C. Kao, S. Kuo, T. Y. Wang, and C. Y. Jang. Productivity diagnosis via fuzzy clustering and classification: an application to machinery industry. *Omega*, 24(3):309–319, June 1996.
- [29] M. M. Trivedi and J. C. Bezdek. Low-level segmentation of aerial images with fuzzy clustering. *IEEE Transactions on Systems, Man and Cybernetics*, 16(4):589–598, July 1986.
- [30] M. C. Clark, L. O. Hall, D. B. Goldgof, L. P. Clarke, R. P. Velthuisen, and M. S. Silbiger. MRI segmentation using fuzzy clustering techniques. *IEEE Engineering in Medicine and Biology Magazine*, 13(5):730–742, November 1994.

- [31] I. Gath and D. Hoory. Fuzzy clustering of elliptic ring-shaped clusters. *Pattern Recognition Letters*, 16(7):727–741, July 1995.
- [32] H. Frigui and R. Krishnapuram. A comparison of fuzzy shell-clustering methods for the detection of ellipses. *IEEE Transactions on Fuzzy Systems*, 4(2):193–199, May 1996.
- [33] C. V. Jawahar and P. K. Biswas A. K. Ray. Detection of clusters of distinct geometry: a step towards generalised fuzzy clustering. *Pattern Recognition Letters*, 16(11):1119–1123, November 1995.
- [34] K. Hirota and W. Pedrycz. Directional fuzzy clustering and its applications to fuzzy modelling. *Fuzzy Sets and Systems*, 80(3):315–326, June 1996.
- [35] R. N. Dave. Robust fuzzy clustering algorithms. In *Second IEEE International Conference on Fuzzy Systems*, pages 1281–1286, San Francisco, March–April 1993.
- [36] S. J. Kia and G. G. Coghill. Unsupervised clustering and centroid estimation using dynamic competitive learning. *Biological Cybernetics*, 67(1):433–443, 1992.
- [37] S. J. Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2):261–272, February 1997.

Chapter 5

Topology-unconstrained contour detection

5.1 Introduction

The main difference between segmentation and edge detection is that the former concerns the detection of regions, according to a given criterion, while the latter concerns the detection of region *boundaries*. These two operations are complementary but the transition from one to another is not always straightforward because the information level is different and the application domains are usually different. Edge detection may be used for example in image enhancement techniques. Adding its weighted second derivative to an image would restore smooth transitions at region boundaries (see shock filters, Section 2.8) or simply enhance textural structures. In addition to its versatility, edge detection processes are usually much faster than image segmentation ones.

This chapter is dedicated to the detection of *contours*, which is intended to separate the different objects of interest from each other and from the background. Edge detection alone is usually not sufficient to efficiently construct these boundaries and is part of a more complex detection scheme. Edges are spatial transitions between different regions, in gray-scale, color, and vector-valued images (texture descriptors for example), but a single region rarely shows a uniform intensity transition along its boundary, which renders the detection difficult. In this chapter a particular class of contour detectors will be investigated, which have no constraint on contour topology, i.e. without constraint on the contour *anatomy*.

A relatively recent approach to contour detection, called *snakes*, is of prime interest in this domain. The basic idea, initially formulated by Kass et al. [1], uses internal and external constraints to make an active contour converge towards a local energy minimum. This energy is the combination of an internal snake energy due to deformation and internal resistance, and an external energy due to the image structure (usually described by its underlying gradient). The sum of both energies must be minimized in order to make the snake converge. This method is therefore an energy minimization based active contour. Recently this technique has been exploited in a large number of applications and improved for better convergence. Geodesic active contours are one successful example and will be presented in § 5.5.3.

This chapter proposes an analysis of different techniques, from low-level gradient-based edge detectors in Sections 5.2 to more sophisticated techniques like active contours in Section 5.4. Edge

detection through curve evolution is presented in Section 5.5 and finally a contour detection technique that combines nonlinear isotropic diffusion and the extraction of iso-level curves considered as contour candidates using morphological flooding is introduced in Section 5.6. The contour selection is done by minimization of a gradient-based energy functional. This technique is best adapted to the detection of single or multiple closed contours, quickly and unsupervised, and without any restriction on the contour topology, i.e. there is no restriction concerning splitting, merging, development of corners, etc.

5.2 Classical edge detection

Edge detection is the detection of *abrupt transitions* or *discontinuities* in the gray-level or vector-valued representation of images. The fundamental question is: how abrupt must be the transition to be considered as an edge? Let us consider the simple 1-D edge model shown in Fig. 5.1, obtained using the sigmoid function given by Eq. 2.21. The first and second derivatives are shown as well: if we suppose that the edge is neither located where the edge “starts” nor where the edge “ends”, but somewhere in between, an immediate choice is to consider the zero crossings of the second-order derivative [2], assuming that the scale is uniform, i.e. uniform gray-level differences (in the case of images) correspond to uniform changes in the visual perception. In 2-D the commonly used approach to detect edge points is to consider the zero-crossings of the image Laplacian:

$$l(\mathbf{x}) = \Delta v = \nabla \cdot \nabla v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} ,$$

where ∇ is the gradient operator. Zero-crossing always form closed contours, which obviously do not always correspond to contours in the image. The Laplacian is seldom used in practice because of its unacceptable sensitivity to noise. Therefore it is combined with a smoothing operator which defines how “strong” edges must be in order to be detected. A typical low-pass filter used for this application has a Gaussian kernel, which does not introduce new edge points for increasing values of σ and which has the minimal product of position and frequency bandwidths. The edge detection operator is then given by:

$$h(\mathbf{x}) = \Delta(G_\sigma * v) ,$$

where G_σ is a Gaussian filter with variance σ and $**$ is the two-dimensional convolution. Instead of convolving the input image and then computing the Laplacian it is preferable to derive the Laplacian of the Gaussian and to apply the resulting filter to the image. This formulation was introduced first by Marr and Hildreth [3]. Since the 2-D Gaussian filter has a separable impulse response¹, i.e. $G_\sigma(x, y) = g_\sigma(x)g_\sigma(y)$, the derivatives may be computed for a single variable only:

$$\begin{aligned} g_\sigma(x) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} , \\ g'_\sigma(x) &= -\frac{1}{\sigma^2} x g_\sigma(x) , \\ g''_\sigma(x) &= \left(\frac{1}{\sigma^4} x^2 - \frac{1}{\sigma^2}\right) g_\sigma(x) , \end{aligned}$$

and the Laplacian in 2-D is given by $g''_\sigma(x)g_\sigma(y) + g''_\sigma(y)g_\sigma(x)$.

¹We make the assumption that the variance σ is identical in both x and y directions.

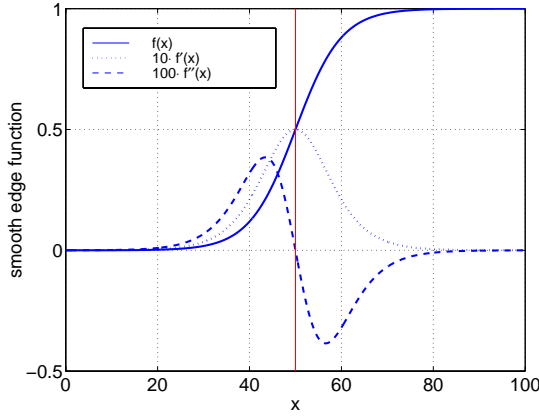


Figure 5.1: Smooth step-edge model with first and second derivatives.

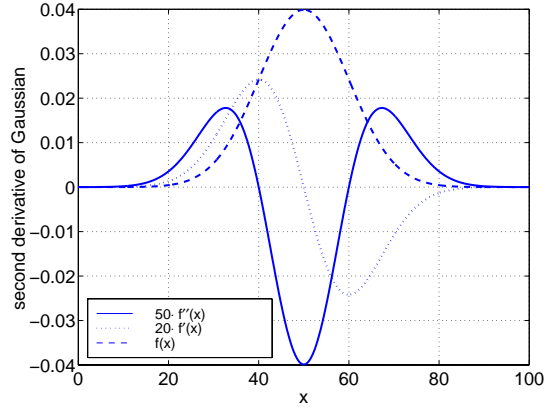


Figure 5.2: Laplacian of Gaussian used for edge detection.

Figure 5.2 shows the cross section of the Laplacian of a Gaussian function. The value of σ will decide how abrupt transitions must be to be considered as edges. The main deficiency of the Laplacian is that it does not contain any curvature-dependent term and is therefore curvature insensitive. It will entirely miss any geometrically nontrivial singular boundaries. A more natural choice then would be a second order directional derivative. Under the assumption that an edge is located in a cross-section in direction of the local gradient, the following equation can be used in place of the Laplacian:

$$\begin{aligned}
 \frac{\partial^2 v}{\partial n^2} &= \nabla \left(\nabla v \cdot \frac{\nabla v}{|\nabla v|} \right) \cdot \frac{\nabla v}{|\nabla v|} \\
 &= \nabla (|\nabla v|) \cdot \frac{\nabla v}{|\nabla v|} \\
 &= \left(\mathcal{H}(v) \frac{\nabla v}{|\nabla v|} \right) \cdot \frac{\nabla v}{|\nabla v|} \\
 &= \frac{1}{|\nabla v|^2} \left\{ \left(\frac{\partial v}{\partial x} \right)^2 \frac{\partial^2 v}{\partial x^2} + 2 \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} \frac{\partial^2 v}{\partial x \partial y} + \left(\frac{\partial v}{\partial y} \right)^2 \frac{\partial^2 v}{\partial y^2} \right\},
 \end{aligned} \tag{5.1}$$

where $\mathbf{n} = \nabla v / |\nabla v|$ and $\mathcal{H}(v)$ is the *Hessian* matrix of v :

$$\mathcal{H}(v) = \begin{pmatrix} \partial^2 v / \partial x^2 & \partial^2 v / \partial x \partial y \\ \partial^2 v / \partial y \partial x & \partial^2 v / \partial y^2 \end{pmatrix}.$$

The above result has been previously used for edge detection in [4] and can be compared to the directional derivatives obtained in Chapter 2 for anisotropic diffusion, where the dependence on local curvature has been shown. However, the complexity of Eq. 5.1 renders this approach very sensitive to numerical approximation. Especially the squared gradient module at the denominator increases the floating point errors during the computation. Therefore the simpler Laplacian formulation is mostly used for edge detection applications. High order numerical approximations to the different derivatives are used in [4] to improve the precision.

Figure 5.3 shows three examples of edge detection operators. In all cases the zero-crossings of the Laplacian have been detected but different approximations to the partial derivatives have

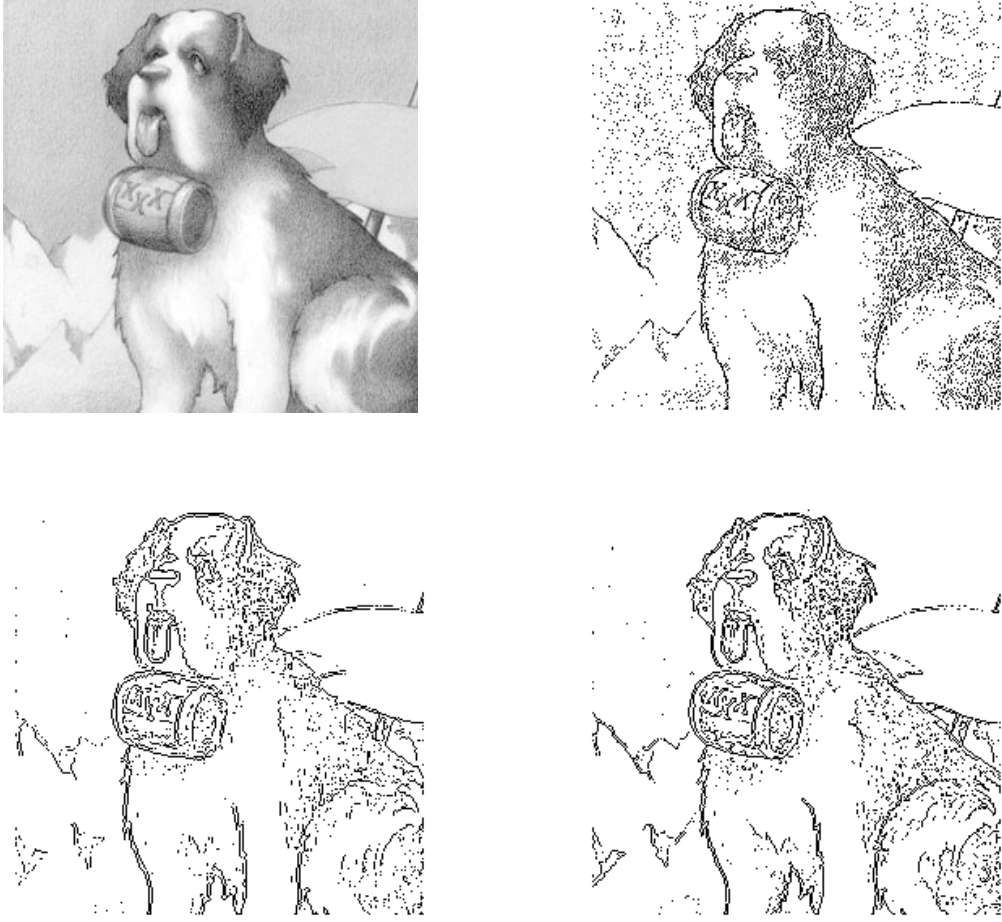


Figure 5.3: The top-left image has been used to illustrate the effect of different numerical approximations to first and second order derivatives for edge detection: finite difference (top-right), cubic interpolation (bottom-left), and the Sobel operator (bottom-right). The zero-crossings are shown in black.

been used. In addition to the Laplacian, gradient thresholding has been introduced in the edge detection scheme in order to keep only zero-crossings that correspond to high gradient values. In the top-right image a finite difference approximation has been used (see § 2.6.4), showing a high sensitivity to spurious details and noise. The bottom-left result has been obtained by locally approximating the image with third order polynomials, involving the convolution with six 5×5 filters [5]. A close result, displayed bottom-right, can be obtained (for high signal-to-noise ratios) using a Sobel operator [2]:

$$S_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{pmatrix} ** \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix},$$

with $S_y = S_x^T$ in matrix notation and the Laplacian is then given by:

$$S_l = \begin{pmatrix} 2 & 4 & 4 & 4 & 2 \\ 4 & 0 & -8 & 0 & 4 \\ 4 & -8 & -24 & -8 & 4 \\ 4 & 0 & -8 & 0 & 4 \\ 2 & 4 & 4 & 4 & 2 \end{pmatrix}.$$

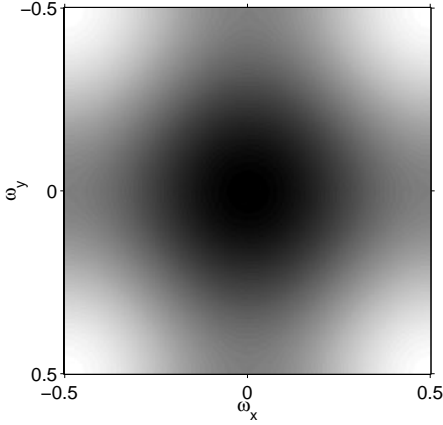


Figure 5.4: Transfer function of the central difference approximation to the Laplacian.

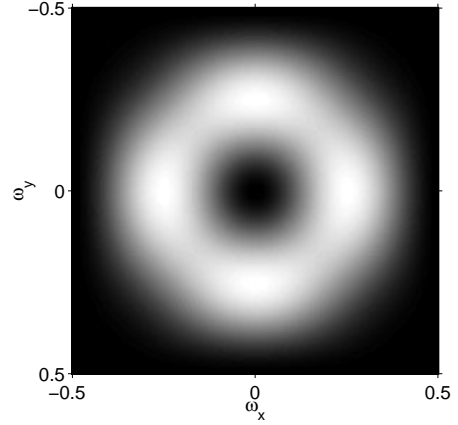


Figure 5.5: Transfer function of the Sobel approximation to the Laplacian.

The Sobel operator combines a central difference filter with a smoothing filter which reduces the influence of noise. Figure 5.4 shows the transfer function (Fourier transform of impulse response) of the finite difference approximation to the Laplacian, involving a symmetrical 3×3 filter. The transfer function of the Laplacian approximated with the Sobel operator is shown in Fig. 5.5. These two results can be compared to the ideal Laplacian, whose transfer function is given by $L(\omega_x) = (j\omega_x)^2 + (j\omega_y)^2 = -\omega_r^2$. The finite difference approximation is closer to this theoretical result, but the low-pass component of the Sobel operator renders the approximation less sensitive to high-frequency noise. Other filters have been proposed in order to approximate spatial derivatives, like the *Roberts*, *Prewitt*, *Kirsch*, or *Frei* operators [5].

Figure 5.6 gives examples of edge detection in presence of noise. The Sobel operator, despite its smoothing term, detects most of the spurious edges introduced by the noise. The bottom-left image shows the result obtained with a Laplacian of Gaussian operator. There are still several spurious edges but the advantage here is that the smoothing term σ can be adapted to the “amount” of noise.

The main deficiencies when combining an edge detector with a Gaussian filter (or any other low-pass filter) can be summarized as follows:

- Second derivatives with zero value rarely exist in digital images. The value is either positive or negative and one has to define a decision criterion to select the edge pixels.
- This approach detects single edge pixels. The construction of continuous edges needs additional processing.

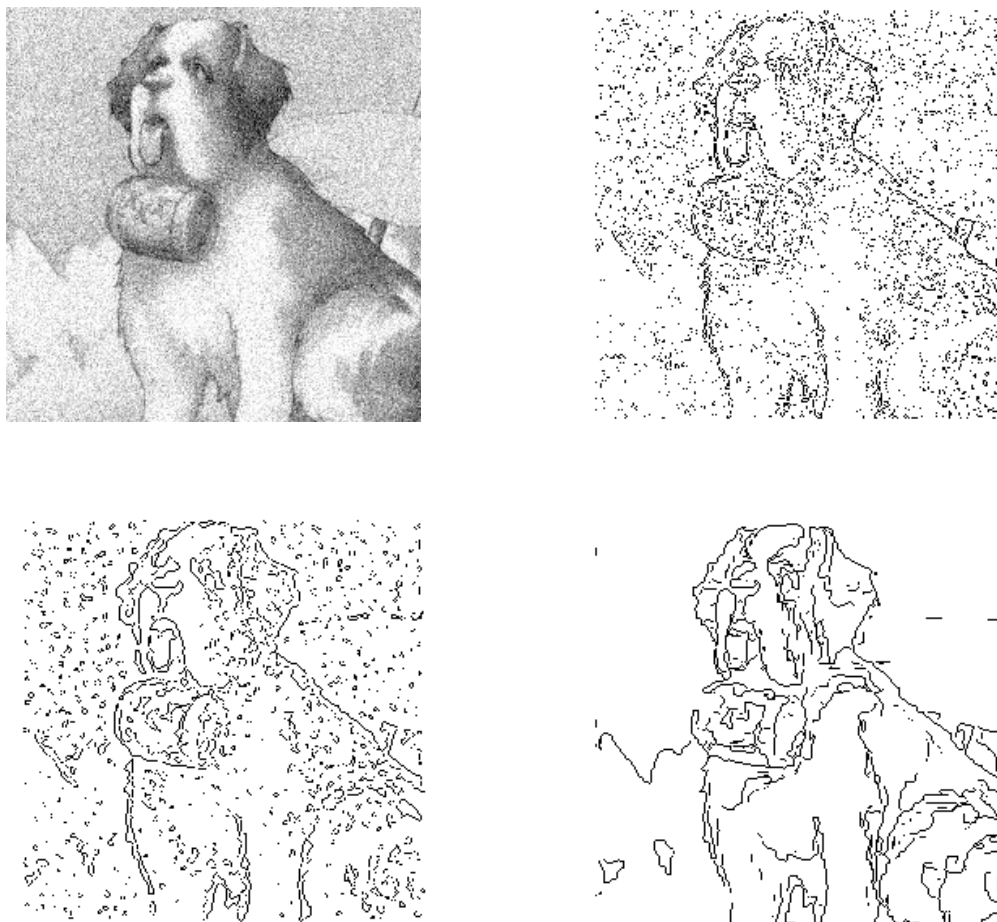


Figure 5.6: The top-left image is corrupted with noise and different approaches are tested to detect edges: Sobel operator (top-right), Laplacian of Gaussian (bottom-left), and Sobel operator after nonlinear isotropic diffusion (bottom-right).

- The use of a Gaussian filter does not preserve the geometry of structures. The appearance of rounded corners is the main consequence, as explained in Chapter 2. This problem can be solved by using nonlinear isotropic diffusion instead of a Gaussian filter, but the final result (using the above edge detection strategy) does usually not justify the increased computational complexity.
- The Gaussian filter has an infinite impulse response that must be limited.
- The choice of the right gradient threshold, if it exists, is not straightforward.

In Fig. 5.6 the bottom-right image has been obtained by combining nonlinear isotropic diffusion and the detection of Laplacian zero-crossings. Even if the result is not perfect, it outperforms clearly the other approaches illustrated in the same figure. When applied to the image without additional noise, as shown in Fig. 5.8, the result is even more impressive. The diffusion result



Figure 5.7: Test image after nonlinear isotropic diffusion.



Figure 5.8: Zero-crossings of the Laplacian using the Sobel operator computed for the image of Fig. 5.7.

is shown in Fig. 5.7, where spurious edges have been suppressed while geometric properties are preserved.

From the above list, two main deficiencies emerge: the geometrical distortion and the fact that not edges but edge pixels are detected. Typically edge detection algorithms are followed by edge linking and other boundary detection procedures designed to assemble pixels into meaningful boundaries. Simple procedures analyze a given neighborhood of each edge pixel and assemble the pixels which have similar value. Another approach is to link points that lie on a curve of specified shape. This result can be obtained by using the *Hough transform* [6]. When considering straight lines given by the general equation $y = ax + b$, the idea is to consider the parameter space ab and to count for every (a, b) coordinate, called *accumulator cell*, the number of edge pixels falling onto the corresponding line. This implies that the parameter space has been previously sampled and quantized. The construction of parameter space ab is called the Hough transform. The immediate limitation of this approach when applied to straight lines is that collinear edges will be linked together. A solution to this problem would be to add two other parameters defining the starting end ending points of every line, but the increase in complexity is non-negligible. The Hough transform can be applied to many other functions [7] with the only limitation that the number of parameters will define the complexity of the search process and finally the resolution and the precision of the solution. The Hough transform is one example of edge linking method and many other approaches have been proposed for the same application, as *graph-theoretic techniques* for example, where edge segments are represented in the form of a graph and boundary detection is performed by searching for low-cost paths [8].

The goal in this section was not to go through all the edge detection literature but to define some key concepts which are used almost in every edge detection formulation. There are many other techniques and the reader can find some references in [9, 10, 11, 12].

The examples given in this section lead to the following general conclusions:

- The use of filters for edge detection requires edge linking and boundary detection phases to

complete the process.

- Finite length filters usually allow for a quick processing and can be easily implemented in hardware. However, their poor accuracy limits the application domains.
- The edge detection is insensitive to local curvature and corners will be smoothed out.

In the following section a computational approach to edge detection that received a great attention in the image processing community and known as Canny's edge detector [13] is presented.

5.3 Canny's edge detector

5.3.1 Edge detection and localization criteria

The analysis presented by Canny in [13] first considers one-dimensional edge profiles, assuming that two-dimensional edges locally have a constant cross-section in some direction. This assumption, however, is not true for corners. The detection problem assumes that an edge is bathed in Gaussian noise and that the signal must be convolved with the impulse response of a given filter. The center of an edge is then marked where the output of the convolution shows a local maximum. The filter design consist of finding the filter which gives the best performance with respect to a number of criteria. In his analysis Canny proposes the following performance criteria:

- *“Good” detection.* There should be a low probability of missing real edges or of falsely detecting edges. These probabilities are monotonically decreasing functions of the output *signal-to-noise ratio* (SNR).
- *“Good” localization.* The located edges should be as close as possible to the center of the true edges.
- *Single response.* Only one response to a single edge should be detected, which is implicitly captured in the first criterion. But the mathematical form of the first criterion does not capture the multiple response requirement.

These intuitive criteria must be captured in a mathematical form that is readily solvable. Let the impulse response of the filter be denoted by $f(x)$ and the edge function itself by $H(x)$. The filter is supposed to have a finite impulse response bounded by $[-W, W]$. The two first criteria are mathematically expressed through the following equations:

$$SNR = \frac{\left| \int_{-W}^W H(-x)f(x)dx \right|}{n_0 \sqrt{\int_{-W}^W f^2(x)dx}},$$

$$Localization = \frac{\left| \int_{-W}^W H'(-x)f'(x)dx \right|}{n_0 \sqrt{\int_{-W}^W (f'(x))^2 dx}},$$

where n_0^2 is the mean-squared noise amplitude per unit length and $f'(x)$ is the derivative of $f(x)$. The detection criterion is expressed through the SNR and the localization through the reciprocal of the standard deviation of the edge location. The localization formula given above is obtained through an approximation to the variance of edge locations in presence of Gaussian noise.

The filter design reduces to the maximization of both quantities or, to simplify the analysis of step edges, the maximization of their product. The elimination of multiple responses is obtained by considering the average distance between zero-crossings of the response of a function to Gaussian noise:

$$d_{zc}(f) = \pi \left(\frac{\int_{-\infty}^{\infty} (f'(x))^2 dx}{\int_{-\infty}^{\infty} (f''(x))^2 dx} \right) .$$

A natural choice is to constraint this distance to be some fraction k of W . The expected number of noise maxima in a region of width $2W$, the filter length, is then given by $N_{\text{noise}} = 2/k$. Fixing k fixes the number of noise maxima that could lead to a false response.

5.3.2 Optimal edge detector

The analytical solution to the criteria derived so far would be difficult and even impossible to establish. However, $f(x)$ being a discrete signal, the evaluation of the different criteria can be done numerically and the optimization can be done directly on the sampled impulse response. This optimization is done by using a *penalty method*, where the constrained optimization is reduced to one or several unconstrained optimizations.

We will now focus on the case where $H(x)$ is the step edge, defined as:

$$H_{\text{step}}(x) = A \int_{-\infty}^x \delta(u) du ,$$

where $\delta(x)$ is the Dirac operator. It is then possible to define two performance measures $\Sigma(f)$ and $\Lambda(f')$ and the optimization criterion reduces to:

$$\Sigma(f)\Lambda(f') = \frac{A^2}{n_0^2} \frac{\left| \int_{-W}^0 f(x) dx \right|}{\sqrt{\int_{-W}^W f^2(x) dx}} \frac{|f'(0)|}{\sqrt{\int_{-W}^W (f'(x))^2 dx}} ,$$

where the A^2/n_0^2 term can be skipped in the maximization procedure. The interesting feature of this criterion is that there is an uncertainty principle relating the detection and localization performance of the step edge detector: it is not possible to improve both simultaneously [13].

The general solution in the range $[-W, 0]$ is given by:

$$f(x) = \{a_1 \sin(\omega x) + a_2 \cos(\omega x)\} e^{\alpha x} + \{a_3 \sin(\omega x) + a_4 \cos(\omega x)\} e^{-\alpha x} + c ,$$

and the solution in the range $[-W, W]$ is immediate, knowing that $f(x)$ is an odd function. The boundary conditions are given by $f(x) = 0$, $\forall x \in \{-W, 0, W\}$, and $f'(x) = 0$, $\forall x \in \{-W, W\}$. These boundary conditions allow for solving the quantities a_1 through a_4 in terms of the unknown constants α , ω , c , and $s = f'(0)$. Numerical methods are used also in this case to solve the optimization problem in three variables α , ω , and $\beta = s/c$.

A rough estimate for the probability of a spurious maximum in the neighborhood of the true maximum can be derived. The following relationship is finally obtained:

$$\frac{|f'(0)|}{\sigma_s} = r \Sigma(f) ,$$

where σ_s is the variance of the second derivative of the response of $f(x)$ to the noise component. Ideally one should have $r = 1$, but since this condition is impossible to fulfill in practice, a filter with r as close as possible to 1 must be found.

The very interesting consequence of this optimization problem is that the edge filter can be approximated by the first derivative of a Gaussian and then:

$$f(x) = -\frac{x}{\sqrt{2\pi}\sigma^3} \exp\left(-\frac{x^2}{2\sigma^2}\right) .$$

The two performance measures can be calculated leading to the following performance index:

$$\Sigma\Lambda = \sqrt{\frac{8}{3\pi}} \approx 0.92 ,$$

and the value of r is:

$$r = \sqrt{\frac{4}{15}} \approx 0.51 .$$

The performance of the first derivative of Gaussian is worse than the optimal operator by about 20 % ($\Sigma\Lambda = 1.12$) and its multiple response measure is worse by about 10 % ($r = 0.576$). A one-dimensional Marr-Hildreth edge detector [3] is almost identical with the operator derived above, except that in this case the edges are maxima in the filter response and not zero-crossings. In the two-dimensional case however, the directional properties of the detector enhance its detection and localization performances compared to the Laplacian.

To estimate the noise component the global histogram combined with two thresholds is used. The thresholding is then done with hysteresis, which means that if part of a contour is above a *high* threshold those points are immediately output, as the entire connected segment of contour which contains the points and which lies above a *low* threshold. This approach reduces the *streaking*, which is the breaking up of a contour caused by the operator output fluctuations above and below a single threshold.

The two-dimensional implementation is discussed next.

5.3.3 Two-dimensional implementation

The extension to two dimensions can be done in a similar way to what is described in Section 5.2. We use directional derivatives to construct the edge detector. The direction is estimated from the normalized smoothed gradient vector. Finally it turns out that at a local maximum of the edge detector we have:

$$\frac{\partial^2 G_\sigma}{\partial n^2} * * v = \frac{\partial^2}{\partial n^2} (G_\sigma * * v) = 0 .$$

This formulation has been previously used in other contour detection schemes, as in [4]. The interesting part in Canny's edge detector is the use of multiple directional filters. This choice is motivated by the fact that, in the case of directional contours, *both* the detection and the localization can be improved by scaling the filter by a factor $s > 1$ in the direction normal to the edge direction. Therefore a "large" filter should be used in a direction normal to the contour direction. The two-dimensional filters are then constructed so that in the edge direction the signal has a derivative of Gaussian shape and in the normal direction a step shape. Several directions are used to compute the different filters (for example six, with an angle of $\pi/6$ between the different filters).

Figure 5.9 shows the comparison between the results obtained with Canny's edge detector ($\sigma = 1$), before and after nonlinear isotropic diffusion. In the case where no noise has been added to the image (top row), the result obtained without diffusion shows more details in the background (mountains), while the dog is covered with many spurious edges. The same experiment has been

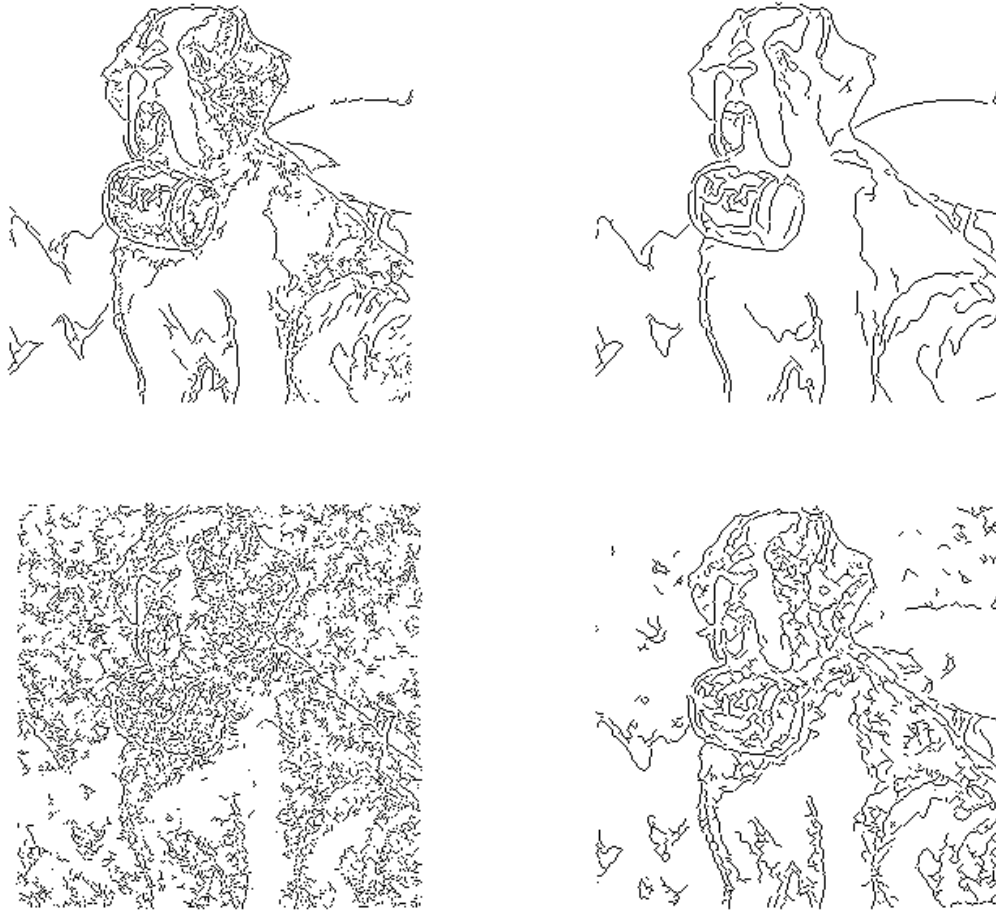


Figure 5.9: The top-left image shows the output of Canny's edge detector for the image shown in Fig. 5.3. The result obtained after a nonlinear isotropic diffusion is shown top-right. The same scheme has been applied in presence of noise and the resulting image is shown bottom-left without pre-filtering and bottom-right with pre-filtering (nonlinear isotropic diffusion).

repeated with additional noise, as shown in the bottom row of Fig. 5.9. Keeping a narrow filter is not adapted and larger filters have been used to obtain the results shown in Figs. 5.10 and 5.11. It is however not always straightforward to choose the appropriate filter size.

Edge detectors are clearly not adapted when objects must be detected. In order to obtain closed contours, except when only part of the object is visible, the best option is to fit a closed contour to the object boundaries. This approach is investigated in the following sections.



Figure 5.10: Result of Canny's edge detector applied to a noisy image with $\sigma = 3$.

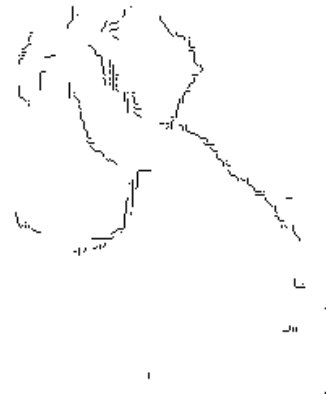


Figure 5.11: Result of Canny's edge detector applied to a noisy image with $\sigma = 6$.

5.4 Active contour models

5.4.1 Introduction

In this section an important boundary detection technique called *snakes*² and which belongs to the particular class of *active contours* is presented. When single or multiple objects must be detected in an image these approaches constitute a powerful alternative to classical edge detection.

Snakes have been initially formulated by Kass et al. [1]. The snake is defined as *an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it towards features such as lines and edges*. Therefore, snakes are active contour models that always minimize their energy functional and therefore exhibit a dynamic behavior. From a more general point of view, snakes are an example of techniques used to match a deformable model to an image by means of energy minimization.

In the following paragraph snake model and its mathematical formulation are presented. Advantages and disadvantages are discussed and a recently proposed generation of active contours is presented in Section 5.5.

5.4.2 Snakes

The basic snake model is a *controlled continuity spline* under the influence of image forces and external constraint forces. The snake is a parameterized planar curve $\mathcal{C}(q) : [0, 1] \rightarrow \mathbb{R}^2$ and the energy functional we will use can be written as:

$$E_{snake} = \int_0^1 \{E_{int}(\mathcal{C}(q)) + E_{ext}(\mathcal{C}(q)) + E_{cons}(\mathcal{C}(q))\} dq, \quad (5.2)$$

where $E_{int}(\mathcal{C}(q))$ is the internal energy due to binding, $E_{ext}(\mathcal{C}(q))$ gives rise to the image force, and $E_{cons}(\mathcal{C}(q))$ gives rise to the external constraint forces. This last term accounts mainly for

²Note that snakes can be used to detect other open curves as well.

user interaction in supervised contour detection systems and will be skipped here. The internal spline energy can be written as:

$$E_{\text{int}}(\mathcal{C}(q)) = \frac{1}{2} \left(\alpha(\mathcal{C}(q)) \left| \frac{d\mathcal{C}(q)}{dq} \right|^2 + \beta(\mathcal{C}(q)) \left| \frac{d^2\mathcal{C}(q)}{dq^2} \right|^2 \right) .$$

The first-order term makes the snake act like a membrane (elasticity) and the second order term makes it act like a thin plate (rigidity). Adjusting $\alpha(\mathcal{C}(q))$ and $\beta(\mathcal{C}(q))$ controls the relative importance of both terms. If $\beta(\mathcal{C}(q)) = 0$ at a point then the snake can become second-order discontinuous and develop a corner.

The energy functional $E_{\text{ext}}(\mathcal{C}(q))$ is needed to attract the snake to salient image features. When these features are a combination of lines, edges, and terminations, the following energy functional can be used:

$$E_{\text{ext}}(\mathcal{C}(q)) = \omega_{\text{line}}(\mathcal{C}(q))E_{\text{line}}(\mathcal{C}(q)) + \omega_{\text{edge}}(\mathcal{C}(q))E_{\text{edge}}(\mathcal{C}(q)) + \omega_{\text{term}}(\mathcal{C}(q))E_{\text{term}}(\mathcal{C}(q)) .$$

The line energy functional can be simply set to $E_{\text{line}}(\mathcal{C}(q)) = v(\mathcal{C}(q))$ and depending on the sign of ω_{line} the snake will be attracted either to light lines (-) or dark lines (+).

The edge energy functional can be set to $E_{\text{edge}}(\mathcal{C}(q)) = -|\nabla v(\mathcal{C}(q))|^2$ and the snake is attracted to contours with large gradients. In order to allow the snake to come to equilibrium on a noisy image, a blurred version of the image is used. The blurring is then progressively reduced. The following energy functional is then used:

$$E_{\text{edge}}(\mathcal{C}(q)) = |G_{\sigma} * \Delta v(\mathcal{C}(q))|^2 ,$$

which is minimum for edge pixels.

The snake is attracted towards the zero-crossings of the Laplacian of the image, as defined in Section 5.2. A large value of σ can be used to approach the energy minimization problem, which should make the snake converge towards object boundaries, and then progressively reduced to reach the local minimum.

The termination functional is obtained through the local curvature, as defined in § 2.6.1:

$$E_{\text{term}}(\mathcal{C}(q)) = \text{curv}(\mathcal{C}(q)) ,$$

where $\text{curv}(\mathcal{C}(q))$ is given by Eq. 2.8. A more appropriate curvature dependent energy functional would be $E_{\text{term}}(\mathcal{C}(q)) = -|\text{curv}(\mathcal{C}(q))|^2$, which is maximum for small curvatures and therefore attracts the snake towards corners. By combining $E_{\text{edge}}(\mathcal{C}(q))$ and $E_{\text{term}}(\mathcal{C}(q))$ one can create a snake that is attracted to edges and terminations.

To derive the evolution equation of snakes only the internal energy E_{int} and the edge energy E_{edge} with $w_{\text{edge}} = 1$ are kept. The two parameters α and β are supposed to be constant. From the calculus of variations, the necessary and sufficient condition for the integral given by Eq. 5.2 to be stationary is that $\partial\mathcal{C}(q, t)/\partial q$ is an admissible function satisfying Euler's differential equation [14]. In the present case this equation is given by:

$$\alpha\mathcal{C}''(q) - \beta\mathcal{C}''''(q) - \nabla E_{\text{ext}} = 0 ,$$

and the solution can be found iteratively by making the snake dynamic, i.e. $\mathcal{C}'''(q) = \mathcal{C}''(q, t)$:

$$\frac{\partial\mathcal{C}(q, t)}{\partial t} = \alpha\mathcal{C}''(q, t) - \beta\mathcal{C}''''(q, t) - \nabla E_{\text{ext}} .$$

In the discrete case, when the snake is built of a finite number N of points (called *snaxels*), the iterative energy minimization process can be written in matrix form as:

$$\begin{aligned} \mathbf{x}_{t+1} &= (\mathbf{P} + \gamma \mathbf{I})^{-1} \left(\mathbf{x}_t - \frac{\partial E_{\text{ext}}}{\partial x}(\mathbf{x}, \mathbf{y}) \right), \\ \mathbf{y}_{t+1} &= (\mathbf{P} + \gamma \mathbf{I})^{-1} \left(\mathbf{y}_t - \frac{\partial E_{\text{ext}}}{\partial y}(\mathbf{x}, \mathbf{y}) \right), \end{aligned}$$

where t is the time index, \mathbf{x} and \mathbf{y} are vectors containing respectively the x and y coordinates of all snake points, \mathbf{I} is the $N \times N$ identity matrix, and \mathbf{P} is an $N \times N$ pentadiagonal matrix given by:

$$\mathbf{P} = \begin{pmatrix} 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \cdots & 0 & \beta & -\alpha - 4\beta \\ -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \cdots & 0 & \beta \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \beta & 0 & \cdots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta \\ -\alpha - 4\beta & \beta & 0 & \cdots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta \end{pmatrix}.$$

The above matrix is computed once, as well as the spatial derivatives of E_{ext} . The matrix size depends on the number of snake points and the number of \mathbf{P} matrices is 1 if the parameters α and β are constant, N if they are not.

Figure 5.12 shows an example of shape detection using the snake formulation. Given the star-shaped object displayed top-left and two different initialization to the snake (a convex³ curve top-right and a non-convex one bottom-left) the same result is obtained and displayed bottom-right. The snake formulation is certainly very efficient to detect the *convex hull* [2] of objects but it is unable to fit deep cavities, even if tuning parameter β may improve slightly the result shown in Fig. 5.12. In general, the main limitations of the basic snake model can be summarized as follows:

- The snake converges to a local energy minimum (ill-posed problem).
- High order derivatives along the curve must be estimated, which degrades the robustness to noise.
- Parameters α and β must be tuned.
- The convergence speed depends on the matrix size, which is equal to the number of snake points used to represent the snake, which in turn sets the resolution.
- There is no solution to changes in topology (splitting, merging, corners, etc.).

A significant number of modifications and improvements were proposed in order to solve the abovementioned limitations. The most interesting propositions will be briefly described and finally we will focus on a more geometrical formulation of active contours presented in the following section.

³A function $f(x)$ with continuous second derivative is called *convex* if $f'' \geq 0$ [15].

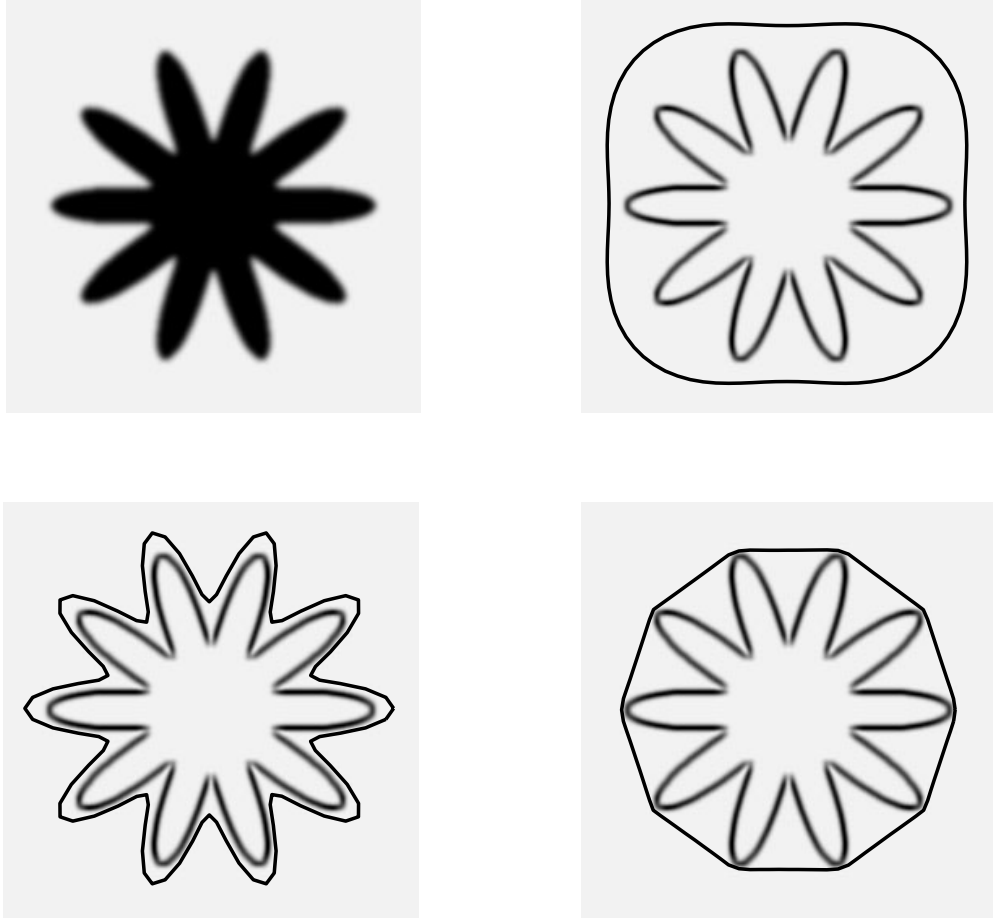


Figure 5.12: The top-left image shows a non-convex shape. Two different initializations to the snake algorithm are proposed top-right and bottom-left, superimposed on the external energy functional. The same result is obtained and displayed bottom right.

In [16] the *balloon model* is introduced. External energy is seen as a potential associated with the external forces, meaning that $F(\mathcal{C}(q)) = -\nabla E_{\text{ext}}(\mathcal{C}(q))$. In order to avoid large jumps across possible minima the external force $F(\mathcal{C}(q))$ is normalized, which simulates a local time step that makes the curve evolve at the same speed everywhere. Then an *internal pressure* is introduced to push the snake outwards so that small edges are passed over. The curve can be therefore considered as a balloon. An additional force $F_{\text{balloon}} = k\mathbf{n}$ is added to the normalized force $F(\mathcal{C}(q))$. The behavior of the introduced model outperforms clearly the original snake formulation, even in presence of sharp corners and non-convex shapes. The extension of the method to 3-D images is introduced in [17]. The detection of changes in topology, however, has not been solved yet.

Modeling and extraction of active contours has been investigated in [18, 19]. The author proposes a complete study of active contour models and formulates the *generalized active contours*. It uses a *regenerative* shape matrix, i.e. the complete chain of contour points can be generated

from an appropriate inversion if any two points are known, which is invariant and unique under rigid motions. The model is combined with *Markov random fields* to model local deformation. The proposed approach performs well for the extraction of contours from noisy images, but it still very sensitive to the initial contour location.

The main limitation in the approaches listed above is that the original snake model is not capable of handling changes in the topology of the evolving contour when direct implementations are performed. The final topology will therefore be the same as that of the initial curve \mathcal{C}_0 . The detection of possible splitting or merging is addressed in [20]. The authors propose to decompose the image domain into a grid of discrete cells and to re-parameterize the snake with a new set of nodes and springs by computing the intersection points with the superposed grid. A simplicial decomposition, also called triangulation, is used. The simplest formulation partitions the space using a uniform cubic grid and the triangulation is obtained by subdividing each cube into $n!$ simplices (reducing to triangles in 2-D), n being the space dimension. The curve is then approximated by all line segments that interconnect the intersection points of the snake with the grid. By assigning to every vertex a sign (positive if inside the snake, negative otherwise), intersecting cells (grid “lines”) can be tracked easily. Since the curve can only cross a cell on two points, changes in topology can be detected easily.

Since their formulation, snakes or active contours have been exhaustively used in contour detection applications, most of the time with an improved formulation. One of the preferred application domain is medical image processing. In [21] an active contour model for biomedical images is presented, with application to magnetic resonance (MR) images. Model-based segmentation of biomedical images for computerized tomography (CT) slides is presented in [22], and a geometric snake model (see Section 5.5) applied to MR images is presented in [23].

Instead of trying to improve the existing model by adding and suppressing terms or features, one should consider the problem with detachment and think about a formulation that fits the intrinsic nature of active contour. This approach and different issues are presented in the following section.

5.5 A geometrical approach to active contours

5.5.1 Contour detection with front propagation

The initial snake formulation presented in the previous section illustrates the powerful concept of active contours. The main drawbacks of this formulation gave rise to a number of new formulations intended to improve the “quality” and precision of convergence. We propose here to reconsider the initial problem and the nature of contours, by introducing a geometrical approach to active contours. As defined in the previous section, snakes are energy minimizing splines and hence underly geometrical problems. The idea then is to consider the active contour as a *curve* that adapts its shape and topology to internal and external constraints. The gradient vector field is a typical constraint used when the curve should fit to salient edges that in turn define the boundaries between objects.

The general equation for curve evolution we will consider is the following:

$$\frac{\partial \mathcal{C}}{\partial t} = F \mathbf{n} ,$$

where F is a speed function and \mathbf{n} is the outward (inward) normal for expanding (collapsing) curves, as illustrated in Fig. 5.13. In order to keep the evolution stable, there is a number of

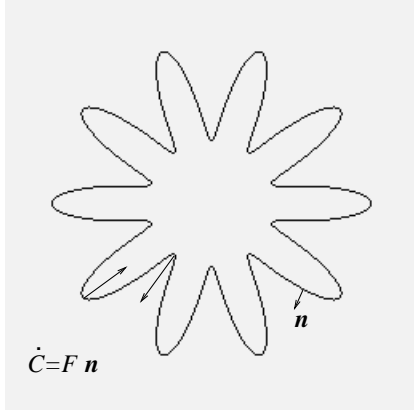


Figure 5.13: Propagating curve under curvature dependent speed function.

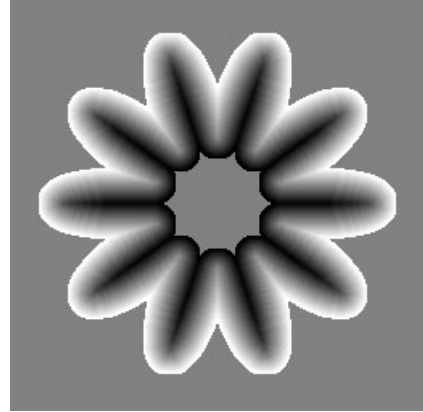


Figure 5.14: Narrow band used in the level set approach.

conditions to fulfill. The first one should guarantee that the *total variation* (total oscillation) decreases as the curve evolves. This measure of “wrinkling” is defined as follows:

$$TV(t) = \int_0^1 |\kappa(q, t)| \sqrt{\left(\frac{\partial x}{\partial q}\right)^2 + \left(\frac{\partial y}{\partial q}\right)^2} dq,$$

and it is shown in [24] to decrease when $\partial F / \partial \kappa < 0$ wherever $\kappa = 0$ (κ is the local curvature). A simple case can be considered to illustrate this result: let $F(\kappa) = 1 \pm \epsilon \kappa$, then $\partial F / \partial \kappa = \pm \epsilon$. In the positive case oscillations grow. This effect corresponds to the backwards diffusion (see Chapter 2) and is unstable.

The second important condition is called *entropy condition* and says that *if the front is viewed as a burning flame, then once a particle is burnt it stays burnt*. In general terms, an entropy condition imposes that no new information can be created during the evolution of the problem. A simple example to illustrate the above definition is to consider a cosine curve propagating with speed $F(\kappa) = 1$: the front develops a sharp corner in finite time and it is not clear how to continue the evolution. Beyond the formation of the discontinuity in the derivative a *weak solution* is needed, so called because the solution weakly satisfies the definition of differentiability⁴. If the moving curve is regarded as an interface separating two regions, as it is the case here, then the weak solution should prevent the front from passing through itself. This weak solution is characterized through the above entropy condition.

The numerical schemes derived to solve the above curve evolution problem use hyperbolic conservation laws to satisfy the entropy condition. Such a scheme is given in the following paragraph for solving a particular approach to curve evolution.

In [25] the authors propose a solution to *shape modeling* through the use of curve evolving under curvature dependent speed and using a *stopping function* to make the curve converge towards object boundaries. The method can be used to detect arbitrary shapes, including shapes with

⁴A solution is said to be a weak solution of a differential equation if it satisfies an integral formulation of the equation.

significant protrusions and in general cases where no a priori assumption is made about the object's topology. In order to improve the computation of evolving curves, a new approach called *level sets theory* [26] is used. This technique and its main issues are presented in the following section.

5.5.2 A level set formulation of curve evolution

Level sets were introduced by Osher and Sethian [26, 27] in a successful attempt to solve (multiple) non-parametric curve evolution problems in a two-dimensional representation. Instead of considering a curve as a set of connected elements in a discrete representation and to follow the evolution in time of every element, with the difficult task of tracking splitting, merging, collapsing, and other changes in the curve topology, the idea is to consider the curve as a propagating front in the two-dimensional space. This approach should solve all the geometrical problems that arise when propagating curves.

In its general formulation, the level set theory is extended to any $(N - 1)$ -dimensional hypersurface Γ , which reduces to a curve when $N = 2$. It is assumed that this hypersurface is propagating along its normal direction with speed F , which is a function of arguments such as curvature or external forces. This propagating interface is embedded as the zero level set of a higher dimensional function $\phi(\mathbf{x}, t) : \mathbb{R}^2 \mapsto \mathbb{R}$. This function is defined as follows:

$$\phi(\mathbf{x}, t = 0) \pm d ,$$

where d is the distance from \mathbf{x} to $\Gamma(t = 0) = \{\mathbf{x} \mid \phi(\mathbf{x}, t = 0) = 0\}$, and the plus (minus) sign is chosen if \mathbf{x} is outside (inside) $\Gamma(t = 0)$. The goal now is to produce an equation for the evolving function $\phi(\mathbf{x}, t)$ that contains the embedded motion of $\Gamma(t)$ as the zero level set $\phi(\mathbf{x}, t) = 0$. Let $\mathbf{x}(t)$ be the trajectory of a point on the propagating front with $\partial\mathbf{x}/\partial t \cdot \mathbf{n} = F(\mathbf{x}(t))$ then the following evolution equation is obtained [26]:

$$\frac{\partial\phi}{\partial t} + F|\nabla\phi| = 0 ,$$

given $\phi(\mathbf{x}, t = 0)$. The term $F|\nabla\phi|$ is known as the *Hamiltonian*.

The level set formulation has the following main properties:

- The evolving function $\phi(\mathbf{x}, t)$ remains a function as long as F is smooth. However the level surface $\phi(\mathbf{x}, t) = 0$ may change topology, break, merge, and form sharp corners as the function evolves.
- Since $\phi(\mathbf{x}, t)$ remains a function as it evolves, numerical simulations can be developed using a discrete grid and substitution of finite difference approximations for the spatial and temporal derivatives.
- Intrinsic geometric properties of the front are easily determined from the level function $\phi(\mathbf{x}, t)$.
- There are no significant changes required to follow fronts in three space dimensions, and therefore propagating surfaces are easily handled.

The applications of level set theory are numerous [27] and include grid generation, image enhancement, noise removal, simulation of physical phenomena like flame combustion and crystal growth [28], and finally shape detection and recognition [25].

To construct a general formulation, let the speed function F be the combination of different contributions:

$$F = F_{\text{prop}} + F_{\text{curv}} + F_{\text{adv}} ,$$

where $F_{\text{prop}} = F_0$ is a constant propagation expansion speed, $F_{\text{curv}} = -\epsilon\kappa$ is the dependence of the speed on the curvature, and finally $F_{\text{adv}} = \mathbf{V}(\mathbf{x}, t) \cdot \mathbf{n}$ is an advection speed (used for example to control the motion through an underlying edge function applied to a particular image), where $\mathbf{n} = \nabla\phi/|\nabla\phi|$ is normal to the front. The level set equation may then be re-written as:

$$\frac{\partial\phi}{\partial t} = -F_0|\nabla\phi| + \epsilon\kappa|\nabla\phi| - \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi . \quad (5.3)$$

The first order convex⁵ scheme that approximates the above equation is given by:

$$\begin{aligned} \frac{\phi_{ij}(n+1) - \phi_{ij}(n)}{\Delta t} = & - \{ \max(F_{0,ij}, 0) \nabla_+ \phi_{ij}(n) + \min(F_{0,ij}, 0) \nabla_- \phi_{ij}(n) \} \\ & + \epsilon\kappa_{ij}(n) \sqrt{(\Delta^x \phi_{ij}(n))^2 + (\Delta^y \phi_{ij}(n))^2} \\ & - \{ \max(u_{ij}(n), 0) \Delta_-^x \phi_{ij}(n) + \min(u_{ij}(n), 0) \Delta_+^x \phi_{ij}(n) \} \\ & - \{ \max(v_{ij}(n), 0) \Delta_-^y \phi_{ij}(n) + \min(v_{ij}(n), 0) \Delta_+^y \phi_{ij}(n) \} , \end{aligned} \quad (5.4)$$

where $\mathbf{V}(\mathbf{x}, t) = (u, v)(\mathbf{x}, t)$ and $\kappa_{ij}(n)$ is the central difference approximation of Eq. 2.8 as defined by Eqs. 2.17, 2.18, and 2.19. The forward and backward-difference approximations are defined in the same way as in § 2.6.4. See that paragraph for details.

In the numerical approximation given by Eq. 5.4 the first term on the right side describes motion in the normal direction and has been obtained through an entropy-satisfying scheme. The second term depending on the curvature, it is a parabolic contribution to the equation of motion and can hence not be obtained by simple upwind scheme [27]. The most straightforward approach is to use central difference approximations to the derivatives. The two last terms correspond to pure passive advection which is approximated through simple upwind scheme. Complete description of the different approximations and extension to second-order convex schemes or non-convex schemes can be found in [27].

In practice, the iterative process defined by Eq. 5.4 becomes very time consuming. To reduce the complexity, the following procedure must be used:

1. Compute the stopping function and its first order derivatives.
2. Construct the initial level function $\phi(\mathbf{x}, t = 0)$.
3. Compute one iteration of Eq. 5.4: at every position on the sampling grid the different speed terms are those of the nearest zero level set element.
4. Extract the zero level set. If it does not change from the previous time step then stop. Otherwise return to step 3.

As a first simplification, one can notice that there is no need to compute the level set evolution over the whole sampling grid. A *narrow band* of samples taken in the neighborhood of the zero level set should be used, as illustrated in Fig. 5.14. The limits of this narrow band must be recomputed

⁵The speed function is supposed to be convex, i.e. F is such that the resulting Hamiltonian $H = F|\nabla\phi|$ is convex.

after a number of iterations. The different speed terms can be taken at the grid location, meaning that other level sets than the zero level step may stop their evolution due to the presence of sharp edges. However, since the level function is recomputed regularly, this does globally not affect the convergence.

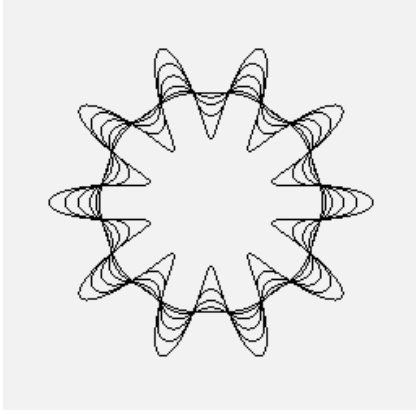


Figure 5.15: Curvature dependent front propagation.

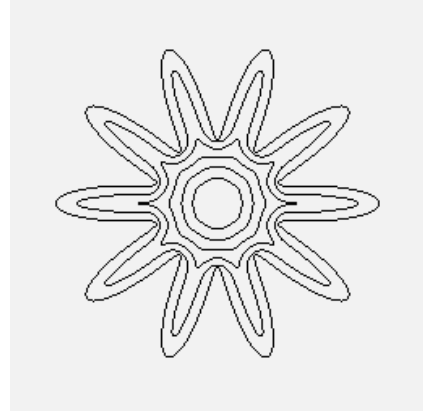


Figure 5.16: Curvature dependent front propagation with constant inward speed term.

Two examples of curve evolution are given in Figs. 5.15 and 5.16 under curvature dependent speed and with additional constant inward motion respectively. In both cases there is no oscillation or collapse of points, and the curve remains smooth during evolution.

In order to fully take advantage of the level set approach and make the link with the active contour concept introduced in the previous section, the following paragraph describes *geodesic active contours*.

5.5.3 Geodesic active contours

Geodesics are minimal distance curves. The minimum distance curve lays in a Riemannian space whose metric is defined by the image content. A particular active contours model lets the curve propagate by means of a velocity that contains two terms, one related to the velocity of the curve and the other shrinks or expands it towards the boundary. The model is given by a geometrical flow (PDE), based on mean curvature motion. This model is characterized by a curve evolution approach instead of an energy minimization one. To derive the geodesic active contour model, the following modifications are performed on the original snake formulation:

- parameter β is set to zero,
- the edge detector $|\nabla v(\mathcal{C}(q))|$ used in $E_{edge}(\mathcal{C}(q))$ is replaced by a function $g^2(|\nabla v(\mathcal{C}(q))|)$ such that $\lim_{r \rightarrow \infty} g(r) = 0$,
- the internal and external energies are equal.

We then consider the following energy functional:

$$E(\mathcal{C}(q)) = \alpha \int_0^1 \left| \frac{d\mathcal{C}(q)}{dq} \right|^2 + \lambda \int_0^1 g^2(|\nabla v(\mathcal{C}(q))|) .$$

It is shown in [29] that minimizing the above energy functional is equivalent to computing the minimum of the following quantity:

$$\mathcal{L}_R = \int_0^{L(\mathcal{C})} g(|\nabla v(\mathcal{C}(q))|) ds ,$$

where $ds = |d\mathcal{C}(q)/dq|dq$ is the arc-length and $L(\mathcal{C})$ is the Euclidean length of \mathcal{C} . The steepest descent method is used to find a local minimum of \mathcal{L}_R . The flow minimizing \mathcal{L}_R is then given by:

$$\frac{\partial \mathcal{C}}{\partial t} = (g(|\nabla v(\mathcal{C}(q))|)\kappa - \nabla g(|\nabla v(\mathcal{C}(q))|) \cdot \mathbf{n})\mathbf{n} ,$$

where κ is the Euclidean curvature and \mathbf{n} is the unit inward normal. Basically the curve is smoothly deforming towards the object. The deformation is due to the curvature term and ∇g attracts the curve towards the objects boundaries. Finally the curve is stopped by $g(|\nabla v(\mathcal{C}(q))|)$.

Geodesic active contours have the following properties:

- they connect the energy and curve evolution approaches of active contours,
- they improve existing curve evolution models,
- there is always a unique, stable and consistent solution,
- they allow simultaneous detection of interior and exterior boundaries in several objects without special contour tracking procedure,
- and they do not require special stopping conditions.

The curve evolution is performed by means of the level set formulation presented in the previous paragraph. By adding to the above equation an extra speed term $\nu g(|\nabla v(\mathcal{C}(q))|)|\nabla \phi|$, $\nu \in \mathbb{R}^+$, that minimizes the area enclosed by the curve, the evolution equation is given by:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= |\nabla \phi| \nabla \cdot \left(g(|\nabla v(\mathbf{x})|) \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu g(|\nabla v(\mathbf{x})|) |\nabla \phi| \\ &= \nu g(|\nabla v(\mathbf{x})|) |\nabla \phi| + g(|\nabla v(\mathbf{x})|) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| + \nabla g(|\nabla v(\mathbf{x})|) \cdot \nabla \phi . \end{aligned} \quad (5.5)$$

The extra speed term increases the convergence speed and can be considered as an area constraint, in the same way that the balloon force does in [16].

Equation 5.5 is the solution of the geodesic problem with an extra area-based speed and constitutes the *geodesic active contour*. The comparison between Eqs. 5.3 and 5.5 leads to the following relationships:

$$\begin{aligned} F_0 &= -\nu g(|\nabla v(\mathcal{C}(q))|) , \\ \mathbf{V}(\mathbf{x}, t) &= -\nabla g(|\nabla v(\mathcal{C}(q))|) , \\ \epsilon &= g(|\nabla v(\mathcal{C}(q))|) . \end{aligned}$$

The different speed terms are all strongly related to the underlying image through $g(|\nabla v(\mathcal{C}(q))|)$. The following function is suggested in [29]:

$$g(|\nabla v(\mathbf{x})|) = \frac{1}{1 + |\nabla v_\sigma(\mathbf{x})|^p}, \quad (5.6)$$

where $v_\sigma(\mathbf{x})$ is a smoothed version of $v(\mathbf{x})$ obtained through Gaussian filtering. The exponent p is either 1 or 2. An alternative would be to use the conductivity function given by Eq. 2.14. In my investigations it has shown to have a better response to edges and to be less sensitive to noise.

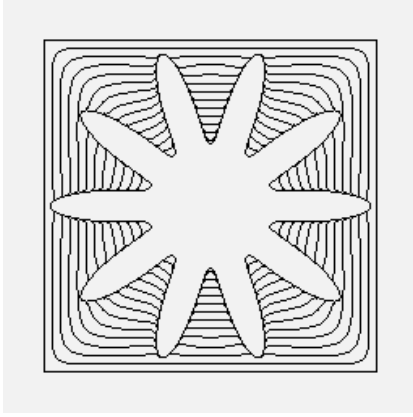


Figure 5.17: Geodesic shape detection with inward motion.

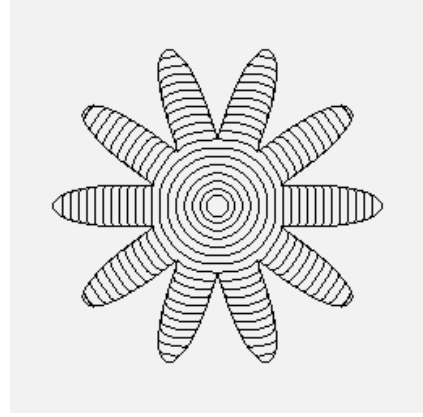


Figure 5.18: Geodesic shape detection with outward motion.

Figures 5.17 and 5.18 show two different evolutions of geodesic active contours applied to the star-shaped object displayed top-left in Fig. 5.12. These results have been obtained using inward motion ($\epsilon > 0$) and outward motion ($\epsilon < 0$) respectively. An application to multiple complex objects detection is shown in Fig 5.19. The original image is shown top-left and a first result is shown top-right. The latter includes shadows, and the stopping function has been scaled (it acts directly on ϵ and F_0) to obtain the result shown bottom-left. The stopping function given by Eq. 5.6 has been used for the different computations, except for that leading to the result shown bottom-right of Fig. 5.12, where the conductivity function given by Eq. 2.14 has been used. In both cases the final contours fit to the different object boundaries. However, the latter function has shown to be better adapted in presence of noise and shadows.

While geodesic active contours are certainly one of the most rigorous formulation of active contours and show excellent performances in detecting arbitrary shapes, the following limitations can be observed:

- the zero level set corresponds to an existing shape or contour only at steady state,
- convergence depends on the stopping function and parameter ν ,
- the process is sensitive to noise and will most of the time require that the image has been previously filtered,

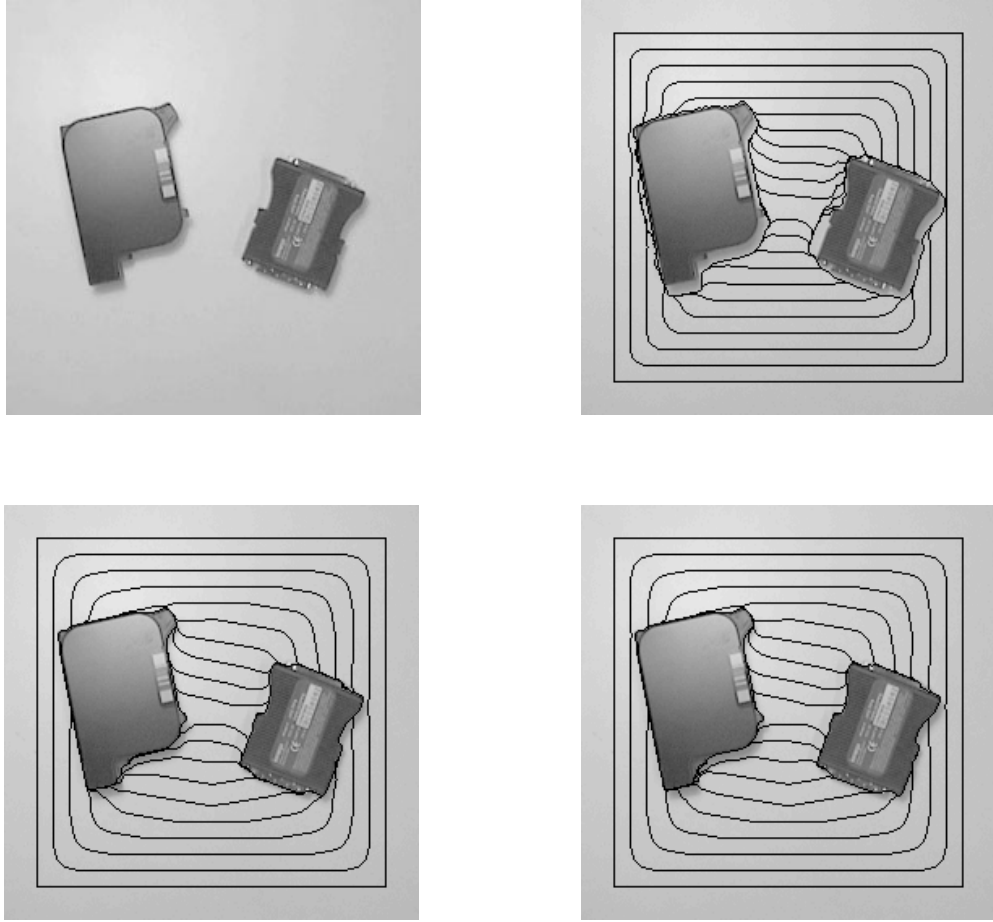


Figure 5.19: Example of geodesic active contours: top-left is the original image, a first example of geodesic curve evolution is shown top-right, where the shadow has been merged with the different objects. By scaling the gradient amplitude, the result shown bottom-left is obtained. The bottom-right image has been obtained by replacing the stopping function given by Eq. 5.6 with the conductivity function given by Eq. 2.14.

- the expanding case must be used when only parts of the objects are visible, meaning often longer convergence time or even false convergence in presence of complex backgrounds,
- the global behavior of snakes has been replaced by a local behavior, which may become a drawback when parts of the objects show a poor contrast with the background, allowing the curve to penetrate into the object,
- when the initial curve is open, it collapses through the end points.

It is usually suggested to filter the original image with a Gaussian low-pass filter. We will not follow this procedure since this type of filter is not well adapted when geometries must be preserved. Following the way suggested in Chapter 2 and using nonlinear isotropic diffusion,

another approach that should solve, under certain circumstances, some of the abovementioned limitations is introduced in the following section.

Several other active contour formulations show a close relationship to the geodesic active contours. In [30] the author extends the front propagation formulation [25] and ends up with the evolution equation of Eq. 5.5. *Gradient vector flow* snakes using dense vector fields derived from images are introduced in [31], and in [32] the geodesic active contour equation is extended to speed up the convergence and solve some of the abovementioned drawbacks.

Finally, the link between the different active contour formulations using PDEs is presented in an excellent tutorial paper by Sapiro [33].

5.6 Nonlinear isotropic diffusion and morphological flooding

In this section a method that provides contours which fit the image morphology is presented. The selection of contours corresponding to object boundaries is then performed by minimizing an energy functional based on gradient information⁶. The contour candidates which are extracted are iso-level curves and can be seen as image *contour lines* and are obtained from a gray-level image.

The following contour extraction scheme is proposed:

- use nonlinear-isotropic diffusion to filter the input image,
- choose a gray-level representation of the filtered image that should render the image shape,
- use morphological flooding to extract the iso-level curves from the gray-level image,
- use morphological opening with geodesic reconstruction to suppress peaks due to light reflection,
- extract contours by slicing [2] and use an energy functional to measure the strength of every contour.

The first step is necessary whatever the approach to contour detection is. Even the geodesic active contour model presented in the previous section performs much better when the image has been previously filtered. The gray-level representation is necessary to obtain the different levels. The important steps are the morphological operations, where flooding ensures that there are no “internal” contours, and opening with geodesic reconstruction suppresses high peaks due to light reflection that may take a large part of the signal’s dynamical range. Finally, the energy functional allows the selection of the best fitting curve.

In the case of color images the nonlinear isotropic diffusion for vector-valued images is used (see § 2.7.6) after changing from the *RGB* to the $L^*u^*v^*$ representation (see § 3.3.3). This combined diffusion makes the different components “mark” each other with their contours. This effect is very useful when trying to extract iso-level contours. The choice of a gray-level image representation is maybe the most sensitive part. However, as will be illustrated later in Chapter 7, the luminance component L^* is shown to be a judicious choice. In the case of the image used so far to illustrate the different active contours, the chrominance component is so poor that the luminance is the only reasonable choice.

⁶Depending on the function used it may become an energy maximization problem.

Once a gray-level image representation has been defined it is modified using morphological flooding. This process can be formulated as an iterative morphological operator using set theory. It is a particular operator, since the input image is a gray-scale image and the output is a binary image. It is therefore necessary to introduce some of the gray-level operators previously defined in Chapter 2 for binary images. Let \mathbb{A} and \mathbb{B} be sets in \mathbb{Z}^2 , with components \mathbf{a} and \mathbf{b} respectively. The *translation* of \mathbb{A} by \mathbf{x} is defined as [2]:

$$(\mathbb{A})_{\mathbf{x}} = \{\mathbf{c} \in \mathbb{Z}^2 \mid \mathbf{c} = \mathbf{a} + \mathbf{x}, \quad \mathbf{a} \in \mathbb{A}\},$$

and the *reflection* of \mathbb{B} is defined as [2]:

$$\hat{\mathbb{B}} = \{\mathbf{x} \in \mathbb{Z}^2 \mid \mathbf{x} = -\mathbf{b}, \quad \mathbf{b} \in \mathbb{B}\}.$$

The *dilation* of \mathbb{A} by \mathbb{B} is defined as [2]:

$$\mathbb{A} \oplus \mathbb{B} = \{\mathbf{x} \in \mathbb{Z}^2 \mid (\hat{\mathbb{B}})_{\mathbf{x}} \cap \mathbb{A} \neq \emptyset\}.$$

With these three definitions we can now give a set formulation to the morphological flooding. Let the input image be described by a function $v : \mathbb{Z}^2 \mapsto \mathbb{R}$, \mathbb{D}_v is the image domain and $\partial\mathbb{D}_v$ is the image border domain. The following operator is used to build the subset of flooded pixels at level $l \in \mathbb{R}$:

$$\mathbb{F}_k^{(l)} = (\mathbb{F}_{k-1}^{(l)} \oplus \mathbb{B}) \cap \{\mathbf{x} \in \mathbb{D}_v \mid v(\mathbf{x}) \leq l\}, \quad k \in \mathbb{N}^*,$$

where \mathbb{B} is a structuring element, usually a cross (4-connected) or a 3×3 square (8-connected). The operator is initialized for $k = 0$ as follows:

$$\mathbb{F}_0^{(l)} = \{\mathbf{x} \in \partial\mathbb{D}_v \mid v(\mathbf{x}) \leq l\},$$

and it has converged when $\mathbb{F}_k^{(l)} = \mathbb{F}_{k-1}^{(l)}$ and we let then $\mathbb{F}^{(l)} = \mathbb{F}_k^{(l)}$ be the set of pixels flooded at level l .

It is straightforward that the complete procedure is not repeated for every level, and for $l_2 > l_1$ the flooded subset for level l_2 is obtained by initializing the procedure with the flooded subset of level l_1 . The only condition is that:

$$\mathbb{F}_0^{(l_2)} = \mathbb{F}^{(l_1)} \cup \{\mathbf{x} \in \partial\mathbb{D}_v \mid v(\mathbf{x}) \leq l_2\}.$$

The above equation ensures that flooding can start at local border minima. The flooding process provides a kind of 2-D *convex hull* of the gray level image, which has the following feature: *there always exist a convex path between the image border and any point in the image*. This hull is obtained with the different subsets $\mathbb{F}^{(l_i)}$, $i = 1, 2, 3, \dots$, and $\cup_i \mathbb{F}^{(l_i)} \equiv \mathbb{D}_v$, by labeling every image pixel in \mathbb{D}_v according to the following rule:

$$h(\mathbf{x}) = l_i, \quad \forall \mathbf{x} \in \{\mathbf{x} \in \mathbb{F}^{(l_i)} \mid \mathbf{x} \ni \mathbb{F}^{(l_{i-1})}\},$$

and

$$h(\mathbf{x}) = l_0, \quad \forall \mathbf{x} \in \mathbb{F}^{(l_0)}.$$

Finally, the iso-level curve energy is computed as follows:

$$\mathcal{E}_l = (1 + E[e(\mathcal{C}_l)])(1 + Var[e(\mathcal{C}_l)]), \quad (5.7)$$

where $e(\mathcal{C}_l)$ is actually the energy measure given by the edge function, $E[e(\mathcal{C}_l)]$ is the expected or mean energy along the curve and $Var[e(\mathcal{C}_l)]$ is the corresponding variance. In order to select the curve with lowest energy and lowest energy variance the above equation is used. The unit offset in both parentheses ensures that if only one of the two measures is zero the resulting energy is non-zero. The energy term $e(\mathcal{C}_l)$ is taken equal to the conductivity function given by Eq. 2.14, with a squared gradient equal to the sum of squared gradients of the different components L^* , u^* , and v^* , thus taking the color difference formula (see Eq. 3.4) into account to measure color derivatives as color distances.

Obviously looking at the iso-level curves in gray-scale images renders the boundary detection sensitive to the presence of shading. A way to reduce this effect is to use gradient information instead of the gray-level amplitude. The “level” information is lost and only edge information remains, but in that case the morphological flooding recovers the image components from the edge function. The global effect, however, is that boundaries are slightly shifted outwards.



Figure 5.20: Gray-level image after nonlinear isotropic diffusion and morphological flooding.

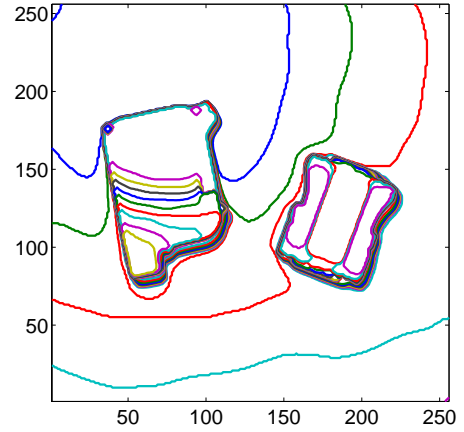


Figure 5.21: Iso-level curves obtained from Fig. 5.20.

The image shown top-left in Fig. 5.19 has been used to illustrate the proposed approach. In Fig. 5.20 the result obtained after nonlinear isotropic diffusion and morphological flooding is shown. The corresponding iso-level curves are shown in Fig. 5.21. The use of a tungsten lamp to illuminate the objects become visible. The influence of this non-homogeneous lightning is reduced by the three component diffusion. Figure 5.22 shows the corresponding energy functional and the position of the minimum. The corresponding contour is superimposed on the original image in Fig. 5.23. Obviously part of the shading has been included in the contour. However, as we be shown in Chapter 7, this method can be successfully and quickly applied to cases where the lightning is more homogeneous, and especially for medical applications. It must be outlined here that the shading has been slightly reduced with the diffusion process, even if the original image shows to have a poor chrominance component. When objects are better differentiated from a spectral point of view, this approach performs much better, even for inspection applications.

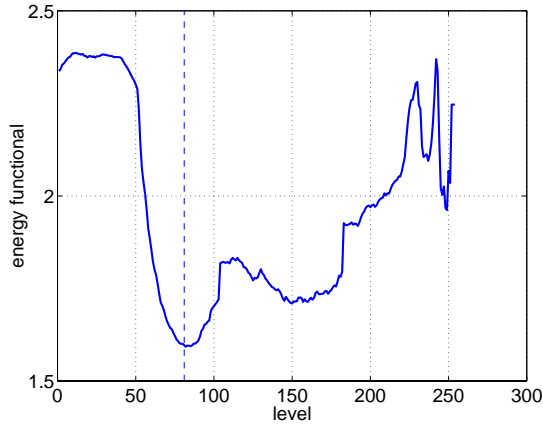


Figure 5.22: Energy functional obtained by computing Eq. 5.7 for each iso-level curve.

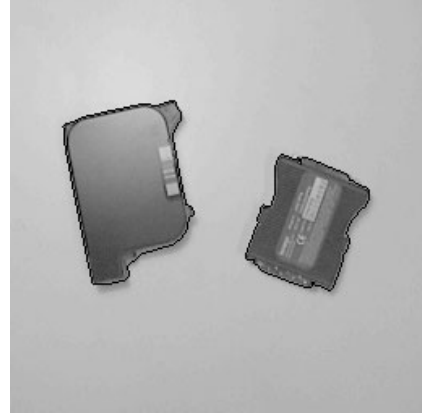


Figure 5.23: Minimum energy iso-level contour superimposed on the original image.

5.7 Summary

In this chapter the detection of multiple closed contours in images, without constraint on the topology, has been investigated. Low-level edge detectors have been presented in Section 5.2 and active contour models have been presented in Sections 5.4 and 5.5. Section 5.3 presents Canny's edge detector. The active contour techniques are very well adapted to the detection of multiple objects without any constraint on the contour topology, especially the geometric formulations of active contours, when compared to parametric snakes. The notion of curve evolution has been presented in § 5.5.1 and a level set formulation to the problem has been described in § 5.5.2. One of the most complete and rigorous formulation of geometric active contours, namely geodesic active contours, has been presented in § 5.5.3. Finally, the detection of multiple closed contours using nonlinear isotropic color diffusion and morphological flooding, used to extract iso-level curves, has been introduced in Section 5.6.

Now that the detection of objects of interest through segmentation and boundary detection has been investigated, the following chapter deals with the classification of objects through symmetry measurements. The schemes that will be presented in Chapter 6 take into account shape, color, and texture information in the symmetry quantification process. Finally these methods will be applied to objects with any "degree" of symmetry.

Bibliography

- [1] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
- [2] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [3] D. Marr and E. Hildreth. Theory of edge detection. *Proceeding of the Royal Society of London*, B207:187–217, 1980.
- [4] R. M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):58–68, January 1984.
- [5] M. Kunt, G. Granlund, M. Kocher, and C. Horne. *Traitement numérique des images*, volume 2 of *Traitement de l'information*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1993.
- [6] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the Association for Computing Machinery (ACM)*, 15(1):11–15, January 1972.
- [7] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [8] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.
- [9] V. S. Nalwa and T. O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):699–714, November 1986.
- [10] N. Saito and M. A. Cunningham. Generalized E-filter and its application to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):814–817, 1990.
- [11] R. M. Haralick and J. S. J. Lee. Context dependent edge detection and and evaluation. *Pattern Recognition*, 23(1-2):1–19, 1990.
- [12] M. Petrou and J. Kitter. Optimal edge detector for ramp edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):483–491, 1991.
- [13] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

- [14] R. Courant and F. John. *Introduction to Calculus and Analysis*, volume 2. Springer Verlag, 1989.
- [15] R. Courant and F. John. *Introduction to Calculus and Analysis*, volume 1. Springer Verlag, 1989.
- [16] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 53(2):211–218, March 1991.
- [17] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.
- [18] K. F. Lai. *Deformable Contours: Modeling, Extraction, Detection, and Classification*. PhD thesis, University of Wisconsin-Madison, 1994.
- [19] K. F. Lai and R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1084–1090, November 1995.
- [20] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proceedings of the Fifth International Conference on Computer Vision, Cambridge, Massachusetts*, pages 840–845. IEEE Computer Society, IEEE Computer Society Press, 20-23 June 1995.
- [21] H. Eviatar and R. L. Somorjai. A fast, simple active contour algorithm for biomedical images. *Pattern Recognition Letters*, 17:969–974, 1996.
- [22] J. M. Pardo, D. Cabello, and J. Heras. A snake for model-based segmentation of biomedical images. *Pattern Recognition Letters*, 18:1529–1538, 1997.
- [23] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209, April 1997.
- [24] J. A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.
- [25] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.
- [26] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [27] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [28] J. A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98:231–253, 1992.
- [29] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February-March 1997.

- [30] S. Kichenassamy, P. Olver, A. Kumar, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 810–815, 20-23 June 1995.
- [31] Ch. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998.
- [32] K. Siddiqi, Y. Bérubé Lauzière, A. Tannenbaum, and W. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Transactions on Image Processing*, 7(3):433–443, March 1998.
- [33] G. Sapiro. From active contours to anisotropic diffusion: connections between basic PDE’s in image processing. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 477–480, 16–19 September 1996.

Chapter 6

Symmetry axes and symmetry quantification

6.1 Introduction

The third logical step in image analysis applications after image filtering and object boundary detection is the computation of one or more features. At this stage the goal is to focus the analysis part on relevant characteristics that should allow differentiation between several groups of objects. In this chapter the problem of symmetry quantification for classification tasks will be investigated. This means that not only symmetrical objects will be processed but that the different approaches developed in the scope of this thesis will be applied to any type of objects. The novelty of the method consists of the concepts of *symmetry measure*, which, unlike other methods, allows us to identify symmetry axes based on an index or “degree” of symmetry.

Symmetry is a *natural characteristic* that objects may have in nature. This feature can be used to reduce the amount of data needed to represent symmetrical objects (ideally by 50 %). In the case of intrinsically symmetrical objects, finding the axis is not a big effort and different existing techniques give access to this information [1, 2, 3, 4]. The problem becomes more tricky when symmetry is not perfect or even inexistent. In the former case, standard methods like the *principal component* decomposition, may give a satisfactory approximation. If symmetry detection must provide more than just the *best fitting symmetry axis*, as it is the case in classification and recognition tasks, more elaborate approaches are needed than simple approximations or methods devoted to almost perfect symmetry.

In this chapter a new approach to symmetry axis computation is introduced, where a symmetry measure is defined first and then optimized for a given object. The different definitions and a review of “standard” approaches to symmetry axis computation are presented first in Sections 6.2 and 6.3. The different approaches to the principal component decomposition are presented in Section 6.4. Symmetry optimization is introduced in Section 6.5. Two schemes, one genetic algorithm and one based on self-organizing maps, are used to compute the axis with highest symmetry. A more general approach is introduced in Section 6.6, where symmetry maps are used to analyze the symmetry of objects. This approach can be seen as a kind of non-reversible transform that maps any image to a symmetry space. Two multi-resolution approaches are proposed to compute quickly this symmetry map.

6.2 Definition and approaches

6.2.1 What is symmetry ?

In geometry, symmetry is *an exact correspondence in position or form about a given point, line, or plane*. From this definition it follows that the extraction of a symmetry axis is possible only for intrinsically symmetrical objects. In general this is not the case in natural images. Even for symmetrical objects, perfect symmetry is impossible to obtain in digital images due to imperfect lighting, digitization, or occlusion. The term of symmetry axis is often used for almost-symmetrical images or objects and corresponds to the axis that minimizes the visual difference between the original and its reflection.

The different approaches to symmetry axis computation can be:

- based on human symmetry perception,
- model-based,
- measure-based.

The former approach is based on the *human visual system* (HVS) and the knowledge we have about how the human brain perceives and analyses symmetrical patterns. However, it appears that this research field has led to a number of often contradictory results [5]. The second approach takes into account a model of the observed objects in order to compute the symmetry axes. This characteristic may be used to facilitate the computation [6]. However there is not always a priori knowledge on what will be analyzed. The last approach, which is investigated in this chapter, defines a symmetry measure and attempts to find the axis that maximizes this function.

6.2.2 Human symmetry perception

When investigating the problem of detecting symmetries in images or measuring the “degree” of symmetry of objects, it may be interesting to follow the way humans perceive symmetry. It is indeed a very interesting research topic, as it is the case for everything related to the HVS and in general to the human intelligence and senses. It is however important to keep at mind that human beings are capable of interpretation, analysis, decision, etc., but there is always a part of subjectivity or uncertainty in the way things are perceived. The important point is to know if we want to mimic the way human work or if we need precise and reproducible (*robust*) measurement methods. In our case, even if human beings are very efficient in the detection of symmetries, the quantification problem is almost impossible to answer.

Several studies have attempted to take into account the human symmetry perception to solve detection problems. A survey of such papers can be found in [5]. What comes out when reviewing the work that has been done in this domain up to now is that a number of characteristics are not clearly established and even sometimes explained in contradictory ways. It is therefore difficult to exploit efficiently the human symmetry perception.

The approach proposed in this chapter is a computational approach to symmetry quantification. The advantage of this approach is that since the core is the symmetry measure, the same schemes can be used later with more HVS-oriented symmetry measures. In order to maintain the level of complexity within acceptable limits, a simple mean squared error measure is adopted in this study, as explained in Section 6.5. A number of techniques used in computer vision will be reviewed first.

The analogy between orientation and symmetry will be discussed, and the principal component decomposition as a quick and efficient way of computing the orientation of objects will be presented in Section 6.4.

6.3 Symmetry in computer vision

Several methods have been proposed to compute the *orientation* of two-dimensional discrete objects. A widely used method is the discrete Karhunen-Loève transform, also called principal component decomposition, which can be used either for binary images [7] or gray level images. In the latter case the pixel coordinates are weighted by the gray level values. Finally, the principal component decomposition can be extended to the spatial frequency domain [8, 9]. In our investigations, the frequency-based principal component approach has been shown to yield results matching the perceived symmetry. However, this approach gives the exact axis orientation while the position is not defined in a unique way. This technique will be described in detail in § 6.4.3.

A method to compute the best symmetry axis called *n-transform* was first proposed in [10]. The *n-transform* F_n of an image or object is obtained by multiplying component-wise each coordinate vector x_i n times by itself, the image being centered at the center of mass. It is shown in [10] that both the number and the location of all the axes of symmetry can be computed from the center of mass of F_{n_i} and $F_{n_{i+1}}$, where n_1, n_2, \dots are the values of n for which the center of mass is not located at the origin. A coefficient of symmetry β is also introduced. The exact location of the symmetry axis may be found by maximizing β , starting from the axis provided by the *n-transform*. For this purpose, β is decomposed into a power series. The complete derivation can be found in [10]. This method is mainly devoted to symmetrical and almost-symmetrical objects. In addition, the equations that must be solved to compute an approximate axis are rather complex.

A method to extract rotational and reflectional symmetry, that is symmetry about a point and an axis respectively, is proposed in [11]. It is based on successive matching of transformed versions of the image in order to obtain the best match. The directional correlation of edge features is used to evaluate the matching. This method may require a large number of iterations to find the symmetry axis location. However, the advantage of this method is that it does not require any knowledge about the centroid and it allows also for the detection of symmetry properties in almost-symmetrical images.

The automatic detection of rotational symmetry is presented in [12]. This method, devoted to shape analysis, is based on the determination of the number of intersections between a closed curve \mathcal{C} and the circle that is centered at the centroid of \mathcal{C} with a radius equal to the mean distance to the curve. This method however works only for symmetrical objects. For non-symmetrical objects it is better to compute the principal axis.

In [13] a method for detecting mirror symmetry using gray level information and local orientation applied to the recognition and tracking of cars is presented. This method, however, assumes that only vertical or near vertical symmetries are of interest. The orientation of the gradient vectors in the gray scale image is used in [14] to compute the symmetry axis orientation. This method, like all gradient-based methods, has been shown to be excessively noise sensitive. Symmetry has been treated as a continuous feature in [2]. They define a symmetry distance that allows for reconstructing symmetry of occluded shapes. Symmetry in planar shapes is discussed in [1].

Other approaches based on shape information and polygonal representation can be found

in [15, 4]. A more general approach to the search for symmetry axes of skewed objects can be found in [16].

Most of the studies do not make use of a symmetry measure for the symmetry axis computation. The introduction of such a measure becomes necessary when dealing with images that might be clearly asymmetrical but where symmetry must be quantified. This approach generates new constraints that must be taken into account. For example, the segmentation problem, which should separate the object from the background, must be solved in advance. The reason is that we use a symmetry measure that *must not* be influenced by the image frame or any smooth window.

6.4 Principal component approaches

6.4.1 Symmetry and orientation

An interesting alternative to symmetry axis computation is principal component decomposition. The idea is to obtain an axis that defines the *orientation* of a given object of interest. The main advantage of such an approach is the significant gain in computation time compared to other more rigorous and complex approaches to symmetry axis computation.

A conceptual limitation of this method is the analogy between symmetry and orientation. The comparison is not always possible and the orientation coincides with a symmetry axis only in particular cases. However, this technique can be used efficiently in numerous applications such as the data dimension reduction presented in § 3.3.2 for color spaces. This is the reason why the basic concepts underlying the principal component decomposition are presented in the following paragraphs.

6.4.2 Spatial formulation

Several methods have been proposed to compute the orientation of two-dimensional discrete objects. The most widely used method is certainly the *Karhunen-Loève transform*, also called *principal component decomposition* or *Hotelling transform*. The two-dimensional formulation is presented next for application to binary and gray-level images.

Given a subset $\mathbb{X} \subset \mathbb{R}^2$ of vectors with weights $w(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, the center or mean value is defined as:

$$\mathbf{m}_{\mathbf{x}} = \frac{\sum_{\mathbb{X}} w(\mathbf{x}) \mathbf{x}}{\sum_{\mathbb{X}} w(\mathbf{x})}. \quad (6.1)$$

Subset \mathbb{X} is oriented along the line for which the following error function is minimum:

$$e(\mathbf{k}_p) = \sum_{\mathbb{X}} w(\mathbf{x}) \left\| (\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T \mathbf{k}_p \right\|^2, \quad (6.2)$$

where each vector $\mathbf{x} \in \mathbb{X}$ is projected on the unit vector \mathbf{k}_p , as shown in Fig. 6.1. This vector is normal to the unit vector \mathbf{k} parallel to the orientation line. This error function is the weighted mean squared distance computed from the line passing through the mean vector $\mathbf{m}_{\mathbf{x}}$ and oriented

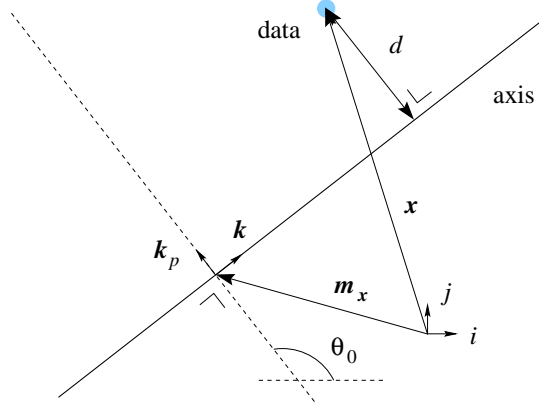


Figure 6.1: The principal component of an n -dimensional data distribution is the axis from which the data variance is minimum.

in direction of \mathbf{k} . The distance term in Eq. 6.2 can be rewritten as:

$$\begin{aligned}
 d^2(x, \mathbf{k}_p) &= \left\| (x - \mathbf{m}_x)^T \cdot \mathbf{k}_p \right\|^2 \\
 &= \left((x - \mathbf{m}_x)^T \cdot \mathbf{k}_p \right) \left((x - \mathbf{m}_x)^T \cdot \mathbf{k}_p \right) \\
 &= \left(\mathbf{k}_p^T \cdot (x - \mathbf{m}_x) \right) \left((x - \mathbf{m}_x)^T \cdot \mathbf{k}_p \right) \\
 &= \mathbf{k}_p^T \left((x - \mathbf{m}_x) (x - \mathbf{m}_x)^T \right) \cdot \mathbf{k}_p,
 \end{aligned}$$

and Eq. 6.2 becomes:

$$\begin{aligned}
 e(\mathbf{k}_p) &= \mathbf{k}_p^T \left\{ \sum_{\mathbb{X}} w \left((x - \mathbf{m}_x) \cdot (x - \mathbf{m}_x)^T \right) \right\} \cdot \mathbf{k}_p \\
 &= \mathbf{k}_p^T \mathbf{A} \cdot \mathbf{k}_p.
 \end{aligned} \tag{6.3}$$

Matrix \mathbf{A} is symmetrical and semi-positive definite. It is the *covariance matrix* of data set \mathbb{X} . Minimizing the error function given in Eq. 6.3 is equivalent to solving the eigenvalue/eigenvector system for \mathbf{A} . Since this matrix is symmetrical, one can write

$$\mathbf{D} = \mathbf{P}^T \mathbf{A} \mathbf{P},$$

where \mathbf{D} is a diagonal matrix whose elements are the eigenvalues $\lambda_{1,2}$ of \mathbf{A} , with $\lambda_1 \geq \lambda_2$ and \mathbf{P} is an orthogonal matrix whose columns are the corresponding orthonormal eigenvectors. Minimizing Eq. 6.3 is equivalent to finding the smallest eigenvalue of \mathbf{A} and the corresponding eigenvector, which is then equal to \mathbf{k}_p . The vector \mathbf{k} being orthogonal to \mathbf{k}_p , it is the eigenvector corresponding to the largest eigenvalue λ_1 . Note that the two eigenvalues correspond to the variance of weighted vectors $\mathbf{x} \in \mathbb{X}$ with respect to the corresponding eigenvector.

Starting from Eq. 6.3 one has to define the weights $w = w(\mathbf{x})$. For our application the set of vectors \mathbb{X} corresponds to the coordinates of the pixels which form the image. The simplest application of the principal component method makes use of binary objects. A mask describes the

shape of the object, which means that the inner structure of the analyzed object is not taken into account, and therefore this method is only shape sensitive. In this case the weights are defined as

$$w(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is in the object,} \\ 0 & \text{if } \mathbf{x} \text{ is outside the object.} \end{cases} \quad (6.4)$$

The notion of principal component is illustrated with two binary objects displayed in Figs 6.2 and 6.3. These two isosceles triangles have a unique symmetry axis which, for the first one, correspond to the orientation line. In the case of Fig. 6.3, the basis is larger than the height and the orientation is finally orthogonal to the symmetry axis. The main limitation of the principal component decomposition applied to symmetry problems is the possible weak relationship between symmetry axis and orientation. However, in numerous cases this approach is very interesting due to its simplicity and the low processing time it requires.

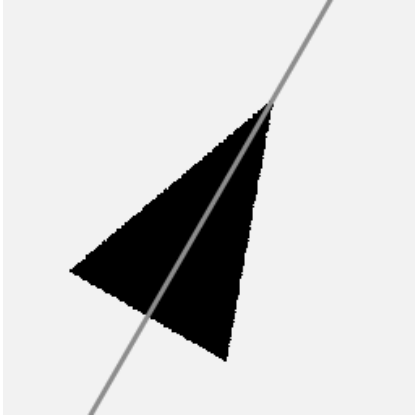


Figure 6.2: Symmetry axis and orientation are identical.



Figure 6.3: Symmetry axis and orientation are different.

The gray level information can be embedded into the principal component decomposition through the following choice of weights:

$$w(\mathbf{x}) = h(\mathbf{x}) v(\mathbf{x}) , \quad (6.5)$$

where $v(\mathbf{x})$ is the intensity function and $h(\mathbf{x})$ a window function. It may be simply given by Eq. 6.4, which correspond to the weight function in the binary case. The influence of the border location on the axis computation is difficult to establish but it is certain that the computation may be very sensitive to shape information, especially for objects with low contrast. A more rigorous approach is to use a soft isotropic window, like a Gaussian window, which has the minimal product of position and frequency bandwidths:

$$h(\mathbf{x}) = \exp \left(\frac{(\mathbf{x} - \mathbf{m}_x)^T (\mathbf{x} - \mathbf{m}_x)}{2\sigma^2} \right) , \quad (6.6)$$

where σ^2 is the variance, or other types of windows used for spectral analysis like the Hamming or Hanning windows for example. The consequence is that the computation of axes does not depend

on the segmentation result. However, this method suffers from an important drawback: the value of σ (or in general the window size parameter) must be adapted to the size of the object of interest in order to reduce the influence of image pixels located outside.

In addition, the principal axis of inertia is sensitive to any linear transformation of the intensity values. A possible modification of Eq. 6.5 that renders the principal axis of inertia insensitive to shifting and scaling of the intensity function is:

$$w(\mathbf{x}) = h(\mathbf{x}) \frac{v(\mathbf{x}) - E[v]}{\sqrt{E[(v(\mathbf{x}) - E[v])^2]}},$$

where $E[.]$ is the expected value. The computation is then less sensitive to low contrast problems.

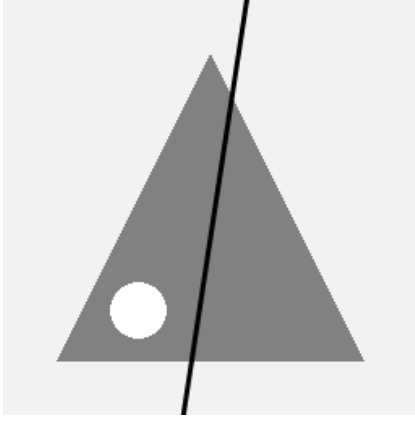


Figure 6.4: The symmetrical object shown above has been corrupted with a white hole.

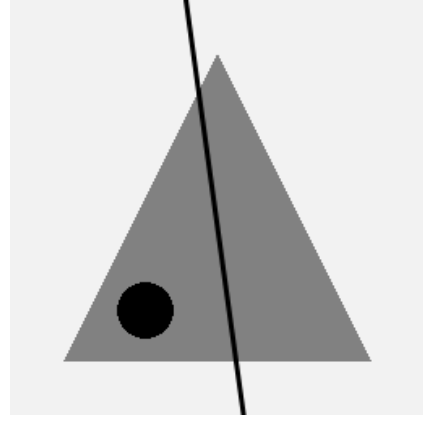


Figure 6.5: The same experiment than that shown in Fig. 6.4 has been done with a black hole.

Figures 6.4 and 6.5 illustrate the drawback when using gray-levels as weights. Both figures show the same light gray triangle corrupted with a white and black hole respectively. While the introduced perturbation is “identical” from a symmetry point of view, the principal component is different. In the former case there is a gain of “weight” on the left side, while there is a loss of “weight” in the latter case. This deficiency will be solved in the frequency domain formulation introduced in § 6.4.3.

This gray level principal component decomposition is identical to the principal axis of inertia computation in mechanics, the gray level being associated to the weight [17]. An alternative may be to consider the square intensity as weight. In that case the weight can be seen as a local energy measure.

The most rigorous principal component formulation searches for the orientation in the Fourier domain. This approach gives more importance to the image structure than in the formulations presented in this paragraph and is described in the following paragraph.

6.4.3 Frequency domain formulation

A number of papers deal with the computation of local orientation for texture classification [18, 8, 9, 19], taking advantage of the fact that the orientation of objects can be computed in the Fourier domain. Knowing that broad signals have a narrow band in the frequency domain and inversely, the principal component in the Fourier domain will correspond to the direction where the signal is narrow. Therefore, the orthogonal direction is considered as the orientation line in the spatial domain. The weights are given by $w(\mathbf{f}) = |\Upsilon(\mathbf{f})|^2$, where $\Upsilon(\mathbf{f}) = \mathcal{F}[v(\mathbf{x})]$ is the image Fourier transform and $\mathbf{f} \in \mathbb{R}^2$ is the frequency coordinate¹. The error function is then given by:

$$e(\mathbf{k}_p) = \mathbf{k}_p^T \left\{ \sum_{\mathbb{X}} |\Upsilon(\mathbf{f})|^2 \left((\mathbf{f} - \mathbf{m}_f)(\mathbf{f} - \mathbf{m}_f)^T \right) \right\} \cdot \mathbf{k}_p \quad (6.7)$$

The approach proposed in [8] solves Eq. 6.7 in the spatial domain using the Parseval relationship [20]:

$$\int f_k f_l |\Upsilon(\mathbf{f})|^2 d\mathbf{f} = \int \frac{\partial v(\mathbf{x})}{\partial x_k} \frac{\partial v(\mathbf{x})}{\partial x_l} d\mathbf{x} , \quad k, l = 1, 2 ,$$

This equation allows for using the partial derivatives of the intensity function to compute the covariance matrix. Using the Parseval relationship in the discrete case, Eq. 6.7 becomes:

$$\begin{aligned} e(\mathbf{k}_p) &= \mathbf{k}_p^T \left\{ \sum_{\mathbb{X}} |\Upsilon(\mathbf{f})|^2 \mathbf{f} \mathbf{f}^T \right\} \cdot \mathbf{k}_p \\ &\cong \mathbf{k}_p^T \left\{ \sum_{\mathbb{X}} h(\mathbf{x}) \left(\frac{\partial v}{\partial \mathbf{x}} \right) \left(\frac{\partial v}{\partial \mathbf{x}} \right)^T \right\} \cdot \mathbf{k}_p , \end{aligned} \quad (6.8)$$

where $h(\mathbf{x})$ is an interpolating and smoothing function with Gaussian shape as given by Eq. 6.6. Since the spatial translation to the center \mathbf{m}_x affects only the signal phase in the Fourier domain, the error function given by Eq. 6.8 is written in the frequency domain with $\mathbf{m}_f = \mathbf{0}$.

Using complex analysis, Eq. 6.8 reduces to the following set of equations:

$$z = \sum_{\mathbb{X}} (\nabla_{\mathbb{C}} v(\mathbf{x}))^2 h(\mathbf{x}) , \quad (6.9)$$

$$\arg z = 2\theta_0 = \arctan \left(2 \frac{\partial v / \partial x \partial v / \partial y}{\partial^2 v / \partial x^2 - \partial^2 v / \partial y^2} \right) , \quad (6.10)$$

$$|z| = 4\pi^2 (e(\mathbf{k}_{\max}) - e(\mathbf{k}_{\min})) , \quad (6.11)$$

where $\nabla_{\mathbb{C}} v(\mathbf{x}) = \partial v / \partial x + j \partial v / \partial y$ is the image gradient in complex notation, and θ_0 is the angle that \mathbf{k}_p forms with axis i (see Fig. 6.1). The two quantities \mathbf{k}_{\max} and \mathbf{k}_{\min} are the smallest and largest fitting errors obtained through Eq. 6.2. If the two error terms are needed separately, the following equations must be used:

$$\begin{aligned} e(\mathbf{k}_{\min}) &= \frac{1}{8\pi^2} \left(\sum_{\mathbb{X}} |\nabla_{\mathbb{C}} v(\mathbf{x})|^2 h(\mathbf{x}) - |z| \right) , \\ e(\mathbf{k}_{\max}) &= \frac{1}{8\pi^2} \left(\sum_{\mathbb{X}} |\nabla_{\mathbb{C}} v(\mathbf{x})|^2 h(\mathbf{x}) + |z| \right) . \end{aligned}$$

A complete derivation of the equations given above can be found in [8].

¹The continuous space formulation is presented first.

The proposed method allows one to find the object orientation in an image. The remaining degree of freedom is the center of gravity, which must correspond to the origin of the smoothing function $h(\mathbf{x})$. This method has been developed to compute the local orientation as a feature for texture classification and therefore $h(\mathbf{x})$ is chosen so that it covers a small region. When using the entire image, one should choose an approximate origin and compute the real center by means of convolution over a small neighborhood. The final center is then chosen to be the point where Eq. 6.11 is maximal, as suggested in [8]. However, in order to reduce the computation time, Eq. 6.1 should be used to compute directly the center of gravity.

The principal component approach based on spatial frequencies has shown to be a robust method. As mentioned above, the center is computed using the gray scale distribution. The axes are then computed using the frequency approach. However, this method depends on the smoothing function $h(\mathbf{x})$, which requires the object to be small and centered in the image in order to take into account most of the structure information.

When using the gray levels as weights, one assumes that white is “heavier” than black. However, the gray level reflects the amount of perceived light and the correspondence with mass is not evident. In the frequency domain approach, the weights correspond to the energy at given frequencies. The notion of weight in this case is more accurate. The two examples given in Figs. 6.4 and 6.5 illustrate the deficiency of the gray level formulation. From Eq. 6.10 it appears that the perturbations introduced in both cases have the same effect on the orientation when using the frequency domain formulation.

Another interesting method for computing the orientation of objects is to fit an ellipse to the object boundary. This approach is presented in the following paragraph.

6.4.4 Orientation through ellipse fitting

A solution to the general problem of fitting conic sections to scattered was proposed in [21]. Recently, a solution to the specific problem of ellipse fitting was proposed in [22]. Given a second order polynomial that represents a generic conic by its zero set:

$$F(\mathbf{a}, \boldsymbol{\chi}) = \mathbf{a}\boldsymbol{\chi} = ax^2 + bxy + cy^2 + dx + ey + f, \quad (6.12)$$

where $\mathbf{a} = (a \ b \ c \ d \ e \ f)$ and $\boldsymbol{\chi} = (x^2 \ xy \ y^2 \ x \ y \ 1)^T$, we define the design matrix $\mathbf{D} = (\boldsymbol{\chi}_1 \ \boldsymbol{\chi}_2 \ \cdots \ \boldsymbol{\chi}_N)^T$ and the scatter matrix $\mathbf{S} = \mathbf{D}^T \mathbf{D}$. It has been shown [22] that the unique elliptical solution is the one that minimizes the algebraic distance over the set of N data points in the least square sense, that is:

$$\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \left\{ \sum_{i=1}^N F(\mathbf{a}, \boldsymbol{\chi}_i)^2 \right\}, \quad (6.13)$$

subject to the constraint:

$$b^2 - 4ac = \mathbf{a}^T \mathbf{C} \cdot \mathbf{a} = -1,$$

with:

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Equation 6.13 can be solved by the rank-deficient generalized eigenvalue system:

$$\mathbf{D}^T \mathbf{D} \mathbf{a} = \mathbf{S} \mathbf{a} = \mathbf{C} \mathbf{a} .$$

The parameters of the ellipse are given by the eigenvector corresponding to the only negative eigenvalue of $\mathbf{S}^{-1} \mathbf{C}$ [22].

The principal component analysis and the ellipse fitting problem have some similarities. The former attempts to minimize the variance with respect to the axes, whereas the latter method attempts to minimize the least square error between the ellipse and the object contour. The two eigenvalues $\lambda_{1,2}$ obtained using the principal component analysis represent the variance along the two corresponding eigenvectors. Thus the ratio $\sqrt{\lambda_1/\lambda_2}$, which corresponds to the ratio of the standard deviations from the two axes, is analogous to the ratio a_1/a_2 , a_1 and a_2 being the two ellipse axes. Since ellipse fitting uses contour information, the process is only shape sensitive and there is no way to integrate texture or color content of objects to the computation.

In order to compute the different ellipse parameters, let us rewrite Eq. 6.12 as follows:

$$F(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + 2\mathbf{u}^T \mathbf{x} + f ,$$

with:

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} , \\ \mathbf{u} &= \begin{pmatrix} d/2 \\ e/2 \end{pmatrix} . \end{aligned}$$

The parameters of the ellipse are then obtained through basic linear algebra:

$$\mathbf{x}_c = -\mathbf{A}^{-1} \mathbf{u} ,$$

and the axes are obtained through the eigenvalue/eigenvector analysis of matrix \mathbf{A} : the eigenvectors define the axes directions and the corresponding eigenvalues $\lambda_{1,2}$ are used to compute the ellipse axes lengths:

$$a_{1,2} = \sqrt{\left| \frac{f}{\lambda_{1,2}} \right|} ,$$

Note that the axis direction can be obtained by using the complex notation and Eq. 6.10, which provides the angle that the main ellipse axis forms with the x axis.

The main advantage of using an ellipse fitting instead of a principal component decomposition is that the shape is approximated by a geometrical figure that is smooth and has exactly two symmetry axes. Hence, the least square deviation from the fitted ellipse may be used as a symmetry or even regularity measure. To analyze the shape symmetry, only the border information is used. A transformation to an elliptical coordinate system can be performed:

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{R}(\mathbf{x} - \mathbf{x}_c) , \\ \mathbf{R} &= \begin{pmatrix} \sin(\alpha)/a_1 & \cos(\alpha)/a_1 \\ \cos(\alpha)/a_2 & -\sin(\alpha)/a_2 \end{pmatrix} , \end{aligned} \tag{6.14}$$

where a_1 and a_2 are the large and small ellipse axes respectively, α is the angle that a_1 forms with the x axis, and \mathbf{x}_c is the ellipse center.

For each border point the angle with respect to the axis $a_1 \geq a_2$ and the relative distance to the ellipse is computed. We obtain a one-dimensional periodical signal in the interval $[-\pi, \pi]$. This signal must be re-sampled since the object boundary samples are not equally spaced. The parameters a_1 and a_2 in Eq. 6.14 act as normalizing terms for the distance with respect to the ellipse size. A low-pass filtering should be applied to suppress small discontinuities and to reduce the influence of small variations in the segmentation result. The symmetry is evaluated by comparing the two sub-signals for the intervals $[-\pi, 0]$ and $[0, \pi]$ [6].

6.4.5 The segmentation problem

The *segmentation problem* consist of knowing if prior knowledge of the object location and boundary is needed or not, i.e. a segmented version of the image is necessary. Some approaches to the computation of symmetry axes do not require this information but since they are usually devoted to symmetrical objects, a symmetry quantification is not provided. If we consider the principal component technique applied to the image pixels that are part of the object of interest, it is difficult to establish what the shape, texture, or even color components in the computation are. In other words do we measure shape orientation or the actual orientation of a pixel distribution? In the frequency formulation the problem is even more complex since we use a Gaussian window. Objects must be centered and be of small size. A translation affects only the signal phase in the frequency domain, but the window is always centered. Finally an affine transform applied to the objects affects directly the axis computation.

The symmetry measure must be computed from the pixels which constitute the object of interest. If all the image pixels are considered, the system will detect the image frame symmetry axes. In addition, the same object at different scales would give different symmetry values. In fact, the larger the object surface, the less influence the border precision has on the fitting function. Only pixels within the object should be used while the their reflection about the axis may be anywhere within the image frame. Pixels reflected outside the image are considered to have a null symmetric value.

In the following section a more rigorous approach to the computation of symmetry axes based on the optimization of a given symmetry measure will be introduced.

6.5 Symmetry optimization

6.5.1 Symmetry measure and parameter space

The proposed approach consists of comparing the original and reflected images in terms of *visual difference*. The problem of quantifying the difference between two images is non trivial because the limit between similar and different objects is subjective. In addition, the symmetry quantification should *ideally* be a uniform measure, which means that a uniform change in the visual similarity should result in a uniform change in the symmetry value. This notion of uniformity has previously been introduced for color spaces [23] and for video sequence coding [24]. The correspondence between the latter and symmetry measure is immediate, since measuring the distortion between an image and the coded/decoded version is a similarity measure. Hence these methods may be used to evaluate the degree of symmetry. This should underline the non-deterministic aspect of measuring symmetry in almost-symmetrical and asymmetrical images. It is difficult to state what

a *good* symmetry axis and a *good* symmetry measure are, and there is no unique way to compute them.

If we suppose that any object is the combination of a symmetric object and an asymmetric object, and that the latter is considered as a noise component, then the *Peak Signal to Noise Ratio* (PSNR) can be used as symmetry measure:

$$\text{PSNR} = 10 \log_{10} \left(\frac{(N_q - 1)^2}{\text{MSE}} \right) ,$$

where N_q is the number of quantization levels in the image and MSE the mean square error between the original image and its reflected version. The latter is given by:

$$\text{MSE} = E[\| \mathbf{v}(\mathbf{x}) - \mathbf{v}(\tilde{\mathbf{x}}) \|^2] ,$$

where $\mathbf{v} : \mathbb{D}_{\mathbf{v}} \mapsto \mathbb{R}^n$ is the vector-valued input image, n the vector dimension, $\mathbb{D}_{\mathbf{v}} \subset \mathbb{Z}^2$ is the object of interest domain, $\mathbf{x} \in \mathbb{D}_{\mathbf{v}}$ is the pixel coordinate in the original image and $\tilde{\mathbf{x}} \in \mathbb{R}^2$ is the symmetric coordinate.

The PSNR is used to define the following symmetry coefficient:

$$\psi = \psi(\varphi, \rho, \mathbf{v}) = 1 - \frac{1}{1 + \text{PSNR}} , \quad (6.15)$$

where φ and ρ are two axis parameters. This measure does not depend on the image mean value but is sensitive to scaling. Using PSNR is not an optimal way to quantify the *visual difference* between two images but this kind of measure is easy to implement and reduces the processing time. In the following text, the symmetry measure will be called *fitting function*.

Before discussing the fitting function optimization, one has to define the two parameters needed to describe the axis line. The usual way of describing a line with a mathematical equation is $y = ax + b$, where a is the slope and b an offset in the y direction.

In order to obtain finite values and uniform scales, the two following parameters are defined:

$$\begin{aligned} \varphi &= \arctan(a) , \\ \rho &= -b \cos(\varphi) , \end{aligned}$$

where $\varphi \in [0; \pi)$ is the angle with the x axis (clock-wise) and ρ is a radial offset, i.e. the normal distance from origin to the axis. The x axis is assumed to be left-right oriented and the y axis top-bottom oriented. The line equation becomes

$$y = \tan(\varphi)x - \frac{\rho}{\cos(\varphi)} ,$$

when $\varphi \neq \pi/2$ and

$$x = \rho ,$$

when $\varphi = \pi/2$. A uniform variation of parameters φ and ρ correspond respectively to a uniform rotation and a uniform lateral shift of the axis.

A binary mask of the object is obtained by segmentation. The image origin is set to the center of gravity (c_x, c_y) of this mask. The angle φ_0 of its principal component [7] and the largest distance ρ_{\max} from the center to the object border are used for normalization. Finally the two following parameters are obtained:

$$\begin{aligned} \hat{\varphi} &= \varphi - \varphi_0 , \\ \hat{\rho} &= \rho / \rho_{\max} , \end{aligned}$$

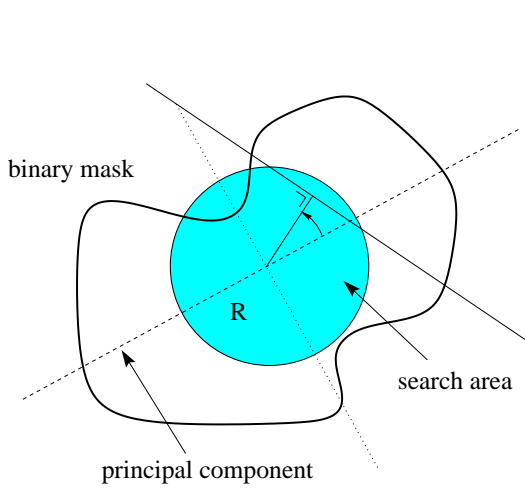


Figure 6.6: Axis parameters used to construct a symmetry map: angle and offset.

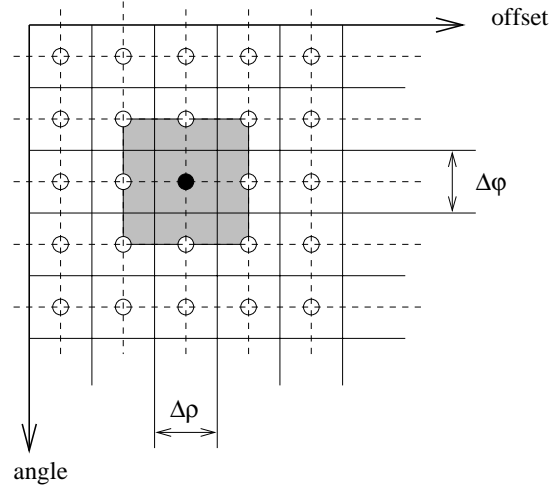


Figure 6.7: Organization of the sampling grid used to compute a symmetry map.

The use of $\tilde{\varphi}$ and $\tilde{\rho}$, as well as taking (c_x, c_y) as origin, renders the processing insensitive to scale change, shifting, or rotation of the object of interest. The different parameters are illustrated in Fig. 6.6: the angle and offset parameters are normalized so that any affine transform applied to the object of interest do not modify the symmetry map. The principal component of the binary mask is used as a reference axis for angles, the center of mass is the origin, and the area of interest has a radius equal to half the distance to the largest distance to the border.

6.5.2 Color and texture descriptors

Since there is no proof that symmetry in shape, color, and texture are strongly correlated, especially in almost-symmetrical and asymmetrical objects, three symmetry parameters should be computed using the corresponding information. Features must be used to describe local color and structure. The goal here is not to find how to describe texture and color information, but how to use this information for symmetry measurements.

Starting with *RGB* color images, one has to compute the frames which will be used for extracting color and texture information. The latter may be simply the combination of the three color input components. From a spectral point of view, however, these components also contain luminance information. It is clear that this component strongly influences our color perception. The following approach has been used in this study:

- use luminance component to compute the texture descriptors,
- use three filtered color components for color description.

To extract these components the uniform color spaces defined in Chapter 3 are used. The color distance formula is then directly embedded which gives more sense to the use of the MSE as a similarity measure.

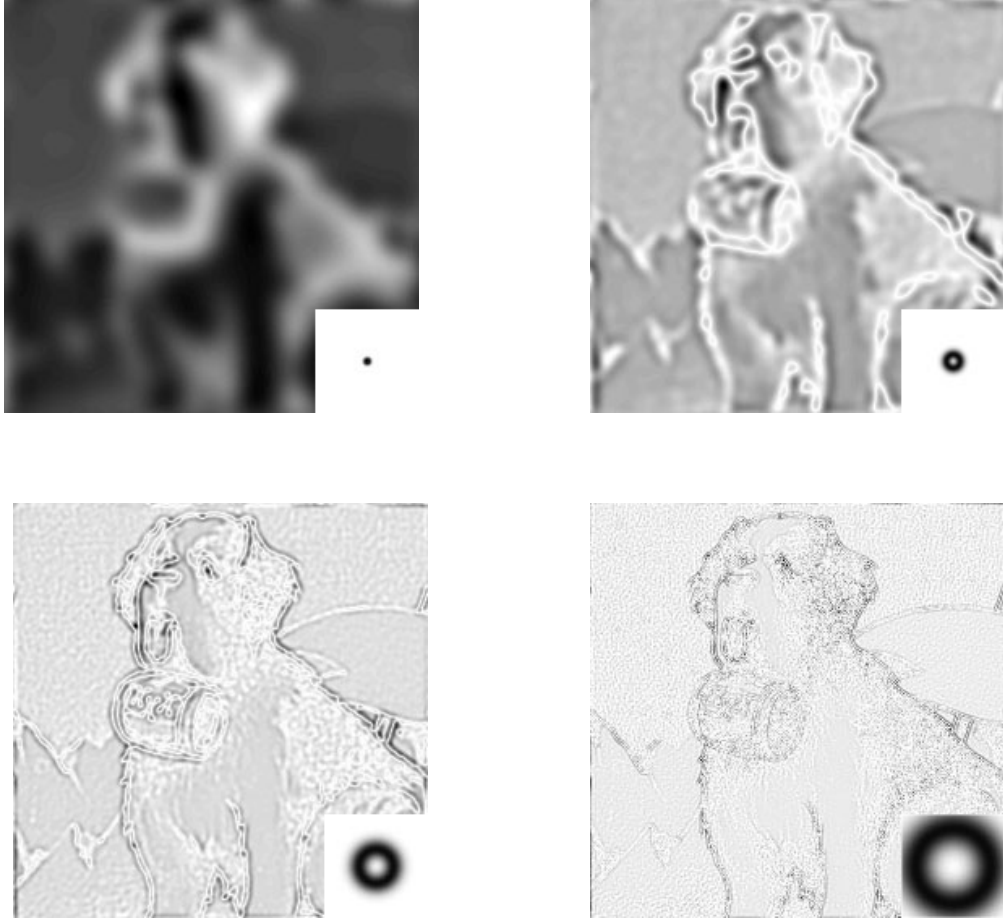


Figure 6.8: Filter bank used to extract texture descriptors.

The use of Gabor filters for the extraction of texture features has shown to be very powerful in terms of texture discrimination [25]. However this approach is not suited in this case, since the filters are directional. Therefore, symmetrical features must have symmetrical directions, which depends on the symmetry axis orientation. Isotropic filters are needed to avoid this dependency. In this study a filter bank built of isotropic Gaussian filters has been used. The equations are derived from the definition of Gabor filters [9]:

$$\begin{aligned}
 G_i(\omega_r) &= \exp\left(-\frac{(\omega_r - \omega_{r_i}^0)^2}{2\sigma_{r_i}^2}\right), \\
 \omega_{r_i}^0 &= \omega_{\min} + \sigma\{1 + 3(2^{i-1} - 1)\}, \\
 \sigma &= (\omega_{\max} - \omega_{\min})/2(2^M - 1), \\
 \sigma_{r_i} &= \sigma 2^{i-1},
 \end{aligned}$$

with M the number of filters, ω_{\max} and ω_{\min} the frequency limits and $1 \leq i \leq M$. ω_r is the radial spatial frequency. Up to now they have shown to be one of the best method for texture

description. Figure 6.8 gives an example of this sub-band decomposition used to describe texture information.

In the two following paragraphs two approaches to the optimization problem are introduced, the first one using a genetic algorithm and the second one self-organizing maps. While these two approaches are fast methods, they only give the global maximum of the symmetry measure. Section 6.6 will introduce methods to obtain all local maxima.

6.5.3 Genetic algorithm approach

Methods to compute the best fitting symmetry axis corresponding to the global maximum of a given symmetry measure will be presented first. In this section a *genetic algorithm* (GA) is used as a heuristic search method.

A GA is an active search method that uses operators inspired by evolutionary processes. The parameters used for the optimization are seen as *alleles* and are part of a *chromosome*. The goal of this technique is to find the best chromosome, i.e. the set of parameters that maximize a given function. The “quality” of the chromosome should improve through the different generations. A new generation is obtained from the current set of chromosomes, all encoding a different combination of parameters, using different operators, typically crossover and mutation. The reproduction stage usually mixes part of two randomly chosen chromosomes. The complete scheme used in this work is shown in Fig. 6.9 for binary valued chromosomes. The extension to real valued chromosomes is discussed in the following paragraph.

Most of the developments in GA deal with binary encoded chromosomes. This means that the different parameters can only take a limited number of values and with a predefined precision. When using real parameters, one possibility is to sample the parameter space and to define the chromosome length according to the required precision [26] (the variables are “encoded”). The direct encoding of real parameters has been also investigated, leading for example to *blend crossover* (BLX- α), which uniformly picks values that lie on a line passing through two parents [27]. *Simulated binary crossover* (SBX) was introduced in [28, 29]. Its search power is similar to that of single-point crossover used for binary GAs, where the search power is defined in terms of the probability to create an arbitrary child from two parents. In this study a modified version of this technique is used.

Crossover is used for producing offsprings from couples of parent chromosomes. In SBX the following spread factor is first defined:

$$\beta = \left\| \frac{c_1 - c_2}{p_1 - p_2} \right\| ,$$

$$P(\beta) = \begin{cases} 0.5(n+1)\beta^n & \text{if } \beta \leq 1, \\ 0.5(n+1)/\beta^{n+2} & \text{otherwise,} \end{cases}$$

where p_1, p_2 are the two parent’s real variable values and c_1, c_2 the produced offsprings. The spread factor is randomly chosen according to the above distribution (see [28] for the detailed procedure) and used to compute c_1 and c_2 :

$$\begin{aligned} c_1 &= 0.5[(p_1 + p_2) + \beta(p_1 - p_2)] , \\ c_2 &= 0.5[(p_1 + p_2) - \beta(p_1 - p_2)] . \end{aligned} \tag{6.16}$$

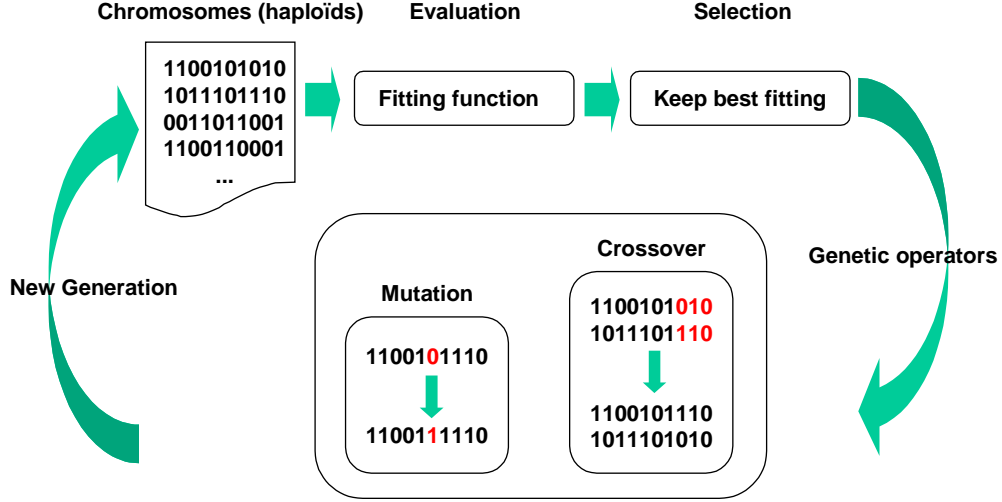


Figure 6.9: Genetic algorithm scheme used in this study.

Let us introduce a new probability distribution for the spread factor β which is the combination between an *attraction* term and a *rejection* term:

$$P(\beta) = \frac{1}{\sqrt{2\pi}(\sigma_a - \sigma_r)} \left\{ \exp\left(-\frac{(\beta - 1)^2}{2\sigma_a^2}\right) - \exp\left(-\frac{(\beta - 1)^2}{2\sigma_r^2}\right) \right\}, \quad (6.17)$$

with $\sigma_r \ll \sigma_a$. The shape of this distribution is close to that obtained with the SBX and has the advantage of producing children different from their parents, which is due to the rejection term in Eq. 6.17. When using the SBX, the probability that the children are exactly equal to the parent is the highest. However, if one assumes that the best fitting parents are kept for the next generation, it is not desirable to duplicate them, because there is then a redundancy in the search process. By keeping the best fitting chromosomes one ensures the convergence to the optimal solution. The normalizing term is not useful for a numerical implementation, which is very easy when using normal distributions [30]. The distribution may also be bounded without modifying the shape (we must have $\beta \in [0; \infty)$). Figure 6.10 plots the original and modified distributions.

The GA works independently for the two axis parameters and is initialized with random values, all located in the search area shown in Fig. 6.6. For every pair of parameters, the image reflection about the corresponding image and the MSE are computed. In order to speed the computation, a different subset of \mathbb{X} (input data) is used for every generation. The consequence is that the less the object is symmetrical, the higher is the introduced uncertainty. Depending on the subset an axis may show a better symmetry than the true symmetry axis. This effect is especially strong

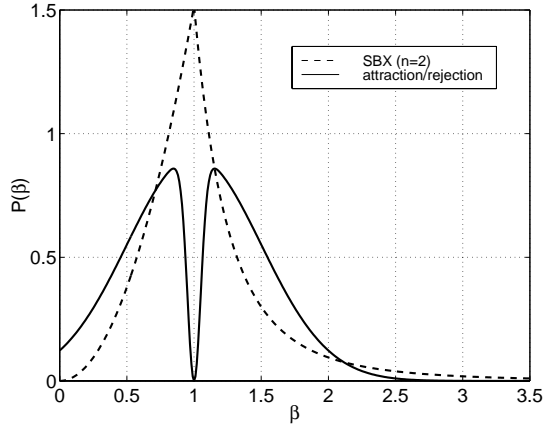


Figure 6.10: SBX distribution and attraction/rejection Gaussian distribution.

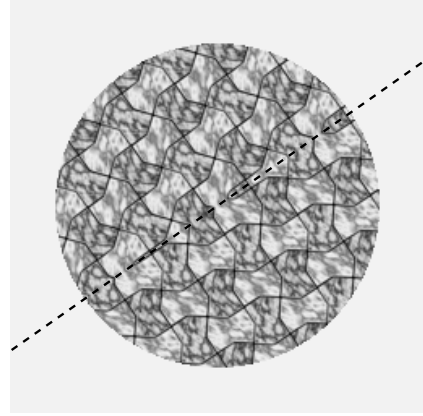


Figure 6.11: Image used to illustrate the different computation methods. The object contour has been chosen circular in order to evaluate the texture descriptor.

when using active search methods like the GA, where a well fitting chromosome may be suppressed when the next generation is created. However this simplification is necessary in order to speed up the processing. The test image shown in Fig. 6.11 has been used to illustrate the GA technique. Since the object of interest has a circular shape, only structural information is exploited using the texture descriptors given in § 6.5.2. Since the symmetry axis in this example is quite evident, the robustness of the technique has been evaluated through its ability to reproduce the result. Figure 6.12 plots the results obtained by running 50 times the GA on the test image. The range of values obtained do not correspond to a visually perceptible change in the axis position.

6.5.4 Self-organizing maps derived approach

Using *self-organizing maps* (SOM) is a way of mapping high-dimensional signal spaces on a two-dimensional array. This projection gives an ordered two-dimensional approximation of an n -dimensional distribution. The two-dimensional map is an array of nodes to which an n -dimensional vector is associated. Organizing this map is an iterative procedure based on *competition* between the different nodes: at each iteration an input vector is randomly chosen and associated with the best fitting node (the reference node) using a specific metric. The vectors associated with the different nodes are updated using the following relationship [31]:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] , \quad (6.18)$$

where $t = 0, 1, 2, \dots$ is the iteration number (time variable), $\mathbf{x}(t)$ is the randomly chosen input vector and $h_{ci}(t)$ is called *neighborhood function*. Subscript i indicates the node number and c the reference node number.

The neighborhood function weights the update term $\mathbf{x}(t) - \mathbf{m}_i(t)$ according to the distance of node i to the reference node c within the map. Different functions may be defined under the

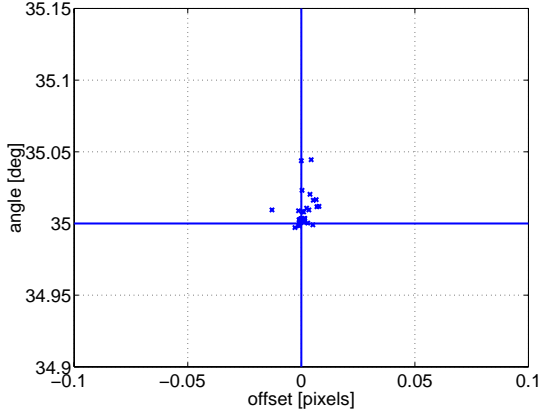


Figure 6.12: The texture-based symmetry axis has been computed 50 times using the GA for the test image shown in Fig. 6.11.

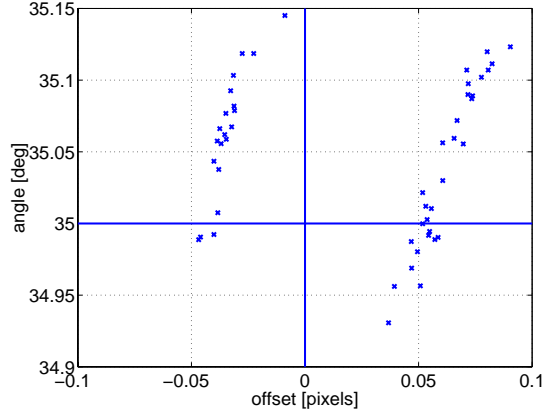


Figure 6.13: The same experiment as in Fig. 6.12 has been repeated for the SOM-based technique.

condition that $h_{ci} \rightarrow 0$ with increasing $\|\mathbf{r}_c - \mathbf{r}_i\|$, where $\mathbf{r}_i \in \mathbb{R}^2$ is the position of node i . It is also necessary for convergence that $\lim_{t \rightarrow \infty} h_{ci}(t) = 0$, i.e. the vector update decreases in time.

Suppose now that the n -dimensional feature space becomes the parameter space (in our case $n = 2$). A randomly chosen set of input vectors is chosen and the fitting function approximated. The closest vector in the map is the reference vector. The nodes are updated using a modified version of Eq. 6.18:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + f(\psi)h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] .$$

$f(\psi)$ is a function of ψ (for example $f(\psi) = \psi$) introduced in order to weight the update according to the fitting function value. The result of this approach is, in our case, an approximation of the SM defined in the next section.

Since we are seeking a unique set of parameters, that is the axis parameters which maximize the fitting function, an alternative approach is to ensure the convergence of all the nodes to a unique solution. This “moving map” allows for searching actively the parameter space with the advantage of improving the search precision when the nodes are close to the optimal parameter vector. In order to achieve this result, the fitting function is evaluated for every vector in the map. The best fitting vector becomes the reference vector.

The experiment performed for the test image of Fig. 6.11 for the GA approach has been repeated with the SOM technique. The result is shown in Fig. 6.13. It can be compared with the one obtained for the GA approach and displayed in Fig. 6.12. The precision is lower but the dispersion is still small enough to be imperceptible. Finally, the SOM approach has the advantage of being much faster than the GA approach.

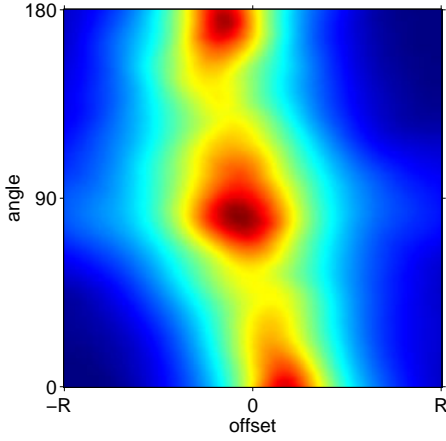


Figure 6.14: Symmetry map obtained by direct computation.

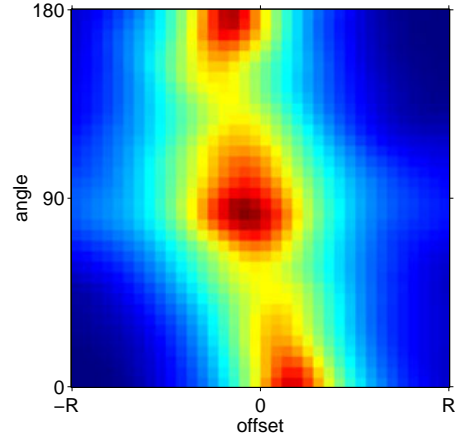


Figure 6.15: The same symmetry map than that shown in Fig. 6.14 has been recomputed with a larger sampling step.

6.6 Symmetry map computation

6.6.1 Direct computation

The next step of the procedure is the computation of the fitting function ψ for $0 \leq \tilde{\varphi} \leq \pi$ and $-0.5 \leq \tilde{\rho} \leq 0.5$. The range of values chosen for $\tilde{\rho}$ has shown to be large enough. The result is a *symmetry map* (SM) where all maxima correspond to local optima of the fitting function and where the global maximum corresponds to the best fitting symmetry axis (i.e. the axis that maximizes the fitting function, as previously defined). For a SM of size $m \times n$, $\psi(i, j) = \psi(\varphi, \rho, \mathbf{v})$ with:

$$\varphi = i \frac{\pi}{m} + \varphi_0, \quad (6.19)$$

$$\rho = (j \frac{1}{n-1} - 0.5) \rho_{\max}, \quad (6.20)$$

and $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$.

Figure 6.14 shows a typical result of symmetry map². For every set of parameters (*offset*, *angle*) the image reflection and symmetry value are computed. The processing time, however, is unacceptable.

In order to reduce the processing time, the sampling precision in the parameter space can be reduced. Figure 6.15 shows the result obtained reducing the precision by a factor 5. The full resolution is obtained by interpolation. The effect is a small shift of the maxima location and therefore the procedure requires the use of local optimization methods [30] (steepest descent, for example) to obtain the exact maxima location. Figure 6.16 shows the result obtained by using a reduced subset (5 %) of input samples. The fitting function is only approximated and the result is a noisy SM. A (very) low-pass filter can be used to smooth out the map, as shown in Fig. 6.17,

²The same object has been used to illustrate the different techniques presented in this section in order to facilitate the comparison.

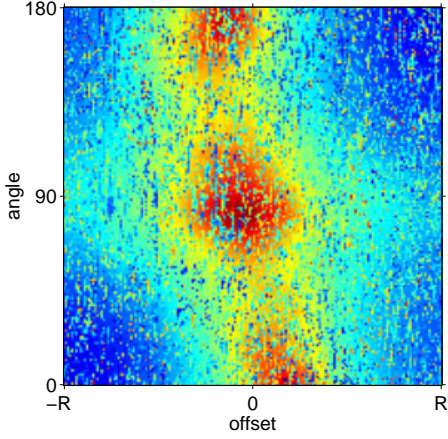


Figure 6.16: Symmetry map computed with a randomly chosen subset of image pixels.

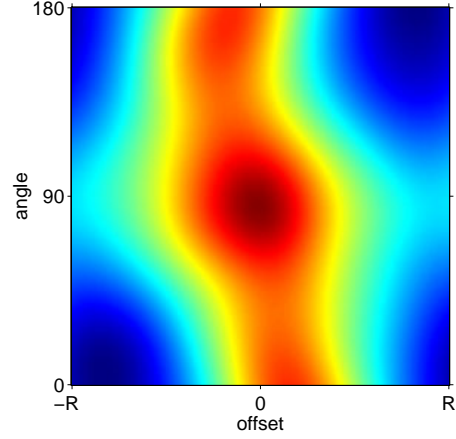


Figure 6.17: The symmetry map shown in Fig. 6.16 has been smoothed out with a Gaussian low-pass filter.

but it introduces a strong border effect. The result is a strong shift of the maxima location. Hence more elaborated methods are needed to construct a SM.

In the next sections two multi-resolution (MR) approaches for computing a SM are presented. From this map, all the local maxima (including the global maximum) can be found precisely using simple gradient information and local optimization methods [30].

6.6.2 Multi-resolution approach

The first multi-resolution approach proposed in this section uses successive refinement of the parameter space in order to obtain the final SM. For a SM of size $m \times n$ the fitting function is approximated by:

$$\psi(i, j) \cong \begin{cases} \sum_{\mathbf{x} \in \mathbb{S}} \psi_{\mathbf{x}}(i + \hat{i}, j + \hat{j}) & \text{if } i \bmod 2^r \equiv 0 \text{ and } j \bmod 2^r \equiv 0, \\ \psi(i - (i \bmod 2^r), j - (j \bmod 2^r)) & \text{otherwise,} \end{cases}$$

where r is the resolution (resolution zero is full resolution), $0 \leq i \leq m - 1$ and $0 \leq j \leq n - 1$. In order to reduce the computation time, only a subset \mathbb{S} of size N of the image samples is used ($\mathbf{x} \in \mathbb{S} \subset \mathbb{D}_{\mathbf{v}}$, where $\mathbb{D}_{\mathbf{v}}$ is the object of interest domain). The two coordinates \hat{i} and \hat{j} are two random numbers having a normal distribution $N(\mu = (2^r - 1)/2, \sigma^2 = \mu^2)$. The introduction of \hat{i} and \hat{j} is intended to give at every sampling location a local value instead of a punctual value. Figure 6.7 shows how the sampling grid is organized.

φ and ρ are obtained using Eqs. 6.19 and 6.20 respectively. This approach yields a smooth SM where each sample gives the local value. After each iteration, only regions having a value higher than a given threshold are kept for further processing.

The second multi-resolution approach presented in the next section uses the true fitting function computed from the input image taken at multiple resolutions.

6.6.3 Pyramidal approach

A pyramidal representation of the input image is used [32, 33]: the image size is successively reduced by a factor two in order to generate a down-sampled version of the previous image. At resolution r (the original image has $r = 0$) the down-sampled image is given by:

$$\mathbf{v}^{(r)}(\mathbf{x}) = \mathbf{v}^{(0)}(2^r(\mathbf{x} + \mathbf{1}) - \mathbf{1}) ,$$

where the image origin is supposed to be $(0, 0)$.

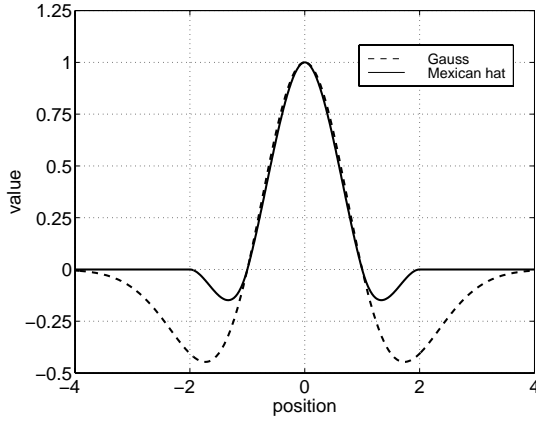


Figure 6.18: Mexican hat filter obtained through a polynomial function and the second derivative of Gaussian (negative).

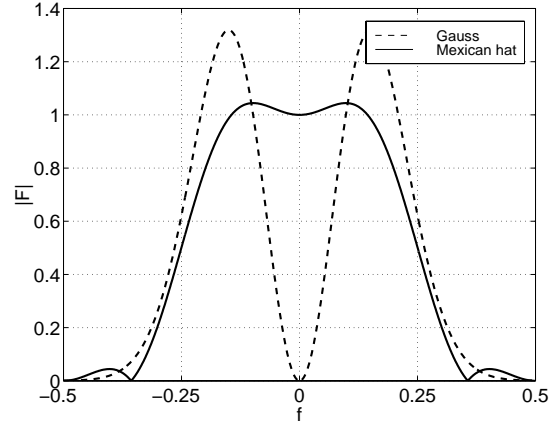


Figure 6.19: Transfer function of the Mexican hat filter and the second derivative of Gaussian.

In order to avoid aliasing a *bicubic interpolation kernel* is used, called *Mexican hat*. The 1-D finite length impulse response is given by [34]:

$$h_3(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 \leq |x| < 1, \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 \leq |x| < 2, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\int_{-\infty}^{\infty} h_3(x) dx = 1 .$$

This filter has an impulse response close to the second derivative of a Gaussian, as can be seen in Fig. 6.18. While these two filters are very similar, the Mexican hat has the advantage of being of finite length and to have a transfer function of low-pass type, as shown in Fig. 6.19. The second derivative of Gaussian is band-pass shaped with a zero mean value (or zero frequency).

The SM is initially computed using a small version of the image, in our case $1/16$ of the initial size (512×512). The obtained SM is a kind of low-pass version of the true SM. Portions of the SM where the fitting function gives results above a given threshold are used to refine the computation. The procedure is repeated until the initial image size has been reached. Figure 6.22 shows the result. The effect of using down-sampled versions of the original image is a blurring

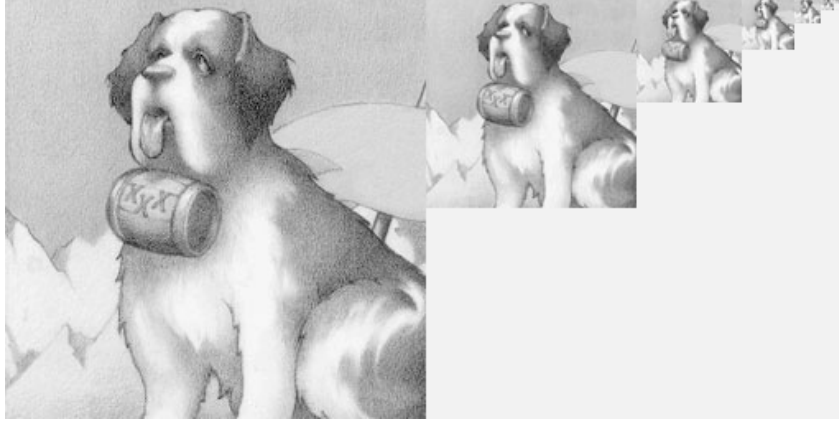


Figure 6.20: Pyramidal decomposition: at every level the previous image is down-sampled by a factor 2.

of the SM which in turn depends on the resolution. At full resolution, the different maxima are precisely located. This result can be compared with that shown in Fig. 6.21 and obtained for the multi-resolution approach. This technique provides a noisy symmetry map due to the random selection of pixels used for the computation. Even if the different maxima are well-defined, it introduces a lot of spurious maxima. Finally, the pyramidal approach shows to be best suited for the computation of symmetry maps, because the regions where local maxima occur the full image resolution is used.

6.6.4 Noise component

This section gives the analytical value of the expected MSE between a *noisy* image and its reflected version. The MSE is given by:

$$E[(v + n - \tilde{v} - \tilde{n})^2] = E[v^2 + n^2 + \tilde{v}^2 + \tilde{n}^2 + 2vn - 2v\tilde{v} - 2v\tilde{n} - 2\tilde{v}n - 2n\tilde{n} + 2\tilde{v}\tilde{n}] , \quad (6.21)$$

where $v(x, y)$ is the image intensity function and $n(x, y)$ the added noise. The tilde denotes the symmetrical value. While $E[v]$ and $E[n]$ remain constant, the values of $E[\tilde{v}]$ and $E[\tilde{n}]$ vary with the symmetry axis. However, for large objects (the pixel population is large), $E[n] = \mu \cong E[\tilde{n}]$ because the statistical distributions of both signals are very close. If the noise has no spatial correlation and is uncorrelated with the image, we can write:

$$\begin{aligned} E[n\tilde{n}] &= E[n]E[\tilde{n}] \cong \mu^2 , \\ E[vn] &= E[v]E[n] = E[v]\mu , \\ E[\tilde{v}\tilde{n}] &= E[\tilde{v}]E[\tilde{n}] \cong E[\tilde{v}]\mu . \end{aligned}$$

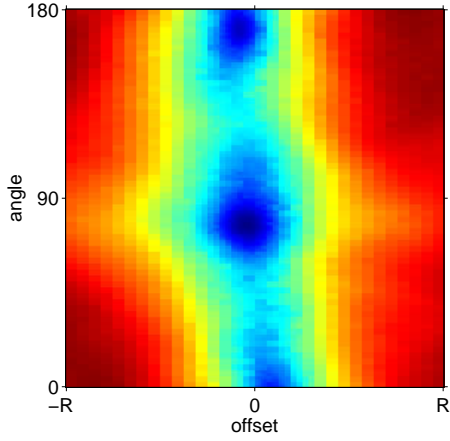


Figure 6.21: Symmetry map obtained using a multi-scale approach.

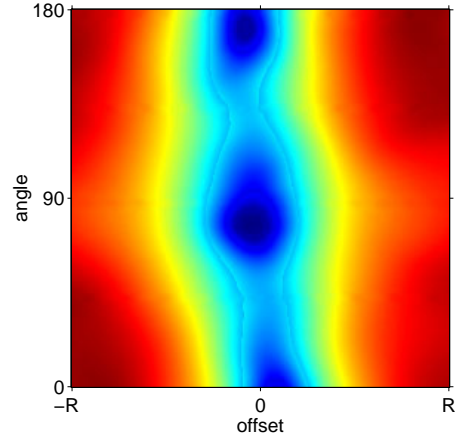


Figure 6.22: Symmetry map obtained using an image pyramid.

Equation 6.21 becomes:

$$\begin{aligned}
 E[(v + n - \tilde{v} - \tilde{n})^2] &\cong E[(v - \tilde{v})^2] + E[(n - \tilde{n})^2] \\
 &\cong E[(v - \tilde{v})^2] + 2E[n^2] - 2\mu^2 \\
 &= E[(v - \tilde{v})^2] + 2\sigma^2.
 \end{aligned}$$

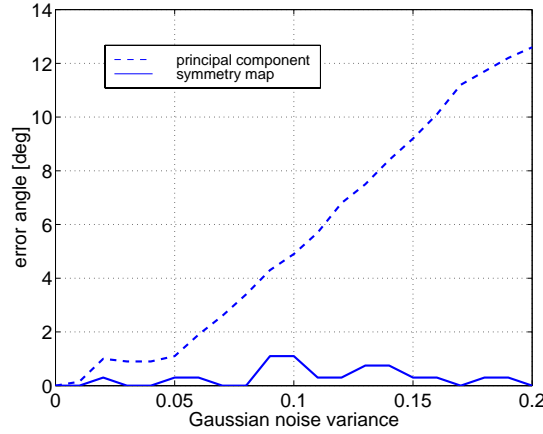


Figure 6.23: Angular deviation from the true symmetry axis obtained for increasing values of the Gaussian noise variance, computed for the test image of Fig. 6.11.

The added noise affects *globally* the SM, which means that existing maxima do not move and new maxima do not appear. This result can be easily extended to vector-valued images. Figure 6.23 plots the angular deviation from the true symmetry axis for increasing Gaussian white noise and for two approaches: the symmetry map and the principal component technique. It

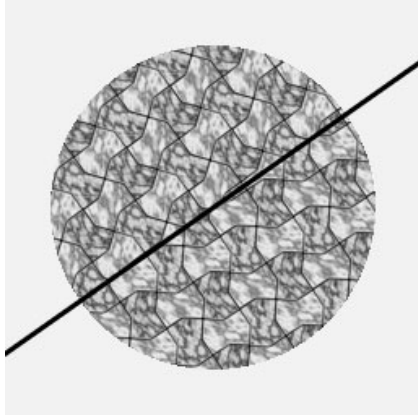


Figure 6.24: Symmetry axis obtained after SM computation using texture descriptors.

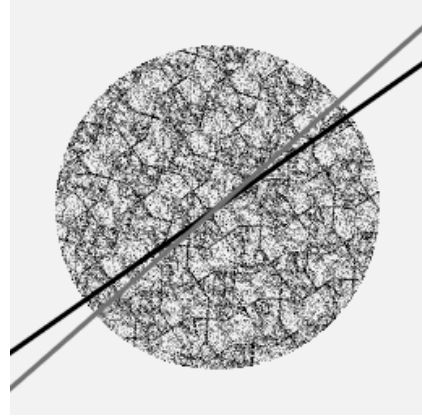


Figure 6.25: Symmetry axis obtained after SM computation (black) using texture descriptors and the image shown in Fig. 6.24 with additional noise. KL result is shown gray.

shows clearly that the former is much more robust to noise, even if small variations mainly due to the discrete nature of the image and SM sampling grid are possible. Figures 6.24 and 6.25 show the result obtained for the test image of Fig. 6.11.

6.7 Summary

In this chapter the problem of finding the highest symmetry of objects in digital images based on shape, color, and texture information has been investigated. The different techniques introduced in this chapter take vector-valued data as input, so that color and texture descriptors can be used as well.

Two approaches have been proposed first to search the axis parameter space (angle and offset) for the global maximum. A fitting function or symmetry measure has been used to evaluate the degree of symmetry for every pair of parameters. A genetic algorithm for real valued chromosomes and a scheme derived from the self-organizing maps theory have been designed for this purpose. They allow the quick computation of the axis that maximizes the symmetry (MSE between original and reflected object in our case).

Symmetry maps (SM) have been defined. They contain the symmetry function for a bounded part of the axis parameter space. Two methods have been given to compute this map with a reasonable processing time, namely a multi-resolution and a pyramidal approach. In the former case the parameter space is sampled with increasing resolution in regions where the fitting function gives high values. The latter approach uses a pyramidal image representation and the image resolution is progressively increased in high symmetry regions of the parameter space. These approaches require a longer processing time but have the advantage to provide a signature that may be used efficiently for classification and recognition tasks.

The use of multi-dimensional images with local color and texture information renders these tools very powerful. The applications are various:

- object recognition,
- database retrieval,
- medical diagnosis.

The SMs are especially interesting because they provide a signature that may be compared with that of other objects and lead to a classification. The analysis of SMs is a way of analyzing an object in a particular space designed for a unique criteria. The processing has some analogy with the Hough transform, except that the different pairs of parameters are not used to count image pixels but to evaluate a fitting function.

Now that the different processing step of the image analysis scheme have been investigated, their application to *digital dermatoscopy* will be presented in the following chapter.

Bibliography

- [1] L. Van Gool, T. Moons, D. Ungureanu, and E. Pauwels. Symmetry from shape and shape from symmetry. *International Journal of Robotics Research*, 15(1):407–424, February 1995.
- [2] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1154–1166, December 1995.
- [3] G. Marola. Using symmetry for detecting and locating objects in a picture. *Computer Vision, Graphics, and Image Processing*, 46:179–195, 1989.
- [4] S. K. Parui and D. Dutta Majumder. Symmetry analysis by computer. *Pattern Recognition*, 16(1):63–67, 1983.
- [5] C. W. Tyler, editor. *Human Symmetry Perception and its Computational Analysis*. VSP, Utrecht, The Netherlands, 1996.
- [6] Ph. Schmid. Symmetry measure for the computer aided diagnosis of pigmented skin lesions. In *Proceedings of the 11th International Symposium in Computer Assisted Radiology and Surgery*, page 998, Berlin, Germany, 25–28 June 1997.
- [7] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [8] J. Bigün. Multidimensional orientation estimation with application to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991.
- [9] J. Bigün. N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):80–87, January 1994.
- [10] G. Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):104–108, January 1989.
- [11] T. Masuda, K. Yamamoto, and H. Yamada. Detection of partial symmetry using correlation with rotated-reflected images. *Pattern Recognition*, 26(8):1245–1253, 1993.
- [12] J.-J. Leou and W.-H. Tsai. Automatic rotational symmetry determination for shape analysis. *Pattern Recognition*, 20(6):571–582, 1987.

- [13] T. Zielke. Intensity and edge based symmetry detection with application to car following. *CVGIP: Image Understanding*, 58(1):177–190, July 1993.
- [14] Ch. Sun. Symmetry detection using gradient information. *Pattern Recognition Letters*, 16:987–996, 1995.
- [15] L. S. Davis. Understanding shape: II. symmetry. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(3):204–212, March 1977.
- [16] S. A. Friedberg. Finding axes of skewed symmetry. *Computer Vision, Graphics, and Image Processing*, 34:138–155, 1986.
- [17] M. R. Spiegel. *Theoretical Mechanics*. Mc Graw-Hill Book Company, New-York, 1967.
- [18] J. Bigün. Frequency and orientation sensitive texture measures using linear symmetry. *Signal Processing*, 29:1–16, 1992.
- [19] J. Bigün. Optimal orientation detection of linear symmetry. In *Proc. First Int. Conf. Comput. Vision*, pages 433–443, London, England, June 1987.
- [20] W. K. Pratt. *Digital Image Processing*. Wiley-Interscience, 2nd edition, 1991.
- [21] F. L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.
- [22] M. Pilu, A. W. Fitzgibbon, and R. B. Fisher. Ellipse-specific direct least-square fitting. In *Proceedings of the International Conference on Image Processing (ICIP-96)*, volume 3, pages 599–602, Lausanne, 16–19 September 1996. The IEEE Signal Processing Society.
- [23] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. Wiley, New-York, 1982.
- [24] Ch. J. van den Branden Lambrecht and O. Verscheure. Perceptual quality measure using a spatio-temporal model of the human visual system. In *Proceedings of the SPIE*, volume 2668, pages 450–461, San Jose, CA, January 28–February 2 1996.
- [25] M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.
- [26] A. H. Wright. Genetic algorithms for real parameter optimization. In G. J. E. Rawlins, editor, *Foundations of genetic algorithms*, pages 205–218, San Francisco, California, 1991. Morgan Kaufmann Publishers.
- [27] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval-schemata. In L. D. Whitley, editor, *Foundations of genetic algorithms 2*, pages 187–202, Vail, Colorado, 1993. Morgan Kaufmann Publishers.
- [28] K. Deb and A. Kumar. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. *Complex Systems*, 9(6):431–454, December 1995.
- [29] K. Deb and A. Kumar. Simulated binary crossover for continuous search space. *Complex Systems*, 9(1):115–148, February 1995.

- [30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, second edition, 1994.
- [31] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, 1995.
- [32] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, COM-31(4):532–540, April 1983.
- [33] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51, 1981.
- [34] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman & Hall Computing, London, UK, first edition, 1993.

Chapter 7

Digital dermatoscopy

7.1 Introduction

In this chapter the different techniques developed in Chapters 2 to 6 will be applied to images of pigmented skin lesions. This application is motivated by the need of *computer aided diagnosis* (CAD) systems to improve the accuracy of malignant melanoma diagnosis. The main goal of such a system is to help the physician in the different processing steps, from the lesion detection to the classification into different types of lesions. The quantification of diagnostic features is an essential and difficult step. A diagnosis scheme is the combination of several parameters whose relevancy has been established by physicians. These results, however, are based on a limited number of cases and have been obtained through subjective evaluation. It is therefore very difficult to know what is the exact contribution of every diagnostic feature to the final diagnosis.

In most of the published work the way diagnostic features are quantified has been neglected. Researchers and especially physicians tried to extract as many features as possible, without caring about the real signification of what they wanted to extract and what they were actually extracting. However, this is a way to quickly propose a complete diagnosis scheme with statistical classification. In this thesis the goal is to solve the lesion detection problem and to focus on a single feature, namely symmetry. It is indeed necessary to make a full analysis of the different features before classification schemes can be established.

The motivations, imaging techniques, diagnosis schemes, and principles of computer aided diagnosis will be presented in Section 7.2. An efficient scheme to remove the hair that may corrupt the image processing results is introduced in Section 7.3. The segmentation and contour detection techniques are validated for their use on images of pigmented skin lesions in Section 7.4. A simple experiment is presented, which shows that contours drawn manually by physicians, even when trained, can not be used as an absolute reference for the validation. However, we can show that the different techniques, when compared to a “group” of experts, give results that are globally close to those obtained by the dermatologists. In Section 7.5 the symmetry quantification is applied to two groups of respectively 50 benign and 50 malignant lesions of melanocytic type. A 6-D symmetry feature vector is extracted from 3 symmetry maps, obtained from shape, color, and texture descriptors (see Chapter 6). A simple linear classification is used to evaluate this approach to symmetry quantification and to compare it to existing techniques.

7.2 Skin cancer diagnosis

7.2.1 Introduction

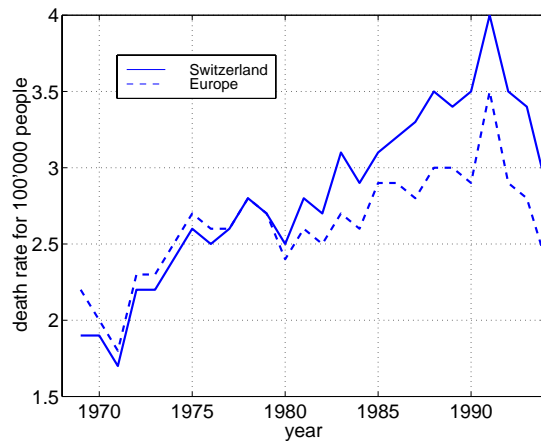


Figure 7.1: Death rate evolution for Switzerland and Europe from 1969 to 1994.

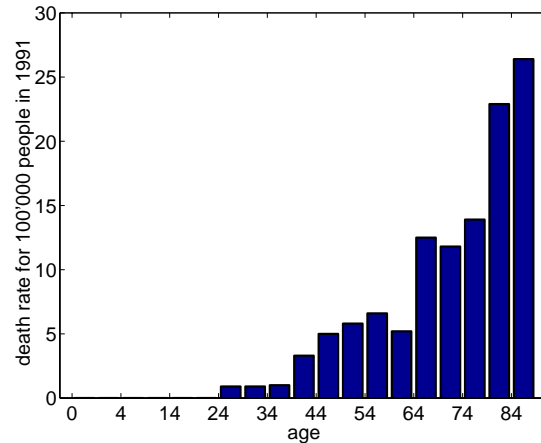


Figure 7.2: Death rate versus age for Switzerland in 1991.

Malignant melanoma (MM) is the third most frequent type of skin cancer and one of the most malignant tumors. It is therefore worrying that the number of new cases has increased much more than for other types of cancer. It is of 10-12 per 100'000 in Europe, 5 times more than 50 years ago, 10-20 per 100'000 in the USA, and 30-40 per 100'000 in Australia, where the climate and sun exposure are especially dangerous [1]. Figure 7.1 shows the death rate versus the year, from 1969 to 1994, for Switzerland and Europe [2]. It is very interesting to note that in both cases the evolution has the same shape, even if since about 1980 the death rate is higher in Switzerland than in Europe. The strong increase in the late 80's is certainly due to the fact that the style of life and the standard of living have changed fundamentally in rich countries since the early 70's. People started traveling a lot, especially in exotic places where the climate is very dangerous for white and Caucasian people, and without any appropriate protection. The small decrease beginning of 1990 may be due to the fact that people has been better informed about the danger of long sun exposure and the ways to protect themselves against sunburn. It will be interesting to see if this tendency will be confirmed in the next years. Figure 7.2 shows the death rate versus the age of people in 1991 [2]. It is visible from this bar graph that people older than 60 are most likely to develop skin cancer. In Europe, northern countries are more concerned than southern countries. The highest rates are in Denmark, Holland, Great-Britain, and Ireland. Countries like Greece, Italy, Spain, and Portugal have 5 to 6 times lower rates. Finally we can say that people born in 1930 have a probability of 1:1500 to develop skin cancer, it is 1:105 for people born in 1990, and it is estimated to be 1:75 for people who will be born in 2000 [1].

The most malignant type of skin cancer is the *malignant melanoma* (MM). It develops from pigmented cells and most of the time on the skin. Similarly to other types of tumors, the development of malignant melanoma is divided into different phases. In the initial phase there is an alteration of the DNA located in the cell nucleus. Among different causes UV light seems to

play an important role in the development of malignant melanoma. In a second phase, there is a proliferation of atypic cells which develop into a *melanoma in situ*, limited to the epidermis. Finally, the cancerous cells extend to the lower skin layers, get connected to the blood vessels and the cancer can spread to other organs. In the first stage the development is horizontal, while in the last stage it is vertical [1].

A malignant melanoma diagnosed at an early stage of its development can be cured without complications. This is not the case when the lesion has started to grow in depth until it reaches the lymphatic system and spreads to other regions of the body. The *early diagnosis of malignant melanoma* is therefore a crucial problem for dermatologists. Knowing that the rate of accurate diagnostics is about 75 % (true positives) [3, 4] it is important to develop efficient schemes for clinical diagnosis and to exploit them fully through computerized systems. The purpose is not to replace physicians but to allow a more robust approach when human judgment is very subjective. Especially *lesion detection* and *feature quantification* are important topics. Other problems such as database management, networking, storage [5], display, etc., are also of prime importance.

The different imaging techniques available for skin cancer diagnosis are presented in the following paragraph.

7.2.2 Imaging

The diagnosis of skin cancer is performed through direct observation with the help of a special magnifying glass called a *dermatoscope* [3]. It is capable of magnifications of 10 to 40 times and is lightly pressed against the skin. Illumination is done by a halogen lamp at an angle of incidence of 20 degrees¹. Slides can be obtained using a standard camera with a special lens or with a digital camera. The former gives images of higher resolution while the latter allows for a direct connection to the computer. Photographic slides must be developed and then digitized, which takes a lot of time. Since digital cameras with high resolution and with high quality color rendering are very expensive, most of the dermatologists work with photographic slides. It has also been shown that the information content (essential to the diagnosis) of digitized slides is either altered by the digitization process [6] or by data compression [7].

The magnified picture of a skin lesion is called *macroscopic* image [3]. It was the first type of images used for skin cancer diagnosis. A number of diagnostic features have been evaluated for macroscopic images and the results led to the development of diagnosis schemes [8]. However, the analysis of pigmented skin lesions through macroscopic images is limited because there is almost no structural and colorimetric information.

The use of *dermatoscopic* images gave a new dimension to skin cancer diagnosis. An oil immersion is used to render the skin translucent. The resulting image reveals all the pigmented structures, different color shades, depending on the pigment depth, and a fuzzy limit between the lesion and the healthy skin. This technique is called *epiluminescence microscopy* (ELM) [3]. Because the information content of dermatoscopic images is much more complex, the visual evaluation of diagnostic features has become a tricky problem. Especially for this type of images efficient image processing techniques must be developed to help physicians in their work.

Similarly to the macroscopic case diagnosis schemes have been established by expert dermatologists based on their experience. One of these schemes will be shortly described in § 7.2.3. Figures 7.3 and 7.4 show respectively a macroscopic and a dermatoscopic image of the same pig-

¹These data are those of the Heine Dermatoscope Delta 10 [3].



Figure 7.3: Macroscopic image of benign melanocytic skin lesion.

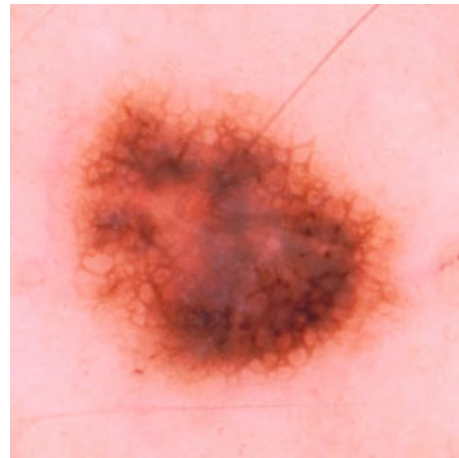


Figure 7.4: Dermatoscopic image of benign melanocytic skin lesion.

mented skin lesion. While the former shows well shaped limits but almost no color shades and no pigmented structures, the latter reveals patterns that will be important for the diagnosis of certain types of lesions.

Beside these two “standard” imaging techniques there are a number of other more complex ones that have been mainly designed to obtain an important parameter: the depth of invasion. Macroscopic and dermatoscopic images indeed give a planar view of the lesion, even if some of the structures underlying the skin surface are revealed by the latter technique. The lesion thickness is an important parameter that allows the physician to determine if there is a vertical growth of the lesion [9]. It is however difficult to obtain this information *in situ*, i.e. without excision of the lesion. A technique developed by Dhawan et al. is called *nevoscopy* [10, 11]. The lesion is transilluminated using fiber optics directed into the surrounding skin. Three images are then picked up by a digital camera at 90, 180 (glancing image), and 45 degrees. The last image is obtained by using two front-silvered mirrors held next to the lesion in a device called *nevoscope*. The images are finally used in a computed tomography algorithm to calculate approximate vertical cross sections of the lesion. The error from the real width (measured on a set of excised lesions) did not exceed 5 % [10]. Unfortunately the apparatus needed in nevoscopy is not handy and quite expensive. It seems that this technique has not been used by anybody else than the inventors, even if the results were very promising.

Two other imaging techniques are still under development: 3-D *ultrasound* devices [12] with sufficient resolution to be used in dermatology and *confocal microscopy* [13]. These techniques, however, need the use of heavy devices, are expensive, and are not yet precise enough for skin cancer diagnosis applications.

In this research we are concerned with dermatoscopic images, even if the future use of other imaging techniques to measure the lesion thickness is not excluded, as explained in § 7.2.4. The different criteria used by dermatologists are listed in the following paragraph.

7.2.3 Clinical diagnosis schemes

Whether to use macroscopic or dermatoscopic images has been investigated in different studies. In [14] the conclusion of the authors is to combine both techniques, which is a reasonable compromise. This study, however, is based on the use of existing schemes and does not take into account the fact that the feature extraction problem has not been fully investigated. Clinical diagnosis is not yet able to provide 100 % accurate results. It is therefore difficult to take any diagnosis scheme as a reference. They constitute a starting point, but not a final goal. In [15] the question is to know if dermatoscopy always improves the diagnosis accuracy. Actually the authors have shown that it can be even deteriorated when used by dermatologists who are not formally trained in this technique. It is undeniable that dermatoscopic images are difficult to analyze and that skin cancer research has not yet fully exploited their potential. It is therefore important to develop image processing techniques that may help in both the validation of dermatoscopic criteria and the clinical diagnosis.

Since we are not concerned with macroscopic images, diagnosis schemes for this type of images are not presented (see [8]). The diagnosis scheme we will use for dermatoscopic images has been established by Stolz et al. [3]. Among different criteria the most relevant have been selected and a scoring system has been established to diagnose MM and other types of skin lesions. The main achievement is the *ABCD rule of dermatoscopy*. The different letters correspond to the following criteria:

- **A**symmetry: in zero, one, or two orthogonal axes (0-2 points). Color, texture, and shape must be taken into account.
- **B**order: the lesion is divided into 8 pieces (radial) which are then labeled as showing a sharp cut-off with the surrounding skin or not (0-8 points).
- **C**olor: the presence of up to six known colors must be detected (white, red, light brown, dark brown, slate blue, and black. 1-6 points).
- **D**ifferential structures: 5 patterns are relevant for specific types of lesions (1-5 points).

The results of this analysis are then weighted to obtain a *total dermatoscopic score* that will be used to classify the lesions. We are not yet interested in this part and the reader should refer to [3, 16] for more information about the scheme and the dermatoscopic criteria. There is already a lot to do to extract efficiently the different parameters, which means that quantification schemes must be established that fully exploit their capability to differentiate benign from malignant melanocytic lesions. It may also be proven in future studies that the combination of only these four parameters is definitely not sufficient to obtain 100 % accurate diagnosis.

The ABCD rule of dermatoscopy reveals the weakness of the evaluation done by physicians. The different criteria have a finite number of states and there is no clear description of what is exactly meant by these criteria and how to make the evaluation. In addition, there is no clear limit between the different states and the feature extraction will entirely depend on the physician's experience. It is important to design extraction schemes that rely on fixed and robust criteria and to provide a real-valued quantification of the different features.

Unfortunately, the feature extraction phase has been almost always neglected. Most of the published work dealing with the computer aided diagnosis of skin cancer extract without coherence as many parameters as possible and then focus on the design of an "efficient" classifier. Before

talking more in details about this problem, we will give a complete scheme of CAD system and the principles of such an approach.

7.2.4 CAD scheme and principles

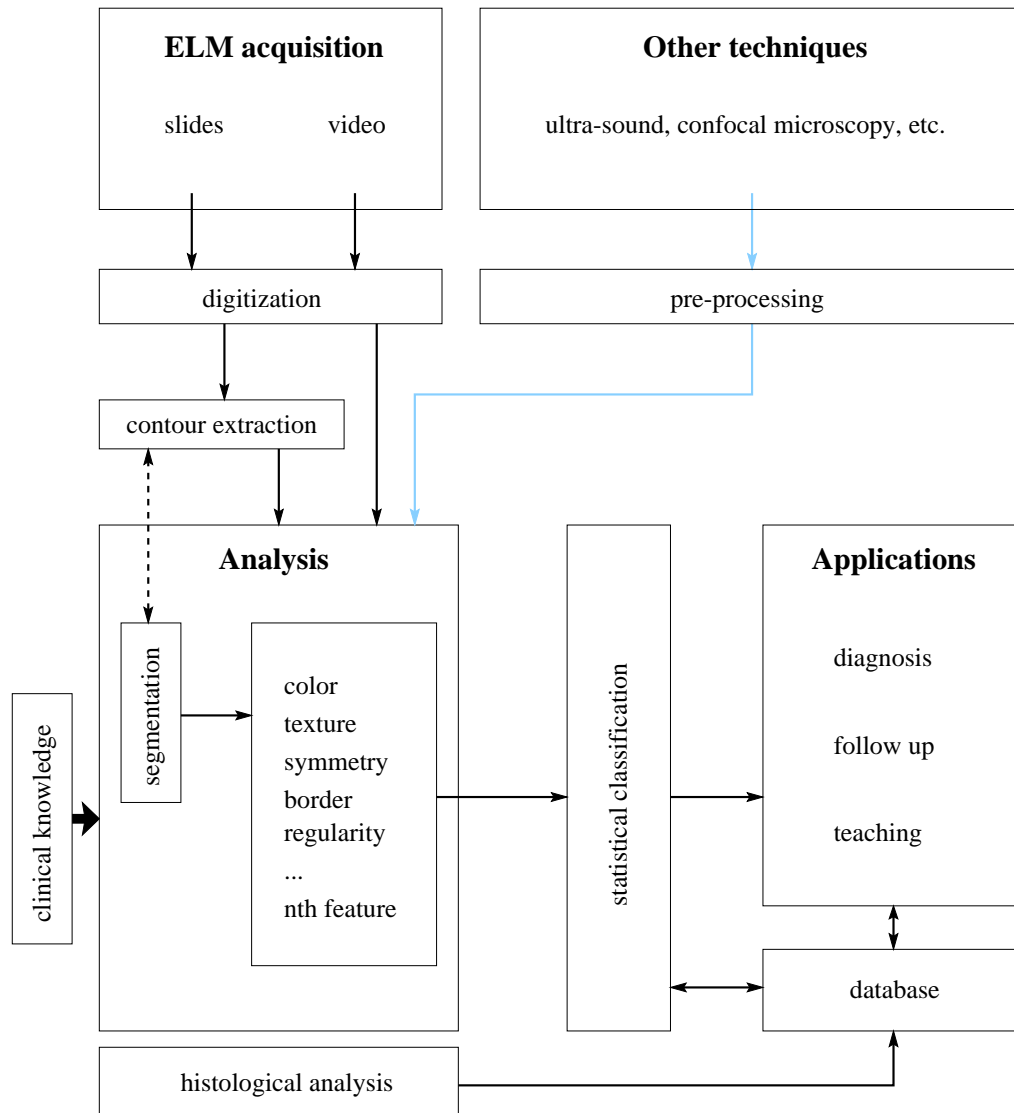


Figure 7.5: Computer aided diagnosis system for skin cancer.

The first step in this research was to establish the scheme of the CAD system for skin cancer shown in Fig. 7.5. The foundation for our work was the clinical knowledge collected by physicians in order to establish diagnosis schemes. The data used as input to our system are digitized dermatoscopic images obtained through ELM, with the possibility, in future extensions, to integrate another acquisition system such as ultra-sound or confocal microscopy, depending on their respective performances. The first processing step is the lesion detection, which can be obtained

either through contour detection or image segmentation techniques. Different features must be quantified for the statistical classification. In parallel to this analysis part, data must be collected. Images of benign and malignant lesions must be collected with their histology² and stored into a database. This collection of images will be used to train the classification scheme and for later consultation by physicians.

The principle of a CAD system is not only to help the physicians during the diagnosis phase, but also to manage the different computing resources. It offers an interface between the dermatologist and the image processing routines, allows the storage and retrieval of unclassified and diagnosed cases, have access to the network and possibly to a database server, and finally to control the digitization process or even a digital camera connected to the computer.

The realization of such a CAD system can be split into several research and development projects:

- true color acquisition system,
- medical diagnosis scheme,
- image processing algorithms,
- database and network management.

In this project the acquisition system is given. The consequence is that the image processing algorithms must be as insensitive as possible to the imperfect acquisition. The medical diagnosis scheme is also in the above list because even if different approaches have been proposed up to now, it is not yet well known what are the most important features and how they should be evaluated. The validation of these schemes is often based on a limited number of lesions and has therefore a poor statistical meaning. It is indeed very difficult to obtain “good quality” images of pigmented lesions, and especially malignant melanoma, with a complete histology. The image processing algorithms are mainly devoted to contour extraction, segmentation, and features extraction, quantification, and analysis steps. In our study we will focus on color segmentation, contour detection, and symmetry quantification. Finally the database and network research are very important since they allow the data management and the connection between physicians from different parts of the world.

In the following paragraph we will shortly review the different publications on computer aided diagnosis of skin cancer.

7.2.5 Feature extraction and classification

The feature extraction part is necessary to every classification system. While several studies are dealing with computer aided diagnosis systems for skin cancer, none of them focuses on the extraction of these parameters. It seems from the literature that this problem has already been solved, which is far from reality, since even from a clinical point of view it is not yet clear how these features should be evaluated and what is their real weight in the diagnosis. There is a kind of excitement in this research domain that pushes everybody to develop as quickly as possible a complete system and to present classification results, even if this part is always performed on a limited number of cases. The design of classifiers, using mainly artificial neural networks, has been investigated in most of the studies dealing with this problem [17, 18, 19, 20, 4, 21, 22, 23, 24].

²The histology is obtained after excision and microscopic inspection.

In [25] the lesion boundary is obtained by simple thresholding of the blue component histogram. This component usually has a weak contrast, while the red component, the closest to the color of the pigment, namely brown, has most of the time a high contrast. It is also difficult to believe that a 1-D histogram thresholding approach, which is among the simplest, can perform well in all the cases. Different features are also quantified, such as the asymmetry using a gray-scale principal component decomposition. The author talk also about texture analysis, but without any details. Very simple parameters, such as area and perimeter, are also extracted in [26]. Some statistical measurements on the different color components are also used, and the lesion is again detected using a global thresholding technique. In [27] the authors try to examine critically the use of computer image analysis but use very simple feature extraction methods, such as the binary principal component for asymmetry. Finally, similar work can be found in [28] and [29]. In the latter case the authors are concerned with texture classification, but without relevant breakthrough.

We must conclude that no real investigation on specific diagnostic features has been performed yet. Most of the studies use the same or very similar techniques and do not provide any improvement in the computer aided extraction and quantification of diagnostic features. We aim at doing so with the complete analysis of symmetries in pigmented lesions. In our study the term asymmetry, used by dermatologists, has been replaced by symmetry, which has a clear meaning. Asymmetry is nothing but the negation of symmetry and the limit between both is a hard limit. An object is perfectly symmetric or it is not. In our case we will deal with almost and non-symmetrical objects and give a quantitative value to this feature.

In the following section we will present a hair removal scheme. Hairs are very often present in dermatoscopic images and corrupt strongly the different image processing steps. It is therefore important to detect its presence and to replace the concerned pixels with an interpolated value. Especially the filtering steps are affected by the presence of hair.

7.3 Hair removal in dermatoscopic images

An important parameter that must be always taken into account in this study is that the images have been obtained from different sources, which means that there may be large variations in terms of image “quality”, and that no particular care has been taken during the acquisition procedure. In addition, there is no standard acquisition system and the images may be digitized slides or numerical images obtained with CCD cameras. In summary, the image processing techniques must work without any restriction on the material (images).

An immediate consequence is that external components that may corrupt the image and hence the image processing results must be localized and removed, masked, or replaced with some interpolated values. One of the most undesirable component present in dermatoscopic images is hair. It is visible as long dark lines that sometimes cross the whole image. Therefore, a typical example of image processing that would be strongly influenced by the hair is the filtering by nonlinear isotropic diffusion, as well as all schemes which may use gradient information. Hair acts as a high and impassable wall on the diffusion process. The image pixels representing hair must be replaced by pixel values obtained through a smooth interpolation with neighboring pixels (located outside the hair).

The goal in this section is not to propose a way to restore hidden information due to the

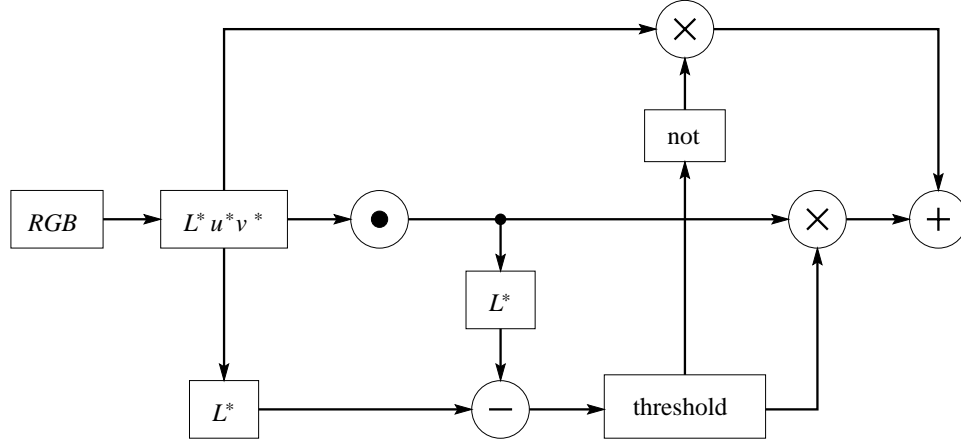


Figure 7.6: Hair removal scheme.

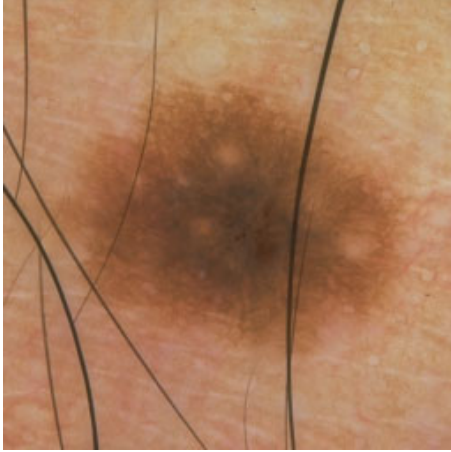


Figure 7.7: Image corrupted with hair.

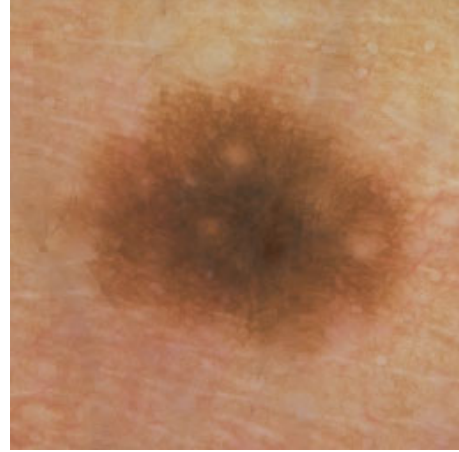


Figure 7.8: Image after hair removal.

presence of hair but to replace the concerned image pixels in such a way that later processing is not influenced. It must be clearly underlined here that any spurious detail introduced at the acquisition level is definitely present and that there is no way to restore hidden information. It is however possible to detect these components and to reduce the (negative) visual effect they cause, which in turn deteriorates the medical diagnosis. A simple, quick and efficient hair removal scheme is presented next.

In our approach we suggest to apply the morphological closing operator [30] with a spherical structuring element to the L^* , u^* , and v^* components, and to threshold the difference between the luminance before and after closing. Hair is indeed very light absorbent and we could experience that a constant threshold can be set up quickly, even when the image “quality” is variable. The selected pixels are replaced with their value after morphological closing, which results in a local interpolation, while other pixels are left unchanged. It may happen that pigmented structures

show to have a similar morphology and a high contrast, but even if parts of them are selected as hair, the introduced modification is usually imperceptible and without effect on the following filtering.

Figure 7.6 shows the complete hair removal scheme. Figures 7.7 and 7.8 show the same pigmented skin lesion respectively before and after hair removal. When looking closer very small color changes are perceptible where we previously had hair. However, this result is sufficient to allow an efficient filtering of the image and even the visual analysis of the lesion by a dermatologist.

Now that the main artifact has been removed from the dermatoscopic images, the different schemes developed in this thesis can be applied and evaluated.

7.4 Segmentation and boundary detection

7.4.1 Introduction

The first processing step towards a complete analysis of pigmented skin lesions is the separation between the lesion and the healthy skin. This can be obtained either by image segmentation or contour detection, and this is why we have investigated both approaches. The techniques developed in Chapters 4 and 5 will be used with dermatoscopic images and typical examples of results will be shown in this section. A number of lesions have been used to compare the results of automatic techniques with those of expert dermatologists.

The contour detection problem is especially tricky in dermatoscopic images where the boundary between lesion and surrounding skin is often difficult to establish. Most of the techniques proposed in the literature are concerned with the contour detection in macroscopic images [31, 32, 33, 34]. The contrast between lesion and skin is much higher and simple methods like thresholding of 1-D histogram may be sufficient to detect the lesion. For dermatoscopic images, more sophisticated techniques must be developed to achieve the separation between lesion and healthy skin. Several schemes taken from the literature have been evaluated in [35, 36] and an edge focusing algorithm in [37]. However, even visual assessment has shown the limitation of the different approaches when the lesion shows different shades in color and a smooth transition to the surrounding skin.

An important problem related to contour detection is the validation of image processing techniques. This problem is especially sensitive when dealing with dermatoscopic images where the goal is not to mimic the physician but to avoid subjective evaluation. It is therefore contradictory to validate contour detection results with contours drawn by physicians, knowing that every human, expert or not, will not be able to reproduce exactly his results. This phenomenon will be illustrated in § 7.4.3. However the validation can be done by showing that automatic techniques give results that are statistically close to those of a group of experts.

In the following paragraph we will first illustrate the different techniques developed in Chapters 4 and 5. Visual assessment is indeed very important in the validation of segmentation and contour detection techniques for dermatoscopic images.

7.4.2 Visual assessment

Different measurements are presented in this and the following paragraph as box and whisker plots [38]. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data.

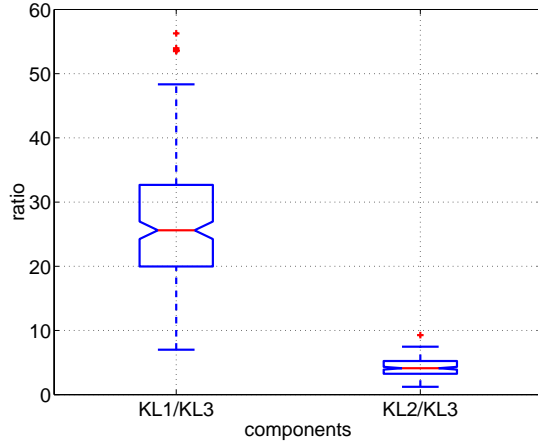


Figure 7.9: Ratios between variance of the different KL components in the RGB color space.

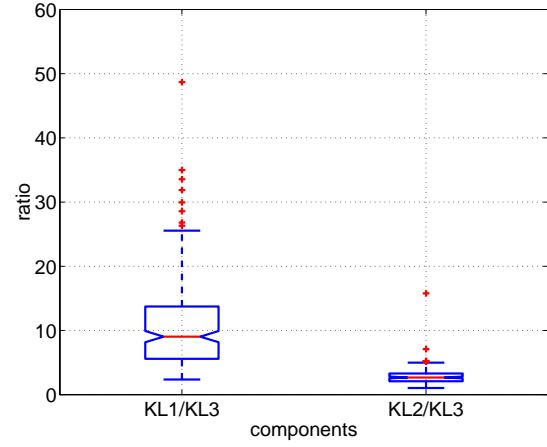


Figure 7.10: Ratios between variance of the different KL components in the $L^*u^*v^*$ color space.

They are computed as a fraction (here 1.5) of the difference between the upper and lower quartiles. However, the whisker limits can not extend beyond the smallest and highest values. Outliers (+) are data with values beyond the ends of the whiskers. Finally, the boxes are notched. Notches represent a robust estimate of the uncertainty about the means for box to box comparison. They are computed from the median value at a distance equal to a fraction of the difference between the upper and lower quartiles normalized by the square root of the data size.

The first plots are shown in Fig. 7.9 and 7.10, where the ratios between the variance of the different KL components have been computed respectively in the RGB and $L^*u^*v^*$ color spaces. They show that if variance is used as a measure of “information content”, then most of the information is projected onto the first two components. This phenomenon is weaker in the $L^*u^*v^*$ space, but computation results have shown that the pixel variance in the third component is mainly due to noise. Two hundred unclassified lesions have been used to produce these results.

Figure 7.11 shows segmentation and contour detection results of dermatoscopic images. The original image is shown top-left, the contour obtained by nonlinear isotropic diffusion and morphological flooding is shown top-right, while the left and right-bottom images are the segmentation results obtained respectively with the FCM and OS-FCM techniques developed in Chapter 4. The latter method gives a better partition of the data and therefore the segmentation result fits better the visible regions. A complex image that contains hair, crusts, and a pigmented lesion with smooth borders and several regions (bluish, dark brown, and light brown) is shown in Fig. 7.12. The segmentation result using the OS-FCM technique is shown bottom-left, the contour detection result is shown top-right, and the bottom-right image shows all the contours corresponding to the different energy minima present in the energy functional used in the contour detection approach (see Chapter 5). It clearly demonstrates that not only the lesion contour but also regions inside the lesion can be detected. Finally the difference between the FCM and OS-FCM techniques is even more visible in Figs. 7.13 and 7.14, where both approaches have been respectively used. The reddish healthy skin gives in the 2-D histogram an elongated and smooth cluster, which is best

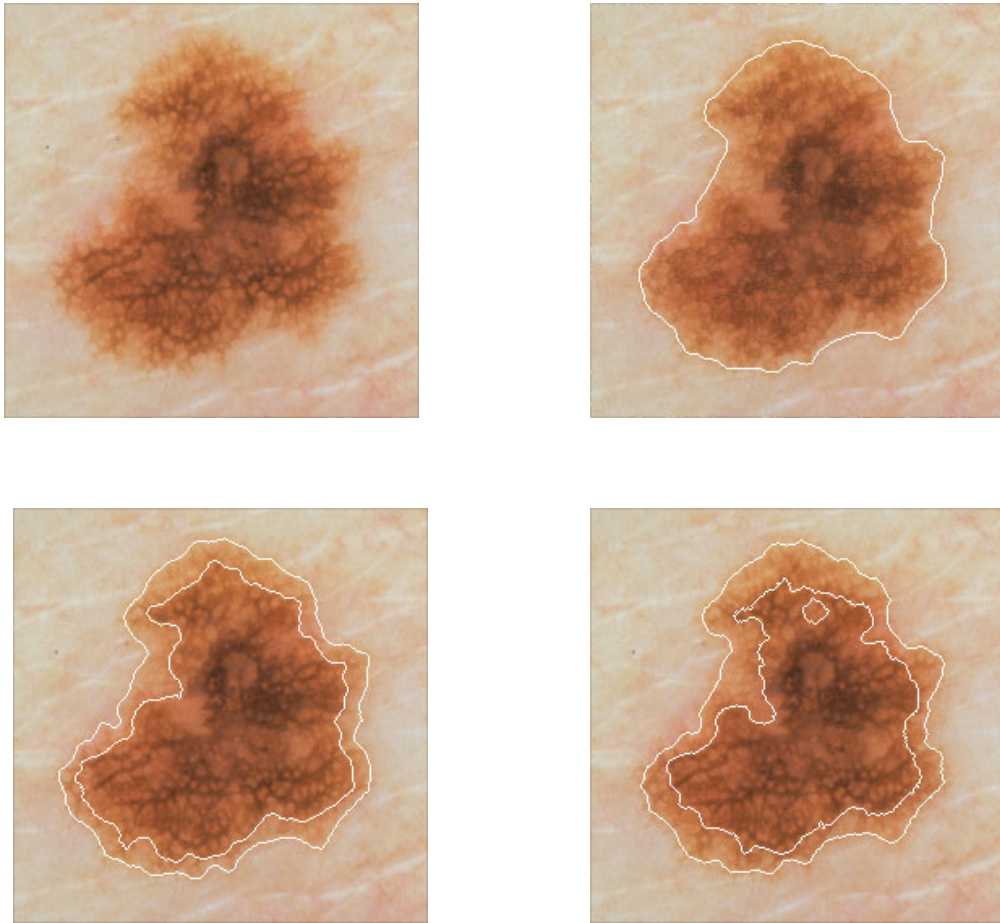


Figure 7.11: Contour detection of pigmented skin lesion. The original lesion is shown top-left, the contour obtained after nonlinear isotropic diffusion and morphological flooding is shown top-right, while the left and right-bottom images are respectively obtained with the FCM and OS-FCM techniques.

handled by the OS-FCM technique.

These different examples demonstrate that the dermoscopic images are very difficult to segment because the boundary between lesion and healthy skin does not show a hard transition that can be uniquely detected. Visual assessment by expert dermatologists was very successful and has shown that the two techniques used in this study always give results that have been considered as acceptable contours.

In the following paragraph we want to compare analytically the different results to the contours drawn by trained dermatologists.

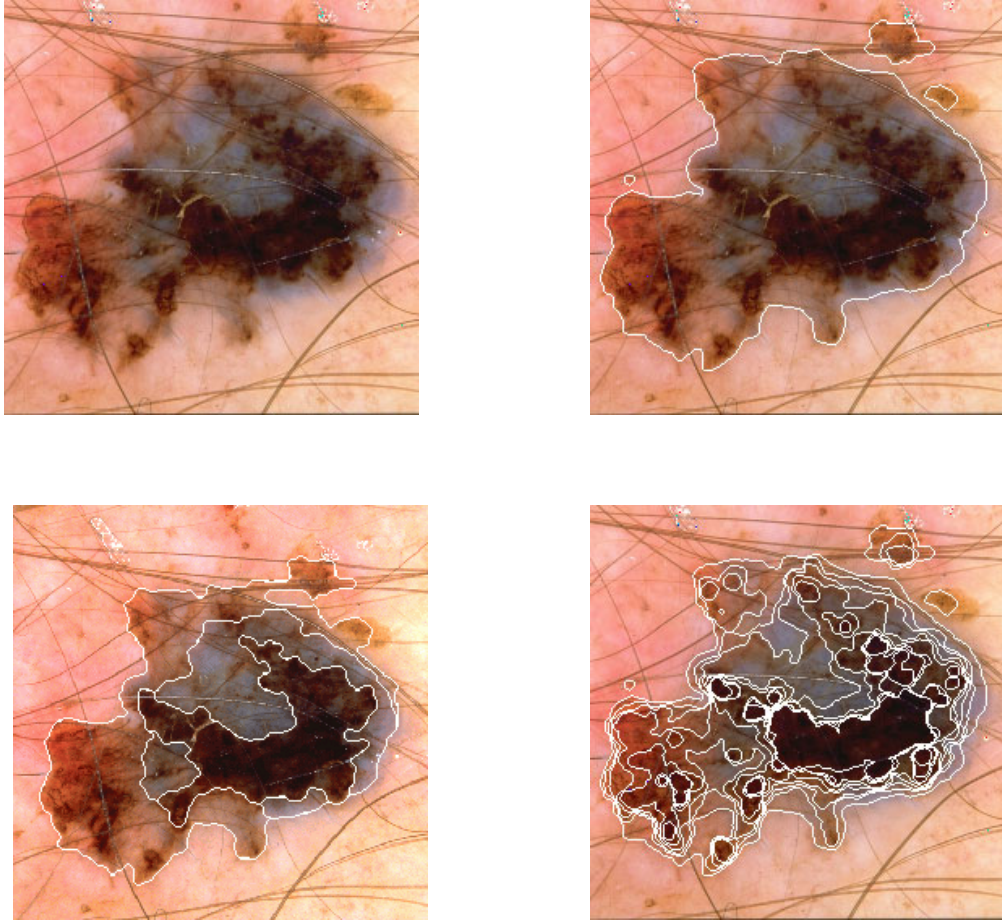


Figure 7.12: Contour detection of complex pigmented lesion. The original lesion is shown top-left, the contour obtained after nonlinear isotropic diffusion and morphological flooding is shown top-right, the bottom-right image shows the different curves corresponding to all local minima of the energy functional, and the segmentation by OS-FCM is shown bottom-left.

7.4.3 Validation

While visual assessment by trained dermatologists has shown that both the segmentation and contour detection methods lead to result that fit the perceived regions and contours, a more rigorous experiment has been performed in order to validate these methods. The idea was to select a number of lesions, actually 25, showing a smooth and fuzzy transition between the lesion and the surrounding skin, and to give them to 5 dermatologists for manual contour drawing. This experiment has been repeated three times for each physician, at several days of interval, in order to evaluate their ability to reproduce results. Two examples of manual drawing obtained from different dermatologists are shown in Figs. 7.15 and 7.16. One can note that the contour location is uncertain in regions where the transition between lesion and healthy skin is very smooth and where the contour is non-convex (regions where the contour penetrates the lesion).

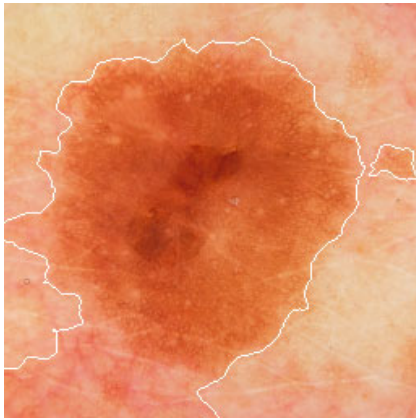


Figure 7.13: Pigmented skin lesion segmented with the FCM technique.

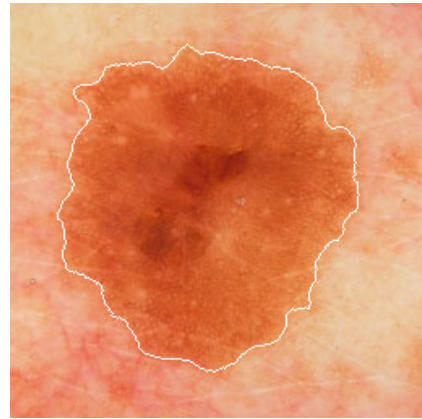


Figure 7.14: Pigmented skin lesion segmented with the OS-FCM technique.

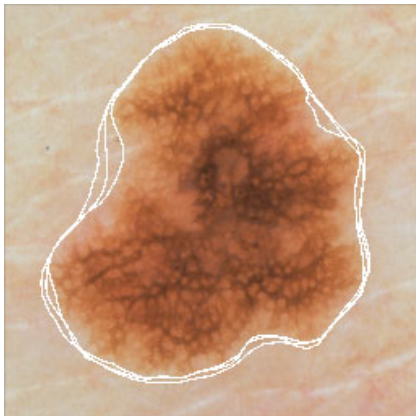


Figure 7.15: Contour drawn by physician (A).

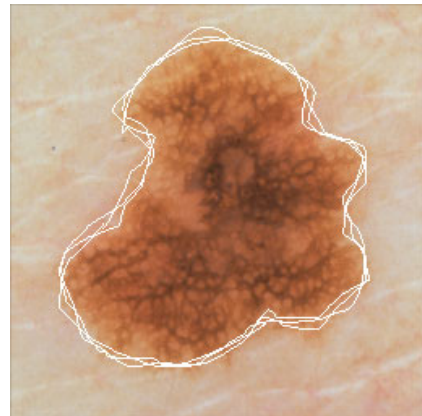


Figure 7.16: Contour drawn by physician (B).

In order to evaluate the constancy of the different results, the following measure has been used:

- Take the processing days pair-wise.
- For every physician and pairs compute two masks, one containing all pixels that have been labeled as being in the lesion at least once, and a second one containing all the pixels that have been labeled as being in the lesion only once.
- Compute the ratio between the number of pixels in the second mask and the number of pixels in the first mask.
- Repeat the procedure with the three masks. In that case the second mask contains all pixels that have been labeled at least once but not all three times.

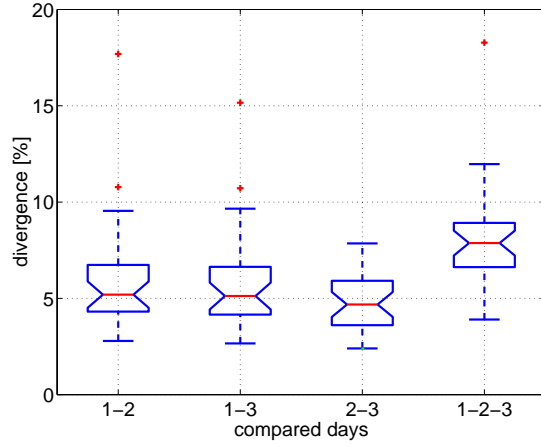


Figure 7.17: Box plots of the different divergence rates obtained for 25 lesions and three drawings performed at different days (A).

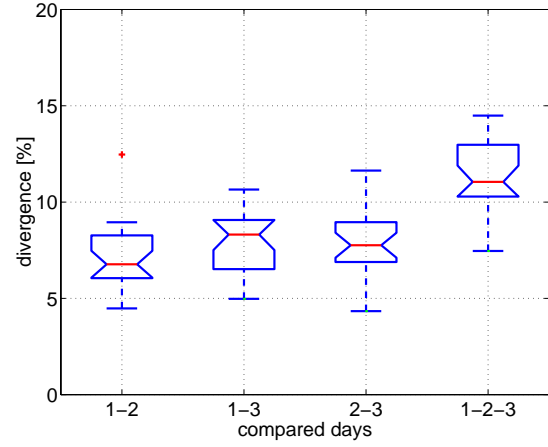


Figure 7.18: Box plots of the different divergence rates obtained for 25 lesions and three drawings performed at different days (B).

The above ratios are equal to zero only when the contours are identical. Figures 7.17 and 7.18 show the box plots corresponding respectively to dermatologist (A) and dermatologist (B). They correspond to the physicians who were respectively the most and less constant in the contour drawing. One can note that when considering the results of all three days together the error is higher, which means that the introduced error is different between the successive days and therefore added in this measure. The immediate and expected conclusion is that hand drawn contours in dermoscopic images do not show sufficient constancy to be used as absolute references to validate image processing techniques. It may be interesting, however, to know if our segmentation and contour detection schemes can be differentiated from a group of experts.

The procedure adopted for the inter-physician measurements is slightly different. A probability image has been computed for every lesion based on the different contour results, included those obtained with our automatic techniques. Such a probability image is shown in Fig. 7.19, where the probability for a given pixel to be inside the lesion is 1 when shown black and 0 when shown white. This representation gives a kind of group statistic and can be used to compare every result taken separately with the other. Only pixels that have a non-zero probability to be inside the lesion are considered. Then, for every lesion, day, and physician, the mean probability that a pixel has been misclassified is computed. This measure allows one to say if an observer has consistent results compared to the group of observers. Figure 7.20 shows the result obtained for both techniques developed in this thesis. The clustering technique shows to give closer results to those of the physician than the contour detection technique. While the latter has a low median value, the misclassification error can become quite high in some cases. This result is certainly due to the fact that the clustering technique uses similar criteria as the physician, while the contour detection technique does not draw the contour by local evaluation but instead chooses between different contour candidates based on a global measure. This can lead to larger divergences from the results provided by physicians.

Figures 7.21 and 7.22 show the results obtained respectively for physician (A) and (B). In both



Figure 7.19: Probability image obtained from the hand drawn and automatic contours.

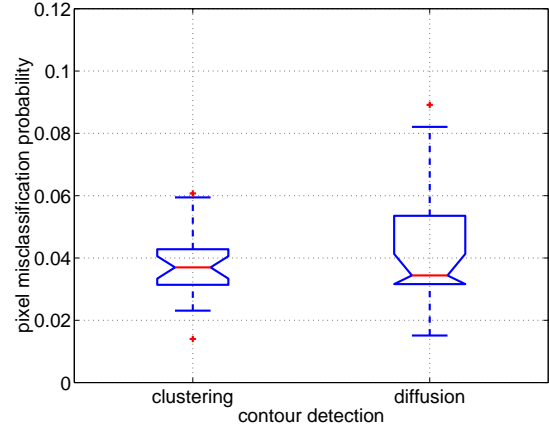


Figure 7.20: Probability that an image pixel has been misclassified using our schemes.

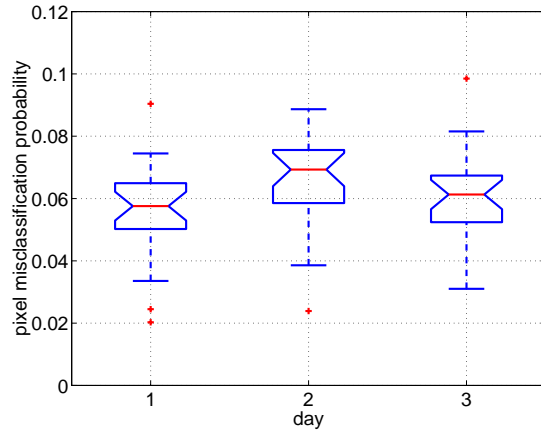


Figure 7.21: Probability that an image pixel has been misclassified (A).

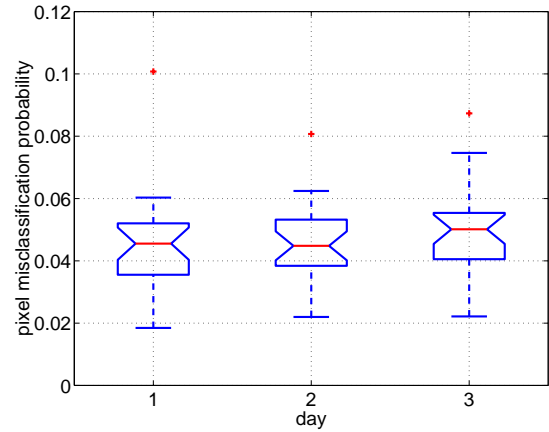


Figure 7.22: Probability that an image pixel has been misclassified (B).

cases the box plots show that the misclassification error is generally higher than that obtained with the automatic techniques. The most interesting thing is that physician (A), who was the most able to reproduce his results, has the worst results when compared to the others. The criteria used by physician (A) to draw a contour are certainly different from those used by the other physician, but he is able to reproduce quite precisely the results. It is now difficult to say if physician (A) is a good reference because he is constant, or a bad one because his results diverge from those of the others.

Finally we will conclude that the validation of image processing techniques for contour detection using a human reference is not adapted for two main reasons: first, the goal of our developments is to avoid human subjectivity in the processing of specific tasks, and the use of hand drawn contours as references is therefore contradictory to this approach. The results obtained in this

section maintain this argument. The most objective way to validate the segmentation and contour detection results is to evaluate the ability of these algorithms to provide the feature extraction part with a lesion mask that does not corrupt the final classification. It is therefore too early to conclude if the techniques developed in this thesis are well adapted to the computer aided diagnosis of malignant melanoma, but we can already affirm that the provided techniques are very promising.

7.5 Symmetry

7.5.1 Introduction

In this study we have focused on symmetry quantification as a classification feature. Shape, color, and texture information are handled separately because there is no proof of a strong correlation between components, especially in almost and non-symmetrical objects. The application to pigmented skin lesions is discussed in this section. It is almost certain, according to clinical research, that symmetry (or asymmetry) considered alone is not sufficient to discriminate between benign and malignant lesions, and even between different types of skin lesions. However we want to show that the way we extract this information performs much better than classical approaches.

As described in § 7.2.5 the symmetry feature has never been really exploited and only very simple approaches have been used up to now. In [39] the authors use the principal component decomposition of a binary mask to compute two orthogonal axes. An asymmetry index is computed for both axes using the following definition:

$$\text{Asymmetry index} = \frac{\Delta A}{A} , \quad (7.1)$$

where ΔA is the non-overlapping area between the original and reflected mask, and A is the area of the original mask. This index is computed for both axes. In [39] only the minimum value is kept. Here we will keep both values.

Another approach that uses the principal component decomposition is given in [25]. The gray-level information is integrated as discussed in Section 6.4. From the principal component the image is rotated so that it coincides with the x axis. Using the notation of § 6.5.1, the asymmetry index is computed as follows [25]:

$$\text{Asymmetry index} = \frac{\sum |v(\mathbf{x}) - \tilde{v}(\mathbf{x})|}{\sum v(\mathbf{x})} , \quad (7.2)$$

where $v(\mathbf{x})$ is the original image and $\tilde{v}(\mathbf{x})$ the reflected version. The index is computed for both axes and the image is masked with a binary mask of the lesion before processing.

We will use both techniques as comparison to our scheme. In the following section some examples will be given first.

7.5.2 Visual assessment

Figures 7.23 and 7.25 give two examples of texture symmetry map obtained for pigmented skin lesions and the corresponding axes are superimposed on the original images in Figs. 7.24 and 7.26 respectively. In the former case the lesion is benign and symmetrical, while in the second case it is malignant and asymmetrical. In any case at least one axis is found, namely the axis that minimize

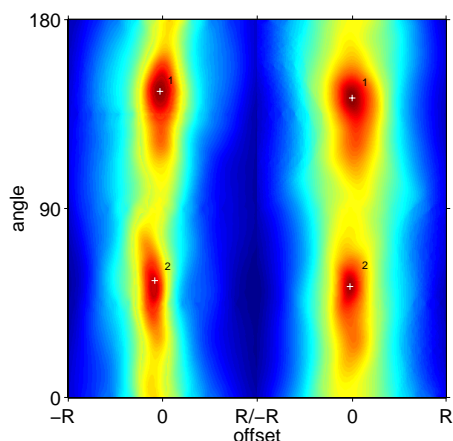


Figure 7.23: Texture (left) and color (right) symmetry maps of symmetric lesion.

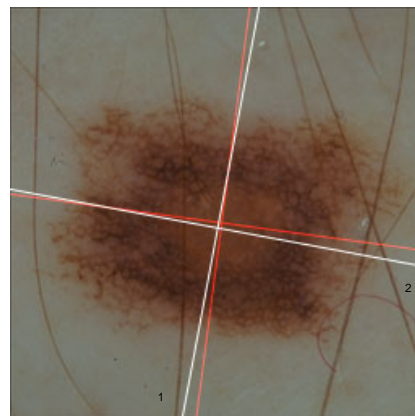


Figure 7.24: Axes obtained from the map of Fig. 7.23 (red for color, white for texture).

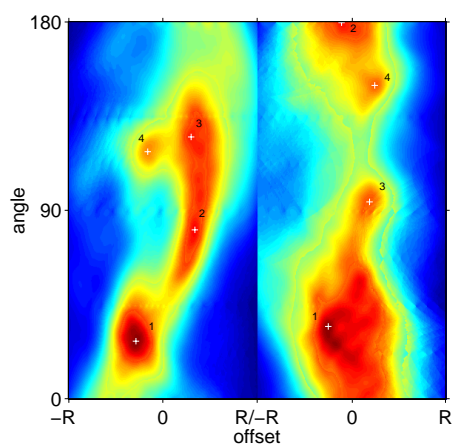


Figure 7.25: Texture (left) and color (right) symmetry maps of asymmetric lesion.

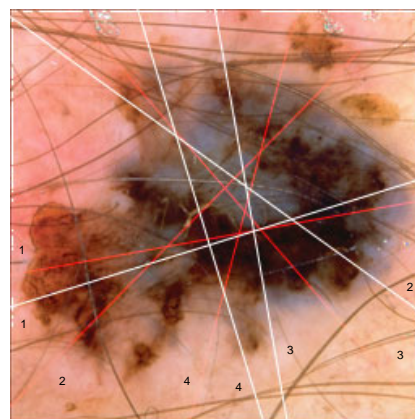


Figure 7.26: Axes obtained from the map of Fig. 7.24 (red for color, white for texture).

the MSE between the original and reflected images. Therefore even asymmetrical images have a symmetry axis but with a low symmetry value. It is interesting to note that when lesions are asymmetrical they have multiple local minima, as can be seen from Fig. 7.26. This characteristic may be exploited in future studies.

A symmetry map contains much more information than the parameters of the axis that minimizes the difference between the original and reflected images. Local minima are also detected, as well as their respective location in the symmetry map. This distribution of minima can be analyzed in order to improve the classification based on symmetry measures. The more information we use, the better we can describe a single object. This kind of analysis is left for future studies.

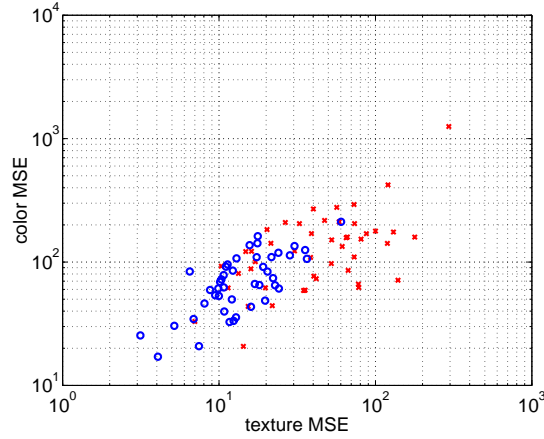


Figure 7.27: Color and texture symmetry MSE of 50 malignant melanoma (x) and 50 naevi (o). The lesion masks have been obtained through the contour detection technique.

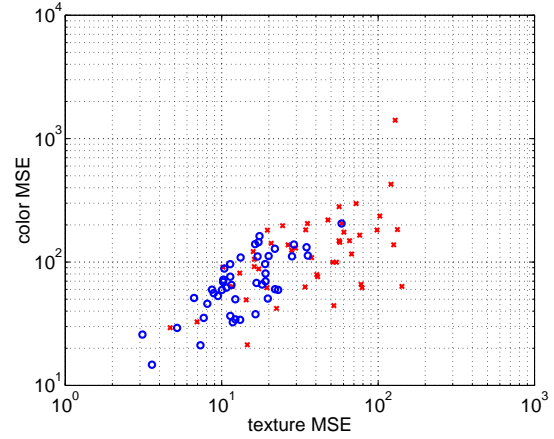


Figure 7.28: Color and texture symmetry MSE of 50 malignant melanoma (x) and 50 naevi (o). The lesion masks have been obtained through the clustering technique.

7.5.3 Symmetry values for diagnostic classification

A linear classifier with training by epoch [40] is used to evaluate the symmetry measure technique. This method uses a gradient descent approach to determine the coefficients of the hyperplane that minimizes the misclassification error. More sophisticated classifier must be developed to end up with an efficient CAD system, under the requirement that more data is available. It will unfortunately take time to collect a significant number of images of both malignant and benign melanocytic lesions with their histology. Symmetry alone can not be used to diagnose malignant melanoma, but it is necessary to exploit fully every diagnostic feature in order to improve the diagnosis accuracy. This is what we want to achieve with our symmetry quantification scheme.

In this paragraph we present some linear classification results that have been obtained with two different approaches: the use of symmetry maps and the principal component decomposition. In the former case three SM are available, for shape, color, and texture symmetry, and in the latter case binary and masked gray-level (luminance component L^*) images have been used. For every approach two parameters have been extracted, the two smallest local minima in the SM case, and the two asymmetry indexes defined in the introduction paragraph for the principal component decompositions, using Eqs. 7.1 and 7.2 respectively. The material used for this initial study is a set of 50 malignant melanoma and 50 benign naevi. In the latter set, 20 lesions were *dysplastic naevi* which are considered as a particular class of benign lesions with a high potential to turn malignant.

Figures 7.27 and 7.28 show the minimum MSE obtained from the texture and color symmetry maps of 50 MM (x) and 50 naevi (o), computed using the lesion masks obtained with our segmentation and contour detection schemes respectively. There is a perceptible influence of the contour detection result on the symmetry map computation, but this did not influence the classification results presented below. Finally we see from this plot that even if the limit between both classes is not clear, the texture component seems to be a better discriminant.

<i>Symmetry features</i>	texture	texture-color	texture-color-shape
<i>Sensitivity</i>	66 %	64 %	78 %
<i>Specificity</i>	90 %	92 %	90 %
<i>Positive predictive value</i>	86.8 %	88.9 %	90.7 %
<i>Negative predictive value</i>	72.6 %	71.9 %	80.4 %
<i>Index of suspicion</i>	76 %	72 %	86 %
<i>Diagnostic accuracy</i>	60 %	59.3 %	72.2 %

Table 7.1: Linear classification of symmetry values for benign and malignant lesions.

<i>Symmetry features</i>	texture	texture-color	texture-color-shape
<i>Sensitivity</i>	76 %	78 %	82 %
<i>Specificity</i>	83.3 %	80 %	80 %
<i>Positive predictive value</i>	88.4 %	86.7 %	87.2 %
<i>Negative predictive value</i>	67.6 %	68.6 %	72.7 %
<i>Index of suspicion</i>	86 %	90 %	94 %
<i>Diagnostic accuracy</i>	69.1 %	69.6 %	73.2 %

Table 7.2: Linear classification of symmetry values for MM and naevi.

A number of measurement can be used to evaluate the classification accuracy:

$$\begin{aligned}
\textit{sensitivity} &= \frac{TP}{TP + FN} , \\
\textit{specificity} &= \frac{TN}{TN + FP} , \\
\textit{positive predictive value} &= \frac{TP}{TP + FP} , \\
\textit{negative predictive value} &= \frac{TN}{TN + FN} , \\
\textit{index of suspicion} &= \frac{TP + FP}{TP + FN} , \\
\textit{diagnostic accuracy} &= \frac{TP}{TP + FP + FN} ,
\end{aligned}$$

where TP is the number of true positives, FN the number of false negatives, TN the number of true negatives, and FP the number of false positives. A *positive* is a lesion that has been classified as being malignant. The *sensitivity* is the proportion of MM that have been classified as MM and the *specificity* the proportion of lesions which are not MM that have been classified as being not MM. The *positive predictive value* is the proportion of lesions classified as MM that are really MM, and inversely for the *negative predictive value*. The *index of suspicion* gives the degree of awareness of the likelihood of a lesion to be a MM. An index higher than 100 % represents an *over-diagnosis* while an index lower than 100 % represents an *under-diagnosis*. The *diagnosis accuracy* is the proportion of cases in which the classification fits the diagnosis.

Table 7.1 gives the different parameter values for different combinations of symmetry features: texture (2-D), texture and color (4-D), and texture, color, and shape (6-D). In Table 7.2 the results obtained without the dysplastic naevi (considered as benign lesions) are given. These results show that the combination of shape, color, and texture improves the separation between benign and

<i>Symmetry features</i>	binary	gray-level
<i>Sensitivity</i>	60 %	70 %
<i>Specificity</i>	82 %	92 %
<i>Positive predictive value</i>	76.9 %	89.7 %
<i>Negative predictive value</i>	67.2 %	75.4 %
<i>Index of suspicion</i>	78 %	78 %
<i>Diagnostic accuracy</i>	50.8 %	64.8 %

Table 7.3: Linear classification of melanocytic lesions using principal component decompositions.

malignant lesions. Especially shape and texture are relevant features, while the color symmetry does not improve significantly the results. Malignant melanoma show to be rather asymmetrical, but this criteria is not sufficient to separate malignant from benign lesions.

An important study that must follow these initial results is the evaluation of different texture and color descriptors. The features used in this study have shown to work well on synthetic images and in noisy environments (see Chapter 6) but may not be optimal for dermatoscopic images.

Table 7.3 gives the results obtained with the same test set but with the principal component approach. A binary mask of the lesion, obtained through the segmentation scheme, is used as input, providing thus a measure of shape symmetry, and to mask a gray-level version of the input images. The binary approach gives weak results while the gray-scale approach improves significantly the classification. These two approaches, however, are more affected by the mask computation, especially the binary approach, which is intuitively appealing.

It is interesting to note from the above results that a very simple approach such as the gray-level principal component decomposition gives slightly better results than the texture symmetry obtained from the SM, taken alone. One possible reason is that the former method uses asymmetry indexes computed from two orthogonal axes, evaluating thus the asymmetry criteria in two orthogonal directions. In the SM case, the two smallest MSE were used, which in several cases correspond to two axes which are far from orthogonal, especially for malignant melanoma. It is therefore important to extend this work with the extraction of all the local minima present in a SM and different criteria such as their respective location. According to what we could observe it may be a very promising issue to the symmetry problem in dermatoscopic images.

In order to employ a more complex classifier, such as a neural network for example, more data is needed. However, this simple procedure illustrates that this approach is encouraging and that the integration of other diagnostic features may lead to a better separation between benign and malign lesions. It is therefore necessary to develop efficient image processing algorithms, which was not the case up to now.

7.6 Summary

This chapter was concerned with the main application of the image processing techniques developed in this thesis, namely digital dermatoscopy. This approach to skin cancer diagnosis uses epiluminescence microscopy to render the skin translucent and to make pigmented structures visible. In Section 7.2 the different notions related to this medical domain have been presented, as well as the notion of computer aided diagnosis system. Prior to the segmentation and feature

quantification steps, the hair removal scheme that has been introduced in Section 7.3 must be used to suppress this disturbing component.

The segmentation and contour detection schemes developed respectively in Chapter 4 and Chapter 5 are illustrated in Section 7.4, followed by a statistical evaluation based on the comparison with hand drawn contours, obtained from a group of trained dermatologists. We could show that a physician, even when trained in this type of diagnosis, can not be used as an absolute reference. Human beings are usually not able to reproduce precisely measurements and the comparison can therefore only be done with a group of experts. In that case the “behavior” of the developed techniques must be as close as possible to that of the group. We could show that the within group error introduced by the automatic schemes, and especially that of the segmentation scheme, is generally lower than that introduced by each physician taken alone.

The symmetry quantification scheme developed in Chapter 6 to compute symmetry maps has been finally evaluated. We could experience that benign lesions have most of the time only two well separated minima (almost orthogonal), while multiple minima are present in SMs of malignant lesions, and have a more chaotic distribution. It has been shown in Section 7.5 that the combination of texture, color, and shape symmetry values improves significantly the separation between benign and malignant lesions. Principal component results have been used as comparison. Finally, we could show that even if a unique feature, namely symmetry, can not be used to diagnose efficiently skin cancer, more can be done to efficiently quantify the different features.

Bibliography

- [1] G. Burg. *Das Melanom*. Serie Gesundheit. Piper/VCH, 1993.
- [2] Office fédéral de la statistique. Mélanome malin de la peau, 1969-1994.
- [3] W. Stolz, O. Braun-Falco, P. Bilek, M. Landthaler, and A. B. Cagnetta. *Color Atlas of Dermatoscopy*. Blackwell Science, 1994.
- [4] F. Ercal, A. Chawla, W. V. Stoecker, H. C. Lee, and R. H. Moss. Neural network diagnosis of malignant melanoma from color images. *IEEE Transactions on Biomedical Engineering*, 41(9):837–845, September 1994.
- [5] A. Kjoelen, S. E. Umbaugh, and M. Zuke. Compression of skin tumor images. *IEEE Engineering in Medicine and Biology*, pages 73–80, May-June 1998.
- [6] D. A. Perednia, J. A. Gaines, and T. W. Butruille. Comparison of the clinical informativeness of photographs and digital imaging media with multiple-choice receiver operating characteristic analysis. *Arch. Dermatol.*, 131:292–297, March 1995.
- [7] H. Kittler, M. Seltenheim, H. Pehamberger, and K. Wolff. Diagnostic informativeness of compressed digital epiluminescence microscopy images of pigmented skin lesions compared with photographs. *Melanoma Research*, 8:255–260, 1998.
- [8] M. C. Mihm A. J. Sober, T. B. Fitzpatrick. Primary melanoma of the skin: recognition and management. *Journal of the American Academy of Dermatology*, 2(3):179–197, March 1980.
- [9] A. Breslow. Thickness, cross-sectional area and depth of invasion in the prognosis of cutaneous melanoma. *Annals of Surgery*, 172(5):902–908, 1970.
- [10] A. P. Dhawan, R. Gordon, and R. M. Rangayyan. Nevoscopy: Three-dimensional computed tomography of nevi and melanomas in situ by transillumination. *IEEE Transactions on Medical Imaging*, MI-3(2):54–61, June 1984.
- [11] A. P. Dhawan, P. Kini, and A. Sicsu. Imaging skin-lesions for detecting skin-cancer through image analysis. In Y. Kim and F. A. Spelman, editors, *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, number 2, pages 388–390, Seattle, Washington USA, November 1989.
- [12] W. Dummer, H.-J. Blaheta, B. C. Bastian, T. Schenk, E.-B. Bröcker, and W. Remy. Preoperative characterization of pigmented skin lesions by epiluminescence microscopy and high-frequency ultrasound. *Arch. Dermatol.*, 131:279–285, March 1995.

- [13] M. Rajadhyaksha and M. Zavislan. Confocal laser microscopy images tissue *in vivo*. *Laser Focus World*, pages 119–127, February 1997.
- [14] H. P. Soyer, J. Smolle, G. Leitinger, E. Rieger, and H. Kerl. Diagnostic reliability of dermoscopic criteria for detecting malignant melanoma. *Dermatology*, 190:25–30, 1995.
- [15] M. Binder, M. Schwarz, A. Winkler, A. Steiner, A. Kaider, P. Wolff, and H. Pehamberger. Epiluminescence microscopy; a useful tool for the diagnosis of pigmented skin lesions for formally trained dermatologists. *Arch. Dermatol.*, 131:286–291, March 1995.
- [16] Z. B. Argenyi. Dermatoscopy (epiluminescence microscopy) of pigmented skin lesions. *Dermatologic Clinics*, 15(1):79–95, January 1997.
- [17] M. Binder, H. Kittler, A. Seeber, and A. Steiner. Epiluminescence microscopy-based classification of pigmented skin lesions using computerized image analysis and an artificial neural network. *Melanoma Research*, 8:261–266, 1998.
- [18] M. Hintz-Madsen, L. Kai Hansen, J. Larsen, E. Olesen, and K. T. Drzewiecki. Design and evaluation of neural classifiers application to skin lesion classification. In *IEEE Workshop on Neural Networks for Signal Processing (NNSP'95)*, pages 484–493, 1995.
- [19] A. Kjoelen, M. J. Thomson, S. E. Umbaugh, R. H. Moss, and W. V. Stoecker. Performance of AI methods in detecting melanoma. *IEEE Engineering in Medicine and Biology*, 14(4):411–416, July/August 1995.
- [20] M. Binder, A. Steiner, K. Wolff M. Schwarz, S. Knollmayer, and H. Pehamberger. Application of an artificial neural network in epiluminescence microscopy pattern analysis of pigmented skin lesions: a pilot study. *British Journal of Dermatology*, 130:460–465, 1994.
- [21] F. Ercal, H.C. Lee, W. V. Stoecker, and R. H. Moss. Skin cancer diagnosis using hierarchical neural networks and fuzzy systems. In *Proceedings of the Artificial Neural Networks in Engineering conference (ANNIE '94)*, volume 4, pages 613–618, November 1994.
- [22] R. R. J. Bostock, E. Claridge, and A. J. Harget. Towards a neural network based system for skin cancer diagnosis. In *IEE Int. Conf. on Neural Networks*, pages 215–219, 1993.
- [23] A. Durg, W. V. Stoecker, J. P. Cookson, S. E. Umbaugh, and R. H. Moss. Identification of variegated coloring in skin tumors; neural networks vs. rule-based induction methods. *IEEE Engineering in Medicine and Biology*, 12(3):71–74, September 1993.
- [24] S. E. Umbaugh, R. H. Moss, and W. V. Stoecker. Applying artificial intelligence to the identification of variegated coloring in skin tumors. *IEEE Engineering in Medicine and Biology*, 10(4):57–62, December 1991.
- [25] D. Gutkiewicz-Krusin, M. Elbaum, P. Szwaykowski, and A. W. Kopf. Can early malignant melanoma be differentiated from atypical melanocytic nevus by *in vivo* techniques? part II. automatic machine vision classification. *Skin Research and Technology*, 3:15–22, 1997.
- [26] H. Ganster, M. Gelautz, A. Pinz, M. Binder, H. Pehamberger, M. Bammer, and J. Krocza. Initial results of automated melanoma recognition. In G. Borgefors, editor, *Proceedings of The 9th Scandinavian Conference on Image Analysis*, pages 209–218, Uppsala, Sweden, June 1995. Swedish Society for Automated Image Analysis.

- [27] P. N. Hall, E. Claridge, and J. D. Morris Smith. Computer screening for early detection of melanoma—is there a future ? *British Journal of Dermatology*, 132:325–338, 1995.
- [28] A. Green, N. Martin, J. Pfitzner, M. O'Rourke, and N. Knight. Computer image analysis in the diagnosis of melanoma. *Journal of the American Academy of Dermatology*, 31:958–964, December 1994.
- [29] J. Kontinen, J. Rönning, and R. M. MacKie. Texture features in the classification of melanocytic lesions. In A. Del Bimbo, editor, *9th International Conference on Image Analysis and Processing (ICIAP'97)*, volume 2 of *Lecture Notes in Computer Science*, pages 453–460, Florence, Italy, September 1997. Springer Verlag.
- [30] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1993.
- [31] F. Ercal, M. Moganti, W. V. Stoecker, and R. H. Moss. Detection of skin tumor boundaries in color images. *IEEE Transactions on Medical Imaging*, 12(3):624–627, September 1993.
- [32] S. E. Umbaugh, R. H. Moss, W. V. Stoecker, and G. A. Hance. Automatic color segmentation algorithms with application to skin tumor feature identification. *IEEE Engineering in Medicine and Biology*, 12(3):75–81, September 1993.
- [33] J. E. Golston, R. H. Moss, W. V., and Stoecker. Boundary detection in skin tumor images: An overall approach and a radial search algorithm. *Pattern Recognition*, 23(11):1235–1247, 1990.
- [34] S. E. Umbaugh, R. H. Moss, and W. V. Stoecker. Automatic color segmentation of images with application to detection of variegated coloring in skin tumors. *IEEE Engineering in Medicine and Biology*, 8(4):43–52, December 1989.
- [35] J. Gao, J. Zhang, M. G. Fleming, I. Pollak, and A. B. Cognetta. Segmentation of dermatoscopic images by stabilized inverse diffusion equations. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 823–827, 1998.
- [36] G. A. Hance, S. E. Umbaugh, R. H. Moss, and W. V. Stoecker. Unsupervised color image segmentation; with application to skin tumor borders. *IEEE Engineering in Medicine and Biology*, 15(1):104–111, January/February 1996.
- [37] W.E. Denton, A. W. G. Duller, and P. J. Fish. Boundary detection for skin lesion: an edge focusing algorithm. In *Image Processing and its Applications*, number 410 in Conference Publication, pages 399–402. IEEE, 4–6 July 1995.
- [38] W. Chase and F. Brown. *General statistics*. John Wiley & Sons, 1992.
- [39] W. V. Stoecker, W. W. Li, and R. H. Moss. Automatic detection of asymmetry in skin tumors. *Computerized Medical Imaging and Graphics*, 16(3):191–197, 1992.
- [40] R. Schalkoff. *Pattern recognition: statistical, structural and neural approaches*. Wiley, 1992.

Chapter 8

Conclusions

8.1 Summary

In this thesis, different image processing tasks related to image analysis have been investigated. Image filtering has been treated in Chapter 2 with an emphasis on edge and geometry preserving techniques. The goal was the preparation of an image before it undergoes a processing like segmentation or contour detection. Irrelevant information must be removed while object contours and boundaries between neighboring regions must be kept unchanged, both in location and shape. Diffusion is a powerful solution to this filtering problem and has been therefore investigated in details. This technique uses the time-varying partial differential equation that controls the diffusion of heat in matter. A formulation reduced to the divergence of the local gradient is used in image processing applications for the diffusion of pixel values. A nonlinear function, called conductivity function, weights the local gradient to control the “amount” of diffusion, which has the desirable effect of reducing and even stop the diffusion through edges with high gradients. The different types of conductivity functions have been presented: a constant value, nonlinear gradient-based functions, and tensors. The diffusion is then respectively isotropic, nonlinear isotropic, and anisotropic. For the application which require edge preserving filtering, the most complicated formulation, namely the anisotropic diffusion, is usually not necessary. Anisotropic diffusion has shown to be very interesting to control precisely the diffusion direction. however, from a computational point of view, the higher complexity of the tensor formulation is not justified by the improvements in the results, when compared to nonlinear isotropic diffusion. The latter has shown to be optimal from an efficiency/complexity point of view. Associated with the right conductivity function, it is very noise resistant, in the sense that the presence of noise has a reduced influence on the convergence of the system.

Different extensions to the basic formulation of nonlinear isotropic diffusion have been introduced for specific applications. A method for scale dependent diffusion has been defined to select the scale at which details should remain in the image. A down-sampled version of the original image, called *pilot*, is used to estimate the local gradient. This means that, depending the scale, structures are considered either as components of the image or as part of an homogeneous region. The use of time-varying conductivity functions has also been developed to allow for switching from a smoothing to an edge enhancing diffusion, which are both controlled by different conductivity functions. The application of diffusion to vector-valued data like color pixels or texture descriptors

has been shortly investigated. The combination of uniform color spaces and diffusion allows the construction of local gradient approximations based on color difference measurements. Finally, shock filters have been presented for edge enhancement applications to show their analogy with backward diffusion as well as the numerical implementation problems these approaches create.

In Chapter 3, basic concepts of colorimetry and color space transformations have been presented. The goal of this chapter was not to present new results obtained in this domain, but to give the necessary background to the development of a color segmentation scheme. The main principles and definitions have been given, as well as the major colorimetric models introduced by the *Commission Internationale de l'Eclairage (CIE)*. The real stimuli *RGB* and the imaginary stimuli *XYZ* have been defined for the 1931 and 1964 standard colorimetric observers. These models have been established by the CIE to integrate the properties of a standard observer into the different color models through color matching experiments. The way to compute the reference white corresponding to different types of illuminations has been introduced and exploited in nonlinear color spaces. Linear colorimetric transformations have been introduced first. They orient the color space along directions with specific characteristics, and to provide easy ways to compute approximations to components like luminance. Nonlinear transformations have then been presented, with a particular emphasis on uniform color spaces, where color difference can be measured using Euclidean distance. It has also been shown that a principal component decomposition does not affect the uniformity of these color spaces, which is a new contribution to this topic.

Image segmentation by color clustering has been investigated in Chapter 4. Among different strategies, the clustering approach has been chosen because it can be applied to any type of data, and can be used for several applications, from data classification to segmentation problems. The discussion started with a short description of probability density estimation methods that may be used to build histograms and then to match the result to a particular model. The *expectation-maximization* (EM) algorithm has been derived to this end, assuming that the data distribution is a mixture of Gaussian functions. This technique can be compared to the *orientation sensitive fuzzy c-means* (OS-FCM) clustering technique that has been developed later in this chapter.

The clustering of data, which aims at the computation of cluster centroids and the partitioning of the data into several classes, has been first presented through classical approaches such as the *basic ISODATA* and *Learning Vector Quantization* (LVQ) techniques. The basic algebra of fuzzy sets and the notion of fuzzy partition space have been presented next to introduce the *fuzzy c-means* clustering technique. In this approach, the distances computed from the data to a particular class centroid are always divided by the distance to the other class centroids. By doing this, one introduces a kind of cluster density parameter, giving thus more weight to isolated clusters. While the fuzzy c-means approach is very powerful to compute cluster centers, it gives poor results in the data partitioning.

The introduction of orientation sensitive fuzzy c-means (OS-FCM) solves the partition problem, especially for Gaussian-like and elliptic shaped distributions. For every cluster a *fuzzy covariance matrix* is computed and used to compute the distance to the data. To conclude this part on fuzzy clustering techniques, the use of a noise class has been proposed which is intended to attract outsiders that may corrupt the clustering process. A geometrical approach to compute the number of classes and their approximate location, needed to initialize the different clustering techniques, has been developed. The last topic in this chapter concerns the removal of small regions using morphological operators.

The contour detection problem has been treated in Chapter 5. Gradient-based approaches have been presented first to introduce the subject. Improvements that can be achieved when combining nonlinear isotropic diffusion with a simple Laplacian operator and the Sobel approximation to derivatives have been illustrated. Canny's edge detector has been presented as well, as it constitutes an elegant analytical approach to the edge detection problem. Since all these edge detectors must be combined with a threshold to avoid the detection of every small transition in the image, closed contours and object boundaries can not be detected efficiently. Therefore, active contour models have been investigated, such as snakes, curve evolution, and geodesic active contours. A numerical approach to curve evolution called *level sets* has been presented. This technique allows one to control an evolving curve with a two-dimensional diffusion formulation, and the curve is then tracked without caring about changes in topology (merging, splitting, corners, etc.). Finally, a new contour detection scheme for color images is presented, which combines multi-frame nonlinear isotropic diffusion, morphological flooding for iso-level contour extraction, and an energy functional to select low-energy contours. Along these contours, the gradient is supposed to have a high mean value and a low variance.

The last theoretical developments concerned symmetry quantification and symmetry axes detection, and have been presented in Chapter 6. This research topic has been quickly reviewed, and a particular approach to symmetry, namely the principal component decomposition, has been presented in more details. The different formulations, taking binary, gray-level, or spatial frequencies as input, have been derived. They suppose that there is a strong correlation between the orientation and a possible symmetry axis. A new approach to symmetry quantification has been presented next, which assumes that a particular symmetry measure must be optimized, i.e. the axis that maximizes the symmetry measure must be computed. Two approaches have been developed to get quickly the result: a genetic algorithm and a scheme derived from self-organizing maps. These two approaches have shown to be robust techniques in the sense that they are able to reproduce results, which is not straightforward for algorithms based on competition. The notion of symmetry map has been introduced, which is a two-dimensional plot of the symmetry measure for two axis parameters, an angle and an offset. This allows the analysis of objects in their "symmetry domain" and to obtain all local maxima and their respective importance and location.

An important aspect of the different techniques that have been developed in Chapter 6 is that they all work with vector-valued data, which means that the input can be a one-component image, such as binary and gray-level images, or a multi-components image such as color images or texture descriptors. The goal is to separate the different information levels which are shape, color, and texture, and to associate with each one a symmetry value. When the symmetry is perfect, these three components match. This is not the case when objects have no real symmetry. Finally, the use of texture descriptors obtained through a filter bank build of isotropic Gaussian band-pass filters has been illustrated on synthetic images in noisy environment, and compared to a principal component approach.

The different techniques presented in Chapters 2 to 6 have been applied to dermatoscopic images in Chapter 7. These images have been obtained through epiluminescence microscopy and are used for the diagnosis of skin cancer. Image processing techniques are needed to separate the lesion and the surrounding healthy skin, and to quantify a number of diagnostic features. The segmentation and contour detection techniques that have been developed respectively in Chapter 4 and Chapter 5 have been illustrated and compared to contour results drawn by hand by experienced physicians. We could show that a unique dermatologist can not be used as an absolute reference

to evaluate the automatic results, but that a group statistics can be established to measure the deviation of any result from those of the group of experts. This measure has been done for the automatic results and for several physicians, and the results have shown that the within-group error obtained in the automatic case is lower than that obtained for the experts.

The symmetry quantification has been evaluated on two sets of 50 benign and 50 malignant lesions. Six symmetry features have been computed (for shape, color, and texture symmetry) by keeping each time the two highest symmetry values in the different symmetry maps. A linear classifier has been used to evaluate the separation between the two classes of lesions. The results have shown that this approach clearly outperforms the principal component approach commonly used.

8.2 Achievements

The different achievements have already been listed in the introduction. In this section we will take them one by one and describe them in more details.

The first achievement is the analysis of diffusion processes and their behavior in a noisy environment. We have given a complete description of these techniques, starting with the isotropic diffusion. This approach is seldom analyzed and often neglected. We have derived the transfer function corresponding to this linear filter and have shown that it can be implemented with a simple 3×3 coefficients filter. This approach is especially interesting for multi-resolution filtering and narrow band low-pass filtering. The nonlinear formulation of isotropic diffusion is presented along with different conductivity functions, and we have shown that one of them has a much better convergence in presence of noise. Finally a number of extensions including the use of a *diffusion pilot* for scale dependent diffusion, time varying conductivity functions for edge smoothing/enhancement combinations, and the application to multi-components images have been introduced.

A new fuzzy clustering technique called *orientation sensitive fuzzy c-means* has been described. It is a powerful extension to the fuzzy c-means clustering technique which integrates the orientation of the different clusters into the different computation steps. This technique improves both the cluster center computation and data partitioning in cases where the different clusters have elliptic shapes.

A new color segmentation scheme based on the analysis of a two-dimensional color histogram and data clustering has been introduced. This scheme integrates the previously developed orientation sensitive fuzzy c-means clustering technique and a geometrical approach to the detection of valid maxima in the histogram. The different steps are given in Section 4.7.

A new contour detection scheme has been presented next. The novelty of this scheme is the introduction of morphological flooding which allows for the extraction of iso-level contours. The use of multi-components nonlinear isotropic diffusion of color images is also introduced and computed first in the contour detection scheme. Finally, the use of an energy functional to select valid contours that takes into account both the mean gradient and its variance along the different contours is also a new contribution.

The major contribution in the domain of symmetry quantification and symmetry axes detection is the optimization approach. We have designed an efficient genetic algorithm to solve this problem, and have shown that self-organizing maps can be used in optimization problems. An important contribution is finally the use of symmetry maps to describe and analyze objects. This transform

gives all the information for a unique criteria.

The notion of symmetry map being new, the techniques we have designed to speed up the process are new as well. Especially the use of pyramidal image representations is a powerful way to improve computation speed.

The application to skin cancer diagnosis is part of a complete computer aided diagnosis scheme. We have proposed a complete and new scheme that integrates clinical approaches to skin cancer detection, imaging techniques, database management, etc.

A solution to the presence of hair in dermatoscopic images has been developed and presented. Simple morphological operators combined with the use of luminance information allows the detection of these artifacts and to replace the concerned pixels with interpolated values.

Finally, new results have been obtained in the framework of computerized skin cancer diagnosis. The two segmentation and contour detection schemes have been validated for this application, and the potential of our symmetry quantification scheme has been demonstrated.

8.3 Possible extensions

The different chapters of this thesis propose complete solutions to some problems. Extensions are possible and desirable in the analysis of skin lesions. While the lesion detection problem has been solved, there is still much to do for the extraction of diagnostic features. In this study we focused on a unique criteria, namely symmetry, to exploit it fully for classification applications. We know from skin cancer research done by physicians that a unique feature is not sufficient to diagnose precisely skin cancer, and that the combination of different criteria is the clue to the early detection of malignant melanoma and other types of skin cancer.

For the symmetry quantification, a deeper analysis of symmetry maps is necessary, and we believe that the classification into benign and malignant lesions can be improved even more. The different maxima in the symmetry map are important feature points, and their respective importance and location must be analyzed and measured. We could observe that benign lesions have less local maxima than malignant lesions, and that the shape, color, and texture maps have similarly distributed maxima in the former case.

Among the different criteria that are used for the clinical diagnosis, one of the most interesting and promising is certainly the detection of specific patterns in dermatoscopic images. This task, which is also the most difficult one, should allow the detection of different pigmented structures and to allow their analysis in terms of regularity, size, etc. Nobody has really started this challenging work yet, and it is desirable that the researcher who will continue this work orients his investigations in this direction.

The design of an efficient computer aided diagnosis system is worth nothing without an efficient user interface. It will be very quickly necessary to put all the different image processing blocks together, and to provide a graphical interface to use them and to display the results. Network facilities and database management are also very important issues that must be investigated in order to build a complete system. In the framework of this research, we could already experience the design of client/server applications in Java, where the idea was to have the graphical interface on the user side, and the database and image processing routines on the server side. This approach seems to be the most promising. Another approach, which uses Matlab as platform, has been evaluated as well. In that case, the different image processing routines can be called from Matlab, and

the results are displayed using Matlab's graphical user interface tools. This solution, however, has the disadvantage of requiring a good knowledge of Matlab and not integrating database facilities.

Finally, the application of the segmentation, contour detection, and symmetry quantification schemes to other types of images would be a very interesting extension. We believe that they may provide powerful tools to solve other types of problems, especially in other medical image analysis applications, such as ultra-sound and magnetic resonance imaging.

Curriculum Vitae

<i>Full name:</i>	Philippe SCHMID-Saugeon
<i>Degrees:</i>	Electrical engineer (ing. él. dipl. EPF) Ph.D.
<i>Current position:</i>	Independent consultant.
<i>Contact numbers:</i>	E-mail: phschmid@schmid-saugeon.com www: http://www.schmid-saugeon.com

Personal data:

<i>Civil status:</i>	married
<i>Date and place of birth:</i>	24th May 1969, Lausanne, Switzerland
<i>Nationality:</i>	Swiss

Education and degrees:

<i>1985-1988</i>	High School: gymnase cantonal de la Cité, Lausanne, Switzerland.
<i>1988-1990</i>	Military Service. Rank: first lieutenant.
<i>1990-1991</i>	University: physics, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
<i>1991-1996</i>	University: electrical engineering, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
<i>1996</i>	Electrical engineering degree (in electronics, analog and digital integrated circuits, antenna design, microwaves, filter design and signal processing): Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
<i>1999</i>	Ph.D. degree in electrical engineering: Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Thesis title: "Segmentation and symmetry measure for image analysis: application to digital dermatoscopy".

Past and present activities:

<i>1993-1996</i>	Member of the teaching committee of the department of electrical engineering, student delegate.
<i>1996-1997</i>	Swiss National Science Foundation proposal entitled "Development and validation of image processing and analysis tools for the quantitative assessment of melanocytic skin lesions". Accepted April 1998 (fund no. 3252-053175).
<i>1996-1999</i>	Research assistant and Ph.D. candidate at the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, under the direction of Prof. Dr. Murat Kunt.
<i>1999-2000</i>	Postdoctoral position at the Video and Image Processing Laboratory, University of California at Berkeley (UCB), USA.

Languages:

<i>French:</i>	native language
<i>German:</i>	second native language
<i>English:</i>	fluent

Programming skills:

<i>Operating systems:</i>	UNIX, Linux, MacOS, and Windows
<i>Programming languages:</i>	Pascal, C, C++, and Java
<i>Design software:</i>	Visual C++
<i>Other languages:</i>	HTML, Perl, Matlab, and LaTeX

Experiences:

From May 1996 to September 1999 I have been a research assistant and Ph.D. candidate at the Signal Processing Laboratory of the Swiss Federal Institute of Technology at Lausanne, Switzerland. During this period I have been an active researcher in the image analysis field. The main results have been obtained in image segmentation and contour detection for color images, and in symmetry quantification for almost and non-symmetrical objects based on color, texture, and shape analysis. These three topics have been widely investigated in order to meet time, precision, and simplicity requirements. The main application domain was the processing of dermoscopic images, obtained by epiluminescence microscopy and used for the diagnosis of skin cancer. In addition to these research fields, I got familiar with web page design, graphical user interface design with Matlab and Java, common gateway interface (CGI) scripts for client/server applications. Beside the image analysis domain I worked for the cross-checking of core experiments in the framework of the MPEG-4 standard. This gave me the opportunity to get familiar with modern image compression schemes. Finally, I had to supervise several undergraduate student projects.

Research interests:

All application domains of image analysis and computer vision, like medical image analysis, robot vision, quality control, guiding systems, aerial and spatial image analysis, and much other, are attractive to me. My interests cover various domains like medical imaging, computer vision, information technologies, networking, and computer science.

Objectives:

My future activity should include both research and project management. The work within multidisciplinary research teams and the contact with people of different origin are essential parameters for my future career because they constitute a stimulating environment. I expect new challenges that will lead to the development of new technologies, which, in turn, will give better access to the growing world of information.

Publications:

- [1] Ph. Schmid, "Segmentation of Dermatoscopic Images by 2D Color Clustering", *IEEE Transactions on Medical Imaging*, vol. 18, no. 2, pp. 164-171, February 1999.
- [2] Ph. Schmid. "Symmetry Axis Computation for Almost-Symmetrical and Asymmetrical Objects: Application to Pigmented Skin Lesions". Medical Image Processing. Submitted paper.
- [3] Ph. Schmid. "Towards a Computer Aided Diagnosis System for Skin Cancer". *IEEE Transactions on Biomedical Engineering*. Submitted paper.
- [4] J. Guilloid and Ph. Schmid. "Validation of Segmentation Techniques for Digital Dermoscopy". *Journal of Investigative Dermatology*. Submitted paper.
- [5] J. Guilloid and Ph. Schmid. "Dermatoscopy". In P. Agache, editor, *Explorations fonctionnelles cutanées: théorie et pratique*. To be published.

- [6] S. Fischer, Ph. Schmid, and J. Guilloid. "Analysis of Skin Lesions with Pigmented Networks". *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 323-326, September 1996.
- [7] J. Guilloid, S. Fischer, Ph. Schmid, D. Salomon, and J.-H. Saurat. "Analyse d'images dermatoscopiques digitalisées". *Proceedings of the 78th Annual Meeting of the Swiss Society of Dermatology and Venereology*, September 1996.
- [8] Ph. Schmid. "Symmetry Measure for the Computer Aided Diagnosis of Pigmented Skin Lesions". *Proceedings of the 11th International Symposium in Computer Assisted Radiology and Surgery*, p. 998, June 1997.
- [9] Ph. Schmid and S. Fischer. "Colour Segmentation for the Analysis of Pigmented Skin Lesions". *Proceedings of the Sixth International Conference on Image Processing and its Applications*, vol. 2, pp. 688-692, July 1997.
- [10] Ph. Schmid and J. Guilloid. "Traitement numérique des images appliqué à la dermatoscopie". *Proceedings of the 80th Annual Meeting of the Swiss Society of Dermatology and Venereology*, September 1998.
- [11] Ph. Schmid. "Lesion Detection in Dermatoscopic Images Using Anisotropic Diffusion and Morphological Flooding". Accepted for publication at ICIP'99.

- [12] Ph. Schmid, "Intra-frame and inter-frame coding results of core experiment T8: Variable-size Lapped Transform Coding". Report No. 96.11, Signal Processing Laboratory of EPFL, 1996.
- [13] Ph. Schmid and G. Menegaz, "Results of CE X1 for efficient coding of homogeneous textures". MPEG-4 report No. m1775, 1997.
- [14] Ph. Schmid, "Digitised Images of Pigmented Skin Lesions: Colour Segmentation using Histogram Information and The Fuzzy c-means Technique". Report No. 97.01, Signal Processing Laboratory of EPFL, 1997.
- [15] Ph. Schmid and G. Menegaz, "Results of CE Z1 for efficient coding of textures". MPEG-4 report No. m1915, 1997.
- [16] Ph. Schmid, "Symmetry Measures for Pigmented Skin Lesions". Report No. 97.05, Signal Processing Laboratory of EPFL, 1997.
- [17] Ph. Schmid, "GUI for the colour segmentation of pigmented skin lesions". Report No. 97.11, Signal Processing Laboratory of EPFL, 1997.