

Exploring the deep structure of images

Citation for published version (APA):

Platel, B. (2007). *Exploring the deep structure of images*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Biomedical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR617003>

DOI:

[10.6100/IR617003](https://doi.org/10.6100/IR617003)

Document status and date:

Published: 01/01/2007

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

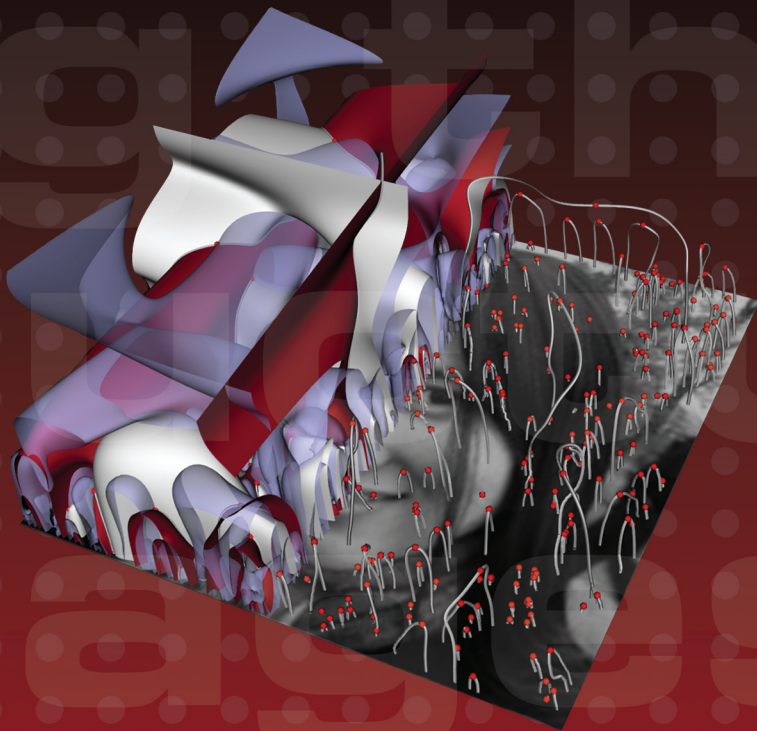
If you believe that this document breaches copyright please contact us at:

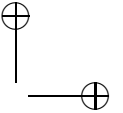
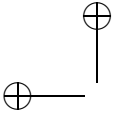
openaccess@tue.nl

providing details and we will investigate your claim.

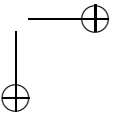
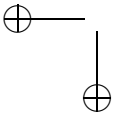
Exploring the Deep Structure of Images

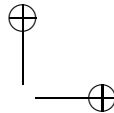
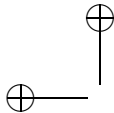
Bram Platel





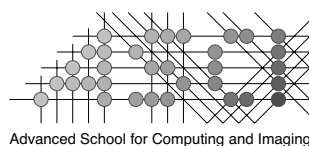
Exploring the Deep Structure of Images





Colophon

This thesis was typeset by the author using $\text{\LaTeX}2_{\epsilon}$. The main body of the text was set using a 11-points Times Roman font. Figures were obtained using Mathematica (a product of Wolfram Research Inc.) Images were included formatted as Encapsulated Postscript (registered trademark of Adobe Systems Inc.). The output was converted to PDF and transferred to film for printing.



This work was carried out in the ASCI graduate school. ASCI dissertation series number 134.

This work was part of the DSSCV project supported by the IST Program of the European Union (IST-2001-35443).

Financial support for the publication of this thesis was kindly provided by the Advanced School for Computing and Imaging (ASCI), and the Technische Universiteit Eindhoven.

The cover has been designed by Oranje Vormgevers, Eindhoven, the Netherlands. The image on the cover represents the critical paths, top-points and zero-crossing surfaces of a portrait. This image was created with ScaleSpaceViz, a program of Frans Kanters.

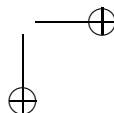
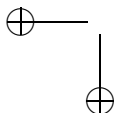
Printed by PrintPartners Ipskamp, Enschede, the Netherlands

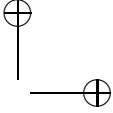
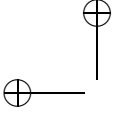
A catalogue record is available from the Library Eindhoven University of Technology

ISBN-10: 90-386-2659-2

ISBN-13: 978-90-386-2659-8

© 2007 B. Platel, Eindhoven, The Netherlands, unless stated otherwise on chapter front pages, all rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the copyright owner.





Exploring the Deep Structure of Images

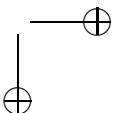
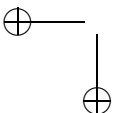
PROEFSCHRIFT

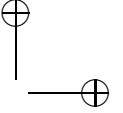
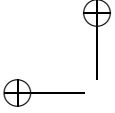
ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 27 februari 2007 om 14.00 uur

door

Bram Platel

geboren te 's-Hertogenbosch



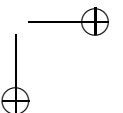
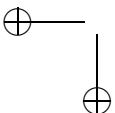


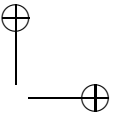
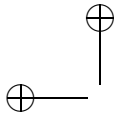
Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. B.M. ter Haar Romeny

Copromotor:

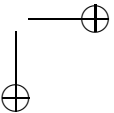
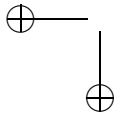
dr. L.M.J. Florack

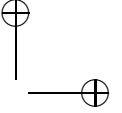
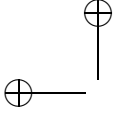




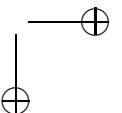
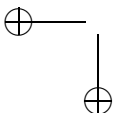
Contents

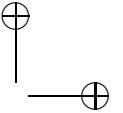
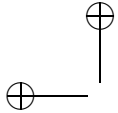
Colophon	ii
Contents	v
1 Introduction and Summary	1
1.1 Introduction	2
1.2 The DSSCV Project	2
1.3 Summary of this Thesis	3
2 Scale Space & The Deep Structure of Images	7
2.1 Introduction	8
2.2 Linear Scale-Space Theory	8
2.2.1 Axiomatic Foundations of Linear Scale Space	9
2.2.2 Generating a Scale-Space Representation	11
2.2.3 Separability	12
2.2.4 α Scale Spaces	13
2.2.5 Sampling the Scale Axis	14
2.2.6 Scale-Space Derivatives	14
2.3 Multi-Scale vs. Multi-Resolution	16
2.4 Catastrophe Theory in Scale-Space	17
2.4.1 Critical Points	18
2.4.2 Critical Curves	18
2.4.3 Detection of Critical Points and Critical Curves	21
2.4.4 Refining the Location of Critical Points	27
2.5 Conclusion	28
3 The Hierarchical Structure of Scale-Space Images	29
3.1 Introduction	30
3.2 Hierarchical Segmentation from Intensity Extrema	30
3.2.1 Extremum Paths and Extremal Regions	30
3.2.2 Scale-Space Hierarchy	32
3.2.3 Remarks	33
3.3 Hierarchical Segmentation from Scale-Space Saddles	35
3.3.1 Scale-Space Saddles	35
3.3.2 Pre-Segmentation	36
3.3.3 Scale-Space Hierarchy	37
3.3.4 Remarks	38
3.4 Hierarchical Segmentation from Attraction Areas	39
3.4.1 The Definition of the Attraction Area	39
3.4.2 Attraction Areas in Scale Space	41
3.4.3 Scale-Space Hierarchy	42



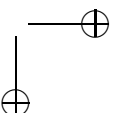
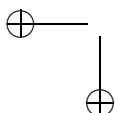


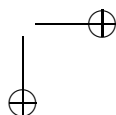
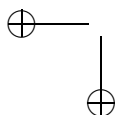
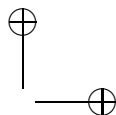
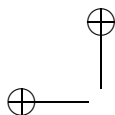
3.4.4	The Algorithm	42
3.4.5	Results	43
3.4.6	Remarks	44
3.5	Other Hierarchical Segmentation Methods	45
3.6	Using the Hierarchical Representation for Matching	45
3.7	Conclusion	46
4	Discrete Representation of Top-Points via Scale Space Tessellation	49
4.1	Introduction	50
4.2	Top-Points as Anchor Points	51
4.3	Stability by Structure	52
4.4	Construction of the Graph	54
4.5	Distance Measure in Scale Space	56
4.6	Overview of the Matching Algorithm	58
4.6.1	Metric Embedding of a Graph	59
4.6.2	Many-to-Many Distribution Based Matching	60
4.7	Experiments	62
4.7.1	Graph Stability under Additive Noise	62
4.7.2	Graph Stability under Within-Class Variation	63
4.8	Conclusions	64
5	Interest Points for Image Matching	69
5.1	Introduction	70
5.2	Harris Detector	70
5.2.1	Harris-Laplace Detector	72
5.2.2	Stability	74
5.2.3	Remarks	74
5.3	SIFT interest-point detector	75
5.3.1	Detection versus Localization	76
5.3.2	Stability	77
5.3.3	Remarks	78
5.4	Top-Points	79
5.4.1	Detection versus Localization	79
5.4.2	Propagation of Errors in Scale Space	82
5.4.3	Noise Propagation for Top-Point Displacement	82
5.4.4	Stability	84
5.5	Experiments	84
5.5.1	Database	84
5.5.2	Repeatability	85
5.6	Summary and Conclusions	87

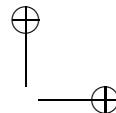
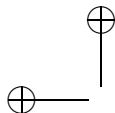




6	Neighborhood Descriptors for Image Matching	89
6.1	Introduction	90
6.2	SIFT Descriptor	90
6.2.1	Descriptor Orientation	90
6.2.2	Descriptor Representation	92
6.2.3	Adaptations of the SIFT Algorithm	94
6.2.4	Distance Measure	94
6.2.5	Remarks	95
6.3	Differential Invariant Descriptors	95
6.3.1	Descriptor Representation	96
6.3.2	Dissimilarity Measure in the Descriptor Space	97
6.3.3	Remarks	98
6.4	Experiments	98
6.4.1	Receiver Operator Characteristics	98
6.4.2	Performance of the Dissimilarity Measure	99
6.4.3	Performance of the Descriptors	100
6.5	Summary and Conclusions	100
7	Object Location and Pose Retrieval	105
7.1	Introduction	106
7.2	Matching Features	106
7.3	Obtaining Pose Coordinates	109
7.4	Clustering in Pose Space	111
7.4.1	Identifying the Number of Clusters	111
7.4.2	Expectation Maximization	112
7.4.3	Normalizing the Pose Space	114
7.4.4	Weighted k -Medians Clustering	115
7.5	Retrieving the Pose	117
7.6	Retrieval examples	119
7.7	Summary and Conclusion	124
	Bibliography	127
	Samenvatting	139
	Dankwoord	143
	Curriculum Vitae	145
	Publications	147





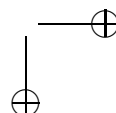
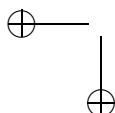


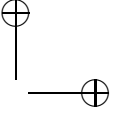
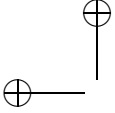
Do not believe in anything simply because you have heard it. Do not believe in anything simply because it is spoken and rumored by many. Do not believe in anything simply because it is found written in your religious books. Do not believe in anything merely on the authority of your teachers and elders. Do not believe in traditions because they have been handed down for many generations. But after observation and analysis, when you find that anything agrees with reason and is conducive to the good and benefit of one and all, then accept it and live up to it.

Buddha

1

Introduction





1.1 Introduction

Typical digital cameras nowadays have resolutions of millions of pixels. Medical imaging devices like CT and MRI scanners produce 3D data sets of hundreds of millions of pixels. The consumer market thrives with new digital HDTV 1080p videos of 1920×1080 pixels at 60 complete frames per second. These examples show that the number of digital images in the world are sheer endless. This still vastly growing amount of digital images results in a big demand for computer algorithms to analyze all this digital information.

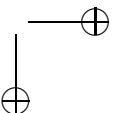
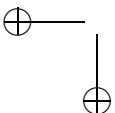
Computer vision is concerned with computer processing of images from the real world. One could for example think of algorithms that detect certain objects in images or of algorithms that assign an object class to an image. A specific example of this would be a computer algorithm that searches through a large database of images to find all images containing a certain object. Nowadays, internet image search engines like e.g. Google Images or Picsearch, use annotations found in the text surrounding an image or in the image file name, as employable search terms in their algorithms. Thus their success rate hinges on non-image attributes rather than image content. It would however be much more efficient and useful if the computer itself could determine which objects are contained in the images. With such an algorithm it would even be possible not to use specific search terms, but to hand the computer an image of an object and ask it to return a number of images containing similar objects.

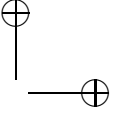
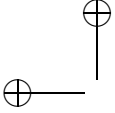
These examples might seem to be trivial tasks at first, but one should keep in mind that the innate human visual system has evolved over millions of years into the most intricate sensory system. On top of these innate recognition capabilities, the visual system is subject to a life long learning. Where we humans easily recognize objects in images, intuitively ignore partial object occlusion or noise and have little problem with identifying objects in different poses, a computer will need a set of intelligent algorithms for its input is merely a record of numbers.

1.2 The DSSCV Project

The work in this thesis has been part of the ‘Deep Structure, Singularities, and Computer Vision’ (DSSCV) project, supported by the IST Program of the European Union fifth framework program (contract IST-2001-35443).

The members of this project consisted of several experts from four European universities, the IT University of Copenhagen (Denmark), the University of Liverpool (United Kingdom), the University of Copenhagen (Denmark) and from the





Eindhoven University of Technology (the Netherlands). The project started on October 1st 2002 and had a duration of 36 months.

The DSSCV consortium was brought together to develop sophisticated representations of images and shapes by syncretizing principles and methods from scale-space theory, singularity theory and algorithmics. The consortium's main goal has been to create new fundamental insights in image understanding as well as to develop elegant, robust and effective algorithms for solving computer vision tasks.

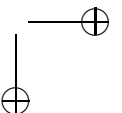
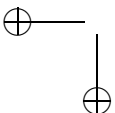
The work in the consortium has lead to the first 'International Workshop on Deep Structure, Singularities and Computer Vision' which took place in Maastricht in 2005 [107].

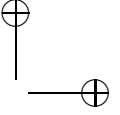
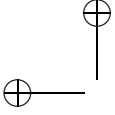
1.3 Summary of this Thesis

This thesis will focus mainly on techniques for automatic image matching. The results from the study are applicable in image analysis applications such as wide baseline matching, texture recognition, object retrieval, robot localization, video data mining, building panorama's object class recognition and object location and pose retrieval. Over the years many different methods have been developed for these tasks and a number of these methods function well in specific areas of image analysis. However a trend in these methods is that they depend more and more on ad hoc parameters and thresholds. These parameters are often not representing a physical entity and it is therefore difficult to assign meaningful values to them. The approach often used to 'solve' this problem is extensive training that exhaustively tries all possible parameters within a certain range and simply choses the parameter set yielding the best results for the training set of images. This results in algorithms that work well for the image modality for which they were trained, but as soon as the input images differ too much from the training set, these algorithms break down.

The goal of the research described in this thesis is to find a general method for image matching, regardless of the type of input images. Our method has to have a reasonable physical motivation and should have as few parameters and thresholds as possible.

To come up with this algorithm old methods are revisited that are rooted in scale-space theory, physics and mathematics. These form the building blocks of our new image matching algorithm. The new algorithm minimizes the amount of parameters and thresholds. Thresholds that will remain have a physical and thus understandable meaning and can be set according to the task at hand.





This thesis is based on a number of published articles. These will be mentioned at the title page of each chapter.

The following shortly summarizes the contents of this thesis.

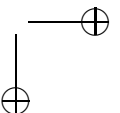
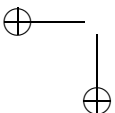
Chapter 2 reviews the principles of scale-space theory and the related deep structure of images. Essential for a general image analysis algorithm is scale invariance. Objects and details exist as meaningful entities only over certain ranges of scale and since it is not known a priori which scale to look at, all scales have to be considered as equally important. Essential operations as blurring and taking derivatives at certain scales are discussed and implementation examples are given. Catastrophe theory is reviewed for the scale-space case. It studies how critical points move as scale is changed, which is essential for each of the proposed algorithms in this thesis. Interest points, in particular top-points, which will be used in the image matching algorithms, are derived directly from catastrophe theory.

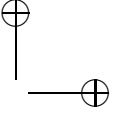
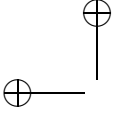
Chapter 3 discusses a number of methods for generating a hierarchical structure from an image. The methods generate a general image segmentation, which does not incorporate any prior knowledge about the image. To overcome the problems that occur in the existing methods a new method is proposed. The DSSCV project conjectured that the hierarchical representation of the deep structure of an image could be used for image description and matching purposes. In this chapter however reasons are given why the proposed hierarchical representations are unfeasible for the task of image matching or object recognition.

Chapter 4 suggests a new representation to overcome the problems that are inherent in the hierarchical representations from chapter 3. The suggested algorithm encodes the scale-space structure of top-points, resulting from catastrophe theory, in a directed acyclic graph. This new representation does allow for the utilization of powerful graph matching algorithms to compare images represented in terms of top-point configurations, rather than using point matching algorithms to compare sets of isolated interest points.

However, the new representation suffers from another artifact, viz. its unpredictable behavior under non-scale-Euclidean transformations like affine or projective transformation. For this reason subsequent chapters do not pursue this algorithm any further, but instead elaborate on an alternative approach based on interest points.

Chapter 5 discusses interest points suitable for image matching. Interest points are characteristic points derived from the image. They should be as invariant as possible under changes in the image. Consider for example the task of locating an object in a scene image (an image containing, among other things, the object).

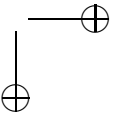
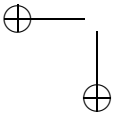


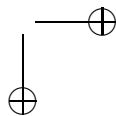
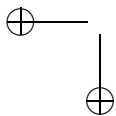
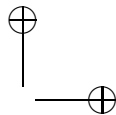
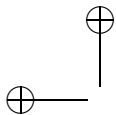


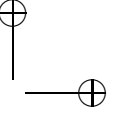
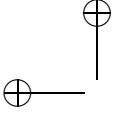
It is important that the interest points in the query image of the object are found in a similar configuration in the scene image. The configuration of interest points should remain similar even if the object in the scene image has been rotated, scaled, exposed to different lighting conditions or affected by some noise. In this chapter the popular Harris, Harris-Laplace and SIFT interest point detectors are discussed and compared to the interest points suggested in this thesis based on catastrophe theory: the top-points. The existing methods are dependent on a large amount of tunable ad hoc parameters. For different types of images these parameters have to be trained on new training sets. The introduced top-points do not have these tunable parameters and yet yield competitive results in a set of repeatability experiments. Top-point localization does not have to be very accurate, since it is possible to refine the localization using local differential image structure. This enables fast detection, without losing the exact location of the top-point. Another beneficial property of the top-points is the ability to analytically predict the sensitivity of each top-point to additive noise. This ability can be used to eliminate unstable interest points before further processing is done.

Chapter 6 describes neighborhood descriptors. For image description and thus image matching purposes, a set of interest points is insufficient. Distinct information has to be added to these points to distinguish them from others. The extra information assigned to each interest point is referred to as a descriptor. Such a descriptor contains information about the local neighborhood of an interest point. In this chapter the popular SIFT descriptor will be discussed and compared to the more mathematically founded differential invariant descriptor. The differential invariant descriptor is based on scale-Euclidean invariant combinations of derivatives. This means that rotation, scaling and translation will not affect the descriptor. This descriptor was deemed to perform poorly in a set of experiments by others, but in this chapter it is shown that when using a proper distance measure between a differential invariant descriptor with only 6 degrees of freedom (d.o.f.), it will even outperform the 128-d.o.f. SIFT descriptor under various circumstances.

Chapter 7 presents the last step in the chain of the image matching algorithm based on interest points and descriptors. It describes the specific problem of object location and pose retrieval. Given a query image of an object the algorithm will try to find the location and pose of this object in a scene image. To do so a point set in a 4-dimensional pose space is constructed, representing the translation, rotation and scaling the query object has undergone in the scene. This point set is given by the matches obtained from the comparison of interest points and descriptors of the object with interest points and descriptors of the scene. From this pose space one or more clusters are obtained, depending on the number of times the object is found in the scene. From these clusters and the features contained in these clusters, the pose of the object in the scene is estimated based on affine or projective transformations.



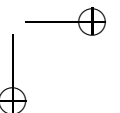
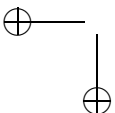


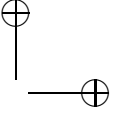
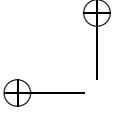


The challenge is to understand the image really on all these levels simultaneously, and not as an unrelated set of derived images at different levels of blurring.

Koenderink (1984)

Scale Space & The Deep Structure of Images





2.1 Introduction

An inherent property of objects in the world and details in images is that they only exist as meaningful entities over certain ranges of scale.

A simple example that is often used to clarify the notion of scale is the concept of a branch on a tree. The branch makes sense from a scale of a few centimeters to a few meters at most. It would be meaningless to discuss the concept of a branch at nanometer or kilometer level. At those scales it is more relevant to talk about the molecules that form the tree, or a forest in which the tree grows.

The fact that objects appear in different ways depending on the scale of observation (e.g. Figure 2.1) implies that the concept of scale as well as the notion of a multi-scale representation are of crucial importance when describing them.

The same holds for image analysis tasks. An image is a physical observable, with an *inner scale* limited by the resolution of the sampling device and an *outer scale* limited by the field of view. Since it is not known a priori at which scale to observe an object or feature in an image, a multi-scale representation is of crucial importance. As Koenderink states in one of the first scale-space papers: “The challenge is to understand the image really on all these levels simultaneously, and not as an unrelated set of derived images at different levels of blurring”.

The goal of this chapter is to review some fundamental results concerning the theory for multi-scale representation called *scale-space theory* and the related *deep structure* of images.

2.2 Linear Scale-Space Theory

Gaussian scale-space was proposed by Iijima [45] in the context of pattern recognition in 1962. Outside of the Japanese scientific community his work went largely unnoticed for a couple of decades, mostly due to the fact that the papers were in Japanese. In western literature scale-space was introduced in 1983 by Witkin [137], in a paper that discussed the blurring properties of one-dimensional signals. The extension to more dimensional signals (e.g. images) was made by Koenderink [63] in 1984.

Since Koenderink’s first seminal paper in 1984 a lot of work has been done in the scale-space community. The interested reader might have a look at some of the following fundamental scale-space books and papers.

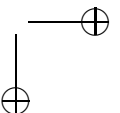
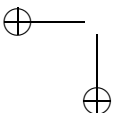


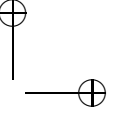
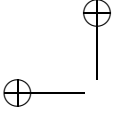


Figure 2.1: The painting “*Gala Contemplating the Mediterranean Sea which at Twenty Meters becomes a Portrait of Abraham Lincoln, Homage to Rothko*”, by Salvador Dali, 1976, illustrates that the scale at which an image is viewed can change the object that is observed.

- The papers by Weickert, Ishikawa and Imija [133, 134] present a review on the early developments in Gaussian scale-space regarding the early Japanese scale-space paper.
- The first book completely dedicated to scale-space was written by Lindeberg [87] in 1994.
- A tutorial introductory book about multi-scale image analysis has been written by ter Haar Romeny [37].
- Florack [24] gives a mathematically rigorous and complete account of image structure.
- A rather complete collection of articles on scale-space theory can be found in the edited volume [128], resulting from the Gaussian scale-space theory workshop in 1996.
- The proceedings of the scale-space conference held every second year starting from 1997 contain many interesting papers on scale-space theory [39, 103, 60, 36, 62].

2.2.1 Axiomatic Foundations of Linear Scale Space

Since the pioneering work by Witkin and Koenderink, a large number of different scale-space formulations have been stated. Most of these formulations are based



on a set of assumptions, that are usually referred to as *scale-space axioms*. An overview of the different scale-space formulations can be found in [88].

In essence these axioms are a mathematical formulation for uncommittedness; stating that there is no prior knowledge and no preference of how to treat a signal or image. From these requirements four basic constraints follow.

- *Linearity*, stating that there is no preferred way to combine observations. Non-linearities would involve some kind of knowledge or memory in a system.
- *Spatial shift invariance*, meaning that there is no preferred location in the image. Any location should be measured in the same fashion.
- *Spatial isotropy*, indicating that there is no preferred orientation.
- *Spatial scale-invariance*, implying that there is no preferred size. Any size of structure, objects, textures, etc. is just as likely as any other.

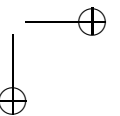
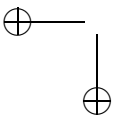
From the axioms of linearity and spatial shift invariance it follows that a blurred version of a signal has to be a convolution of the signal with an aperture function.

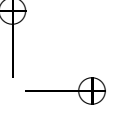
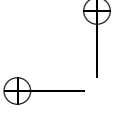
The derivation of the Gaussian scale space has been accomplished in many ways. Some examples of ways to derive the Gaussian scale space are given here. The exact and complete derivation will not be explained in this thesis, for this the reader is referred to one of the following references.

- *Dimensional analysis*, a formulation by Florack [28], and continued work by Pauwels [110] shows that the class of allowable scale-space kernels can be restricted under weak conditions by combining the earlier mentioned conditions of linearity, shift invariance, rotational invariance and semi-group structure with scale invariance. From physics this is known as dimensional analysis, which states that a function that relates physical observables has to be independent on the choice of dimensional units.
- *Causality*, as derived by Koenderink [63], defines that it should be possible to trace every gray-level at coarse scale to a corresponding gray-level at a finer scale. In other words this implies that new level surfaces

$$\{(\mathbf{x}; t) \in \mathbb{R}^N \times \mathbb{R} : L(\mathbf{x}; t) = L_0\} \quad (2.1)$$

must not be created in the scale-space representation when the scale parameter t is increased. By combining causality with the notions of isotropy and





homogeneity, it is shown that the scale-space representation has to obey the diffusion equation.

$$L_t(\mathbf{x}, t) = \Delta L(\mathbf{x}, t) \quad (2.2)$$

A similar result is given by Yuille and Poggio [141].

- *Regularisation*, Florack [22] and Nielsen[102], derived a series of regularisation filters $e^{-(\omega^2 t)^p}$ for $p \in \mathbb{N}$, the Gaussian kernel is one of them.
- *Entropy*, Nielsen [101] derived the Gaussian kernel by expressing the notion of uncommittedness in a statistical way using the entropy of the observed signal.
- *Biological Inspiration*, the human visual system samples the outside world on multiple scales. On the retina lie *receptive fields*, groups of receptors assembled in such a way that they form a set of apertures of widely varying size. Together they measure a scale space of every image [25, 26, 37, 38, 44]. Using the Gaussian kernel can simulate this effect as argued by Koenderink [64, 67, 68, 69, 70, 71, 72]. Young [139, 140] reports that cells in the visual cortex have receptive field profiles that agree with Gaussian derivatives.

2.2.2 Generating a Scale-Space Representation

A brief review of the complete procedure for embedding an image into a Gaussian scale-space is as follows:

Given an N -Dimensional image $I(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}$, the scale-space representation $L(\mathbf{x}, t) : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}^N$ is defined such that the representation at ‘zero scale’ is equal to the original image

$$L(\mathbf{x}, 0) = I(\mathbf{x}) \quad (2.3)$$

and the representations at coarser scales are given by convolution of the given image with Gaussian kernels of successively increasing width

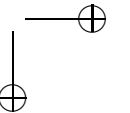
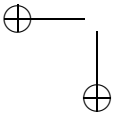
$$L(\mathbf{x}, t) = (g_t * I)(\mathbf{x}), \quad \text{with } g_t(\mathbf{x}) = g(\mathbf{x}, t) = (2.5). \quad (2.4)$$

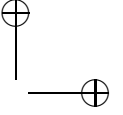
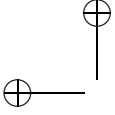
where $g : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is the Gaussian kernel

$$g(\mathbf{x}, t) = \frac{1}{(4\pi t)^{N/2}} e^{-\mathbf{x}^T \mathbf{x} / (4t)}. \quad (2.5)$$

In terms of explicit integrals the result of the convolution operation ‘ $*$ ’ is written

$$L(\mathbf{x}, t) = \int_{\xi \in \mathbb{R}^N} g(\xi, t) I(\mathbf{x} - \xi) d\xi, \quad (2.6)$$





Scale space is named after the space that is formed by looking at an image at all different scales simultaneously. When we stack all these scales we get an extra dimension, the scale dimension, as demonstrated in Figure 2.2 and 2.3.



Figure 2.2: An image at successive, exponentially increasing, scales.

Mathematica™ Implementation

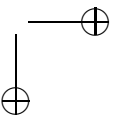
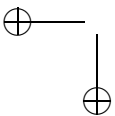
Implementation in Mathematica™ is done by using the MathVisionTools package [2]. Calculating a scale-space image at scale t is easily done as follows:

```
<< MathVisionTools`  
Lt = GaussianDerivative[{t, 0}, {t, 0}][image];
```

2.2.3 Separability

Although sometimes mentioned as one of the scale-space axioms, separability is merely a technically useful property of the Gaussian kernel. The N -Dimensional Gaussian kernel $g : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ can be written as the product of N one-dimensional kernels $g_1 : \mathbb{R} \rightarrow \mathbb{R}$.

$$g(\mathbf{x}, t) = \prod_{i=1}^N g_1(x_i, t). \quad (2.7)$$



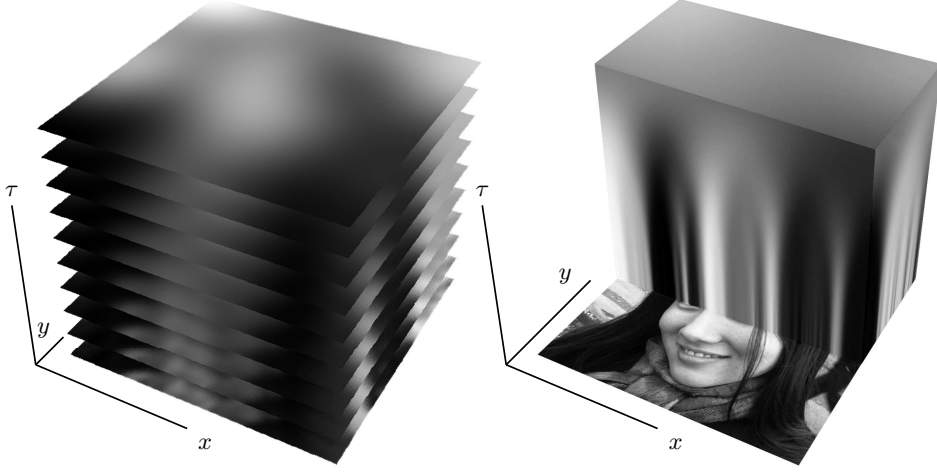


Figure 2.3: A scale space originating from Figure 2.2. Left: a stack of images at successive scales. Right: a cut through the scale space shows the images at successive scales.

In terms of explicit expressions for the Gaussian kernels this can be written as

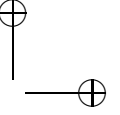
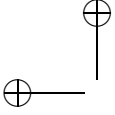
$$\frac{1}{(4\pi t)^{N/2}} e^{-\mathbf{x}^T \mathbf{x} / (4t)} = \prod_{i=1}^N \frac{1}{\sqrt{4\pi t}} e^{-x_i^2 / (4t)}. \quad (2.8)$$

The separability is beneficial in terms of computational efficiency when implementing the smoothing operation by convolutions in the spatial domain. Assume that the smoothing operation is implemented by some discrete approximation and that the discrete convolution mask has width M along each dimension. In the standard case the number of operations for every point is M^N . By using the separability property of the Gaussian kernel, the amount of operations per point reduces to MN .

2.2.4 α Scale Spaces

Work by Florack [22], Nielsen [101], Pauwels [110] and Duits [19] argues that Gaussian scale space is just an instance of a more general linear scale space. Florack has observed that there exists a whole parameterized class of possible filters, namely the ones of which the Fourier Transform equals

$$\mathcal{F}(\omega, t) = e^{-t \|\omega\|^{2\alpha}}. \quad (2.9)$$



Which yields the Gaussian scale space for $\alpha = 1$. Duits has proven that one has to restrict the parameter interval to $\alpha \in (0, 1]$ and has scrutinized the special case $\alpha = \frac{1}{2}$ leading to the so called Poisson scale space. For implementation in the spatial domain Poisson filtering is an alternative to Gaussian filtering in the sense that typical properties are maintained, besides the fact that all scale space axioms are satisfied. Drawbacks however are that the separability property is lost in all cases except for the Gaussian case and that the spatial convolution filters are heavy-tailed. Both result in longer calculation times. For implementation in the Fourier domain, all α scale spaces are equally straightforward in practice.

2.2.5 Sampling the Scale Axis

The scale parameter $\sigma (= \sqrt{2t})$ gives a local length parameter for the level of resolution. The parameter σ can be parameterized with a dimensionless parameter τ as shown by Florack [31]. Scale invariance implies that $\frac{d\sigma}{d\tau}$ is proportional to σ . Without generality loss $\frac{d\sigma}{d\tau} = \sigma$ may be taken, with $\sigma|_{\tau=0} = \epsilon$, where ϵ is a ‘hidden scale’ carrying the dimension of a length. This parameter is limited by the image modality. The dimensionless parameter τ is called the *natural scale parameter*, $\sigma = \epsilon e^\tau$ where τ can be any real number, even negative.

2.2.6 Scale-Space Derivatives

Derivatives in images are often used when computing image descriptors such as features or differential invariants (Chapter 6). Since differential geometry is a natural framework for describing geometric relations, a large number of formulations in terms of derivatives are found in vision problems. Calculating derivatives from discrete data is not a well-posed problem. Derivative estimators enhance noise and it is obvious that some sort of smoothing is necessary. This is immediately obvious when looking at a dithered image as in Figure 2.4.

The scale-space representation provides a well-defined way to deal with the notion of scale in the computation of derivatives. A partial derivative

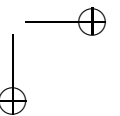
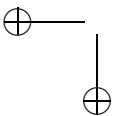
$$L_{x^\alpha} = \partial_{x^\alpha} L = \partial_{x_1^{\alpha_1}} \dots \partial_{x_N^{\alpha_N}} L \quad (2.10)$$

of the scale-space representation can, because of its linearity, be written as

$$\partial_{x^\alpha} (I * g_t)(\mathbf{x}) = (I * \partial_{x^\alpha} g_t)(\mathbf{x}) = (I * g_{tx^\alpha})(\mathbf{x}), \quad (2.11)$$

which implies that

$$L_{x^\alpha}(\mathbf{x}, t) = \int_{\xi \in \mathbb{R}^N} g_{x^\alpha}(\xi, t) I(\mathbf{x} - \xi) d\xi. \quad (2.12)$$



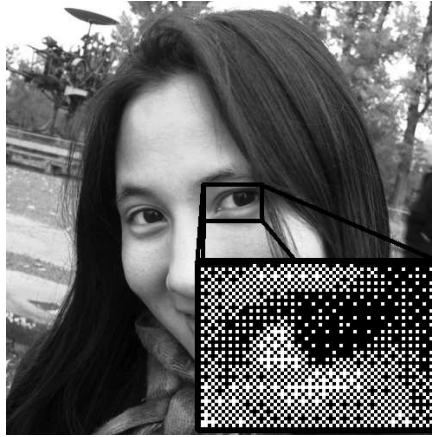


Figure 2.4: A zoomed-in part of a dithered image demonstrates the need for smoothing image data when taking derivatives.

This means that differentiation is done by convolution with a Gaussian derivative kernel. Examples of these kernels are given in Figure 2.5 and their effect is demonstrated in Figure 2.6.

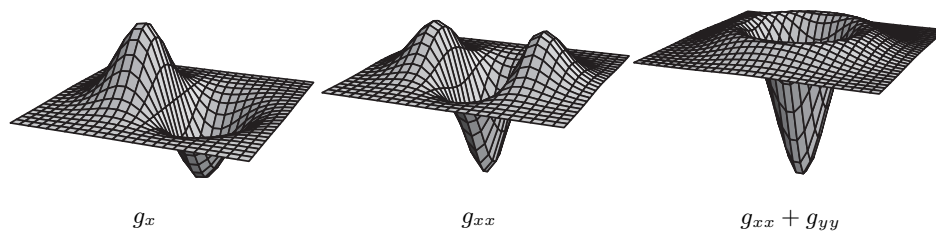


Figure 2.5: Some Gaussian derivative kernels.

Mathematica™ Implementation

Creating the images for Figure 2.6 is done by evaluating the following code:

```
 $L_x = \text{GaussianDerivative}[\{t, 1\}, \{t, 0\}][\text{image}];$   

 $L_{xx} = \text{GaussianDerivative}[\{t, 2\}, \{t, 0\}][\text{image}];$   

 $\Delta L = L_{xx} + \text{GaussianDerivative}[\{t, 0\}, \{t, 2\}][\text{image}];$ 
```



Figure 2.6: Results obtained by convolution of a test image with the Gaussian derivative kernels from Figure 2.5.

2.3 Multi-Scale vs. Multi-Resolution

Next to the scale-space representation which is generated by smoothing and preserves the same grid-size at all scales, a pyramid representation that reduces the number of grid points from one layer to the next is used for multi-resolution purposes. This representation is popular in many computer vision tasks for its speed.

A pyramid representation of an image is a set of successively smoothed *and* sub-sampled representations of the original image. This representation is organized in such a way that the number of pixels decreases with a constant factor (usually 2^N for an N -dimensional signal) from one layer to the next. A pyramid representation of a two-dimensional image is obtained by setting each value of a pixel at a coarser scale $L^{(k+1)}$ to be an average of a neighborhood of pixel values in the finer scale representation $L^{(k)}$. By recursively applying this operation and stacking the resulting representations on top of each other a pyramid of L is generated. This is illustrated in Figure 2.7.

There are some advantages of a pyramid representation. The most important one is that it leads to a rapidly decreasing image size. This reduces the computational work both in the actual generation of the representation as well as in the subsequent processing. The memory requirements are small and the generation is simple and easily implementable, even on hardware.

There are however also many disadvantages of the pyramid representation. The most important disadvantage, for this work, is that the pyramid is generated by an algorithmic process, which makes theoretical analysis very cumbersome. Moreover the pyramid representation has a very coarse discretization along the scale axis,

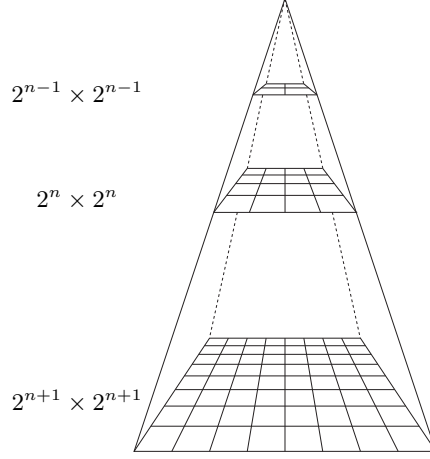


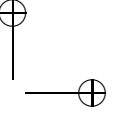
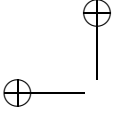
Figure 2.7: A pyramid representation is obtained by successively reducing the image size by combined smoothing and subsampling.

whereas in the scale-space case the scale parameter is continuous. This makes it very hard to match image structures across scales. Due to their implementation pyramids are not translation invariant, which implies that the representation changes when the image is shifted.

2.4 Catastrophe Theory in Scale-Space

In many practical situations families of functions are encountered that depend on *control parameters*. An example of this is the scale parameter in a scale-space representation of an image. Catastrophe theory is the study of how critical points (Section 2.4.1) of a function change as the control parameters change. The theory was originated in the 1960's with the work of Thom [129, 130]. The interested reader can find more information on general catastrophe theory in one of the following books [3, 34, 112, 117]. In this thesis the survey of catastrophe theory will be restricted to the theory applicable to scale-space representations of two-dimensional images.

Koenderink [69, 64, 67, 65, 66] was the first to point out that Thom's classification theorem could be applied to scale-space representations of images. In a scale-space representation there is only one control parameter, the isotropic inner scale. The fact that a scale space is considered which is constrained by the isotropic diffusion equation further restricts the way in which critical points behave through



scale. Damon [13, 14] probably gives the most comprehensive account on this subject, but many others have investigated the theory as well, like Griffin [35], Johansen [49, 50, 52], Kuijper [74, 73], Lindeberg [85, 84] and Loog [91]. Florack and Kuijper's work on the topological structure of scale-space images [33] gives a very clear and thorough account on the behavior of critical points in scale space for generic images.

2.4.1 Critical Points

A critical point of a function $f : \mathbb{R}^N \rightarrow \mathbb{R}$ is a point at which the gradient vanishes,

$$\nabla f = \mathbf{0}.$$

This occurs typically at isolated points at which the Hessian (i.e. the matrix of second order derivatives)

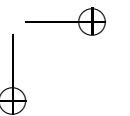
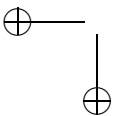
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.13)$$

has nonzero eigenvalues. Critical points for which these conditions hold are referred to as *Morse critical points*. Critical points for which one of the eigenvalues of the Hessian is zero are referred to as *non-Morse critical points* or *top-points*. These top-points will prove to be very useful as characteristic points of an image in the following chapters of this thesis.

For two-dimensional images critical points can best be illustrated by looking at the gray-value images as a height plot. The critical points; saddles, minima and maxima correspond to passes, dales and peaks of the virtual landscape respectively. This is illustrated in Figure 2.8 (note that the input image is blurred to reduce the amount of critical points).

2.4.2 Critical Curves

While varying a control parameter in a continuous fashion, a Morse critical point will move along a *critical curve*. At isolated points on such a curve one of the eigenvalues of the Hessian may become zero, so that the Morse critical point turns into a non-Morse critical point.



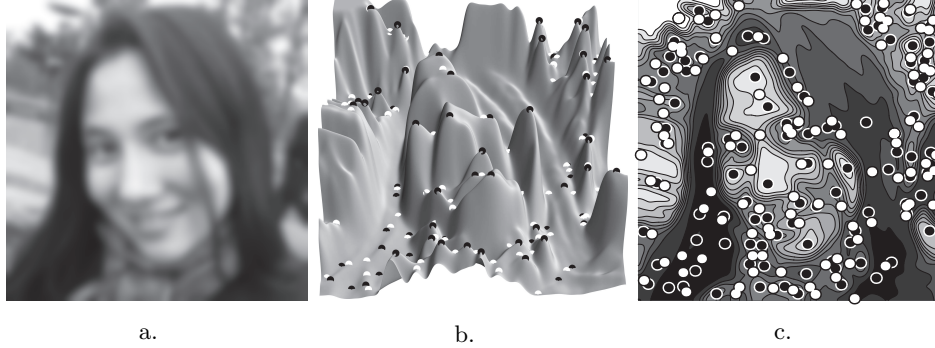


Figure 2.8: a. Blurred image, b. Image a. as a height plot with annotated extrema (black) and saddles (white), c. Contour plot of image a. showing saddles and extrema.

After a small perturbation Morse critical points may move and the corresponding function values may change, but their type will stay the same: if i eigenvalues of the Hessian were negative prior to the perturbation, then this will remain the same after the perturbation. A non-Morse critical point however changes qualitatively when perturbed. In general, a non-Morse critical point will split into a number of Morse critical points. This event is called a *morsification*.

When considering only *generic* situations, and limiting the input images to be ‘typical’, catastrophe theory for scale-space representations simplifies a lot. The only generic morsifications in scale-space are *creations* and *annihilations* of pairs of Morse hypersaddles of opposite *Hessian signature* (also known as *topological charge*)

$$q = \text{sign}(\det \mathbf{H}). \quad (2.14)$$

The topological charge of a non-Morse critical point equals the sum of charges of all Morse critical points involved in the morsification. Figure 2.9 demonstrates the two generic catastrophes in isotropic scale-space.

In one-dimensional signals, creations are prohibited by the diffusion equation and only annihilations occur between minima and maxima in the signal for increasing scale. In the scale-space representation of two-dimensional images, annihilations and creations occur between maxima and saddle points, and minima and saddle points (as demonstrated in Figure 2.10). In three-dimensional scale-space representations of images there are two distinct types of hypersaddles, one with a positive and one with a negative Hessian signature. Also minima and maxima have opposite topological charges in this case, so there are various possibilities for annihilation that are consistent with charge conservation. However, only the catastrophes in which one and only one eigenvalue of the Hessian matrix changes, occur generically.

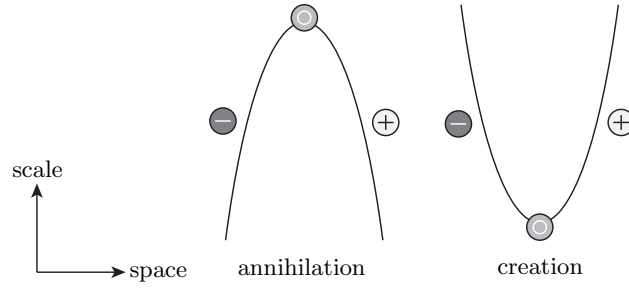
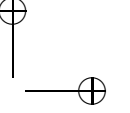
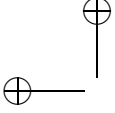


Figure 2.9: The generic catastrophes in isotropic scale-space. Left: annihilation of a pair of Morse critical points. Right: creation of a pair of Morse critical points. The involved points have opposite Hessian signature.



Figure 2.10: Critical curves and critical points for an image. Saddle curves are visualized darker than extrema curves. Visualization done in ScaleSpaceViz [55, 56]



2.4.3 Detection of Critical Points and Critical Curves

There are many ways to detect critical points in images, some of the methods will be discussed here.

Topological Number

The concept presented in the following can be studied in more detail in [53, 54, 23, 46]. The *topological number* is a natural quantity that classifies singularities of scalar images of any dimension.

Suppose P is a point in the image (either singular or regular) and S_P is a closed hypersurface, topologically equivalent to a $(d-1)$ -dimensional sphere, containing point P and not containing any singularities except possibly P itself.

Because S_P does not contain any singularities, the normalized gradient vector field

$$\xi_i = \frac{L_i}{\sqrt{L_j L_j}} \quad (2.15)$$

is well defined on the surface S_P .

At a non-singular point Kalitzin [54] defines a $d-1$ form:

$$\Phi = \xi_{i_1} d\xi_{i_2} \wedge \dots \wedge d\xi_{i_N} \varepsilon^{i_1 i_2 \dots i_N} \quad (2.16)$$

Where ε is the *Levi-Civita* or *permutation tensor* defined as

$$\varepsilon^{i_1 i_2 \dots i_N} = \begin{cases} 0, & \text{if any two labels are the same} \\ 1, & \text{if } i_1, i_2, \dots, i_N \text{ is an even permutation of } 1, 2, 3 \\ -1, & \text{if } i_1, i_2, \dots, i_N \text{ is an odd permutation of } 1, 2, 3. \end{cases} \quad (2.17)$$

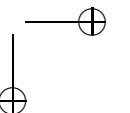
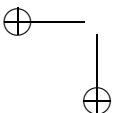
Substituting (2.15) in (2.16) yields the following expression:

$$\Phi = \frac{L_{i_1} dL_{i_2} \wedge \dots \wedge dL_{i_N} \varepsilon^{i_1 i_2 \dots i_N}}{(L_j L_j)^{N/2}}. \quad (2.18)$$

The topological number ν_S for the hyper-surface S_P surrounding point P is defined as

$$\nu_{S_P} = \oint_{\mathbf{x} \in S_P} \Phi(\mathbf{x}). \quad (2.19)$$

For two-dimensional images the topological number (2.19) is also referred to as the *winding number*. The winding number represents the number of times the



normalized gradient turns around its origin, as a test point circles around a given contour (illustrated in Figure 2.11). In two dimensions Φ (2.16) describes the angle between the normalized gradients in two neighboring points. Integration of this angle along a closed contour yields the winding number associated with this contour.

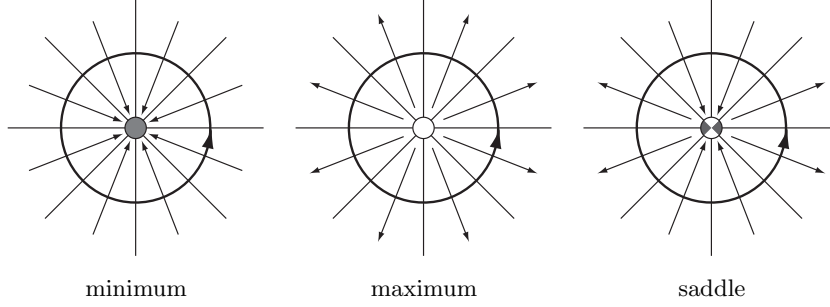


Figure 2.11: The gradient vectors and the winding-number path for a two-dimensional minimum, maximum and a saddle respectively.

The winding number of any closed contour is an integer multiple of 2π . For regular points the winding number is zero. For local extrema the winding number is $+2\pi$ and for saddle points the winding number is -2π .

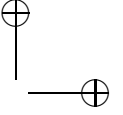
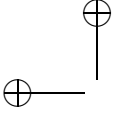
The winding number can be calculated from (2.19) and (2.16), but in two dimensions it is more convenient to use a complex number representation.

Let $z = x + iy$ and $\bar{z} = x - iy$ be the complex conjugated couple of coordinates in the two-dimensional image space and let $L(z, \bar{z})$ be the image in this notation. Then the complex function $W = (L_x + iL_y)/2 \equiv \partial_{\bar{z}}L(z, \bar{z})$ represents the gradient vector field in complex coordinates.

In accordance with (2.16) $\Phi(\mathbf{x})$ can now be written in complex notation as:

$$\begin{aligned}
 \Phi(A) &= \xi_x d\xi_y - \xi_y d\xi_x = \frac{L_x dL_y - L_y dL_x}{L_x L_x + L_y L_y} \\
 &= \text{Im} \frac{(L_x - iL_y)d(L_x + iL_y)}{L_x L_x + L_y L_y} \\
 &= \text{Im}(\bar{W}dW/(W\bar{W})) = \text{Im}(dW/W) \\
 &= \text{Im}(d \ln W).
 \end{aligned} \tag{2.20}$$

Since $\ln W = \ln |W| + i \arg W$ the last equation justifies the interpretation of Φ as a relative angle change of the gradient field.



The winding number approach for two-dimensional images is easily implementable for all scale levels and is fast to calculate. An implementation example is given in the MathematicaTM Implementation box.

MathematicaTM Implementation

The following module uses the winding number approach to locate extrema and saddles in a two dimensional image. A small four pixel neighborhood is used for the calculation of the gradient vectors.

```
windingnumber2D[image_, t_]:=Module[{},
  complexgradient = GaussianDerivative[{t, 1}, {t, 0}][image]
    + i GaussianDerivative[{t, 0}, {t, 1}][image];
  v = RotateLeft[complexgradient, #]&/@{{0, 0}, {-1, 0}, {-1, -1}, {0, -1}};
  wn =  $\frac{1}{2\pi}$  Plus@@Arg[v/RotateLeft[v]]//Round;
  Position[wn, #]&/@{1, -1}];
```

The location of extrema and saddle points in the image at scale t can now be calculated.

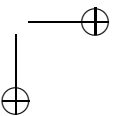
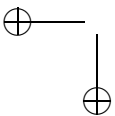
```
{extrema, saddles} = windingnumber2D[image, t];
```

code adopted from [37]

Hexagonal Grid

Detection of the critical points using a *hexagonal grid* for two-dimensional images, has been demonstrated by Blom [8]. In the algorithm the differences in intensity between the pixel that is analyzed and its neighbours are considered. Pixels in the neighbourhood that have a greater intensity than the pixel that is considered are marked as positive pixels and the pixels in the neighborhood that have a lower intensity are marked as negative. The considered pixel is then classified by counting the number of sign changes that are observed while walking over the neighbourhood pixels. This results in the following rules (demonstrated in Figure 2.12):

- Zero sign changes, the point is a maximum or minimum
- Two sign changes, the point is a regular point
- Four sign changes, the point is a saddle point



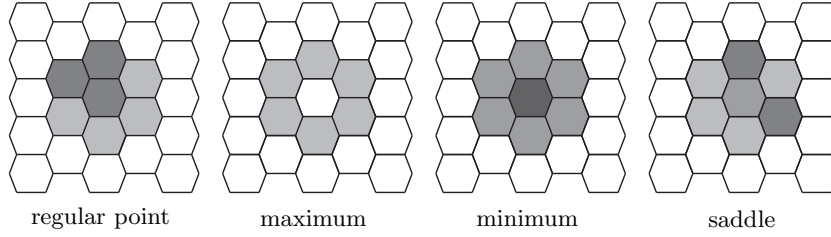


Figure 2.12: Different types of pixels on a hexagonal grid. The regarded pixel is in the middle. Points with a lower intensity are darker, and with a higher intensity are lighter.

Since digital images are usually stored on a rectangular grid, the hexagonal lattice has to be constructed out of this data. Horn [42] suggests to move the row above the pixel of interest half a pixel length to the left and the row below half a pixel distance to the right to construct a hexagonal lattice. Blom [8] shifts every odd row half a pixel length to the right to construct a hexagonal lattice. Blom's method is implementable by considering a six neighborhood of a pixel. The construction of the different types of grids is shown in Figure 2.13.

The critical point detector based on Blom's hexagonal grid is easily implementable and very fast as demonstrated in the MathematicaTM Implementation box. It is however more stable to resample the odd rows of a rectangular grid. Instead of taking samples at distances $0, d, 2d, \dots$ (d denotes the grid distance), samples are now taken at $\frac{1}{2}d, \frac{3}{2}d, \frac{5}{2}d, \dots$. The resampling can for example be done with interpolation polynomials. This resampling will however slow the critical point detection algorithm down significantly.

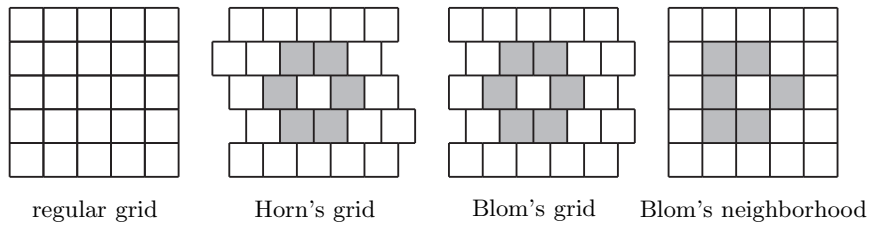
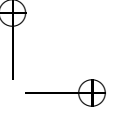
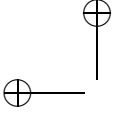


Figure 2.13: A hexagonal grid is constructed from a regular grid by shifting the rows half a pixel. The way in which the rows are shifted is different between authors. The neighborhood of a pixel in Blom's grid is easily represented in a rectangular grid as seen on the right.



Mathematica™ Implementation

The following module uses the hexagonal grid suggested by Blom to detect extrema and saddles in a two dimensional image.

```
hexnumber2D[image_, t.]:=Module[{},  
  imscaled = GaussianDerivative[{t, 0}, {t, 0}][image];  
  imstack = (RotateLeft[imscaled, #] - imscaled)&/@  
    {{-1, 0}, {-1, 1}, {0, 1}, {1, 0}, {0, -1}, {-1, -1}};  
  hex = Transpose[imstack, {3, 2, 1}];  
  countsignchanges:=Plus@@Abs[Sign[#] - RotateLeft[Sign[#]]]/2&;  
  nsignchanges = Map[countsignchanges, hex, {2}];  
  Position[nsignchanges, #]&/@{0, 4}]
```

The location of extrema and saddle points in the image at scale t can now be calculated.

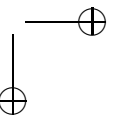
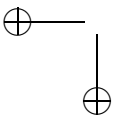
```
{extrema, saddles} = hexdetector2D[image, t];
```

Zero Crossings of the Gradient

Another method for finding critical points in an image uses zero crossings of the first order derivatives to locate points with a zero gradient. For a two-dimensional image this is done by intersecting the level lines $L_x = 0$ with the level lines $L_y = 0$, as demonstrated in Figure 2.14.

This method can be used on the scale-space representation of an image to find all the critical curves. By intersecting the level surfaces $L_x = 0$ and $L_y = 0$, the critical curves are obtained as shown in Figure 2.15. By intersecting the critical curves with the level surface $\det \mathbf{H} = 0$, in which \mathbf{H} is the Hessian matrix (2.13), the non-Morse critical points (top-points) can be located on the curves (Figure 2.15). This algorithm has been implemented by Kanters [55, 56].

A drawback of the zero-crossings algorithm is that it does not automatically distinguish between the different types of critical points. This can however simply be calculated afterwards, by observing the local neighborhood of such a detected critical point.



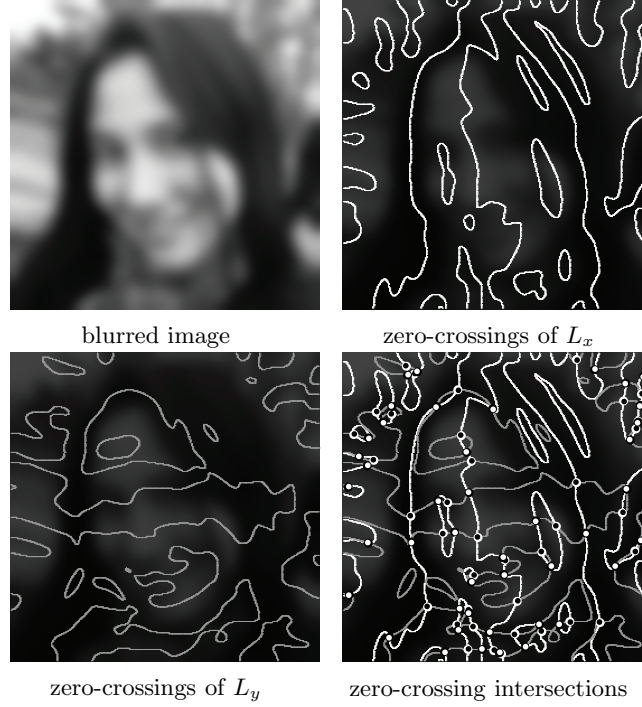


Figure 2.14: Crossing level lines with $L_x = 0$ with level lines with $L_y = 0$ yields the critical points at the intersections.

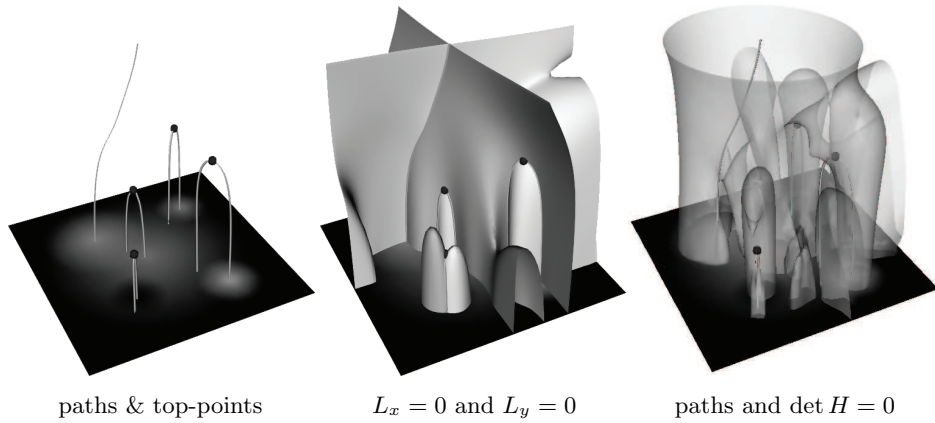
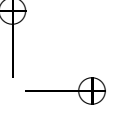
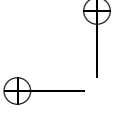


Figure 2.15: Intersecting level surface $L_x = 0$ with level surface $L_y = 0$ yields the critical curves. Intersecting the critical paths with the level surface $\det H = 0$ yields the top-points.



2.4.4 Refining the Location of Critical Points

The described methods for finding critical points operate at pixel precision. For a critical point \mathbf{x}_c any of the above described detectors obtains an approximate location \mathbf{x}_a . The sub-pixel location \mathbf{x}_c of a critical point can be calculated by evaluating a Taylor expansion at the detected location \mathbf{x}_a . For a two-dimensional image the Taylor expansion of the first order derivatives in the critical point is

$$\begin{aligned} L_x(\mathbf{x}_c) &= L_x(\mathbf{x}_a) + L_{xx}(\mathbf{x}_a)(x_c - x_a) + L_{xy}(\mathbf{x}_a)(y_c - y_a) \\ &\quad + \frac{1}{2}L_{xxx}(\mathbf{x}_a)(x_c - x_a)^2 + L_{xxy}(\mathbf{x}_a)(x_c - x_a)(y_c - y_a) \\ &\quad + \frac{1}{2}L_{xyy}(\mathbf{x}_a)(y_c - y_a)^2 + \dots \end{aligned} \quad (2.21)$$

and

$$\begin{aligned} L_y(\mathbf{x}_c) &= L_y(\mathbf{x}_a) + L_{xy}(\mathbf{x}_a)(x_c - x_a) + L_{yy}(\mathbf{x}_a)(y_c - y_a) \\ &\quad + \frac{1}{2}L_{xyx}(\mathbf{x}_a)(x_c - x_a)^2 + L_{xyy}(\mathbf{x}_a)(x_c - x_a)(y_c - y_a) \\ &\quad + \frac{1}{2}L_{yyy}(\mathbf{x}_a)(y_c - y_a)^2 + \dots \end{aligned} \quad (2.22)$$

Since the first-order derivatives at the true location of the critical point $L_x(\mathbf{x}_c)$ and $L_y(\mathbf{x}_c)$ are exactly zero and the Gaussian derivatives up to any order can be calculated at the estimated location \mathbf{x}_a , Equations (2.21, 2.22) can be solved to yield the true sub-pixel location \mathbf{x}_c of the critical point.

Florack and Kuiper [33] have shown that it is possible to calculate a vector pointing to the location of a non-Morse critical point (top-point) from a point sufficiently close by. If (x_a, y_a, t_a) denotes the approximate location of a top-point it is possible to find the true location of the top-point $(x_a + \xi, y_a + \eta, t_a + \tau)$ by:

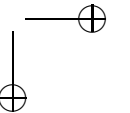
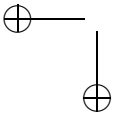
$$\begin{bmatrix} \xi \\ \eta \\ \tau \end{bmatrix} = -\mathbf{M}^{-1} \begin{bmatrix} \mathbf{g} \\ \det \mathbf{H} \end{bmatrix}, \quad (2.23)$$

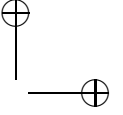
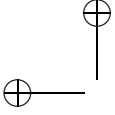
where

$$\mathbf{M} = \begin{bmatrix} \mathbf{H} & \mathbf{w} \\ \mathbf{z}^T & c \end{bmatrix}, \quad (2.24)$$

$$\mathbf{g} = \nabla u, \quad \mathbf{H} = \nabla \mathbf{g}, \quad \mathbf{w} = \partial_t \mathbf{g}, \quad \mathbf{z} = \nabla \det \mathbf{H}, \quad c = \partial_t \det \mathbf{H}, \quad (2.25)$$

in which \mathbf{g} and \mathbf{H} denote the image gradient and Hessian matrix, respectively, and in which all derivatives are taken in the point (x_a, y_a, t_a) .





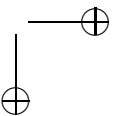
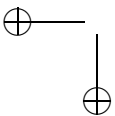
In Section 5.4.1 this theory will be used to determine the exact location of top-points in the scale-space representation of a two-dimensional image.

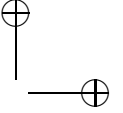
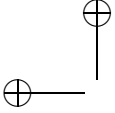
2.5 Conclusion

A multi-scale representation is essential for scale-invariant operations. In this chapter the basics of this linear scale space have been discussed. The most suitable kernel for generating a linear scale-space representation has shown to be the Gaussian kernel. The kernel obeys all scale-space axioms and its separability allows for fast calculation.

In the scale-space representation of an image critical points can be identified. Catastrophe theory describes the movement of these critical points as the scale parameter changes. Morse critical points move through scale forming critical curves and eventually annihilating with critical points of opposite Hessian signature in a non-Morse critical point called a top-point. These top-points will show to be important as image descriptors and key-points in the next chapters.

The location of critical points and top-points can be determined up to pixel precision by fast detection algorithms. A sub-pixel location can afterwards be calculated from the local differential image structure. This enables fast detection without losing the exact location of Morse and non-Morse critical points.

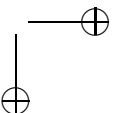
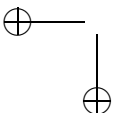


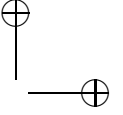
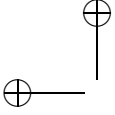


I didn't fail the test, I just found 100 ways to do it wrong.

Benjamin Franklin (1706–1790)

The Hierarchical Structure of Scale-Space Images





3.1 Introduction

There exists a great need for general segmentation and classification methods that work regardless of the input image. Most segmentation and classification methods are designed for a specific combination of image modality and specific need, and do not work in a different context. These tools are often based on existing methods endowed with various tunable parameters and more or less ad hoc choices for these parameters.

In this chapter three methods are discussed that can represent images in a hierarchical way using the deep structure of an image and exploiting the results from catastrophe theory for scale-space representations explained in the previous chapter.

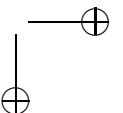
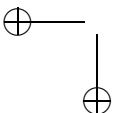
The idea of a hierarchical image description based on multiple scales and regions is nicely explained by a quote from Lifshitz and Pizer [82]: “A face might be described as a light spot containing a light spot (a reflection from the forehead) and three dark spots (the mouth and the regions of the two eyes). In turn the eye regions would be described as containing a dark spot (the eyebrow), a light spot (the eyelid), and a dark spot (the eye), with the latter containing a light spot (the eyeball) which itself contains a dark spot (the iris) which finally contains a yet darker spot (the pupil).”.

3.2 Hierarchical Segmentation from Intensity Extrema

The method presented here was suggested by Lifshitz and Pizer [83] in 1990. This is the first seminal paper suggesting a hierarchical representation of the deep structure of an image, based Koenderink’s work [63]. The aim of their research was to create a computer algorithm to segment grayscale images into regions of interest by constructing a tree structured region descriptor. The hierarchical descriptor introduced by Lifshitz and Pizer is based on the behavior of extrema in the scale-space representation of the image and is referred to as the *extremum stack*.

3.2.1 Extremum Paths and Extremal Regions

By using catastrophe theory as explained in Section 2.4, the paths of extrema are tracked in the scale-space representation of an image. Progressively blurring an image causes extrema to move continuously and eventually to annihilate as they



collide with saddles. The resulting paths are called critical curves as explained in detail in Section 2.4.2. Lifshitz and Pizer disregard the saddle part of the critical curves and focus only on the path that a minimum or a maximum follows through scale. This path they refer to as the *extremum path*. A set of these extremum paths is illustrated in Figure 3.1.



Figure 3.1: A small set of extremum paths of an image. The catastrophe points are indicated with white dots.

For the sake of segmentation a region is assigned to each extremum path. When moving along an extremum path to coarser scales, the intensity change is monotonic (increasing for minima and decreasing for maxima). On an extremum path each point can be associated with a level surface (an iso-intensity surface in the scale-space representation) of that point's intensity. A set of these level surfaces for a simple image containing two light blobs and an image containing a light and a dark blob is given in Figure 3.4. A level surface originating from a point on an extremum path surrounds the extremum in the original image, as demonstrated in Figure 3.2. The pixels in the original image that are encapsulated by the level surface of an extremum form an *extremal region*.

An extremum disappears after its annihilation. The amount of blurring necessary for an extremum to annihilate is said to be a good measure of importance of the

extremal region (this however in practice is not always the case, extrema in flat regions are mostly due to noise but can go up to high levels of scale). The intensity of the topmost point on an extremum path is referred to the path's annihilation intensity. This is the intensity of the level surface that forms the boundary of the associated extremal region, as illustrated in Figure 3.2 for annihilating-point P . The extremal regions can be seen as a *pre-segmentation* of an image, i.e. a segmentation without using a priori knowledge of the image.

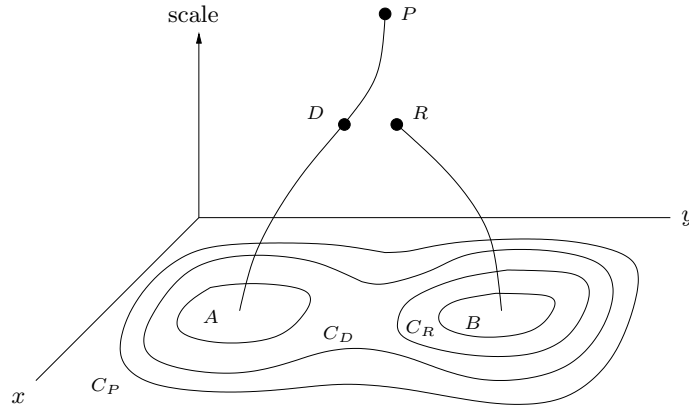


Figure 3.2: Extremum paths and associated isointensity contours for points on these paths (illustration adapted from [83]).

3.2.2 Scale-Space Hierarchy

According to Lifshitz and Pizer, when an extremum annihilates another region's level surface encloses the annihilated extremum, as illustrated in the left picture of Figure 3.3. As will be explained in the next section, this process of level surface enclosement occurs already earlier in scale. Nonetheless a containment relation among extremal regions is induced by this process. The set of extremal regions together with their containment relations can be represented as an *extremal region tree* in which the nodes represent extremal regions. A node is the child of another node if the extremal region that it represents is immediately contained in the extremal region represented by the parent. The root of the description tree represents the entire image. This is illustrated in Figure 3.3.

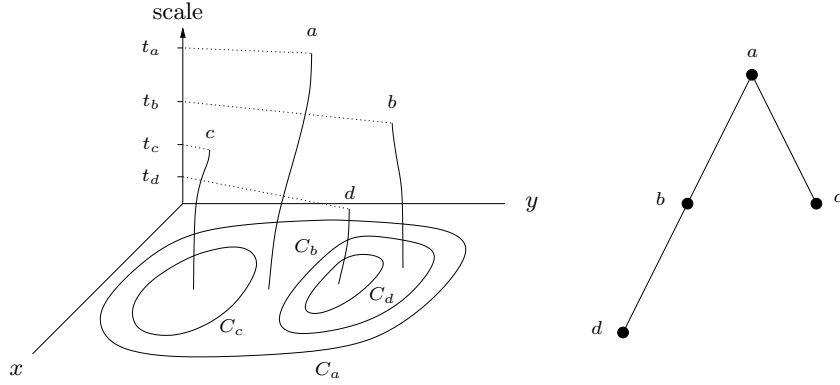


Figure 3.3: Left: Extremum paths with their extremal regions. Right: The associated extremal region tree. (Illustration adapted from [83]).

3.2.3 Remarks

Lifshitz and Pizer presented a basic and general approach for multiscale hierarchical segmentation. In their paper they show some results using the segmentation algorithm on abdominal CT images. Simmons et al. [124] evaluated the performance of the extremum stack for the case of neurological magnetic resonance imaging data and obtained some reasonable results.

Simmons et al. also report *false extrema* being temporarily created due to the discrete nature of the image. These ‘false extrema’ however emerge from the generic creation events discussed in Section 2.4.2. These events are not due to discretization but are generic events occurring commonly in two- or higher dimensional images. Creations of extrema indeed complicate the extremal tree construction. Including the creations in the hierarchical representation would generate a graph structure rather than a tree, but it would be feasible.

According to Lifshitz and Pizer each pixel (non-extremum point) in the original image can be associated with an extremum path by linking the point to the closest point with the same intensity at the next scale level in the scale-space representation and continuing this linking through scale until an extremum path is reached. This idea was introduced by Koenderink in [63]. The path created by this linking is called an *isointensity path*. It is however not the case that following a level surface through scale will unambiguously lead to one extremum. In their article Lifshitz and Pizer note this and mention “nonextrema that seem to escape from the extremal region”. In the article these ‘escaping’ extrema are said not to occur in the images they considered. It appears that Lifshitz and Pizer

have missed the fact that the topology of the level surfaces through extrema on the extremum path changes significantly when the intensity of a connected scale-space saddle in the image is passed. The properties of these scale-space saddles will be discussed in the next section. The change in the topology of level surfaces, occurring after passing a scale-space saddle is demonstrated in Figure 3.4. This shows that extremal regions are merged after passing a scale-space saddle and not after the annihilation of the extrema. The structure becomes exceedingly complex when more structure in the image is involved. Since multiple extremal paths can cross the level surface it is ambiguous to assign a pixel in the original image to a single extremal path by tracking its isointensity path. This problem remains unnoticed in Lifshitz and Pizer's approach as they track the isointensity path in the direction of steepest ascent, this will assign a non-extremum to a single extremum path.

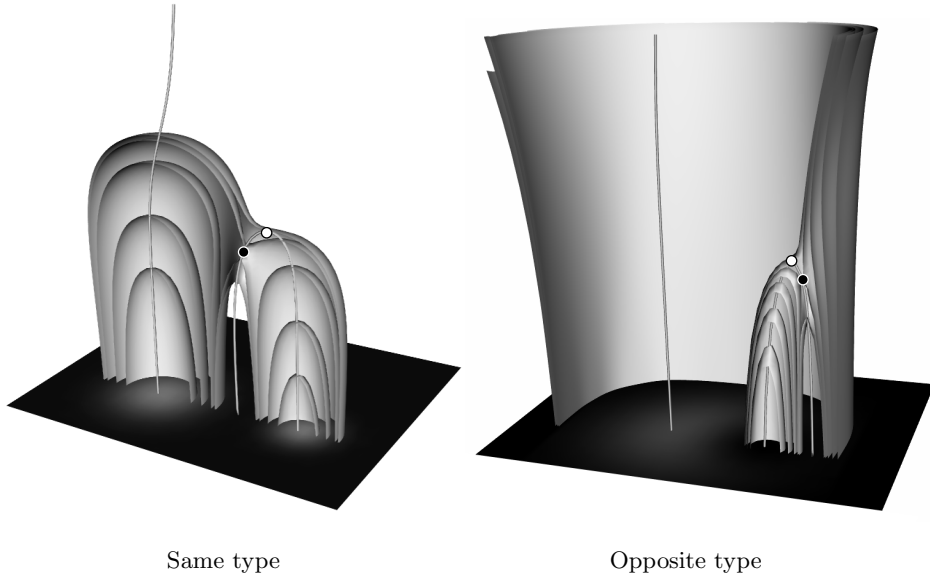
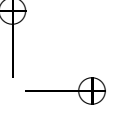
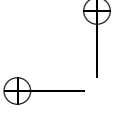


Figure 3.4: A number of level surfaces through the extrema on an extremum path at different scales is illustrated. Up until the scale at which a scale-space saddle (black dot) occurs, all the level surfaces are dome shaped and contain only the extremum path they belong to. Level surfaces between the scale of the scale-space saddle point and annihilation (white dot) scale contain more than one extremum path.

Simmons et al. note that the segmentation performance of the extremum stack is poor for elongated structures and that structures are often merged prematurely. The isotropic Gaussian kernel quickly blurs away line structures. Since the importance of segments is defined by the annihilation scale of the extremum, these segments are assigned a very low importance. Segments belonging to extrema with



high annihilation scales are not necessarily important. Extrema, originating from noise, can go up to high scales in areas of equal gray value. This demonstrates that the annihilation scale alone of an extremum is not a successful measure of importance.

3.3 Hierarchical Segmentation from Scale-Space Saddles

Kuijper [75, 78, 76, 77] proposes to use the nesting of special level surfaces in the scale-space representation of an image to build a hierarchy of critical points and to create a pre-segmentation.

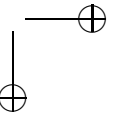
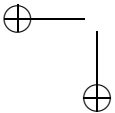
3.3.1 Scale-Space Saddles

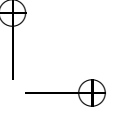
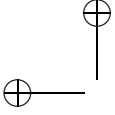
Kuijper's method focusses on scale-space critical points, these points are precisely Koenderink's 'hitherto unnoticed singularities' [65]. Scale-space critical points are points with vanishing gradient with respect to both spatial and scale direction [65].

$$\begin{cases} \nabla L(\mathbf{x}, t) = \mathbf{0} \\ \Delta L(\mathbf{x}, t) = L_t(\mathbf{x}, t) = 0. \end{cases} \quad (3.1)$$

As discussed in Section 2.4.2, each critical curve contains branches representing spatial critical points. These branches are connected at catastrophe points, where two spatial critical points are annihilated or created. For two-dimensional images the two branches connected at catastrophe points necessarily are a *saddle branch* and an *extremum branch*. Kuijper uses this nomenclature in his work. Note that the extremum branch is equivalent to Lifshitz and Pizer's extremum path.

On an extremum branch the intensities change monotonically when scale is increased. This implies that the scale-derivative L_t will never change sign on the extremum branch and thus will never reach the zero value. Intensities of saddle points however do not change monotonically for increasing scale and scale-space saddles can be found on the saddle branch. The number of scale-space saddles on a saddle branch is undetermined; a saddle branch of a critical curve can contain zero, one, or multiple scale-space saddles.





3.3.2 Pre-Segmentation

Level lines in two-dimensional images are often referred to as isophotes. Generally these isophotes are Jordan curves, i.e. they do not intersect one another, and they are closed (if they do not end at the boundary of the image). Isophotes can intersect themselves in a saddle point. As a consequence the image is separated into regions bounded by the isophotes through these saddle points. This separation can be extended to an arbitrary dimension.

Kuijper has extended this idea to the scale-space case, where the regions are separated by level surfaces through scale-space saddle points.

The causality principle (Section 2.2.1) states that no new level lines can be created when scale is increased. This implies that level surfaces in a scale-space representation have an open end towards the original (zero scale) image. As a result all the level surfaces in a Gaussian scale-space representation are dome shaped.

Level surfaces through scale-space saddles have two typical shapes. Kuijper defines the level surface, that intersects the extremum branch of the critical curve on which the scale-space saddle is situated, to be the *critical surface*, the other part of the level surface he calls the *dual surface* as illustrated in Figure 3.5.

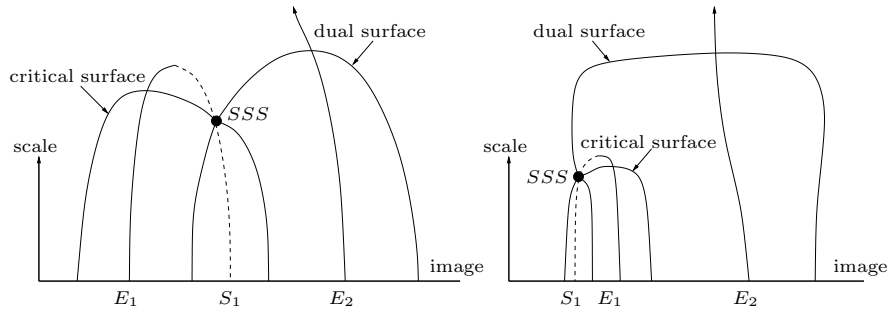
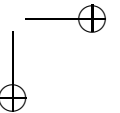
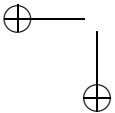
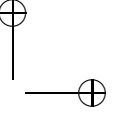
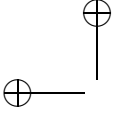


Figure 3.5: Schematic drawing of the critical and dual surface through a scale-space saddle (SSS). Where the critical curve is given by the extremum (solid line) and saddle branch (dotted line) connecting extremum E_1 to saddle S_1 . Left: the critical curves involved belong to the same type of extremum. Right: the critical curves involved belong to different types of extrema.

The shape of the level surface is determined by the type of the extrema that intersect the tops of the level surface domes connected at the scale-space saddle. The two types of level surfaces through scale-space saddles are illustrated in Figure 3.5 and Figure 3.6. If the critical curves involved both belong to the same type of extremum (both maxima or both minima), the resulting level surface con-





sists of two dome-like shapes positioned next to each other. If the critical curves involved have opposite types of extrema, the resulting level surface consists of two dome-like shapes where one is positioned within the other.

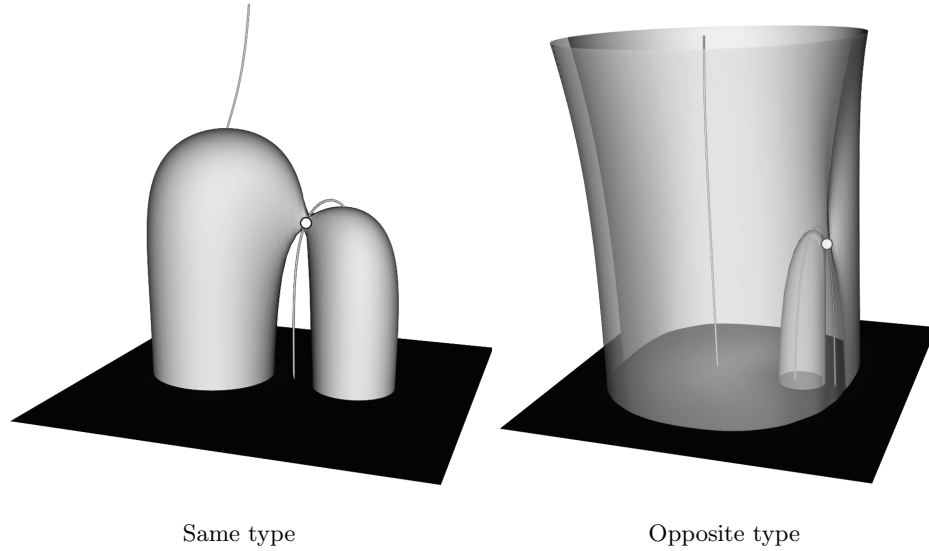


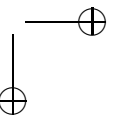
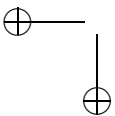
Figure 3.6: Visualization of the level surface through a scale-space saddle (white dot). Left: the critical curves involved belong to the same type of extremum. Right: the critical curves involved belong to different types of extrema. The level surface has been made partially transparent for visualization purposes.

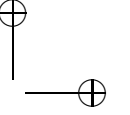
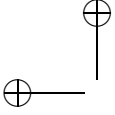
Kuijper defines a *scale-space segment* as the part of a scale-space representation that is bounded by the critical surface.

3.3.3 Scale-Space Hierarchy

As described in the previous section the level surfaces through scale-space saddles define segments in the image. These segments contain at least one extremum. If a segment contains more than one extremum it will contain subsegments. This indicates that the segments through the scale-space saddles induce both a hierarchy and a segmentation of the image.

The following algorithm, using the nesting of the level surfaces through scale-space saddles, was proposed by Kuijper to define the scale-space hierarchy. Assuming that the critical curves and scale-space saddles have already been located, the algorithm follows the steps of Algorithm 1.





Algorithm 1 Hierarchy from scale-space saddles

- 1: For each annihilating extremum, find its level surface through the scale-space saddle (both the critical and dual surface), if no scale-space saddle is present take the saddle on the saddle branch at the finest scale.
 - 2: Label to each extremum branch the dual surface it intersects, sorted by intensity.
 - 3: Start with the remaining extremum at the coarsest scale as the root of the hierarchical graph.
 - 4: Follow the extremum branch to finer scale until it intersects a dual surface.
 - 5: Split into two branches, one branch containing the existing extremum and one containing the extremum that is assigned to the critical surface that belongs to the intersected dual surface.
 - 6: Continue these steps until all extrema are added to the graph (Figure 3.7).
-

3.3.4 Remarks*Missing Segments*

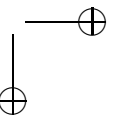
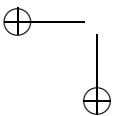
The major drawback of the method presented by Kuijper is that for some scales there are no segments defined for the extrema. This means that it is not a genuinely coarse to fine method as demonstrated in Figure 3.8.

The problem arises from the fact that the critical part of the isointensity surface through a scale-space saddle does not continue all the way up until the annihilation scale, but ends on the extremum branch before the catastrophe takes place. This means that for the image at scales between the top of the scale-space segment (the critical part of the level surface) and the annihilation scale, there does exist a blob caused by the extremum, yet there is no segment that belongs to it. Moreover the segment is defined for scales below the scale-space saddle but for scales above the scale-space saddle the segment is defined by the scale-space saddle that lives at a finer scale. Therefore this method is not truly coarse to fine.

The pre-segmentation as proposed by Kuijper is topologically not correct for these scales. This is an undesirable situation that should be taken care of in a neat, principled way.

Critical Curves without Scale-Space Saddles

To identify a segment with an extremum in the problematic case where a critical curve does not contain a scale-space saddle, Kuijper proposes to use the saddle at the lowest scale on the considered critical curve. This is quite an arbitrary choice. Why would this be a good choice, and if it is, why not take the saddles at the lowest scale for every critical curve and just ignore the scale-space saddles



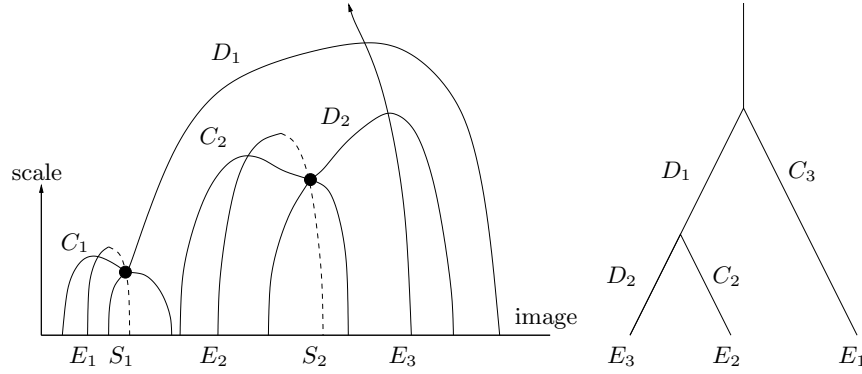
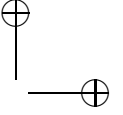
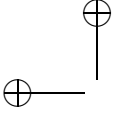


Figure 3.7: Schematic drawing of Kuijper's graph-building algorithm. Start from extremum E_3 , the path is intersected by the dual surface D_1 . D_1 belongs to critical surface C_1 and C_1 belongs to extremum E_1 . Therefore E_1 is a branch of E_3 . E_3 intersects dual surface D_2 , D_2 belongs to C_2 and C_2 belongs to extremum E_2 . Therefore E_2 is a branch of E_3 . The graph is shown on the right.

altogether?

3.4 Hierarchical Segmentation from Attraction Areas

Because of the shortcomings of the algorithms discussed in Section 3.2 and 3.3, a new method is presented in this chapter. This new algorithm is based on a so called *attraction area* and it combines the strong points from the previously discussed methods by Lifshitz and Pizer [83] and Kuijper [75].

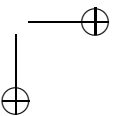
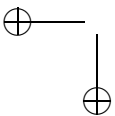
By definition the topology of all the isophotes within an attraction area is the same, i.e. they all encircle a single extremum. One could say that such isophotes are attracted to the extremum, hence the name attraction area.

The attraction area is similar to the support region of a greylevel blob as described by Lindeberg [87].

3.4.1 The Definition of the Attraction Area

The attraction area is defined as the largest possible area around an extremum, bounded by an isophote, that only encircles that extremum. This implies that this is an isophote that originates from a saddle point in the image¹ as shown in

¹Note that the attraction area of an extremum does not have to originate from a saddle that lies on the same critical curve.



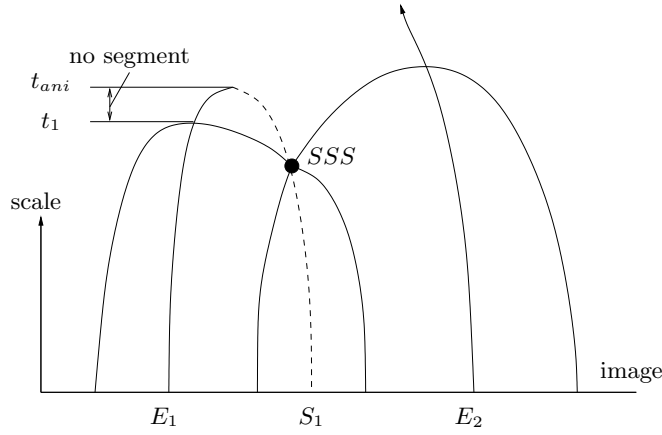


Figure 3.8: For a range of scales there is no segment defined for the extremum. For the scales above the scale space saddle there is a non causal segment.

Figure 3.9.

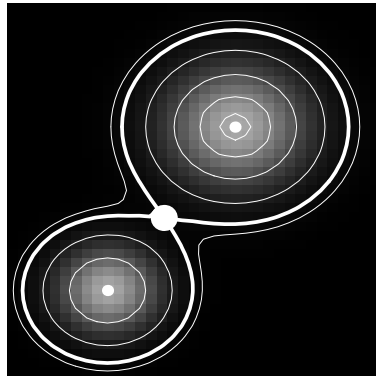
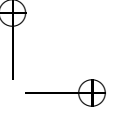
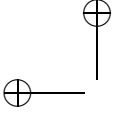


Figure 3.9: Isointensity contours around two extrema, the largest isophote that separates the two extrema is the isophote through the saddle. Larger isophotes encircle both extrema.

The notion of an attraction area is meant to delimit ‘meaningful’ regions containing only one extremum, but is not necessarily itself ‘meaningful’ (what is and what is not depends on a user-defined model).



3.4.2 Attraction Areas in Scale Space

Attraction areas are defined for every scale, they change form gradually through scale, except when they are involved in creations or annihilations. Attraction areas tend to become smaller through scale up until the point where they annihilate. An attraction area can, after the annihilation of its extremum, be absorbed in the remainder of the image (the part of the image that has no attraction area), in that case the attraction area disappears as shown in Figure 3.10a. In the other case it will be absorbed in the attraction area of another extremum. If an attraction area is absorbed by another attraction area after its extremum annihilated, it will suddenly change form. The remaining attraction area will contain the area where the just annihilated extremum was located. This is demonstrated in Figure 3.10b. When an extremum is created the same events can happen as demonstrated for the annihilation case, but in opposite scale direction. An extremum can be created out of the remainder of the image as shown in Figure 3.10c, or it can be created out of an existing attraction area. In that case the attraction area will split in two separate attraction areas, as demonstrated in Figure 3.10d.

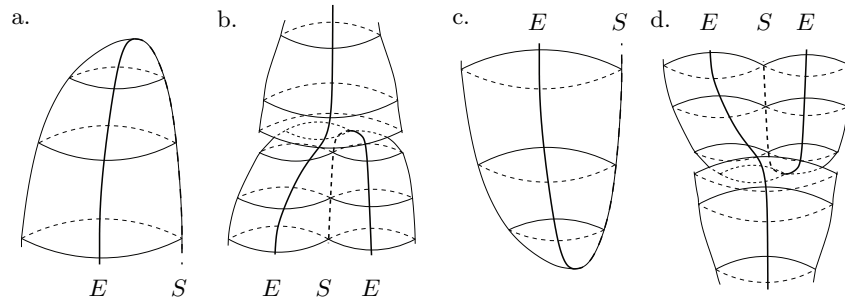
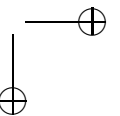
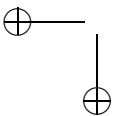
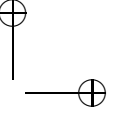
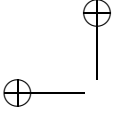


Figure 3.10: Different changes in attraction areas after an extremum is annihilated or created. Also the critical curves are drawn. (Illustration adapted from Lindeberg [87]).

Thus catastrophe scales are the only scales at which sudden form changes occur, as opposed to Kuijper's counterintuitive segments.

Attraction areas do not cover the image domain. In order to obtain a full covering as well as a natural hierarchy one will have to introduce compound attraction areas, containing more than one extremum, consistent with the unfolding of image structure over scale. The elementary attraction areas discussed here pertain to the leaves or single branches of a hierarchical graph at a fixed scale, whereas the compound attraction areas would correspond to subgraphs.





3.4.3 Scale-Space Hierarchy

The events illustrated in Figure 3.10 imply a certain hierarchy in scale space of the attraction areas. Every attraction area in the image can be seen as a single branch of the graph, and the remainder (the part of the image that has no attraction area) as the root of the graph.

From the following rules the hierarchical tree can be built:

- An attraction area is connected to another attraction area if it, after the annihilation of its extremum, is absorbed in that attraction area.
- An attraction area is connected to another attraction area if its extremum is created from that attraction area.
- An attraction area is connected to the root of the graph if it, after the annihilation of its extremum, is absorbed in the remainder of the image.
- An attraction area is connected to the root of the graph if its extremum is created from the remainder of the image.

So the entire hierarchy is defined by events occurring at creations and annihilations. Due to creations followed by annihilations the graph may contain closed loops. An example in which a creation followed by an annihilation takes place is the dumbbell example demonstrated in Figure 3.11 and 3.10.

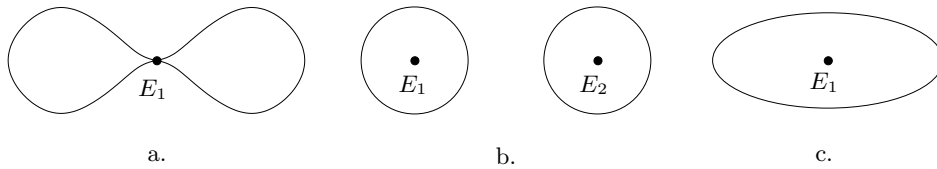
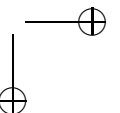
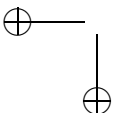


Figure 3.11: a) A schematic drawing of a dumbbell image consisting of two blobs with a bright ‘bridge’ in between them. b) After blurring, the middle extremum splits into two extrema. c) When blurred even more these two merge again.

3.4.4 The Algorithm

An algorithm to find the pre-segmentation and the scale space hierarchy is easily made using the theory of the previous sections. Assuming that the critical curves have already been located, the algorithm could be described by the steps explained in Algorithm 2.



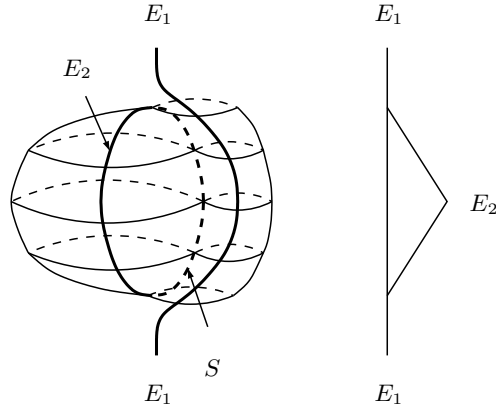
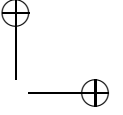
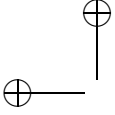


Figure 3.12: The critical curves and attraction areas of the dumbbell example from Figure 3.11. Through scale extremum E_2 is created from the remainder of the image and later it annihilates again, resulting in a closed loop in the hierarchical graph shown on the right.

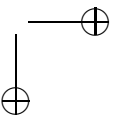
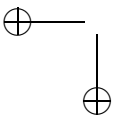
Algorithm 2 top-point graph construction procedure

- 1: For every extremum find its attraction area just before it annihilates and look at the attraction areas after its annihilation.
 - 2: Start from coarse scale and use the remaining extremum as the root of the graph.
 - 3: Link extrema to the root or to each other, depending on their behavior as illustrated in Figure 3.10.
 - 4: Continue these steps until the finest scale is reached.
-

Note that this algorithm is truly coarse-to-fine, unlike Kuijper's algorithm, which insists on finding scale-space saddles at lower scales.

3.4.5 Results

The algorithm using attraction areas for finding the multiscale hierarchy has been implemented in MathematicaTM. The results for a simple test image are shown in Figure 3.13.



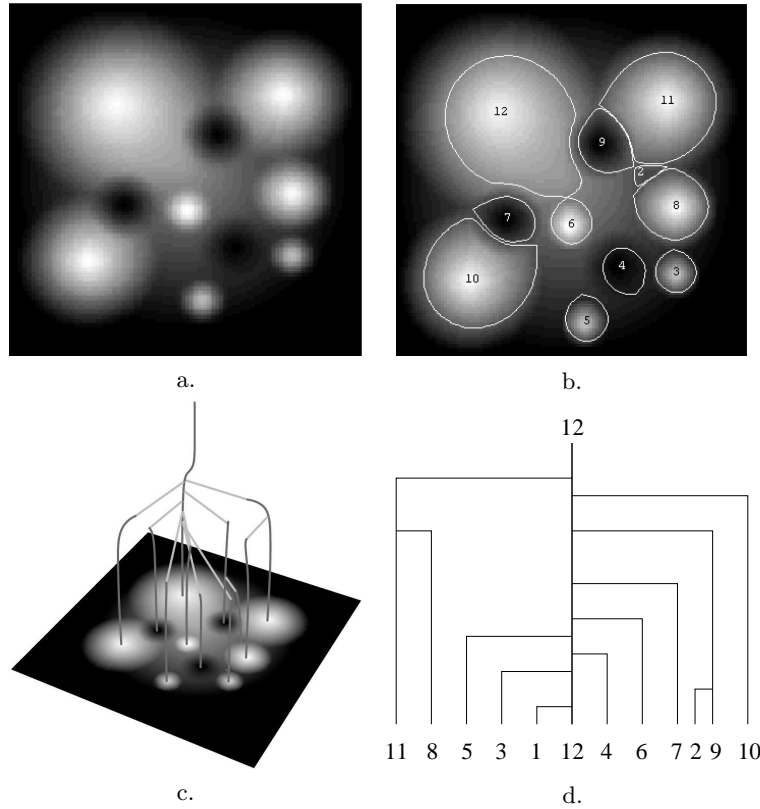
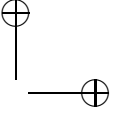
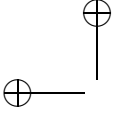


Figure 3.13: a) The input image. b) The segments (attraction areas) at fine scale. The extrema are numbered. c) The hierarchical graph represented by linking the extrema paths. d) The resulting hierarchical graph.

3.4.6 Remarks

The attraction areas method does not have the problems that the previously discussed methods (Section 3.2 and 3.3) had. The attraction areas for instance exist up until the extrema they belong to annihilate. This means that the algorithm is genuinely coarse-to-fine. This in contrast to the segments assigned to the scale-space saddles in Kuijper's approach which disappeared before reaching the annihilation scale.

The algorithm can be implemented more easily and faster than the method proposed by Kuijper, because the attraction areas don't have to be calculated for every scale, but only twice per annihilation (for a scale before and after the anni-



hilation scale).

3.5 Other Hierarchical Segmentation Methods

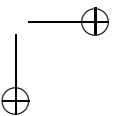
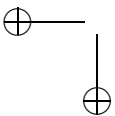
Next to the afore mentioned methods, other algorithms exist that extract a multi-scale hierarchical segmentation from the scale-space representation of an image. The most popular one being the *watershed segmentation* [132], a multi-scale watershed algorithm using the gradient magnitude to define the watersheds is defined by Fogh Olsen and Nielsen in [108]. This multi-scale watershed method generates a hierarchical segmentation and in [106] Fogh Olsen summarizes the possible transitions generically occurring in this hierarchical tree representation. The watershed segmentation algorithm is truly coarse-to-fine.

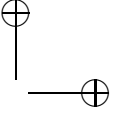
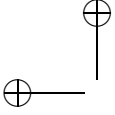
Somchaipeng et al. [126, 127] have recently introduced yet another method for generating a multi-scale hierarchy. This method uses energy maps to define a linking scheme between top-points in scale space. The method however is very slow and the complexity of the algorithm is exponential with the number of critical points. This makes the algorithm unusable for images containing more than 7 critical points. In general most images contain thousands of critical points. This means that in its current form Somchaipeng's method is not applicable.

3.6 Using the Hierarchical Representation for Matching

Apart from image segmentation the previously described methods might be used for image matching. A feasibility study on matching images using the hierarchical representations of the previously described methods has been conducted. To match hierarchical representations, a cost function has to be created that describes the cost of transforming one hierarchical representation into another one. The lower the cost, the more alike two representations are and thus the more alike the two images are that are represented by the hierarchical representations.

Problems that arise in matching the hierarchical representations of images are plenty. The biggest one being the sensitivity of the representation to change in the image, like noise, different lighting and transformations. A small perturbation of the image can already result in a cascade of changes in the hierarchical representation of the image. These changes have to be modeled and a cost function used for matching the representations has to be created. This cost function must be able to handle these differences by assigning a low cost to changes caused by





small perturbations of the image. On the other hand this cost function has to assign high costs when something is structurally different in the image. The task of creating such a cost function seems to be infeasible as it is unpredictable what cascade of changes occurs when the image is perturbed.

By a thorough removal of unstable nodes from the hierarchical representation it might be possible to obtain a certain robustness to small perturbations. However more rigorous image changes like occlusion, re-lighting, contrast change and non-scale-Euclidean transformations also have an unpredictable and far going impact on the hierarchical representation.

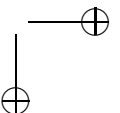
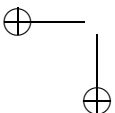
These characteristics make it unfeasible to use the hierarchical representations presented in this chapter for image matching.

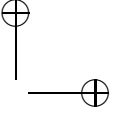
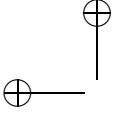
3.7 Conclusion

A number of methods for generating a hierarchical graph embodying the singularities of the deep structure of an image have been discussed. The method of Lifshitz and Pizer [83] has problems assigning segments to single extrema in a topologically sound way. It appears that Lifshitz and Pizer have missed the fact that the topology of the level surfaces through extrema on the extremum path changes significantly when the intensity of a connected scale-space saddle is passed. Also using the annihilation scale of a critical curve as a measure of importance of the segment assigned to the path is not correct. Unstable critical points in flat areas in the image can go up to high scales. On the other side important, but thin structures are blurred away at low scales, and thus are assigned a very low importance.

The method suggested by Kuijper [75] yields a topology based hierarchy, but is difficult to implement, not strictly coarse-to-fine and conceptually questionable. The segmentation resulting from Kuijper's algorithm is not complete, there are scales at which extrema are apparent in the image, but have no segments assigned to them.

To overcome the shortcomings of the two existing methods, a new method was suggested. This new method uses attraction areas to define segments in the image at different scales and to find a scale-space hierarchy. The proposed method has none of the shortcomings of the discussed existing methods. The attraction areas exist up until the extrema they belong to annihilate, in contrast to the segments assigned to scale-space saddles that disappear before reaching the annihilation scale. It however remains to be seen whether the new algorithm is free of other

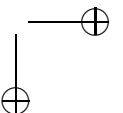
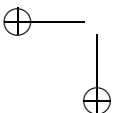


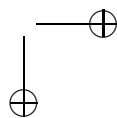
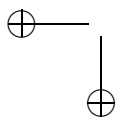
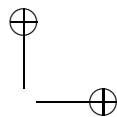
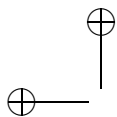


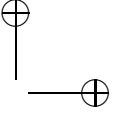
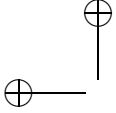
shortcomings.

Model knowledge has to be implemented in the hierarchical graphs either deterministically or probabilistically, to make the segmentation more usable. Much still has to be investigated on this matter.

Using the hierarchical representations presented in this chapter for image matching seems to be unfeasible as the representations are highly sensitive to small perturbations in the image, occlusion, re-lighting, contrast change and non-scale-Euclidean transformations. A small change in the image can result in a cascade of changes in the hierarchical representation. Creating a cost function for matching the representation, that can sensibly take into account these induced changes, appears to be impossible. In the following chapters more robust image representations and matching algorithms, based on the deep structure of images, will be discussed.







Do not follow where the path may lead. Go instead where there is no path and leave a trail.

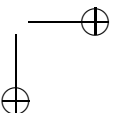
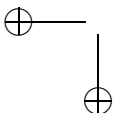
Ralph Waldo Emerson (1803–1882)

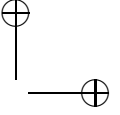
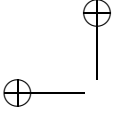
4

Discrete Representation of Top-Points via Scale Space Tessellation

This chapter is based on:

B. Platel, F. Demirci, A. Shokoufandeh, L.M.J. Florack, F.M.W. Kanters, B.M ter Haar Romeny and S.J. Dickinson, Discrete Representation of Top Points via Scale Space Tessellation. In *Scale Space and PDE Methods in Computer Vision*, pages 73–84, 2005



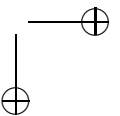
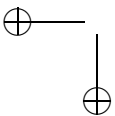


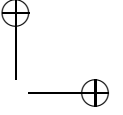
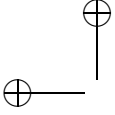
4.1 Introduction

Previous research has shown that top-points (singular points in the scale-space representation of generic images) have proven to be valuable sparse image descriptors that can be used for image reconstruction [57, 104], image matching [58, 111] and are potentially useful for motion extraction [32, 48]. In previous work, images were compared using a point matching scheme which took into account the positions, scales, and differential properties of corresponding top-points [58, 57]. The underlying matching framework was based on the Earth Mover's Distance, a powerful, many-to-many point matching framework. However, treating the points as an unstructured collection ignores the neighborhood structure that may exist within a given scale or across scales. Grouping certain top-points together explicitly encodes the neighborhood structure of a point, effectively enriching the information encoded at a point, information that can be exploited during both indexing [122] and matching [123].

In this paper, an unstructured set of top-points is taken and a neighborhood structure is imposed on them. Inspired by the methods described in the previous chapter, the scale-space structure of a set of top-points is encoded in a *directed acyclic graph* (DAG). Specifically, the position-based grouping of the top-points provided by a Delaunay triangulation is combined with the scale-space ordering of the top-points to yield a directed acyclic graph. This new representation allows for the utilization of powerful graph matching algorithms to compare images represented in terms of top-point configurations, rather than using point matching algorithms to compare sets of isolated top-points. The matching algorithm draws on recent work in many-to-many graph matching [61, 15, 16], which reduces the matching problem to that of computing a distribution-based distance measure between embeddings of labeled graphs.

The new construction is described by first introducing top-points as anchor points for the representation. Next, a measure for the stability of these top-points is introduced that will be used to prune unstable top-points. Section 4.4 describes the construction of the DAG through a Delaunay triangulation scheme. Section 4.5 proposes a scale-space distance measure. Section 4.6 reviews the many-to-many DAG matching algorithm, which will be used to evaluate the construction. Section 4.7 describes the experiments conducted on the new graph construction. The first experiment examines the stability of the construction under Gaussian noise of increasing magnitude applied to the original images. The second experiment examines the invariance of the graph structure to within-class image deformation, which may include minor displacements of points both within and across scales.





4.2 Top-Points as Anchor Points

As explained in Section 2.4, top-points are points at which creation and annihilation events take place. A top-point is a critical point at which the determinant of the Hessian (i.e. the matrix of second order derivatives) degenerates (as discussed in Section 2.4.1):

$$\begin{cases} \nabla L = \mathbf{0} \\ \det(\mathbf{H}) = 0. \end{cases} \quad (4.1)$$

One way to obtain these top-points is by means of zero-crossings in scale space. This involves derivatives up to second order and yields sub-pixel results. Other, more elaborate methods, can be used to find or refine the top-point positions. For details, the reader is referred to Section 2.4.3. In Figure 4.1, the critical paths and their top-points are shown for a picture of a face.

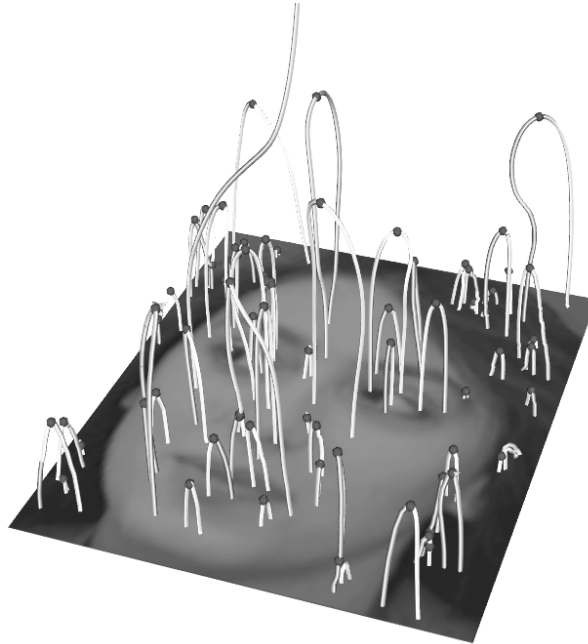
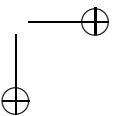
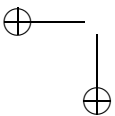
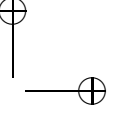
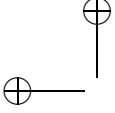


Figure 4.1: Critical paths and top-points of a face.





4.3 Stability by Structure

It is obvious that the positions of extrema at very fine scales are sensitive to noise. This, in most cases, is not a problem. Most of these extrema are blurred away at fine scales and won't affect the matching scheme at slightly coarser scales. However, problems do arise in areas in the image that consist of almost constant intensity. One can imagine that the positions of the extrema (and as a result the critical paths and top-points) are very sensitive to small perturbations in these areas. These unstable critical paths and top-points can continue up to high scales since there is no structure in the vicinity to interact with. This is why merely the scale of an anchor point in scale-space cannot be a good measure of importance or stability as often seen in literature. To account for these unstable top-points, a measure of stability is needed, so that unstable points can either be given a low weight in the matching scheme, or they can be disregarded completely.

A top-point is more stable in an area with a lot of structure. The amount of structure contained in a *spatial* area around a top-point can be quantified by the *total (quadratic) variation* (TV) norm over that area:

$$TV(\Omega) \stackrel{\text{def}}{=} \frac{\sigma^2 \int_{\Omega} \|\nabla u(\mathbf{x})\|^2 dV}{\int_{\Omega} dV} \quad (4.2)$$

The TV-norm is calculated over a circular area Ω with radius $\lambda\sigma$ around a top-point at position (\mathbf{x}_c, t_c) .

$$\Omega : \|\mathbf{x} - \mathbf{x}_c\|^2 \leq \lambda^2 \sigma^2. \quad (4.3)$$

Note that the size of the circle depends on the scale $\sigma = \sqrt{2t}$ and on the dimensionless scaling factor λ , Figure 4.2.

To get rid of the arbitrary parameter λ , consider a critical point u (from here on $u(\mathbf{x}_c, t_c)$ will be denoted as u and Einstein summation convention will be used for simplicity). A spatial first-order Taylor series for the gradient $u_i(\mathbf{x})$ yields:

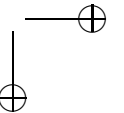
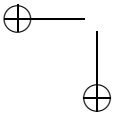
$$u_i(\mathbf{x}, t) = u_{ij}(\mathbf{x} - \mathbf{x}_c)^j. \quad (4.4)$$

Note that at a critical point the first order derivatives are zero. This result can be used in the denominator of the TV-norm expression (4.2), yielding

$$\sigma^2 \int_{\Omega} u_i(\mathbf{x}, t) u_i(\mathbf{x}, t) dV = \sigma^2 u_{ij} u_{ik} \int_{\|\mathbf{x} - \mathbf{x}_c\|^2 \leq \lambda^2 \sigma^2} (\mathbf{x} - \mathbf{x}_c)^j (\mathbf{x} - \mathbf{x}_c)^k dx. \quad (4.5)$$

The r.h.s. of Equation (4.5) can be converted to polar coordinates using

$$\begin{cases} x^1 &= x_c^1 + r \cos \phi \\ x^2 &= x_c^2 + r \sin \phi. \end{cases} \quad (4.6)$$



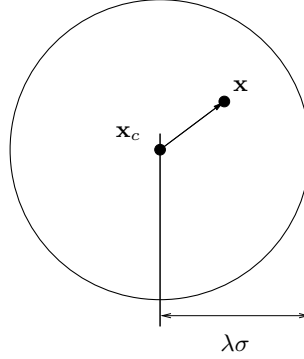
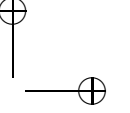
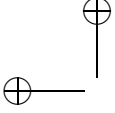


Figure 4.2: Integration area around critical point \mathbf{x}_c .

In polar coordinates Equation (4.5) can be written as

$$\sigma^2 u_{ij} u_{ik} \int_0^{\lambda\sigma} \int_0^{2\pi} r^3 \begin{pmatrix} \cos^2 \phi & \cos \phi \sin \phi \\ \cos \phi \sin \phi & \sin^2 \phi \end{pmatrix}_{jk} dr d\phi. \quad (4.7)$$

Which simplifies to

$$\sigma^2 u_{ij} u_{ij} \begin{pmatrix} \pi & 0 \\ 0 & \pi \end{pmatrix}_{jk} \int_0^{\lambda\sigma} r^3 dr = \pi \sigma^2 u_{ij} u_{ij} \left[\frac{1}{4} r^4 \right]_0^{\lambda\sigma} = \frac{\pi}{4} \lambda^4 \sigma^6 \text{trace}(\mathbf{H}^2). \quad (4.8)$$

The numerator of the TV-norm Equation (4.4) is defined by the area Ω which equals

$$\int_{\|\mathbf{x}\|^2 \leq \lambda^2 \sigma^2} d\Omega = \pi \lambda^2 \sigma^2. \quad (4.9)$$

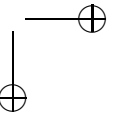
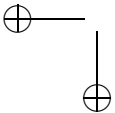
The TV-norm (4.2) with substitution of Equations (4.8, 4.9) thus becomes:

$$TV(\lambda) = \frac{\sigma^2 \int_{\Omega} \|\nabla u\|^2 dV}{\int_{\Omega} dV} = \frac{1}{4} \lambda^2 \sigma^4 \text{trace}(\mathbf{H}^2) + (\text{h.o.t.}), \quad (4.10)$$

where h.o.t. stands for terms of $\mathcal{O}(\lambda^2)$ and higher. From this the *differential TV-norm* can be defined by the following limiting procedure:

$$tv \stackrel{\text{def}}{=} \lim_{\lambda \rightarrow 0} 4 \frac{1}{\lambda^2} TV(\lambda) = \sigma^4 \text{trace}(\mathbf{H}^2). \quad (4.11)$$

The proportionality factor 4 is irrelevant for our purposes and introduced merely for convenience. The normalization factor $1/\lambda^2$ is needed prior to evaluation of the



limit since $TV(\lambda) = \mathcal{O}(\lambda^2)$. Equation (4.11) has been referred to by Koenderink as *deviation from flatness*, which can indeed be seen to be the differential counterpart of Equation (4.2). It is now possible to calculate a stability measure for a top-point *locally* by using only its second order derivatives. This stability measure can be used to weigh the importance of top-points in the matching scheme, or to remove any unstable top-points by thresholding them on their stability value. The latter is demonstrated in Figure 4.3.

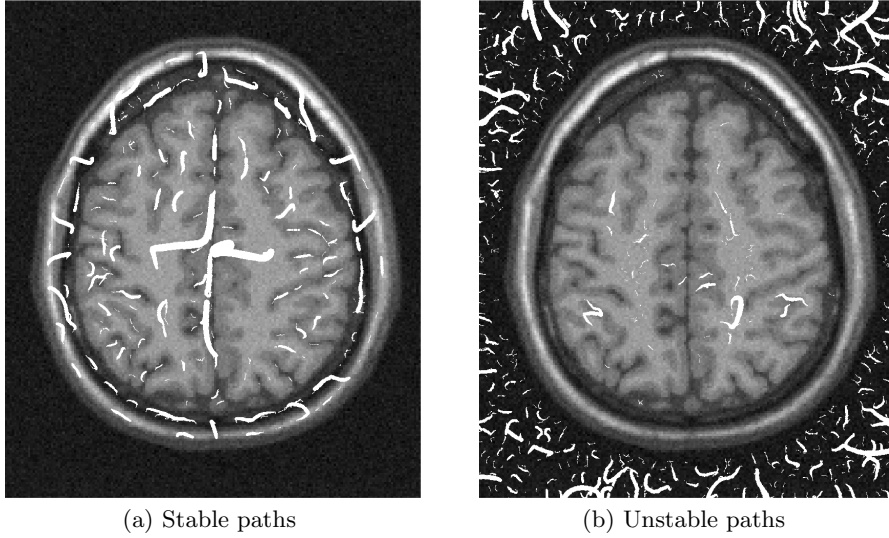
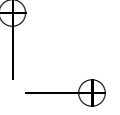
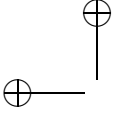


Figure 4.3: Spatial projection of critical paths of an MR brain scan image. The paths are filtered by thresholding the stability measure (Equation (4.11)) of their top-points. Most instabilities occur in flat regions, as expected.

4.4 Construction of the Graph

The goal of the proposed graph construction is two-fold. First, it has to encode the neighborhood structure of a set of points, explicitly relating nearby points to each other in a way that is invariant to minor perturbations in point location. Moreover, when local neighborhood structure does indeed change, it is essential that such changes will not affect the encoded structure elsewhere in the graph (image). The Delaunay triangulation imposes a position-based neighborhood structure with exactly these properties [113]. It represents a triangulation of the points which is equivalent to the nerve of the cells in a Voronoi tessellation, i.e. that triangulation of the convex hull of the points in the diagram in which



every circumcircle of a triangle is an empty circle [105], Figure 4.4.

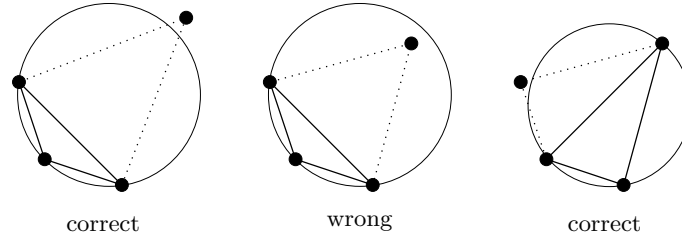


Figure 4.4: The Delaunay triangulation has the property that the circumcircle of every triangle does not contain any points of the triangulation.

Examples of a Delaunay triangulation and a Voronoi diagram are given in Figure 4.5. The edge set of the resulting graph will be based on the edges of the triangulation.

The second goal is to capture the scale-space ordering of the points to yield a directed acyclic graph, with coarser scale top-points directed to nearby finer scale top-points.

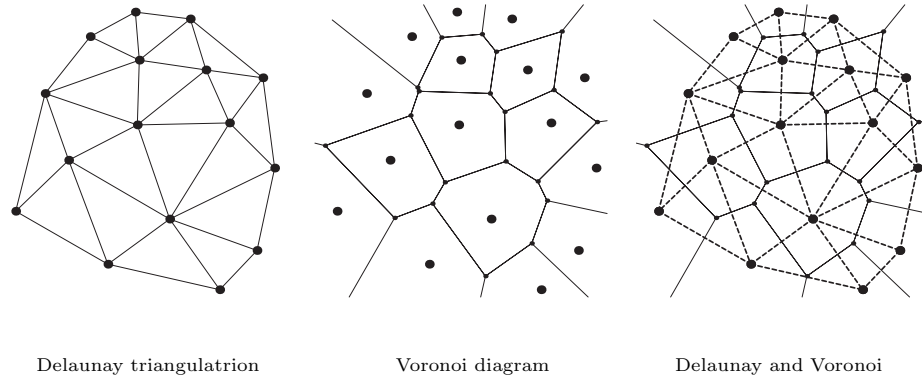
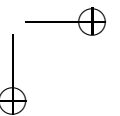
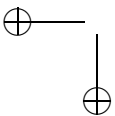
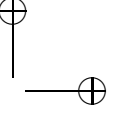
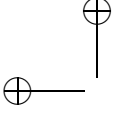


Figure 4.5: The Delaunay triangulation and Voronoi diagram of a small pointset.

The first step in constructing graph G is the detection of top-points and critical paths as explained in Section 2.4.3. The root of G , denoted as v_1 , will correspond to the single critical path that continues up to infinity; note that the top-point associated with this critical path lies at infinite scale σ_∞ and in the center of gravity of the image [91]. All other nodes in G , denoted as v_2, \dots, v_n , correspond to the detected top-points and their corresponding critical paths. v_2, \dots, v_n are ordered in decreasing order of the scale at which they are detected, e.g., v_2 is detected at





a coarser scale than v_n for $n > 2$. As the Delaunay triangulation of the points is built, the DAG is simultaneously constructed. Beginning with the root, v_1 , it gives a singleton point in the Delaunay triangulation, and a corresponding single node in G . Next, at the scale corresponding to v_2 , v_1 's position is projected down to v_2 's level, and the 'triangulation' is recomputed. In this case, the 'triangulation' yields an edge between v_1 and v_2 . Each new edge in the triangulation yields a new edge in G , directed from coarser top-points to finer top-points; in this case, a directed edge in G is added from v_1 to v_2 . This process is continued with each new top-point, first projecting all previous top-points to the new point's level, recomputing the triangulation, and using the triangulation to define new directed edges in G . A summary of this procedure is presented in Algorithm 3.

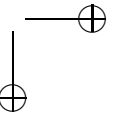
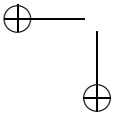
The construction is illustrated for a simple image in Figure 4.6. In the top two frames in the left figure, the transition in the triangulation from v_2 (point 2) to v_3 (point 3) is shown; the root is annotated as point 1. In the upper right frame, the triangulation consists of three edges; correspondingly, G has three edges: $(1, 2), (1, 3), (2, 3)$, where (x, y) denotes an edge directed from node x to node y . In the lower left figure, point 4 is added to the triangulation, and the triangulation recomputed; correspondingly, edges $(1, 4), (2, 4), (3, 4)$ are added to G (note that $(1, 2)$ is no longer in the triangulation, but remains in G). Finally, in the lower right frame, point 5 is added, and the triangulation recomputed. The new edges in the triangulation yield new edges in G : $(2, 5), (4, 5), (1, 5)$. The right side of Figure 4.6 illustrates the resulting graph (note that the directions of the edges are not shown). Figure 4.7 is the result of applying this construction to the face of Figure 4.1.

Algorithm 3 top-point graph construction procedure

- 1: Detect the critical paths.
 - 2: Extract the top-points from the critical paths.
 - 3: Label the extremum path continuing up to infinity as v_1 .
 - 4: Label the rest of the nodes (critical paths, together with their top-points) according to the scale of their top-points from high scale to low as v_2, \dots, v_n .
 - 5: For $i = 2$ to n evaluate node v_i :
 - 6: Trace the critical paths down to the scale of the considered node v_i .
 - 7: Calculate the 2D Delaunay triangulation of all the extrema at that scale.
 - 8: All connections to v_i in the triangulation are stored as directed edges in G .
-

4.5 Distance Measure in Scale Space

The distance between two points in scale space cannot simply be measured by the Euclidean distance. The scale-space representation of an N -dimensional image is



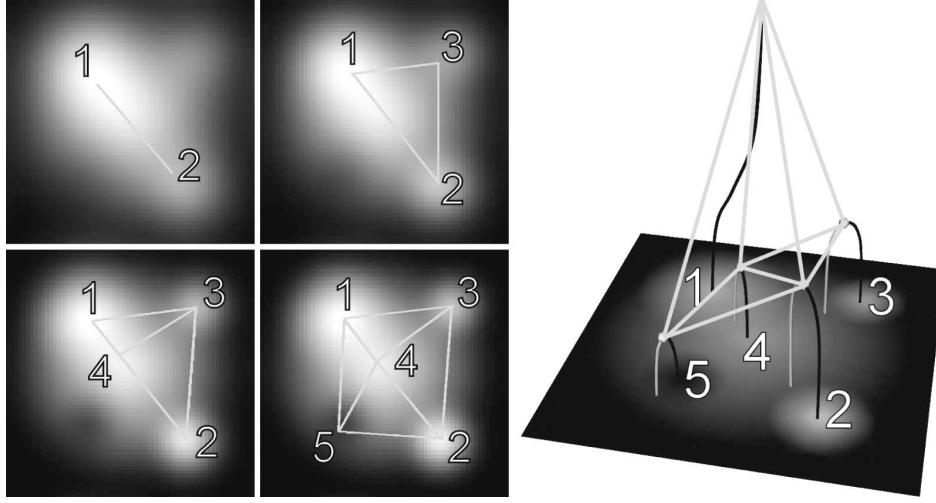


Figure 4.6: The DAG obtained from applying Algorithm 3 to the critical paths and top-points of the face in Figure 4.1.

not a simple Euclidean $(N + 1)$ -dimensional space.

To handle this, Eberly [20] proposed a metric that depends on a parameter $\rho > 0$, weighing the relative importance of spatial and scale measurements. The metric is defined by:

$$ds^2 = \sum_{i=1}^N \frac{dx_i^2}{\sigma^2} + \frac{1}{\rho^2} \frac{d\sigma^2}{\sigma^2}, \quad (4.12)$$

in which x_i represent the spatial positions of a point in scale space and σ represents the scale of the point. For simplicity only two-dimensional images will be considered.

In order to define a distance S between two points (x_1, y_1, σ_1) and (x_2, y_2, σ_2) , a geodesic connecting these points has to be found. By following van Wijk et al. [135] the distance S between two arbitrary points can be written as:

$$\begin{aligned} S &= (r_2 - r_1)/\rho \\ r_i &= \ln(\sqrt{b_i^2 + 1} - b_i), \quad i = 1, \dots, N \\ b_i &= \frac{\sigma_2^2 - \sigma_1^2 - (-1)^i \rho^2 R^2}{2\sigma_i \rho R}, \quad i = 1, \dots, N \end{aligned} \quad (4.13)$$

For two points exactly above each other $(x_1, y_1) = (x_2, y_2)$, $S = |\ln(\sigma_2/\sigma_1)|/\rho$. This metric can be used to measure distances between the vertices in the proposed

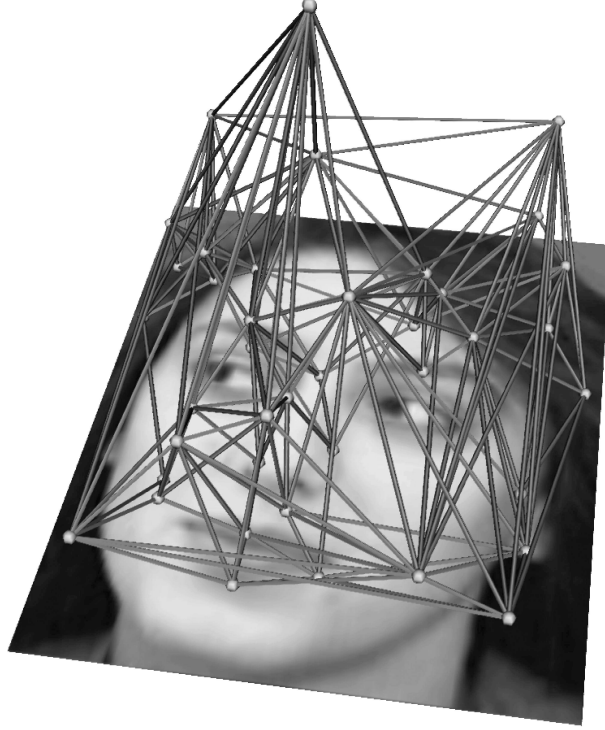
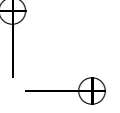
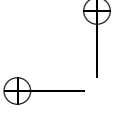


Figure 4.7: Visualization of the DAG construction algorithm. Left: the Delaunay triangulations at the scales of the nodes. Right: the resulting DAG (edge directions not shown).

directed acyclic graph. However the introduced tunable parameter $\rho(> 0)$, has to be found experimentally (we use $\rho = 0.7$ as it was found to be the best performing parameter in a set of experiments).

4.6 Overview of the Matching Algorithm

The matching algorithm is based on the metric-tree representation of labeled graphs and their low-distortion embeddings into normed vector spaces via spherical coding [15, 16, 17, 61]. The advantage of this embedding technique is that it prescribes a single vector space into which both graphs are embedded. This two-step transformation reduces the many-to-many matching problem to that of computing a distribution-based distance measure between two such embeddings.



To compute the distance between two sets of weighted vectors, a variation of Earth Mover's Distance under transformation sets is used. For two given graphs, the algorithm provides an overall similarity (distance) measure.

4.6.1 Metric Embedding of a Graph

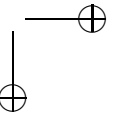
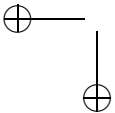
The problem of many-to-many graph matching has often been studied in the context of edit-distance [90, 121], where one graph is transformed into the other by a minimal set of re-labelings, additions, deletions, merges and splits of nodes and edges. The edit-distance however has a number of drawbacks. 1) it is computationally expensive especially for graphs (polynomial time algorithms are only available for trees); 2) the current tree-edit distance algorithms do not incorporate edge weights; and 3) a meaningful cost of an edit operation is hard to obtain.

For the matching of the directed acyclic graphs proposed here a metric embedding is used. Consider a mapping $f : \mathcal{A} \rightarrow \mathcal{B}$, where \mathcal{A} is a set of points in the original metric space, with distance function $\mathcal{D}(\cdot, \cdot)$ and \mathcal{B} is a set of points in the (host) d -dimensional k -normed space $\|\cdot\|_k$. For any pair $p, q \in \mathcal{A}$ it is given that

$$\frac{1}{c} \mathcal{D}(p, q) \leq \|f(p) - f(q)\| \leq \mathcal{D}(p, q) \quad (4.14)$$

for a certain parameter c , known as the *distortion*. From Equation (4.14) it is obvious that the closer c is to 1, the better the target set of points \mathcal{B} mimics the original set \mathcal{A} .

The embedding of $(\mathcal{A}, \mathcal{D})$ in a k -normed space $(\mathcal{B}, \|\cdot\|_k)$ simplifies problems defined over difficult metric spaces to problems over easier normed spaces. The embedding of the directed acyclic graph, resulting from Algorithm 3 on page 56, is described in detail in [15, 16, 17, 61]. In short, the $n \times n$ distance matrix \mathcal{D} , defined as the shortest-path distance metric for the directed acyclic graph, is approximated by a tree metric \mathcal{T} . This approximation or fitting problem is known as the *numerical taxonomy* problem. The resulting metric tree \mathcal{T} will in turn be encoded into a Euclidean space of fixed dimensions by means of *spherical coding* [16, 17]. The dimension of the target space has a direct effect on the quality of the embedding. Specifically, as the dimensionality of the target space increases, the quality of the embedding will improve. Still, there exists an asymptotic bound beyond which increasing the dimensionality will no longer improve the quality of the embedding [17] (see Figure 4.8).



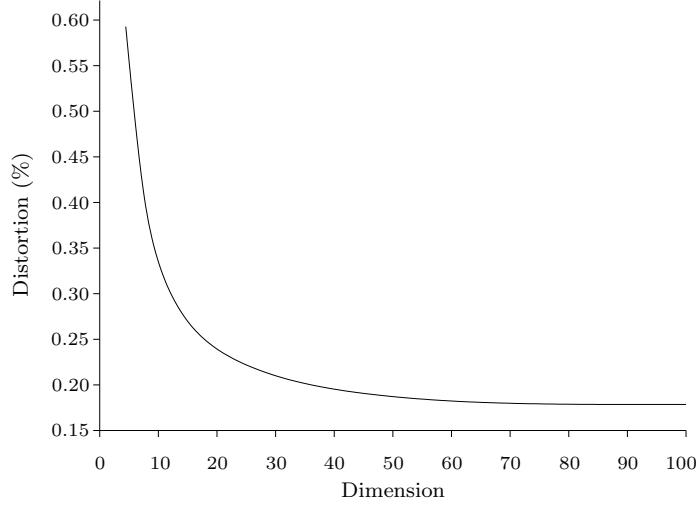


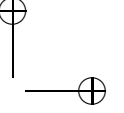
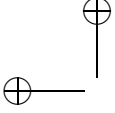
Figure 4.8: Trade-off between distortion and dimension. Increasing the dimensionality of the embedding space will reduce the the distortion. However, this trend does not continue indefinitely to produce isometric embeddings (image adopted from [17]).

4.6.2 Many-to-Many Distribution Based Matching

By embedding the directed acyclic graphs into a normed space, the problem of many-to-many matching of graphs has been reduced to the problem of many-to-many matching of weighted distributions of points in a normed space.

Given a pair of weighted distributions in the same normed space, the Earth Mover's Distance (EMD) [12, 116, 58, 15] framework is then applied to find an optimal match between the distributions. The EMD transforms a given distance measure between single features (the *ground distance*) to a distance measure for distributions of features. The main advantage of the EMD is the fact that it can handle many distances and allows partial matches in a natural way. This property allows the EMD to deal with unequal clusters and noisy data sets.

The computation of the EMD is derived from the well-known *transportation problem* [1], also known as the Monge-Kantorovich problem [114], in which the optimal value determines the minimum amount of 'work' required to transform one distribution into the other. This resembles the minimal amount of energy needed to transport piles of earth from one site to fill up a set holes on another site, hence the name 'earth mover's distance' (Figure 4.9).



Formally, let $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$ be the first distribution with m points, and let $Q = \{(q_1, w_{q_1}), \dots, (q_n, w_{q_n})\}$ be the second distribution with n points. Let $D = d_{ij}$ be the ground distance matrix, where d_{ij} represents the ground distance between points p_i and q_j . The objective is to find a flow matrix F , describing the flow between points p_i and q_j that minimizes the overall work:

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad (4.15)$$

subject to the following list of constraints:

$$f_{ij} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n, \quad (4.16)$$

$$\sum_{j=1}^n f_{ij} \leq w_{p_i}, \quad 1 \leq i \leq m, \quad (4.17)$$

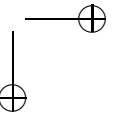
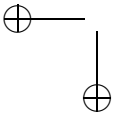
$$\sum_{i=1}^m f_{ij} \leq w_{q_j}, \quad 1 \leq j \leq n, \quad (4.18)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left(\sum_{i=1}^m w_{p_i}, \sum_{j=1}^n w_{q_j} \right). \quad (4.19)$$

Constraint (4.16) allows moving mass from P to Q and not vice versa. Constraint (4.17) limits the amount of mass that can be sent by the clusters in P to their weights. Constraint (4.18) limits the clusters in Q to receive no more mass than their weights; and constraint (4.19) forces to move the maximum amount of mass possible.

The optimal value of the objective function, $\text{EMD}(P, Q)$, defines the Earth Mover's Distance between the two distributions.

The standard EMD formulation assumes that the two distributions have been aligned. However a translated and rotated version of a graph embedding is also a graph embedding. To accommodate pairs of distributions that are not rigidly embedded, Cohen and Guibas [12] extended the definition of the EMD, to allow one of the sets to undergo a transformation. Their iterative process called 'an optimal flow and an optimal transformation' achieves a local minimum of the objective function. More details on the computation of the optimal transformation can be found in [15, 61].



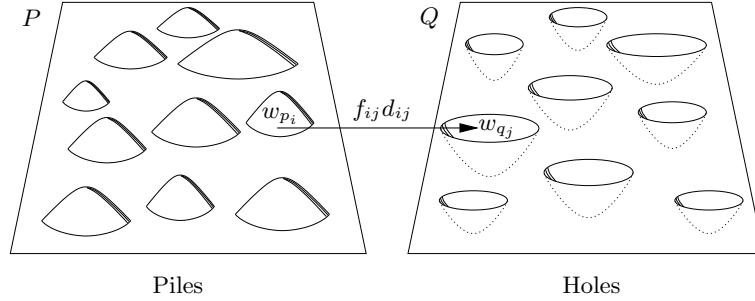


Figure 4.9: Illustration of the earth movers distance algorithm.

4.7 Experiments

To evaluate the robustness of the construction, the sensitivity of the construction to two types of perturbations is explored. The first is the sensitivity of the construction to noise in the image, while the second is within-class deformation. Experiments are conducted using a subset of the Olivetti Research Laboratory face database. The database consists of faces of 20 people with 10 faces per person, for a total of 200 images; each image in the database is 112×92 pixels. The face images are in frontal view and differ by various factors such as gender, facial expression, hair style, and presence or absence of glasses. A representative view of each face is shown in Figure 4.11. Invariance of a graph to noise or within-class deformation requires a measure of graph distance, so that the distance between the original and perturbed graphs can be computed. For the experiments reported here, this distance is computed using the many-to-many graph matching algorithm explained in Section 4.6.1. Note that the developed algorithm is a general algorithm and is in no way specifically designed for face recognition. Therefore it has not been compared to state-of-the-art face recognition algorithms. These experiments are presented only as a proof of concept.

4.7.1 Graph Stability under Additive Noise

To test the robustness of the graph construction, the stability of the graphs is examined under additive Gaussian noise at different signal levels applied to the original face images. For this experiment, the database consists of the original 200 unperturbed images, while the query set consists of noise-perturbed versions of the database images. Specifically, for each of the 200 images in the database, a set of query image was created by adding 1%, 2%, 4%, 8%, and 16% of Gaussian noise. Next, the similarity between each query (perturbed database image) and

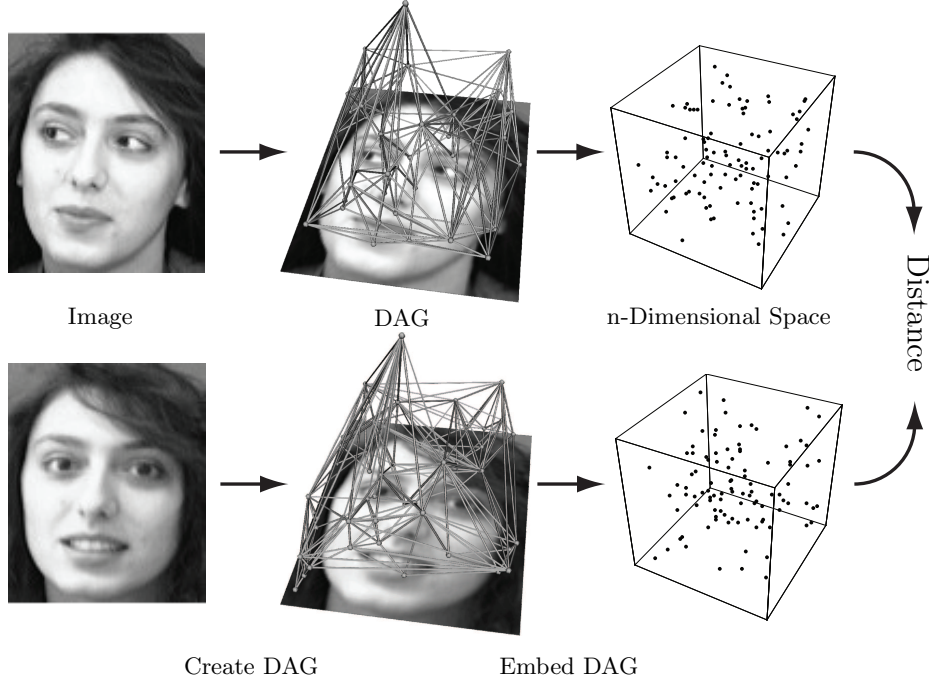
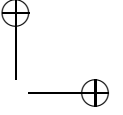
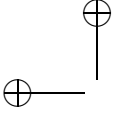


Figure 4.10: For a given face, first its DAG is created according to Section 4.4 (Create DAG). After this each vertex of the DAG is embedded into a vector space of prescribed dimensionality using a deterministic spherical coding (Embed DAG). Finally, the distance between the two distributions is calculated by the modified Earth Mover’s Distance under transformation.

each image in the database was computed. The trial is scored as correct if its distance to the face from which it was perturbed is minimal across all database images. This amounts to 40,000 similarity measurements for each noise level, for a total of 200,000 similarity measurements. The results show that the recognition rate decreases down to 96.5%, 93%, 87%, 83.5%, and 74% for 1%, 2%, 4%, 8%, and 16% of Gaussian noise, respectively. These results indicate a graceful degradation of graph structure with increasing noise.

4.7.2 Graph Stability under Within-Class Variation

To test the stability of the graph construction to within-class variation (e.g., different views of the same face), first the faces in the database are grouped by individual; these will represent our categories. Next, the first image (face) from



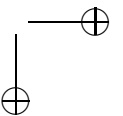
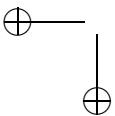
each group is removed (the query) and compared to all remaining database images. The image is then put back in the database, and the procedure is repeated with the second image from each group, etc., until all 10 face images of each of the 20 individuals have been used as a query. If the graph representation is invariant to within-class deformation, resulting from different viewpoints, illumination conditions, presence/absence of glasses, etc., then a query from one individual should match closest to another image from the same individual, rather than an image from another individual. The results are summarized in Table 1 of Figure 4.12.

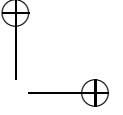
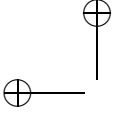
The magnitudes of the distances are denoted by shades of gray, with black and white representing the smallest and largest distances, respectively. Due to symmetry, only the lower half of distance matrix is presented. Intra-object distances, shown along the main diagonal, are very close to zero.

To better understand the differences in the recognition rates for different people, a subset of the matching results was randomly selected among three people in the database, as shown in Table 2, Figure 4.12. Here, the (i, j) -th entry shows the actual distance between face i and face j . It is important to note that the distance between two faces of the same person is smaller than that of different people, as is the case for all query faces. In the experiments, one of the objectives was to see how various factors, such as the presence or absence of glasses, affects the matching results for a single person. Accordingly, a set of images was taken from the database of one person, and the distances have been calculated for each image in this set. The results show that images with the same factors (e.g. wearing glasses or not wearing glasses) are more similar to each other than to others. Table 3 of Figure 4.12 presents a subset of the results. As can be seen from the table, images of the same person with glasses (images 1 and 4) are more similar than those of the same person with and without glasses (images 2 and 3). Still, in terms of categorical matching, the closest face always belongs to the same person. Although these results are encouraging, further evaluation on a larger database needs to be investigated to be more conclusive.

4.8 Conclusions

Imposing neighborhood structure on a set of points yields a graph, for which powerful indexing and matching algorithms exist. In this chapter, a method for imposing neighborhood structure on a set of scale-space top-points was presented. Drawing on the Delaunay triangulation of a set of points, a graph was constructed whose edges are directed from top points at coarser scales to nearby top-points at finer scales. The resulting construction is stable to noise, and within-class variability, as reflected in a set of directed acyclic graph matching experiments.





The results clearly indicate that the graph perturbation due to within-class deformation, including facial expression changes, illumination change, and the presence/absence of glasses is small compared to the graph distance between different classes.

The experiments are however conducted on a database of images which contains little real life object pose variations. The graph construction has to be tested, and made robust against, non-scale Euclidean transformations like affine or projective transformations. More research has also to be done on the effect occlusion and re-lighting (brightness, contrast and lighting changes) have on the graph construction and thus on the matching results.

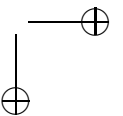
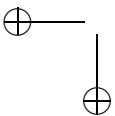




Figure 4.11: The Olivetti Research Laboratories face database.

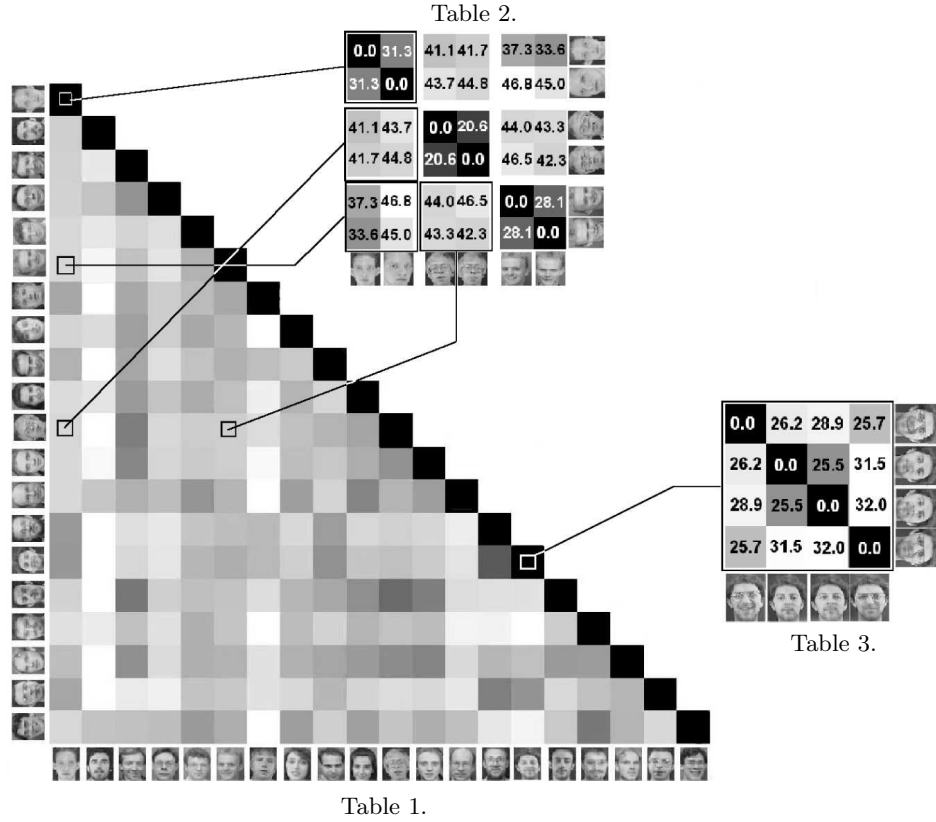
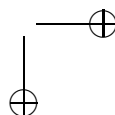
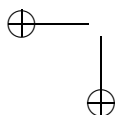
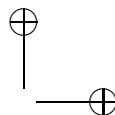
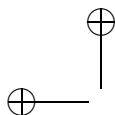
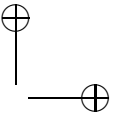
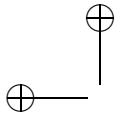


Figure 4.12: Distances in the figure are denoted by numbers and by gray values. The darker the lower the distance. Table 1: Matching results of 20 people. The rows represent the queries and the columns represent the database faces (query and database sets are non-intersecting). Each row represents the matching results for the set of 10 query faces corresponding to a single individual matched against the entire database. The intensity of the table entries indicates matching results, with black representing maximum similarity between two faces and white representing minimum similarity. Table 2: Subset of the matching results with the pairwise distances shown. Table 3: Effect of presence or absence of glasses in the matching for the same person. The results clearly indicate that the graph perturbation due to within-class deformation, including facial expression changes, illumination change, and the presence/absence of glasses is small compared to the graph distance between different classes.





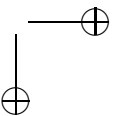
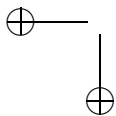
There is always a way to do it better... find it!

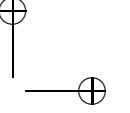
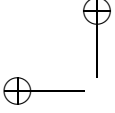
Thomas A. Edison (1847–1931)

Interest Points for Image Matching

This chapter is partly based on:

B. Platel, E.G. Balmachnova, L.M.J. Florack, and B.M. ter Haar Romeny, Top-Points as Interest Points for Image Matching. In *European Conference on Computer Vision (ECCV)*, pages 418–429, 2006





5.1 Introduction

In most image matching, tracking and recognition applications, a set of local features of an image are computed. These features contain local information of the image. Information that preferably does not change when the considered image or object is transformed, corrupted by noise or slightly re-lighted (meaning a difference in brightness, contrast and/or overall lighting conditions). To calculate such a set of characteristic features, a set of characteristic points in the image is needed. These characteristic points are called *interest points*.

These interest points should be as invariant as possible under changes of the image or object in the image. In the case of object retrieval for example, the object to be retrieved could be rotated, scaled, exposed to different lighting conditions or affected by some noise. In the case of group transformations, the set of interest points in the scene image should be transformed in the same way as the object is transformed in the scene. Noise and re-lighting should change the locations of the interest points as little as possible.

There exist many different interest-point detectors. Schmid et al. [119] give an extensive overview of different types of detectors and test their so-called repeatability. In this chapter three interest-point detectors are reviewed: the Harris detector, which scores best in the comparison of Schmid et al.; the SIFT interest-point detector by Lowe [92, 94], probably the most widely used feature detector algorithm in image matching and object retrieval nowadays; and new interest points based catastrophe theory, the top-points.

5.2 Harris Detector

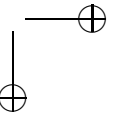
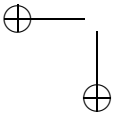
In 1988 Harris & Stephens [40] introduced an interest-point detector which is still very popular in image matching, tracking and recognition applications. Their approach is based on the second moment matrix.

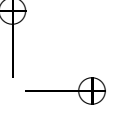
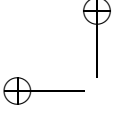
The non linear and dimensionless second moment matrix is often used for feature detection or description of local image structures. This matrix is defined by

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \quad (5.1)$$

for two-dimensional signals/images.

The second moment matrix $\mu(\mathbf{x}, \sigma_I, \sigma_D)$ describes the gradient distribution in a local neighborhood of a point. The gradient derivatives are determined by a local





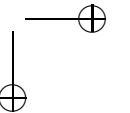
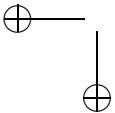
scale σ_D , referred to as the *derivation scale*. The derivatives are averaged in the neighborhood of the point by smoothing with a Gaussian window of size σ_I , the *integration scale*. If both eigenvalues of the second moment matrix, α , β , are small the windowed image region is of approximately constant intensity. If one eigenvalue is large and the other is small, the local image patch is ridge or edge shaped. If both eigenvalues are large, this indicates a corner in the local image patch. Harris & Stephens [40] use this property to extract points for which both eigenvalues are significant. They consider a measure they call the *corner response* defined as:

$$R(\mathbf{x}, \sigma_I, \sigma_D) = \det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - \gamma \text{trace}^2(\mu(\mathbf{x}, \sigma_I, \sigma_D)). \quad (5.2)$$

where γ is a tunable parameter. Harris and Stephens use $\text{trace}(\mu(\mathbf{x}, \sigma_I, \sigma_D))$ and $\det(\mu(\mathbf{x}, \sigma_I, \sigma_D))$ to avoid the explicit eigenvalue decomposition of the second moment matrix $\mu(\mathbf{x}, \sigma_I, \sigma_D)$. Since

$$\begin{aligned} \text{trace}(\mu(\mathbf{x}, \sigma_I, \sigma_D)) &= \alpha + \beta \\ \det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) &= \alpha\beta. \end{aligned} \quad (5.3)$$

The corner response R is positive in corner regions, negative in edge regions and small in flat regions. The well known and widely used interest-point detector, named the *Harris detector* [40] is based on the maxima of this corner response. The effect of the corner response operation and the Harris detector are shown in Figure 5.1.



Mathematica™ Implementation

The following module calculates the values of the second moment matrix (Equation (5.1)) for each image pixel (ie. a list $\{L_x^2, L_x L_y, L_y^2\}$ is returned).

```
secondmomentmatrix[image_,  $\sigma I$ _,  $\sigma D$ _] := Module[{  
  tI =  $\frac{1}{2}\sigma I^2$ ; tD =  $\frac{1}{2}\sigma D^2$ ;  
  Lx = GaussianDerivative[{tD, 1}, {tD, 0}][image];  
  Ly = GaussianDerivative[{tD, 0}, {tD, 1}][image];  
  windowedLxLx = GaussianDerivative[{tI, 0}, {tI, 0}][Lx2];  
  windowedLyLy = GaussianDerivative[{tI, 0}, {tI, 0}][Ly2];  
  windowedLxLy = GaussianDerivative[{tI, 0}, {tI, 0}][LxLy];  
  Map[Partition[#, 2]&,  $\sigma I^2 * \text{Transpose}[\{\text{windowedLxLx}, \text{windowedLxLy},$   
    windowedLyLy], {3, 1, 2}], {2}]];
```

A set of parameters has to be defined. Note that the image size is 500×376 pixels.

```
 $\sigma I$  = 3;  $\sigma D$  = 0.7 $\sigma I$ ;  $\gamma$  = 0.06;
```

The second moment matrix is calculated by

```
 $\mu$  = secondmomentmatrix[image,  $\sigma I$ ,  $\sigma D$ ];
```

The determinant and trace of the second moment matrix can be calculated

```
det $\mu$  = Map[Det[#]&,  $\mu$ , {2}];
```

```
trace $\mu$  = Map[Tr[#]&,  $\mu$ , {2}];
```

Finally the corner response R (Equation (5.2)) can be calculated

```
R = det $\mu$  -  $\gamma$  trace $\mu$ 2;
```

5.2.1 Harris-Laplace Detector

The original Harris interest-point detector operates at a single image scale. Mikolajczyk and Schmid [98] proposed an extension to the Harris detector to make it scale invariant. They named their new approach the *Harris-Laplace detector*.

Their approach consist of two steps: a multi-scale point detection and an iterative selection of scale and location. In the first step a scale-space representation is built for the corner response function (Equation (5.2)) for pre-selected scales $\sigma_n = \xi^n \sigma_0$, where ξ is the scale factor between successive levels (set to 1.4 according to [89, 92]). At each level of the representation the interest points, detected as the local maxima of the corner response, are extracted with an experimentally chosen value of $\gamma = 0.06$. A threshold is used to reject the maxima with a small corner

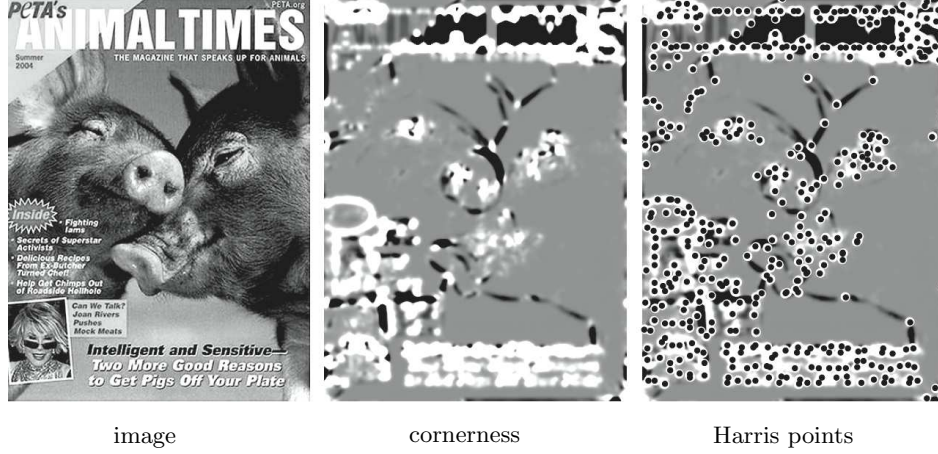


Figure 5.1: Illustration of the corner response of an image as defined in Equation (5.2) and calculated as demonstrated in the Mathematica implementation box. The corresponding maxima of the corner response, known as the Harris points, are plotted on the right.

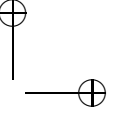
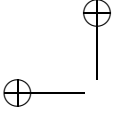
response, as they are less stable under variations in imaging conditions. Since the second moment matrix $\mu(\mathbf{x}, \sigma_I, \sigma_D)$ depends on two scales, the derivation and integration scale, Mikolajczyk and Schmid propose to calculate it using

$$\begin{aligned}\sigma_I &= \sigma_n \\ \sigma_D &= s\sigma_n\end{aligned}\tag{5.4}$$

where s is set to 0.7. In the second step they use an iterative algorithm that simultaneously detects the location and scale of interest points. This algorithm uses extrema of the absolute value of the scalenormalized *Laplacian-of-Gaussian* (LoG) to select the scale of the interest points. The LoG is defined as:

$$\text{LoG}(\mathbf{x}, \sigma_n) = \sigma_n^2 \nabla^2 L,\tag{5.5}$$

Where L is short for $L(\mathbf{x}, \sigma_n) = g(\mathbf{x}, \sigma_n) * I(\mathbf{x})$, the image I at scale σ_n . Points for which the LoG response attains no extremum and for which the response is below a threshold are rejected. For an initial point \mathbf{x} with scale σ_x , the iteration steps are described in Algorithm 4. Their approach results in a set of interest points in scale space that are spatial maxima of the corner response and extrema in scale of the Laplacian-of-Gaussian function. A typical set of Harris-Laplace points of an image is shown in Figure 5.2.



Algorithm 4 Harris-Laplace detector

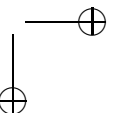
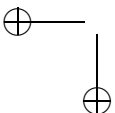
- 1: Find the local extremum over scale of the LoG for point $\mathbf{x}^{(k)}$, otherwise reject the point. The investigated range of scales is limited to $\sigma_x^{k+1} = t\sigma_x^{(k)}$ with $t \in [0.7, \dots, 1.4]$.
 - 2: Detect the spatial location $\mathbf{x}^{(k+1)}$ of a maximum of the Harris measure nearest to $\mathbf{x}^{(k)}$ for the selected σ_x^{k+1} .
 - 3: Goto Step 1 if $\sigma_x^{k+1} \neq \sigma_x^k$ or $\mathbf{x}^{(k+1)} \neq \mathbf{x}^{(k)}$.
-

5.2.2 Stability

The only measure of stability that is used for the original Harris detector is the value of the corner response function (Equation (5.2)). The higher this value the more the point is said to be ‘stable’ against variations in the image. For this reason a threshold is set on the corner response function to filter out unstable Harris points [40]. For the Harris-Laplace detector the value of the Laplacian-of-Gaussian (Equation (5.5)) is also taken as a measure of stability for the interest points in scale space. In this case a second threshold is used to filter out interest points with a low Laplacian-of-Gaussian response [98, 100].

5.2.3 Remarks

The Harris detector is still very popular in methods that require interest points (like matching, tracking, motion estimation, etc.), but do not require scale invariance. If scale invariance is required the Harris-Laplace approach might be considered. According to Mikolajczyk and Schmid, the Harris-Laplace approach provides a compact and representative set of interest points in scale space, which is competitive with state-of-the-art detectors [98, 100]. However the approach contains a great number of tunable parameters, thresholds and assumptions. Tunable parameters as γ of the corner response Equation (5.2) and s to relate σ_D to σ_I in Equation (5.4) have to be set experimentally. Thresholds to ignore ‘instable’ interest points, set for the corner response and the Laplacian-of-Gaussian, are chosen from experience. The scale step ξ for constructing the initial scale-space representation and t for the iterative approach, are also chosen from experience or duplicated from documented values from others. After setting these parameters to ‘appropriate’ values the algorithm appears to perform well. Defining these parameters however can be a tedious and exhaustive task which has to be done separately for each application that is going to use the Harris-Laplace detector. The type of images used, as well as the rendering details, can affect settings as well.



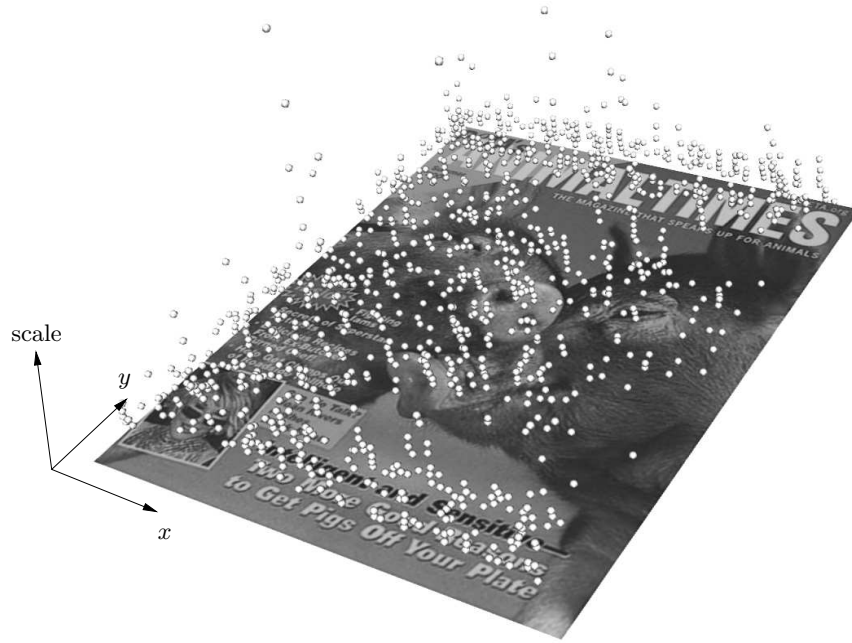


Figure 5.2: A set of Harris-Laplace points for a magazine cover image.

5.3 SIFT interest-point detector

The *Scale Invariant Feature Transform* (SIFT) proposed by Lowe [92, 94] is an algorithm to transform image data into scale-invariant coordinates relative to local features. The SIFT is nowadays perhaps the most frequently used image descriptor due to its stability and calculation speed. In this section the interest points used for the SIFT are discussed.

The location of scale-space extrema of the scale-normalized Laplacian-of-Gaussian (Equation (5.5)), as studied by Lindeberg [87] have shown to be useful interest points [95]. Lowe uses the *Difference-of-Gaussian* (DoG) function to approximate the scale-normalized Laplacian-of-Gaussian (Equation (5.5)). The DoG can be calculated as the difference of two nearby scales separated by a constant multiplicative factor k :

$$\text{DoG}(\mathbf{x}, \sigma) = L(\mathbf{x}, k\sigma) - L(\mathbf{x}, \sigma). \quad (5.6)$$

The approximation is used simply for its calculation speed. Apart from this

approximation a pyramid like approach (Section 2.3) is used to further speed up the interest point detection process. The interest points defined as local extrema of the scale-space representation of the DoG function are detected by simply comparing a pixel to its 26 neighbors in the representation (Figure 5.3). If all neighbors have a value lower or higher than the central pixel, this pixel is marked as an interest point.

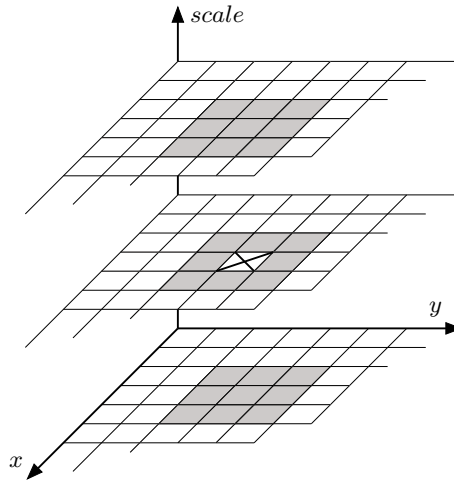


Figure 5.3: Extrema in the scale-space representation of the Difference-of-Gaussian images are detected by comparing a pixel (marked with \times) to its 26 neighbors in a 3×3 pixel region at the current and adjacent scales (marked with gray).

A typical set of SIFT interest points of an image is shown in Figure 5.4.

5.3.1 Detection versus Localization

The initial implementation of Lowe's approach [92] simply located the interest points at the location and scale of the detected extrema as previously explained. However in Brown and Lowe [10] a method for refining the discrete location of the interest points is proposed. Brown and Lowe use the same method as described in Section 2.4.4; they use a Taylor expansion (up to quadratic terms) of the Difference-of-Gaussian function $\text{DoG}(\mathbf{x}, \sigma)$, shifted so that the origin is at the sample point:

$$\text{DoG}(\mathbf{x} + \hat{\mathbf{x}}) = \text{DoG}(\mathbf{x}) + \nabla \text{DoG}(\mathbf{x}) \cdot \hat{\mathbf{x}} \quad (5.7)$$

Brown suggests that the derivatives of DoG are approximated by using differences of neighboring sample points.

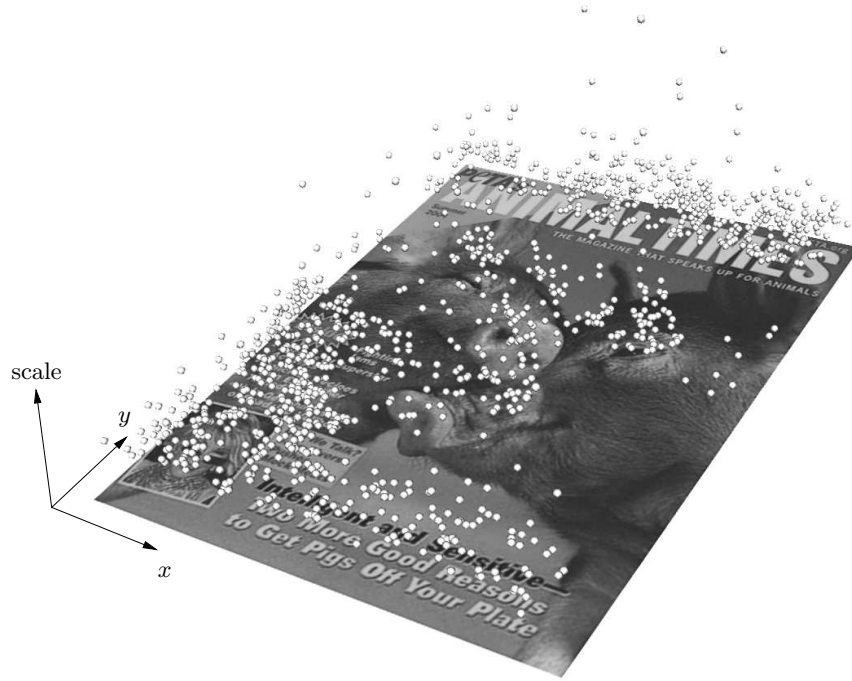
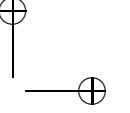
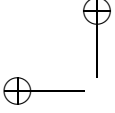


Figure 5.4: A set of SIFT interest points for a magazine cover image.

5.3.2 Stability

Without further information interest points for which $|\text{DoG}(\hat{\mathbf{x}})|$ is less than 0.03 are discarded in Lowe's algorithm [94]. The value of $|\text{DoG}(\hat{\mathbf{x}})|$ approximates the absolute value $|L_{xx} + L_{yy}|$ which in some sense describes the second order structure in a point. A threshold on this value will discard interest points in regions with low contrast.

Next to rejecting interest points with low contrast the SIFT algorithm also rejects interest points on edges. The DoG response along edges is strong, even if the location of an interest point along the edge is poorly determined and thus unstable to small perturbations. Such an interest point will have a large principle curvature across the edge, but a small one perpendicular to the edge. The principle curvatures of the DoG response can be computed from the Hessian matrix \mathbf{H} ,



computed at the scale of the interest point:

$$\mathbf{H} = \begin{bmatrix} \text{DoG}_{xx} & \text{DoG}_{xy} \\ \text{DoG}_{xy} & \text{DoG}_{yy} \end{bmatrix} \quad (5.8)$$

Where the derivatives in Lowe's algorithm are again estimated by taking the differences of neighboring sample points. The eigenvalues of \mathbf{H} are proportional to the principle curvatures of DoG. By using the rules of Equation (5.3) it can be avoided to calculate the eigenvalues directly. Let α be the eigenvalue with the largest magnitude and β be the smallest one. Let r be the ratio between the largest and the smallest eigenvalue $r = \alpha/\beta$. Then,

$$\frac{\text{trace}(\mathbf{H})}{\det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}, \quad (5.9)$$

which depends only on the ratio of the eigenvalues rather than their individual values. The quantity $(r + 1)^2/r$ is at a minimum when the two eigenvalues are equal and it increases with r . Therefore to check if the ratio of principle curvatures is below some threshold $t = \alpha_t/\beta_t$ it suffices to check

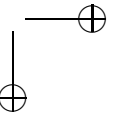
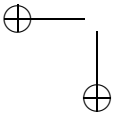
$$\frac{\text{trace}(\mathbf{H})}{\det(\mathbf{H})} < \frac{(t + 1)^2}{t}. \quad (5.10)$$

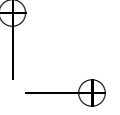
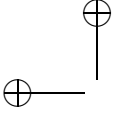
5.3.3 Remarks

The SIFT interest-point detector is currently the most commonly used scale-invariant interest-point detector. Not only due to its fast calculation speed and the availability of libraries and source code of the software, but also due to its excellent performance. In detailed experimental comparisons [95] it was found that the extrema of the scale-normalized LoG (Equation (5.5)) produce the most stable interest points compared to a range of other possible image functions, such as the gradient, Hessian and Harris-Laplace detector. The detector by Lowe, based on the DoG function, yields approximately the same results as the LoG detector.

It is surprising that simply taking the differences of neighboring sample points suffices for taking derivatives up to even third order. One would expect that this would lead to accuracy problems in the derivatives.

Due to its popularity and performance, the SIFT interest-point detector is the best algorithm to compare our own interest-point detector with.





5.4 Top-Points

Top-points are based on scale space and catastrophe theory (Section 2.4.1), and are invariant under gray value scaling and offset as well as scale-Euclidean transformations. The noise behavior of top-points can be described in closed-form, which enables the accurate prediction of the stability of the points [47, 6]. It is possible to retrieve the exact location of a top-point from any coarse estimation through a closed-form vector equation which only depends on local derivatives in the estimated point [33]. All these properties make top-points highly suitable as interest points for invariant matching schemes.

Top-points can for instance can be found from the original image, but any scalar differential entity, e.g. the Laplacian can be used as input for the top-point detection algorithm. The input for our algorithm will be referred to as $u(\mathbf{x})$.

A top-point is a critical point in the scale-space representation of the input $u(\mathbf{x})$, for which the determinant of the Hessian becomes zero. A top-point of a two-dimensional input function is thus defined as a point for which

$$\begin{cases} u_x &= 0, \\ u_y &= 0, \\ u_{xx}u_{yy} - u_{xy}^2 &= 0. \end{cases} \quad (5.11)$$

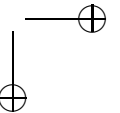
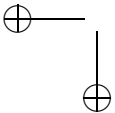
A typical set of top-points of an image is shown in Figure 5.5.

Note that the extrema of the normalized Laplacean scale space as introduced by Lindeberg [86], and used by Lowe [94] in his SIFT interest-point detector (Section 5.3), lie on the critical paths of the Laplacean image. Multiple of such extrema may exist on the extremum branch of a critical path, whereas there is only one top-point per annihilating extremum/saddle pair, Figure 5.6.

5.4.1 Detection versus Localization

Critical paths are detected by following critical points through scale. Top-points are found as points on the critical paths with horizontal tangents. A number of detection algorithms have been discussed in Section 2.4.3.

This method for finding critical points operates at pixel precision. For a critical point \mathbf{x}_c the detector obtains an approximate location \mathbf{x}_a . Recall from Section 2.4.4 that the sub-pixel location \mathbf{x}_c of a critical point can be found such that (5.11) holds to any desired precision. Florack and Kuiper [33] have shown that it



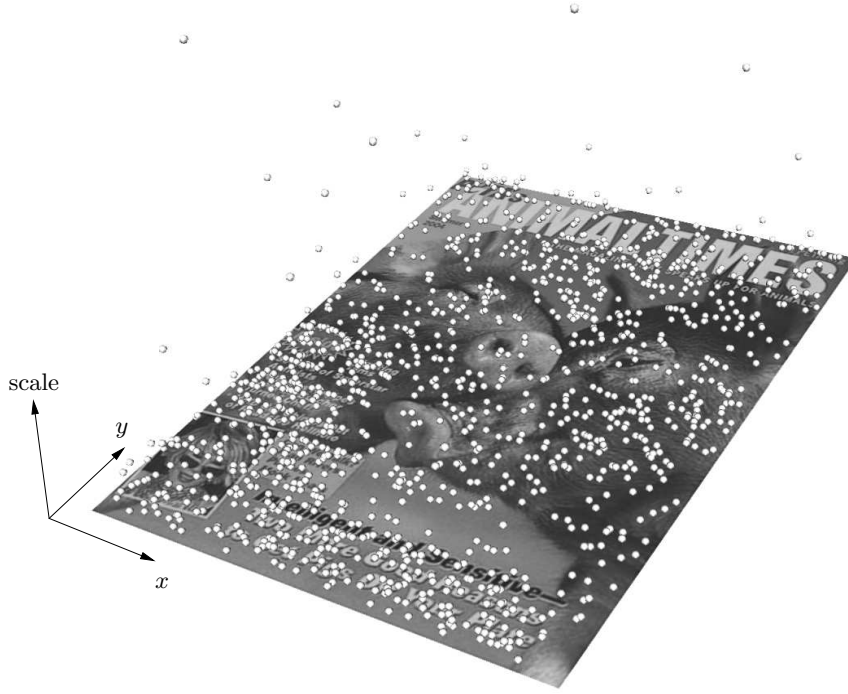


Figure 5.5: A set of top-points for a magazine cover image.

is possible to calculate a vector pointing to the location of a non-Morse critical point from a point sufficiently close by. If (x_a, y_a, t_a) denotes the approximate location of a top-point it is possible to find the true location of the top-point $\mathbf{x}_c = (x_a + \xi, y_a + \eta, t_a + \tau)$ by:

$$\begin{bmatrix} \xi \\ \eta \\ \tau \end{bmatrix} = -\mathbf{M}^{-1} \begin{bmatrix} \mathbf{g} \\ \det \mathbf{H} \end{bmatrix}, \quad (5.12)$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{H} & \mathbf{w} \\ \mathbf{z}^T & c \end{bmatrix}, \quad (5.13)$$

$$\mathbf{g} = \nabla u, \quad \mathbf{H} = \nabla \mathbf{g}, \quad \mathbf{w} = \partial_t \mathbf{g}, \quad \mathbf{z} = \nabla \det \mathbf{H}, \quad c = \partial_t \det \mathbf{H}, \quad (5.14)$$

in which all derivatives are taken in the point (x_a, y_a, t_a) .

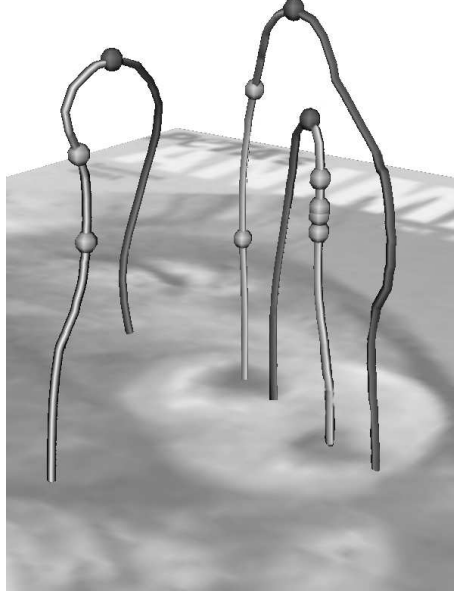


Figure 5.6: A set of critical paths with corresponding top-points (dark bullets), and extrema of the normalized Laplacian (light bullets).

For the two dimensional image case, the matrix \mathbf{M} can be constructed by using:

$$\mathbf{g} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \quad (5.15)$$

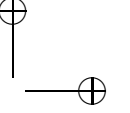
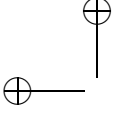
$$\mathbf{H} = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{xy} & u_{yy} \end{bmatrix}, \quad (5.16)$$

$$\mathbf{w} = \begin{bmatrix} u_{xxx} + u_{xyy} \\ u_{xxy} + u_{yyy} \end{bmatrix}, \quad (5.17)$$

$$\mathbf{z} = \begin{bmatrix} u_{xxx}u_{yy} + u_{xx}u_{xyy} - 2u_{xy}u_{xxy} \\ u_{xxy}u_{yy} + u_{xx}u_{yyy} - 2u_{xy}u_{xyy} \end{bmatrix}, \quad (5.18)$$

$$c = (u_{xxx} + u_{xxy})u_{yy} + (u_{yyy} + u_{xyy})u_{xx} - 2(u_{xxy} + u_{xyy})u_{xy}. \quad (5.19)$$

With this result the exact location of a top-point can be calculated from local differential information at an estimate location of a top-point. This will also prove useful for stability analysis in the next sections.



5.4.2 Propagation of Errors in Scale Space

Given a set of measurements in scale space $x \in \mathbb{R}^n$ it is possible to calculate the propagation of errors in a function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ if the measurements are perturbed with additive noise $N \in \mathbb{R}^n$.

In this section, the rules are discussed for the determination of the precision or reliability of a compound ‘measurement’ f in terms of the precision of each constituent x_i . This subject is known as the propagation of errors [109].

Suppose that derived property f is related to measured properties x_1, \dots, x_n by the functional relation

$$f = f(x_1, \dots, x_n) \quad (5.20)$$

The function is assumed to be sufficiently regular.

Suppose that all x_1, \dots, x_n are random and possibly correlated between each other. In this case the propagation of the variance of f can be approximated as

$$\langle (f(x_1, \dots, x_n) - f(\bar{x}_1, \dots, \bar{x}_n))^2 \rangle \approx \sum_{i=1}^n \sum_{j=1}^n \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \langle x_i - \bar{x}_i x_j - \bar{x}_j \rangle, \quad (5.21)$$

where all derivatives are calculated for the mean vector $(\bar{x}_1, \dots, \bar{x}_n)$.

5.4.3 Noise Propagation for Top-Point Displacement

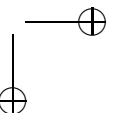
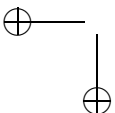
An analysis of how noise added to an image propagates to the location of top-points can be made from the previously discussed theory. For the top-point displacement given in Equation (5.12), the random variables (x_1, \dots, x_n) of Equation (5.21) are the derivatives of the noise term $(N_x, N_y, N_{xx}, \dots, N_{yyyy})$ and the computed ‘measurement’ f is a vector of displacements

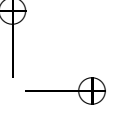
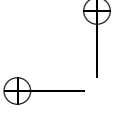
$$[\xi(N_x, \dots, N_{yyyy}), \eta(N_x, \dots, N_{yyyy}), \tau(N_x, \dots, N_{yyyy})]^T$$

in scale space. This is the vector from Section 5.4.1, pointing to the exact location of a top-point, that depends on derivatives up to forth order.

The mean vector $(\bar{N}_1, \dots, \bar{N}_n)$ is zero, therefore the mean displacement is zero as well

$$\begin{bmatrix} \bar{\xi} \\ \bar{\eta} \\ \bar{\tau} \end{bmatrix} = \begin{bmatrix} \xi(\bar{N}_1, \dots, \bar{N}_n) \\ \eta(\bar{N}_1, \dots, \bar{N}_n) \\ \tau(\bar{N}_1, \dots, \bar{N}_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (5.22)$$





Since the actual image v is obtained by adding noise N to the fiducial image u , i.e. $v = u + N$, for every i

$$\frac{\partial f_\alpha}{\partial N_i} = \frac{\partial f_\alpha}{\partial v_i}. \quad (5.23)$$

The following equation describes how the perturbation affects f , using Einstein summation convention for repeated indices:

$$f_\alpha(v) - f_\alpha(u) \approx \delta f_\alpha \equiv \left. \frac{\partial f_\alpha}{\partial v_\beta} \right|_{v=u} N_\beta \quad (5.24)$$

The covariance matrix of the displacement vector $f(N_x, \dots, N_{yyyy}) = [\xi, \eta, \tau]^T$, can be calculated by:

$$\langle \delta f_\alpha, \delta f_\beta \rangle = \frac{f_\alpha}{\partial v_\gamma} \frac{f_\beta}{\partial v_\zeta} \langle N_\gamma N_\zeta \rangle. \quad (5.25)$$

The momentum $M_{m_x, m_y, n_x, n_y}^2 = \langle N_{m_x, m_y} N_{n_x, n_y} \rangle$ of Gaussian derivatives of correlated noise, in case for which the spatial noise correlation distance τ is much smaller than scale t , is given by [47]

$$M_{m_x, m_y, n_x, n_y}^2 \simeq \langle N^2 \rangle \left(\frac{\tau}{2t} \right) \left(\frac{-1}{4t} \right)^{\frac{1}{2}(m_x + m_y + n_x + n_y)} Q_{m_x + n_x} Q_{m_y + n_y}, \quad (5.26)$$

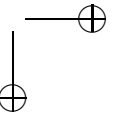
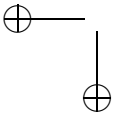
where

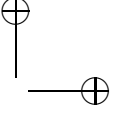
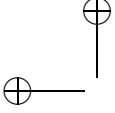
$$Q_n = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \text{ odd} \\ \prod_{i=1}^{n/2} (2i-1) & \text{if } n \text{ is even} \end{cases} \quad \text{e.g.} \quad \begin{array}{|c|c|c|c|c|} \hline n & 0 & 2 & 4 & 6 \\ \hline Q_n & 1 & 1 & 3 & 15 \\ \hline \end{array} \quad (5.27)$$

For simplicity $\tau = 1/2$ is taken, note that this is just a factor in the momentum. In this case Gaussian derivatives of the first and the second order have the following correlation matrix:

$$C = (\langle N_i N_j \rangle)_{ij} = \begin{pmatrix} 4t_0 & 0 & 0 & 0 & 0 \\ 0 & 4t_0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 3 \end{pmatrix} \frac{\langle N^2 \rangle}{(4t_0)^3}, \quad (5.28)$$

where $(N_1, \dots, N_5) = (N_x, N_y, N_{xx}, N_{xy}, N_{yy})$. A similar matrix can be derived up to any order by using Equation (5.26), but is not written here as it would be too big.





5.4.4 Stability

The stability of a top-point can now be expressed by a covariance matrix of spatial and scale displacements induced by additive noise. Since top-points are generic entities in scale space, they cannot vanish or appear when the image is only slightly perturbed. It is assumed that the noise variance is ‘sufficiently small’ in the sense that the induced dislocation of a top-point can be investigated by means of a perturbative approach. It can be shown that the displacement depends on derivatives up to fourth order evaluated at the top-point, and on the noise variance. For more detailed formulas (and experimental verifications) the reader is referred to [6].

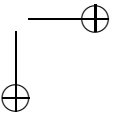
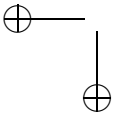
The advantage of this approach is that variances of scale-space displacements can be predicted theoretically and in analytically closed-form on the basis of the local differential structure at a given top-point, cf. Figure 5.7 for a schematic illustration. The ability to predict the motion of top-points under noise is valuable when matching noisy data (e.g. one may want to disregard highly instable top-points altogether).

A threshold on the stability could be set for example on the determinant of the covariance matrix of the displacements or each eigenvalue of the covariance matrix could be analyzed separately.

5.5 Experiments

5.5.1 Database

For the experiments a data set containing transformed versions of 12 different magazine covers (Figure 5.8) is used. The covers contain a variety of objects and text. The data set contains rotated, zoomed and noisy versions of these magazine covers as well as images with perspective transformations. For all transformations the ground truth is known, which enables us to verify the performance of different algorithms on the database. Mikolajczyk’s data set used in [98, 99] is not suitable for our purposes, as we require ground truth for genuine group transformations not confounded with other sources of image changes, such as changes in field of view. To our knowledge Mikolajczyk’s data set does not provide this.



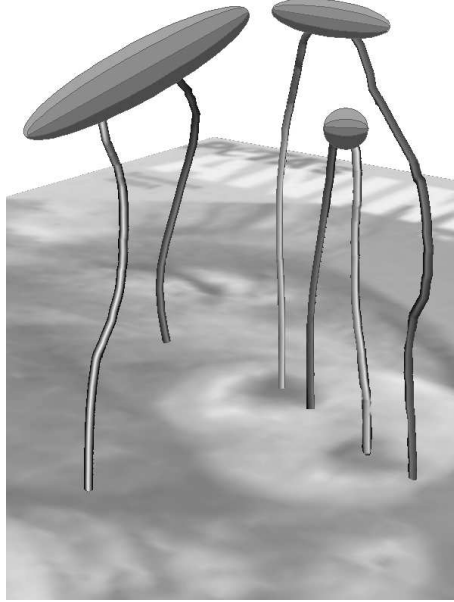


Figure 5.7: The ellipses at the top-points schematically represent the variances of the scale-space displacement of each top-point under additive noise of known variance.

5.5.2 Repeatability

Schmid et al. [119] introduced the so-called repeatability criterion to evaluate the stability and accuracy of interest points and interest-point detectors. The repeatability rate for an interest-point detector on a given pair of images is computed as the ratio between the number of correct point-to-point correspondences and the minimum number of interest points detected in the images ($\times 100\%$).

If the interest point in the perturbed image has moved less than a distance of ϵ pixels away from the position where it would be expected when following the transformation, the point is marked as a repeatable point (typically $\epsilon \approx 2$ pixels).

Experiments show the repeatability of top-points under image rotation (Figure 5.9) and additive Gaussian noise (Figure 5.10). Image rotation causes some top-points to be lost or created due to the resampling of the image. In the Gaussian noise experiment we demonstrate that by using the stability variances described in Section 5.4.4 the repeatability of the top-points can be increased. The top-points are ordered on their stability variances. From this list 100%, 50% and 30% of the most stable top-points are selected for the repeatability experiment re-

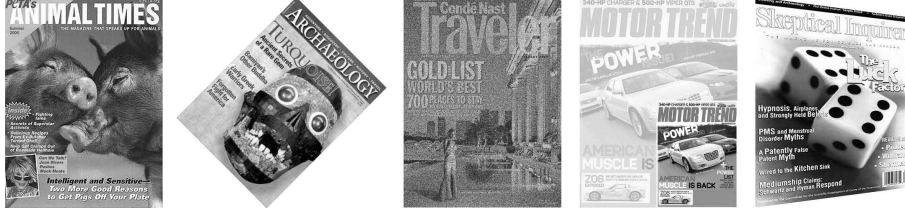


Figure 5.8: A selection of data set images. From left to right: unchanged, rotated, added noise, scaled, changed perspective.

spectively. The stability is calculated as the determinant of the covariance matrix for top-point stability (Equation 5.25). From Figure 5.10 it is apparent that discarding instable points increases the repeatability significantly. The repeatability of the top-point detector is compared to that of the SIFT interest-point detector. In Figure 5.10 can be seen that when we apply a threshold on our stability measure (the SIFT keypoints have already been thresholded on stability as explained in Section 5.3.2) we slightly outperform the SIFT interest-point detector for the noise case. Both algorithms perform worst for a rotation of 45 degrees. On the average taken over the entire database of 45 degree rotated images the repeatability of the SIFT interest points is 78%. Our top-point interest-point detector showed a repeatability rate of 85% when thresholded on stability. The high repeatability rate of the top-points enables us to match images under any angle of rotation and under high levels of noise.

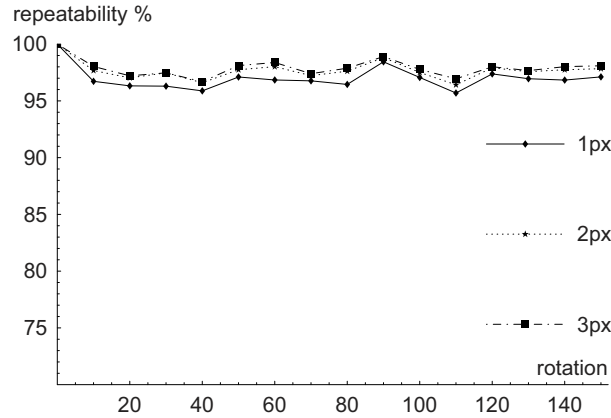


Figure 5.9: The repeatability rate of the top-points for different angles of rotation for different ϵ .

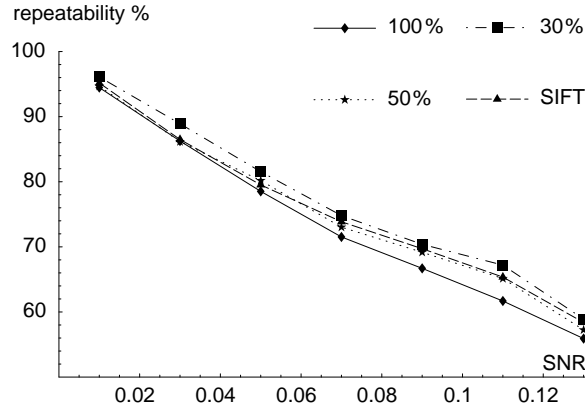


Figure 5.10: The repeatability rate of the top-points and SIFT interest points for additive Gaussian noise expressed in signal to noise ratio. By selecting 100%, 50% or 30% of the most stable points the repeatability is increased.

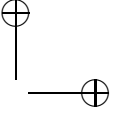
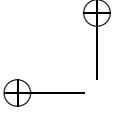
5.6 Summary and Conclusions

The most popular interest point detectors such as the Harris detector and the Scale Invariant Feature Transform (SIFT) are heavily based on tunable parameters and thresholds. These parameters vary for different images. It is unclear how most of these parameters are affected by the image properties, the other parameters and the algorithm. These parameters are in practice set by choosing the best parameters for a training set of experiments. This means that these parameters are valid only for images like the ones in the training set. For different types of images the parameters have to be trained on new training sets.

To overcome the cumbersome task of tuning the parameters for the interest point detectors, a new type of interest point, the top-point, has been introduced. These top-points are highly invariant interest points, suitable for image matching. Top-points are versatile as they can be calculated for every generic function of the image.

From their definition, it is apparent that top-points are invariant under gray value scaling and offset. Next to this the top-points are also invariant to scale-Euclidean transformations (rotation, scaling, translation).

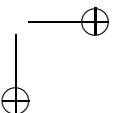
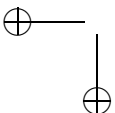
The top-points however are in theory not invariant to affine or projective transformations just like the interest-point detectors mentioned earlier, but in practice they show to be fairly robust to small affine or projective transformations. The

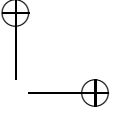
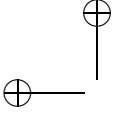


behavior of all before mentioned interest points under large affine or projective transformations however is unpredictable.

The sensitivity of top-points to additive noise can be predicted analytically, which is useful when matching noisy images. Top-point localization does not have to be very accurate, since it is possible to refine its position using local differential image structure. This enables fast detection, without losing the exact location of the top-point.

The repeatability of the top-points has proven to be better than the widely used SIFT interest points in a set of experiments. In the next chapter more comparisons will be made with regard to the SIFT performance.





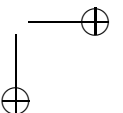
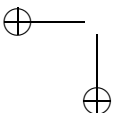
Things should be as simple as possible, but not simpler.

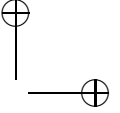
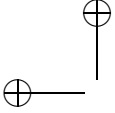
Albert Einstein (1879–1955)

Neighborhood Descriptors for Image Matching

This chapter is partly based on:

B. Platel, E.G. Balmachnova, L.M.J. Florack, and B.M. ter Haar Romeny, Top-Points as Interest Points for Image Matching. In *European Conference on Computer Vision (ECCV)*, pages 418–429, 2006





6.1 Introduction

For image matching purposes a set of points of interest, as discussed in the previous chapter, is insufficient by itself. Distinct information has to be added to these points. This extra information assigned to each interest point is referred to as a *descriptor*.

Local photometric descriptors computed for interest regions have proven to be particularly useful in applications as wide baseline matching [118, 138, 131], object recognition [7, 92, 115, 17], texture recognition [79, 80], image retrieval [58, 111], robot localization [120], video data mining [125], building panorama's [11] and object class recognition [21, 96].

In this chapter the *Scale Invariant Feature Transform* (SIFT) descriptor will be discussed and compared to the more mathematically founded *differential invariant* descriptor. Many more local descriptors exist, for an extensive overview and evaluation of local descriptor, the interested reader is referred to Mikolajczyk and Schmid [99].

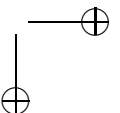
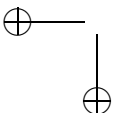
In a set of experiments the performance of top-points together with the differential invariant descriptors are demonstrated and compared to the performance of the SIFT algorithm.

6.2 SIFT Descriptor

The most popular and widely used algorithm for the extraction of interest points and local descriptors is the *Scale Invariant Feature Transform* (SIFT) by Lowe [92, 94]. Lowe's algorithm not only finds the interest points discussed in Section 5.3, but also computes a local descriptor for each interest point. This descriptor will be discussed here in detail. In short, Lowe's algorithm measures local gradients in a region around each interest point. These gradients are transformed into a representation that is robust to significant levels of local shape distortion and change in illumination.

6.2.1 Descriptor Orientation

To make a descriptor rotation invariant it has to be aligned with a certain orientation obtained in the interest point. At the location of an interest point a



distinctive orientation can be calculated, e.g. by calculating the gradient angle

$$\theta(x, y) = \tan^{-1}(L_y/L_x). \quad (6.1)$$

For which the derivatives L_x and L_y are calculated at the scale and location of the interest point. Lowe proposes ‘a more stable method’ for defining one or multiple orientations to an interest point. He forms an orientation histogram from the gradient orientations of sample points in a square region around the interest point (as illustrated in Figure 6.1). The size of the square region in which the gradient orientations are calculated is dependent on the scale of the interest point. The orientation histogram has 36 bins covering the 360 degree range of orientations. Every sample that is added to the histogram is weighted by its gradient magnitude

$$m(x, y) = \sqrt{L_x^2 + L_y^2} \quad (6.2)$$

and by a Gaussian-weighted circular window with a standard deviation that is 1.5 times the scale of the interest point (Figure 6.1). Peaks in this orientation histogram correspond to dominant directions of local gradients (Figure 6.2). The highest peak in the histogram is detected and for any other peak that is within 80% of the highest peak, a new interest point with that orientation is created. Therefore, for locations with multiple peaks of similar magnitude, multiple interest points will be created at the same location and scale, but with different orientations. According to Lowe about 15% of the interest points are assigned multiple orientations and these extra orientations are said to contribute significantly to the stability of the matching algorithm. The location of the histogram peaks are found with more precision by interpolation of the histogram.

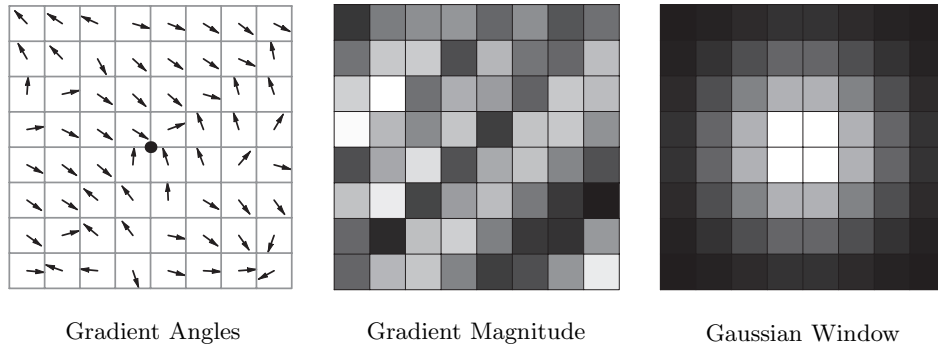


Figure 6.1: In a region around the interest point (center dot) gradient orientations are computed and weighted with their gradient magnitude and a Gaussian window.

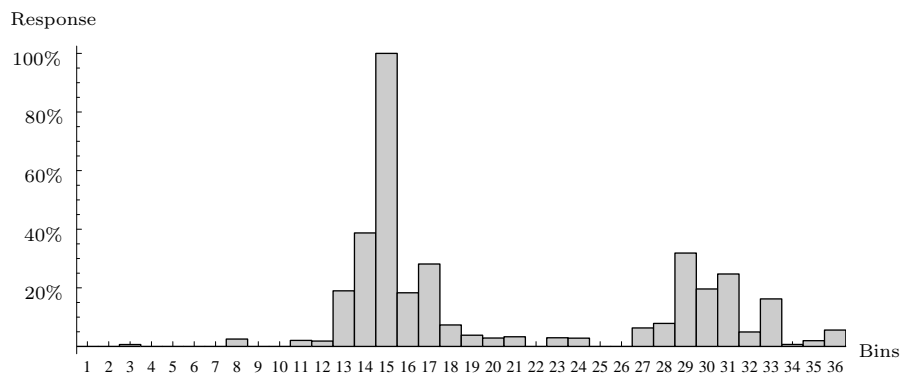


Figure 6.2: The gradient orientation histogram for the interest point of Figure 6.1.

6.2.2 Descriptor Representation

Figure 6.3 illustrates the computation of the interest point descriptor. First the gradient orientations and magnitudes are sampled in a region around the interest point (Figure 6.3a). This region is oriented in the direction of the angle calculated in the previous section. The gradient angles in the region are made rotation invariant by subtracting the interest point orientation (Figure 6.3b).

The gradient magnitudes in the sample points are used as a weighting function of the gradient angles in the region around the interest point. Then a Gaussian weighting function with a standard deviation of 1.5 times the scale of the interest point is applied in the same manner as in Figure 6.1. The purpose of this Gaussian window according to Lowe is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far away from the center of the descriptor (Figure 6.3c).

The weighted gradient angles are accumulated into orientation histograms, summarizing the contents over 4×4 subregions. The orientation histograms have eight orientation bins (Figure 6.3d). To avoid boundary affects in which the descriptor abruptly changes as a sample shifts smoothly from being within one histogram to another, Lowe uses interpolation over the samples.

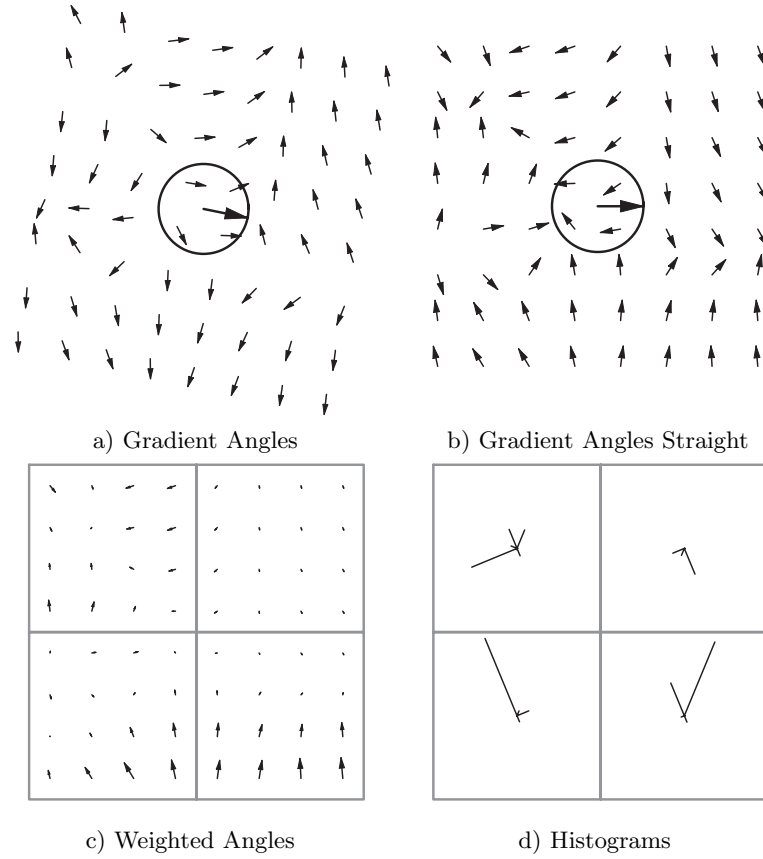
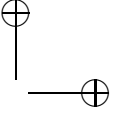
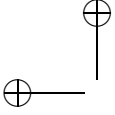


Figure 6.3: a) In an oriented region around the interest point, gradient orientations are computed. b) The gradient angles in the region are made rotation invariant by subtracting the interest point orientation. c) The gradient magnitudes and a Gaussian window are used as weights for the gradient angles. d) The weighted gradient angles are accumulated into orientation histograms, summarizing the contents over 4×4 sub-regions. The figure shows eight directions for each histogram with the length of a line corresponding to the weight of each histogram entry. Note that this figure shows a 2×2 descriptor array computed for an 8×8 set of samples. In Lowe's article 4×4 descriptors are calculated from a 16×16 sample array.

The interest point descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the lines in the histograms of Figure 6.3d. The figure shows a 2×2 array of orientation histograms, whereas in Lowe's article [92] he shows that a 4×4 array of histograms with 8 orientation bins each, achieves the best results for matching purposes. Therefore



the SIFT interest point descriptor is a $4 \times 4 \times 8 = 128$ element feature vector.

This 128 element feature vector is modified to reduce the effects of illumination change. First the vector is normalized to unit length. A change in image contrast in which each pixel value is multiplied by a constant will multiply gradients by the same constant, so this contrast change will be cancelled by the vector normalization. A brightness change in which a constant is added to each image pixel will not affect the gradient values as they are computed as derivatives (or pixel differences in Lowe's case).

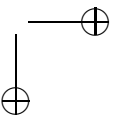
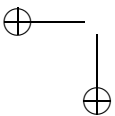
To avoid problems with camera saturation or other large changes in relative magnitudes for some gradients, the influence of large gradient magnitudes are reduced by thresholding the values in the unit feature vector. Values larger than 0.2 are chopped to 0.2 after which the vector is renormalized. The value of 0.2 was determined experimentally [94].

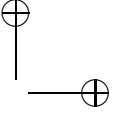
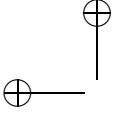
6.2.3 Adaptations of the SIFT Algorithm

Ke and Sukthankar [59] 'improved' the SIFT descriptor by applying *Principle Component Analysis* (PCA) to the normalized gradient patch. They named their descriptor *PCA-SIFT* and demonstrate in a small set of experiments that their descriptor is more distinctive, more compact (36 features instead of 128) and more robust to image deformations than the standard SIFT descriptor. Their method has to be trained however for a specific type of images. Mikolajczyk and Schmid [99] introduced the *Gradient Location-Orientation Histogram* (GLOH), another adaptation of the SIFT descriptor. They compute the SIFT descriptor on a log-polar grid with three bins in radial direction (the radius set to 6, 11, 15) and 8 in angular direction, resulting in 17 bins. The center bin is not divided in angular regions. The gradient orientations are quantized in 16 bins, yielding a 272 bin histogram. The size of this descriptor is reduced with PCA. For their method, a training set is used to estimate the covariance matrix for the PCA. In their article they show superior performance of their GLOH descriptor over the SIFT and PCA-SIFT descriptor.

6.2.4 Distance Measure

The dissimilarity between the SIFT, PCA-SIFT and GLOH descriptors is calculated by using the Euclidean distance on the feature vectors, as suggested by the authors [94, 59, 99].





6.2.5 Remarks

In 2003 the SIFT descriptor was found to be the best performing descriptor compared to steerable filters, differential invariants, moment invariants and cross-correlation for different types of interest points, in an article by Mikolajczyk and Schmid [97]. Lowe has a US patent on his method at the University of British Columbia [93]. The popularity of the SIFT algorithm is not only due to its performance and speed, but also the availability of implementations in many programming languages has led to its wide spread acceptance. The newer descriptors like PCA-SIFT and GLOH seem to be able to improve the performance of the original descriptor, but still use the same machinery.

Even though the SIFT descriptor has been found to be robust and very distinctive, it contains many assumptions and experimentally set thresholds and values. The amount of samples, histograms and histogram bins are experimentally defined. The Gaussian weighting function is arbitrarily set to 1.5 times the scale of the interest point. Thresholds and cut off values for the histograms are experimentally set and all the derivatives are found by simply taking pixel differences of sample values in Lowe's pyramid-like approximation of the scale-space representation of the image (Section 2.3).

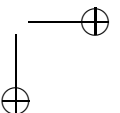
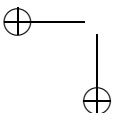
6.3 Differential Invariant Descriptors

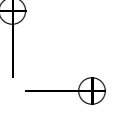
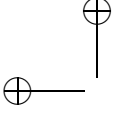
To avoid the large number of 'ad hoc' parameters set in popular descriptors, a more fundamental approach is taken in this section. By using well defined image derivatives at the location of an interest point an approximation is made of the point's neighborhood.

The properties of local derivatives (*local jet*) were investigated by Koenderink and van Doorn [68]. Florack et al. [27, 28, 29, 30] derived differential invariants, which combine components of the local jet to obtain rotation invariance.

The simple and mathematically nice nature makes the *differential invariants* the preferred choice of interest point descriptors. Since the mathematics behind them is well known it will be possible to calculate a sensible distance between the descriptors.

The performance of the differential invariants was the worst among many other local descriptors in Mikolajczyk and Schmid's performance evaluation paper [99]. We conjecture that Mikolajczyk and Schmid's implementation and definition of the differential invariants, together with an experimentally defined distance mea-





sure might have lead to their bad evaluation results. In Section 6.4.3 it is shown that for similar experiments as the ones Mikolajczyk and Schmid used, but incorporating a new dissimilarity measure, the differential invariants out perform the SIFT descriptors.

6.3.1 Descriptor Representation

As interest point descriptor, irreducible differential invariants up to third order will be used, as thoroughly described by Florack [28, 29]. The differential invariants are invariant to rigid transformations. By suitable scaling and normalization, invariance to spatial zooming and intensity scaling is obtained as well, but the resulting system has the property that most low order invariants vanish identically at the top-points of the original (zeroth order) image, and thus do not qualify as distinctive features. Thus when considering top-points of the original image other distinctive features will have to be used. In Chapter 4 the embedding of a graph connecting top-points is used as a descriptor. This proved to be a suitable way of describing the global relationship between top-points of the original image. In this chapter the Laplacian of the input function is used as input for our top-point detector. This results in a detector that is focussed on blob-like structures, like the Laplacian-of-Gaussian interest point detectors used in the SIFT descriptor. For top-points of the Laplacean image the following set of equations holds:

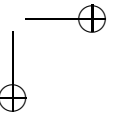
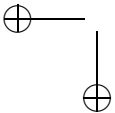
$$\begin{cases} \partial_x \Delta u &= u_{xxx} + u_{xyy} &= 0, \\ \partial_y \Delta u &= u_{xxy} + u_{yyy} &= 0, \\ \det \mathbf{H}(\Delta u) &= (u_{xxx} + u_{xyy})(u_{xxy} + u_{yyy}) - (u_{xxy} + u_{xyy})^2 &= 0. \end{cases} \quad (6.3)$$

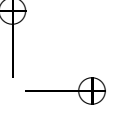
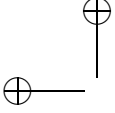
Where $u(\mathbf{x}, 0)$ represents the original image. For this case the non-trivial, scaled and normalized differential invariants up to third order are collected into the column vector given by (6.4), again using summation convention:

$$\begin{pmatrix} \sigma \sqrt{u_i u_i} / u \\ \sigma u_{ii} / \sqrt{u_j u_j} \\ \sigma^2 u_{ij} u_{ij} / u_k u_k \\ \sigma u_i u_{ij} u_j / (u_k u_k)^{3/2} \\ \sigma^2 u_{ijk} u_i u_j u_k / (u_l u_l)^2 \\ \sigma^2 \varepsilon_{ij} u_{jkl} u_i u_k u_l / (u_m u_m)^2 \end{pmatrix}. \quad (6.4)$$

Where ε is the *Levi-Civita* or *permutation tensor* defined as

$$\varepsilon^{i_1 i_2 \dots i_N} = \begin{cases} 0, & \text{if any two labels are the same} \\ 1, & \text{if } i_1, i_2, \dots, i_N \text{ is an even permutation of } 1, 2, 3 \\ -1, & \text{if } i_1, i_2, \dots, i_N \text{ is an odd permutation of } 1, 2, 3. \end{cases} \quad (6.5)$$





Note that the derivatives are extracted from the original, zeroth order image, but evaluated at the location of the top-points of the image Laplacian. This is, in particular, why the gradient magnitude in the denominator poses no difficulties, as it is generically nonzero at a top-point of the image Laplacian.

The resulting differential feature vector guarantees manifest invariance under the scale-Euclidean spatial transformation group, and under linear gray value rescalings.

The normalization proposed in (6.4) may cause instabilities if close to a ‘non-generic’ point where $\nabla u \approx 0$ (in addition to Equation (6.3)). This however will most likely occur in ‘unstable’ top-points, since the lack of differential structure indicates a flat area around the interest point. This implies that these top-points anyway do not contribute to the matching, as they will be discarded based on their lack of stability.

6.3.2 Dissimilarity Measure in the Descriptor Space

To compare features of different interest points, a distance or dissimilarity measure is needed. The most often used measures in literature are the Euclidean and Mahalanobis distance. If \mathbf{x}_0 and \mathbf{x} are two points from the same distribution which has covariance matrix Σ , then the Mahalanobis distance is given by

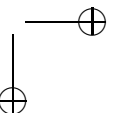
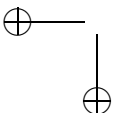
$$d(\mathbf{x}_0, \mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{x}_0)^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}_0)} \quad (6.6)$$

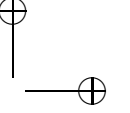
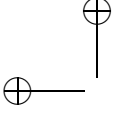
and is equal to the Euclidean distance if the covariance matrix Σ is the identity matrix. The advantage of the Mahalanobis distance is that it can be used to measure distances in non-Euclidean spaces. The drawbacks however are that the covariance matrix has to be derived by using a large training set of images, and that the covariance matrix is the same for every measurement.

By using the perturbative approach from Section 5.4.2 and using the set of differential invariants, a covariance matrix can be calculated for each interest point descriptor separately as follows. Given a set of measurements in scale space $u \in \mathbb{R}^n$ the propagation of errors in a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ can be calculated, if the measurements are perturbed with noise N , $v = u + N \in \mathbb{R}^n$. The following equation describes how the perturbation affects f , using Einstein summation convention for repeated indices:

$$f_\alpha(v) - f_\alpha(u) \approx \delta f_\alpha \equiv \left. \frac{\partial f_\alpha}{\partial v_\beta} \right|_{v=u} N_\beta \quad (6.7)$$

The covariance matrix of f can be expressed as:





$$\langle \delta f_\alpha \delta f_\beta \rangle = \frac{\partial f_\alpha}{\partial v_\gamma} \frac{\partial f_\beta}{\partial v_\zeta} \langle N_\gamma N_\zeta \rangle \quad (6.8)$$

Where the invariants are used for functions f_α and the set up to third order derivatives as v_β .

The noise matrix $\langle N_\gamma N_\zeta \rangle$ is given in Section 5.4.3 Equation (5.26).

This makes it possible to use the Mahalanobis distance (6.6) to calculate the dissimilarity between two feature vectors using the covariance matrix $\Sigma_{\mathbf{k}_o}$ derived specifically for the descriptor \mathbf{k}_o , where $d(\mathbf{k}_o, \mathbf{k})$ close to zero means very similar, and $d(\mathbf{k}_o, \mathbf{k}) \gg 0$ very dissimilar. Note that this makes the dissimilarity measure asymmetric: $d(\mathbf{k}_o, \mathbf{k}) \neq d(\mathbf{k}, \mathbf{k}_o)$. This however does not pose problems since we are only matching unidirectionally, viz. object to scene.

6.3.3 Remarks

The differential invariants up to third order give a six valued descriptor vector. Compared to the 128 valued descriptor of the SIFT algorithm this is a very small amount. The algorithm uses Gaussian derivatives, no window functions and the dissimilarity measure is derived from the behavior of the differential invariants under noise.

6.4 Experiments

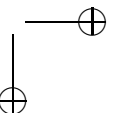
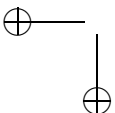
6.4.1 Receiver Operator Characteristics

For the performance evaluation of the dissimilarity measure a similar criterion as the one used in [99] is used. This criterion is based on *Receiver Operating Characteristics* (ROC) of *True Positive Rate* (TPR) versus *False Positive Rate* (FPR). Two points are said to be similar if the distance between their feature vectors is below a threshold t . The value of t is varied to obtain the ROC curves.

Given two images representing the same object the TPR is the number of correctly matched points with respect to the number of possible matches.

$$\text{TPR} = \frac{\# \text{correct matches}}{\# \text{possible matches}} \quad (6.9)$$

A possible match is a match for which the dissimilarity or distance between the descriptors is below a threshold t . The condition for calling a match correct is



the same as in Section 5.5.2; if the interest point in the perturbed image has moved less than a distance of ϵ pixels away from the position where it would be expected when following the transformation, the point is marked as a repeatable point (typically $\epsilon \approx 2$ pixels).

The FPR as defined in [99] is calculated as:

$$\text{FPR} = \frac{\# \text{incorrect matches}}{(\# \text{object points})(\# \text{scene points})} \quad (6.10)$$

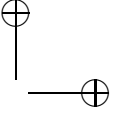
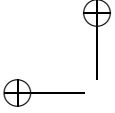
where the object is the original image and the scene a transformed version of the original image (e.g. Figure 6.4).



Figure 6.4: Examples of the transformations used in the experiments, from left to right: the original image, 45 degree rotated, added Gaussian noise, perspective transformed

6.4.2 Performance of the Dissimilarity Measure

To evaluate the performance of the dissimilarity measure defined in Section 6.3.2, ROC curves have been generated for a set of experiments. For comparison ROC curves for the Mahalanobis and Euclidean distance measures are plotted. The covariance matrix for the Mahalanobis distance was obtained by training on the data set itself, so that it may be regarded ‘optimal’ in the sense that no better results are likely to be obtained in practice when using a different training set. In Figure 6.5 the mean ROC curves for three experiments are shown (the ROC curves are averaged over the 12 magazine cover experiments). In experiment a. the images in the database are matched to a 50% scaled down version of the same images. In experiment b. the images in the database are matched to noisy versions of the same images. In experiment c. the images in the database are matched to the 45 degree rotated versions of the same images (e.g. Figure 6.4). In all the experiments it is obvious that the new dissimilarity measure greatly improves the performance of the matching algorithm.



6.4.3 Performance of the Descriptors

To evaluate the performance of the differential invariant descriptors defined in Section 6.3 ROC curves have been calculated for a similar set of experiments. For comparison ROC curves of the SIFT algorithm are plotted, these are generated by using the authors own publicly available program. The SIFT descriptor consist of a 128 feature long vector containing information about the gradient angles in the neighborhood of the interest points. The experiments in Figure 6.6 show superior performance of the differential invariant descriptor over the SIFT descriptor. The difference becomes even more evident if only stable top-points are used as interest points. Note that the used SIFT algorithm automatically filters unstable interest points as described in Section 5.3.2.

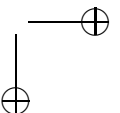
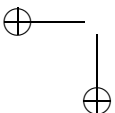
In a different set of experiments the performance of both algorithms has been tested under perspective change. For small perspective changes the differential invariants in the top-points perform slightly better than the SIFT algorithm. However this performance decreases for larger perspective changes. The SIFT detector outperforms the differential invariants in this case. This is probably due to the higher order information used in the differential invariants which is more affected by perspective or affine changes than the first order information used in the SIFT descriptor.

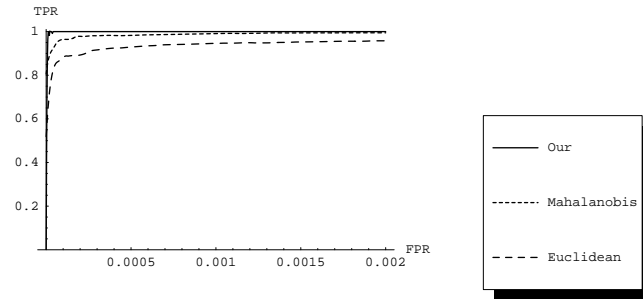
6.5 Summary and Conclusions

As descriptors for our interest points we use a feature vector consisting of only six normalized and scaled differential invariants.

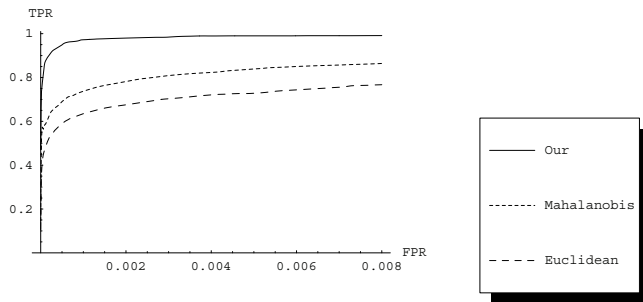
A dissimilarity measure was derived based on the noise behavior of the differential invariant features. This measure significantly increases performance over the popular Mahalanobis and Euclidean distance measures.

For scale-Euclidean transformations as well as additive Gaussian noise our algorithm (6 features in vector) has proven to outperform the SIFT (128 features in vector) approach. However for large perspective changes the SIFT algorithm performs better probably due to the lower order derivatives used for the feature vector.

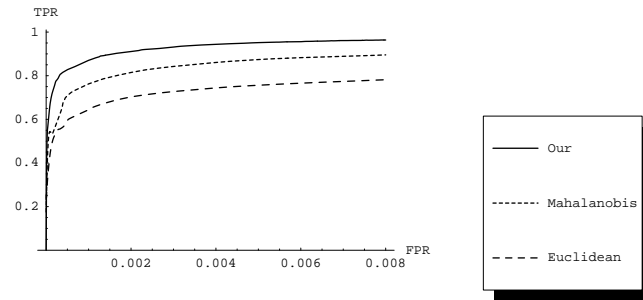




(a)



(b)



(c)

Figure 6.5: **a.** mean ROC curve for 50% scaling. **b.** mean ROC curve for 5% additive Gaussian noise. **c.** mean ROC curve for 45 degree rotation.

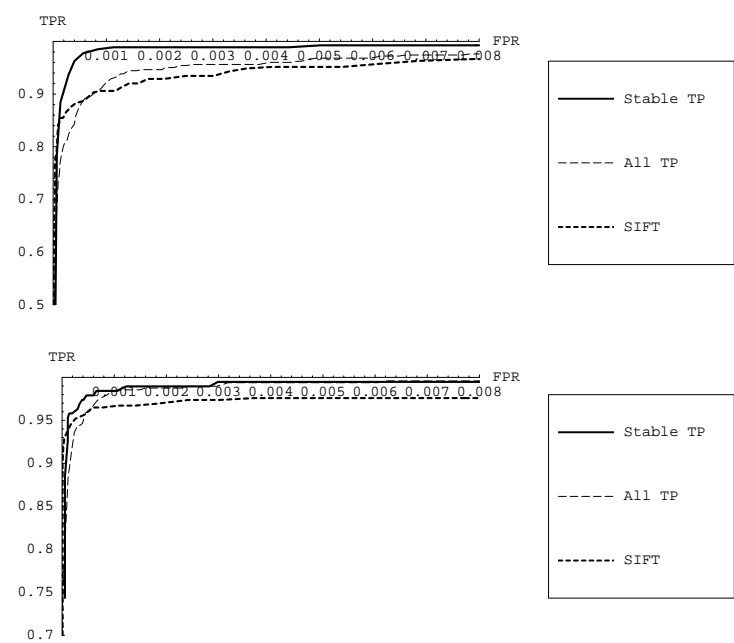


Figure 6.6: a. ROC curve for 45 degree rotation. b. ROC curve for 5% additive Gaussian noise.

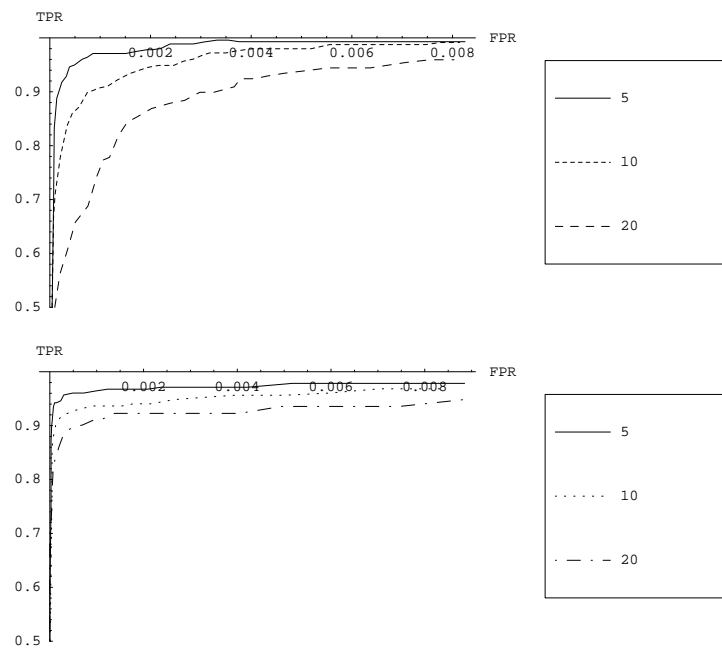
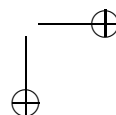
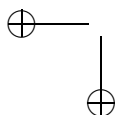
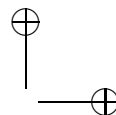
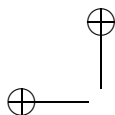
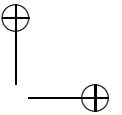
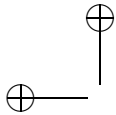


Figure 6.7: ROC curves perspective change for 5, 10 and 20 degrees for: **a.** Our interest points and differential invariants **b.** SIFT interest points and features.



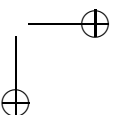
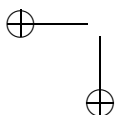


Science is facts; just as houses are made of stones, so is science made of facts; but a pile of stones is not a house and a collection of facts is not necessarily science.

Henri Poincaré (1854–1912)

7

Object Location and Pose Retrieval



7.1 Introduction

In this chapter the previously discussed interest points (Chapter 5) and detectors (Chapter 6) are used for the task of ‘object location and pose retrieval’. The task is to obtain the location and pose of an object contained in a scene image (Figure 7.1). This is done by matching the interest points and descriptors of an image of the query object to the interest points and descriptors of an image of a scene.

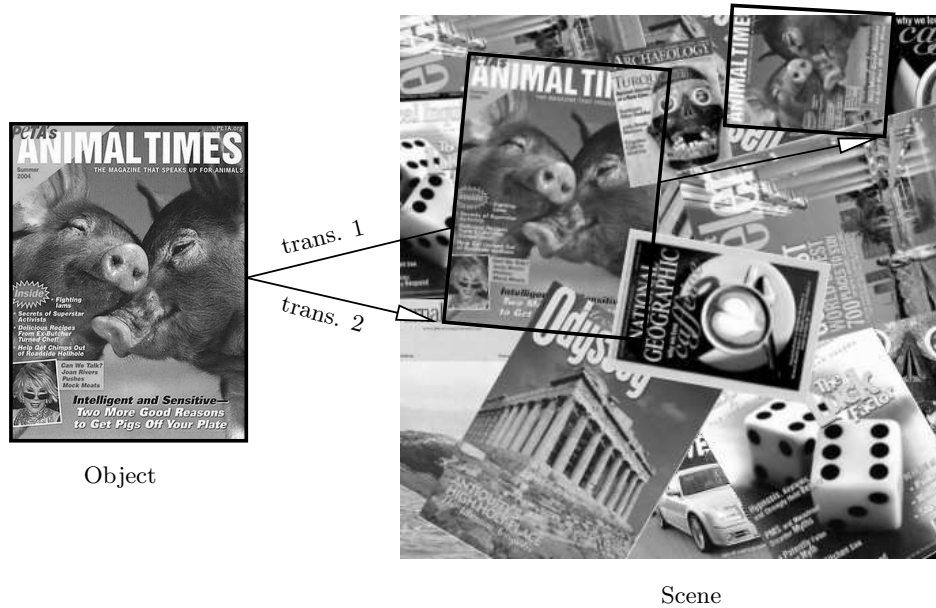


Figure 7.1: The transformation the object in the scene has undergone with respect to the query object is denoted as ‘trans.’. The task of an ‘object location and pose retrieval’ algorithm is to find this transformation.

The complete algorithm is summarized a flowchart (Figure 7.2) and it will be explained step by step in this chapter.

7.2 Matching Features

For both the query object image and the scene image the interest points together with their descriptors are calculated. In each *feature*, the location of the interest point (x, y, σ) , the gradient angle at the location of the interest

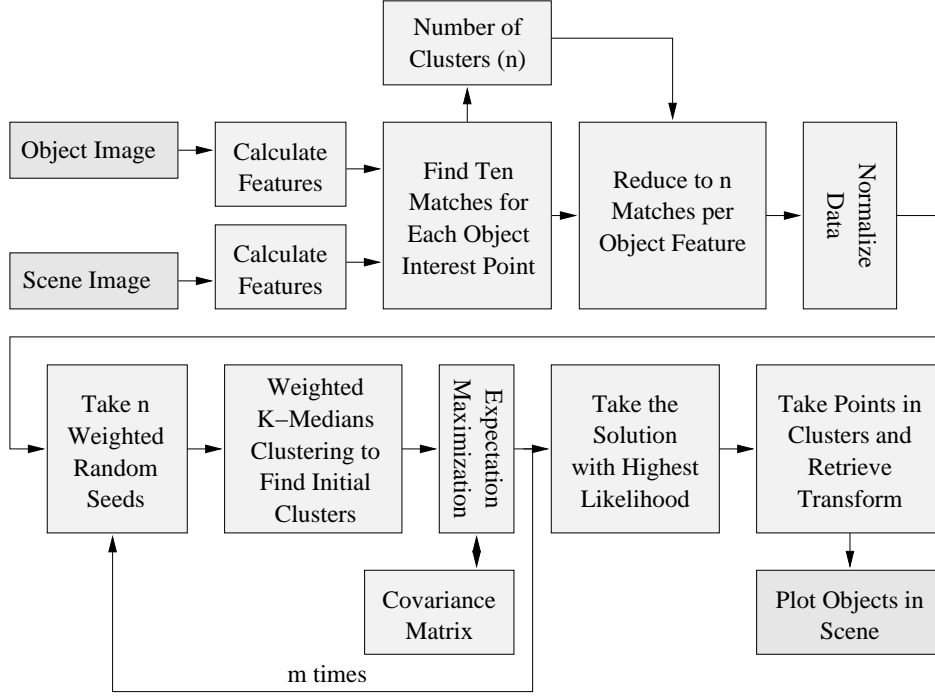


Figure 7.2: Flowchart describing the complete ‘object location and pose retrieval’ algorithm.

point $\theta(x, y, \sigma) = \tan^{-1}(L_y/L_x)$, and the descriptor information (f_1, f_2, \dots, f_n) is stored as illustrated in Figure 7.3. Note that θ is well defined for interest points where $\nabla L \neq \mathbf{0}$. For interest points where $\nabla L = \mathbf{0}$ another intrinsic angle could be used, e.g. the orientation of annihilation, but this occurs so rarely in stable interest points that it can therefore be ignored.

To retrieve the location of the object in the scene, the features of the query object image have to be matched to the features of the scene image. This is done by exhaustively calculating the dissimilarity or distance between all object features and all scene features. For the SIFT, PCA-SIFT and GLOH descriptors (Section 6.2), the distance between descriptors is calculated by simply taking the Euclidean distance between each object descriptor $(f_{i1}^o, f_{i2}^o, \dots, f_{in}^o)$ and each scene descriptor $(f_{j1}^s, f_{j2}^s, \dots, f_{jn}^s)$. For our differential invariant descriptor (Section 6.3) the dissimilarity measure

$$d(\mathbf{f}^o, \mathbf{f}^s) = \sqrt{(\mathbf{f}^s - \mathbf{f}^o)^T \Sigma^{-1} (\mathbf{f}^s - \mathbf{f}^o)} \quad (7.1)$$

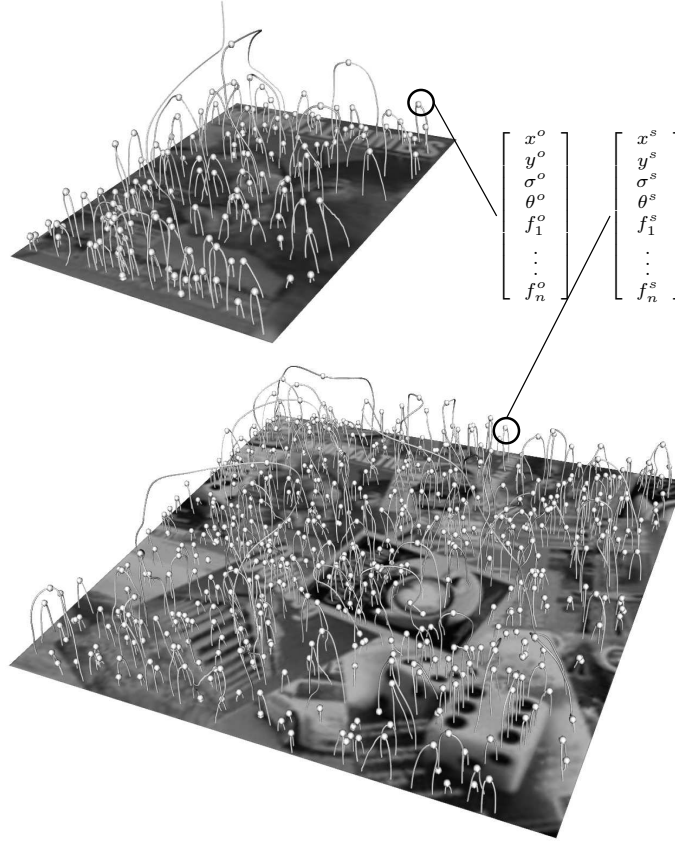


Figure 7.3: The interest points (positions on the critical paths) and features, detected for a query image and a scene image containing the object.

is used as described in Section 6.3.2. Recall that the covariance matrix Σ in (7.1) is calculated analytically for each descriptor by considering the propagation of noise in the differential invariants [9, 5].

By setting a threshold on the dissimilarity, a set of matches between object and scene features can be found. Note that since a threshold is set on the dissimilarity measure, the feature match is not necessarily one to one, but can also be one to none or one to many.

Another possible way of finding matches is to consider all the possible matches to avoid having to set an arbitrary threshold. The drawback of this is that the number of matches to be considered is very large. Images generally contain thousands

of features, which means that there would be millions of matches to consider. To avoid this large amount of data, the n best (most similar) matching scene features are considered per object feature. If the number of times the object is expected in the scene is known, n can be set accordingly. Otherwise an upper bound has to be set on the amount of times an object is expected in a scene. The number of times the object is contained in the scene can then later be estimated as explained in Section 7.4.1. Figure 7.4 illustrates the 100 ‘best’ matches for a simple object retrieval task.



Figure 7.4: The 100 matches with shorted distances for a simple object retrieval task. The features of the query object on the left match to the features of two identical objects in the scene on the right.

7.3 Obtaining Pose Coordinates

Every object feature that has been matched to a scene feature is stored as a pair. For simplicity let's assume feature \mathbf{f}^o matches to scene feature \mathbf{f}^s . Then both

features combined form the following feature pair

$$\left(\begin{bmatrix} x^o \\ y^o \\ \sigma^o \\ \theta^o \\ f_1^o \\ \vdots \\ f_n^o \end{bmatrix} \begin{bmatrix} x^s \\ y^s \\ \sigma^s \\ \theta^s \\ f_1^s \\ \vdots \\ f_n^s \end{bmatrix} \right). \quad (7.2)$$

Note that in the coordinates $(x^o, y^o, \sigma^o, \theta^o)$ and $(x^s, y^s, \sigma^s, \theta^s)$ refer to different bases in the object and scene respectively. The pose of an object in a scene can be described by a transformation. At this stage one single point match is considered and a scale-Euclidean transformation is retrieved. This means that such a point match describes a translation $(\Delta x, \Delta y)$, a scaling λ and a rotation $\Delta\theta$.

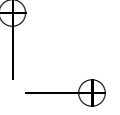
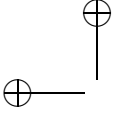
From the information contained in the pair of matched features (7.2) the pose coordinates $(\Delta x, \Delta y, \lambda, \Delta\theta)$ can be extracted in three steps. In step 1 the scaling factor λ is obtained as $\lambda = \sigma^s / \sigma^o$. In step 2 the rotation $\Delta\theta$ is found by taking the difference between the gradient angles θ^s and θ^o . In step 3 the translation $(\Delta x, \Delta y)$ is obtained such that correspondence in spatial position is obtained, i.e.

$$(x^s, y^s) = (\lambda(x^o \cos \Delta\theta - y^o \sin \Delta\theta) - \Delta x, \lambda(x^o \sin \Delta\theta + y^o \cos \Delta\theta) - \Delta y). \quad (7.3)$$

Note that the first two steps commute and can be combined, but the third step does not commute with the previous steps and must therefore be carried out after steps one and two.

The result is a set of coordinates $(\Delta x, \Delta y, \lambda, \Delta\theta)$ for each object-scene feature pair. These coordinates describe points in a 4-dimensional space. This space will be referred to as the *pose space*, as points in this space contain information on the pose of the object in the scene. Note however that these 4 parameters only approximate the full pose for a 3D object and also do not account for any non-rigid transformations.

Each point in pose space is weighted with a weight that is dependent on the distance between the matched features that made this point. Assume point \mathbf{x} in pose space originated from features \mathbf{f}^o and \mathbf{f}^s , then it's weight is defined as $w_x = 1/d(\mathbf{f}^o, \mathbf{f}^s)$ where $d(\mathbf{f}^o, \mathbf{f}^s)$ is defined in Equation (7.1). It is assumed that the distance between stable features $d(\mathbf{f}^o, \mathbf{f}^s)$ is never exactly zero. In this way weak point correspondences will have less influence on the inferred global correspondence as will be discussed in the next section.



7.4 Clustering in Pose Space

As explained in the previous section, each matching object-scene feature pair yields a point in pose space which contains information about the transformation an object in the scene has undergone with respect to the image of the query object. Due to noise, re-lighting, transformations other than scale-Euclidean and miss-matches, these points are not nicely centered but are spread out in pose space. This typically results in one or more clusters in this 4-dimensional space around the locations that describe the pose of the object in the scene. The number of clusters depends on the number of times an object appears in the scene. Since the data is clustered in pose space, an algorithm has to be used to identify these clusters.

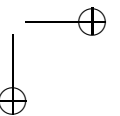
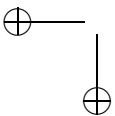
In the case of the SIFT algorithm, Lowe [92, 94] uses a Hough transform [43, 4] to identify possible clusters in pose space. In his algorithm the pose-space is divided into very broad bins. The orientation bin is 30 degrees wide, the scale bins have width of factor 2 and the location bins have a width of 0.25 times the training image dimension. The Hough transform is used to identify clusters with at least 3 entries in a bin. This gives many false responses and needs careful post-processing of the results to eliminate false poses.

Therefore a different method for clustering in pose space is proposed. This method is more sophisticated and less prone to give false responses. The method presented in this chapter uses the *Expectation Maximization* (EM) algorithm [18] to find the clusters.

7.4.1 Identifying the Number of Clusters

Knowing the number of clusters is necessary for the initialization of the proposed clustering algorithms. For some applications the number of object matches in the scene will be known beforehand, but for many other applications the number of times the query object appears in the scene is unknown.

A method for finding the number of clusters is introduced here. In Section 7.3 it was mentioned that by setting an upper bound n on the number of times an object is expected in a scene, an arbitrary threshold on the dissimilarity measure could be avoided. This implies that for each object feature, the n best (most similar) matches are found and stored together with their weight. Thus for all N



object features the following data is stored:

$$\begin{array}{ll}
 \text{object feature 1} & \{(x_1^1, w_{x_1}^1), (x_2^1, w_{x_2}^1), \dots, (x_n^1, w_{x_n}^1)\} \\
 \text{object feature 2} & \{(x_1^2, w_{x_1}^2), (x_2^2, w_{x_2}^2), \dots, (x_n^2, w_{x_n}^2)\} \\
 \vdots & \vdots \\
 \text{object feature N} & \{(x_1^N, w_{x_1}^N), (x_2^N, w_{x_2}^N), \dots, (x_n^N, w_{x_n}^N)\}
 \end{array} \tag{7.4}$$

Where x is the pose space point $(\Delta x, \Delta y, \lambda, \Delta \theta)$ and w_x is the weight of the point and where x_1, \dots, x_n are sorted such that $(w_{x_1} \geq w_{x_2} \geq \dots \geq w_{x_n})$.

The algorithm for finding the number of clusters uses the data as described in Algorithm 5.

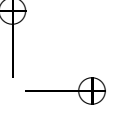
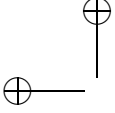
Algorithm 5 Find number of clusters

Take the p highest first weights w_{x_1} .
Calculate $\sum_{i=1}^p w_{x_i}^i$ and store the result r_1
Take the p highest first plus second weights $w_{x_1} + w_{x_2}$
Calculate $\frac{1}{2} \sum_{i=1}^p w_{x_i}^i + w_{x_2}^i$ and store the result r_2
...
Take the p highest weights $\sum_{j=1}^n w_{x_j}$
Calculate $\frac{1}{n} \sum_{i=1}^p \sum_{j=1}^n w_{x_j}^i$ and store the result r_n
Calculate $q_i = r_i - r_{i+1}$ for $i = 1, \dots, n-1$
Now find $\text{argmax}_i(q_i)$, the solution corresponds to the number of clusters.

7.4.2 Expectation Maximization

The EM algorithm is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on so-called unobserved latent variables. EM iteratively alternates between performing an expectation (E) step, which computes the expected value of the latent variables, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters given the data and setting the latent variables to their expectation.

The assumption is made that the clusters in pose space together with the outliers can be modeled as samples drawn from a mixture of Gaussian distributions, a so called *Gaussian Mixture Model* (GMM), combined with a constant distribution to model outliers. Let $\mathbf{C} = \{c_1, \dots, c_k\}$ be the set of Gaussian mixture labels, where k is the number of mixture components and let b denote the uniform distribution that models the outliers. The mixture model is a parametric family of functions that is completely described by the parameter vector Φ . The goal is to estimate



the parameter vectors $\hat{\Phi}$ for which the posterior probability distribution $P(\Phi|\mathbf{X})$ is maximized. In the case of clustering in pose space $\mathbf{X} = \{x_t \in \mathbb{R}^4; t = 1, \dots, T\}$, where each x_t represents a point in pose space which is weighted with a weight w_x as explained in Section 7.3.

According to Bayes rule the probability of the parameter vector for the mixtures Φ , given the found pose space samples X can be expressed as

$$P(\Phi|\mathbf{X}) = \frac{P(\mathbf{X}|\Phi)P(\Phi)}{P(\mathbf{X})}. \quad (7.5)$$

The estimation of parameters is independent of the evidence probability $P(\mathbf{X})$, since it is the same for all parameters. The a posteriori estimation needs knowledge about the a priori distribution $P(\Phi)$ of the parameters. Since no assumptions can be made about the a priori distribution of the parameters, it is assumed that all parameters occur with the same likelihood. In this case the optimization reduces to the maximization of the likelihood of $P(\mathbf{X}|\Phi)$. The maximum likelihood (ML) estimation of the parameters is: $\hat{\Phi} = \text{argmax}_{\Phi} P(\mathbf{X}|\Phi)$.

For the exact derivation of the expectation maximization algorithm the reader is referred to e.g. [18, 41, 136], for the case considered here the algorithm works as follows. First the algorithm has to be initialized with an initial guess per mixture. Each Gaussian mixture is defined by its weight ω_c , its location μ_c and its covariance matrix that defines its shape Σ_c . So for each Gaussian mixture:

$$\Phi_c = \begin{bmatrix} \omega_c \\ \mu_c \\ \Sigma_c \end{bmatrix}. \quad (7.6)$$

and for the uniform distribution modeling the outliers only a weight is set $\Phi_b = \omega_b$. In the Expectation step a membership function h_{xc} for all $\mathbf{x} \in \mathbf{X}$ and $c \in \mathbf{C}$ is calculated which defines the membership of a sample \mathbf{x} to a mixture c .

For the Gaussian mixtures

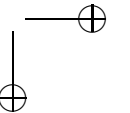
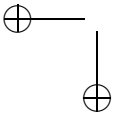
$$h_{xc} = \frac{P(\mathbf{x}|c, \Phi^{(j)})P(\mathbf{x}|\Phi^{(j)})}{P(\mathbf{x})} \quad (7.7)$$

$$= \frac{\frac{\omega_c^{(j)}}{4\pi^2 \sqrt{\det(\Sigma_c^{(j)})}} e^{\frac{1}{2}(\mathbf{x}-\mu_c^{(j)})^T (\Sigma_c^{(j)})^{-1} (\mathbf{x}-\mu_c^{(j)})}}{\sum_{c=1}^k \frac{\omega_c^{(j)}}{4\pi^2 \sqrt{\det(\Sigma_c^{(j)})}} e^{\frac{1}{2}(\mathbf{x}-\mu_c^{(j)})^T (\Sigma_c^{(j)})^{-1} (\mathbf{x}-\mu_c^{(j)})} + \omega_b} \quad (7.8)$$

and for the uniform distribution modeling the outliers

$$h_{xb} = \frac{P(\mathbf{x}|b, \Phi^{(j)})P(\mathbf{x}|\Phi^{(j)})}{P(\mathbf{x})} \quad (7.9)$$

$$= \frac{\omega_b}{\sum_{c=1}^k \frac{\omega_c^{(j)}}{4\pi^2 \sqrt{\det(\Sigma_c^{(j)})}} e^{\frac{1}{2}(\mathbf{x}-\mu_c^{(j)})^T (\Sigma_c^{(j)})^{-1} (\mathbf{x}-\mu_c^{(j)})} + \omega_b}. \quad (7.10)$$



After calculating the membership functions h_{xc} and h_{xb} new parameters $\Phi^{(j+1)}$ are calculated in the Maximization step as follows:

$$\omega_c^{(j+1)} = \frac{1}{\sum_{\mathbf{x} \in \mathbf{X}} w_x} \sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xc} \quad (7.11)$$

$$\omega_b^{(j+1)} = \frac{1}{\sum_{\mathbf{x} \in \mathbf{X}} w_x} \sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xb} \quad (7.12)$$

$$\mu_c^{(j+1)} = \frac{\sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xc} \mathbf{x}}{\sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xc}} \quad (7.13)$$

$$\Sigma_c^{(j+1)} = \frac{\sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xc} (\mathbf{x} - \mu_c)(\mathbf{x} - \mu_c)^T}{\sum_{\mathbf{x} \in \mathbf{X}} w_x h_{xc}} \quad (7.14)$$

The E and M step are iterated until convergence of the log-likelihood is reached. The log-likelihood gives a measure on the fit of the data to the proposed mixture model.

$$L(\mathbf{X}, h|\Phi) =$$

$$\sum_{\mathbf{x} \in \mathbf{X}} \left(\sum_{c=1}^k \left(h_{xc} \log\left(\frac{\omega_c}{4\pi^2 \sqrt{\det(\Sigma_c)}}\right) - \frac{1}{2} (\mathbf{x} - \mu_c)^T (\Sigma_c)^{-1} (\mathbf{x} - \mu_c) \right) + h_{xb} \log(\omega_b) \right) \quad (7.15)$$

The EM algorithm maximizes the log-likelihood each iteration. After a number of iterations the log-likelihood does not change anymore and the EM algorithm has converged to a local ‘best fit’.

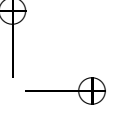
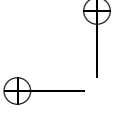
Since the EM algorithm converges to a local maximum of the log-likelihood function, it is dependent on a ‘good’ initialization of the parameters Φ . Especially the mean of the Gaussian mixtures μ_c has to be initialized accurately. The EM algorithm is run for a number of initializations of the parameters Φ and the result that yields the highest log-likelihood is chosen as the best model for our pose data. The initial guess of the location of the mixtures will be found by means of a simple and fast k -medians clustering algorithm as explained in Section 7.4.4.

7.4.3 Normalizing the Pose Space

To be able to identify clusters without having to adjust parameters for every image size the pose space is normalized. The angle dimensions are bounded by

$$(-\pi < \Delta\theta \leq \pi).$$

The maximum and minimum scaling factor λ have to be set manually. These parameters depend on the application in which the matching algorithm is used.



For the purposes in this chapter the maximum and minimum scaling factors are set as

$$(0.2 \leq \lambda \leq 5).$$

This means that matches can be found for objects in the scene that range in scale from 5 times smaller to 5 times larger than the query object. The spatial dimension Δx is bounded by

$$(-\lambda_{max}\sqrt{xdim_o^2 + ydim_o^2} < \Delta x < xdim_s + \lambda_{max}\sqrt{xdim_o^2 + ydim_o^2})$$

and Δy by

$$(-\lambda_{max}\sqrt{xdim_o^2 + ydim_o^2} < \Delta y < ydim_s + \lambda_{max}\sqrt{xdim_o^2 + ydim_o^2})$$

as illustrated in Figure 7.5, where $(xdim_o, ydim_o)$ and $(xdim_s, ydim_s)$ are the object and scene dimensions respectively.

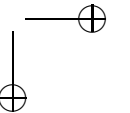
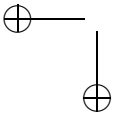
With these boundaries the pose space is normalized. All pose coordinates for which $\lambda > \lambda_{max}$ and $\lambda < \lambda_{min}$ are removed from the point set. All other coordinates are normalized such that $\Delta x, \Delta y, \lambda$ and $\Delta \theta \in [0, 1]$. The advantage of this is that the initialization for the covariance of the Gaussian mixtures modeling the clusters, and the weight of the uniform distribution modeling the outliers, can be set regardless of the dimensions of the object and scene images.

7.4.4 Weighted k -Medians Clustering

To make an estimation of the initial means μ_c for the Gaussian mixture model used in the expectation maximization algorithm, a weighted k -medians clustering approach is used. The k -medians algorithm is similar to the well known k -means algorithm, but is less sensitive to outliers.

The k -medians algorithm is randomly initialized with k initial cluster centroids, where k is the number of clusters. Every weighted pose space point is assigned to the cluster whose centroid is closest to that point (measured with the Euclidean distance). This gives k sets of weighted points.

For these point sets new centroids are calculated by taking the weighted median (Figure 7.6) for each set of sample points. The pose space points are now assigned to the closest new centroid and these two steps are alternated until there is no change in the assignment of the pose space points. The resulting medians are used as initial values for the means μ_c of the Gaussian mixtures in the EM algorithm. Since the weighted k -medians are found with random initialization the results may vary. This is why the EM algorithm is run for a number of different initializations



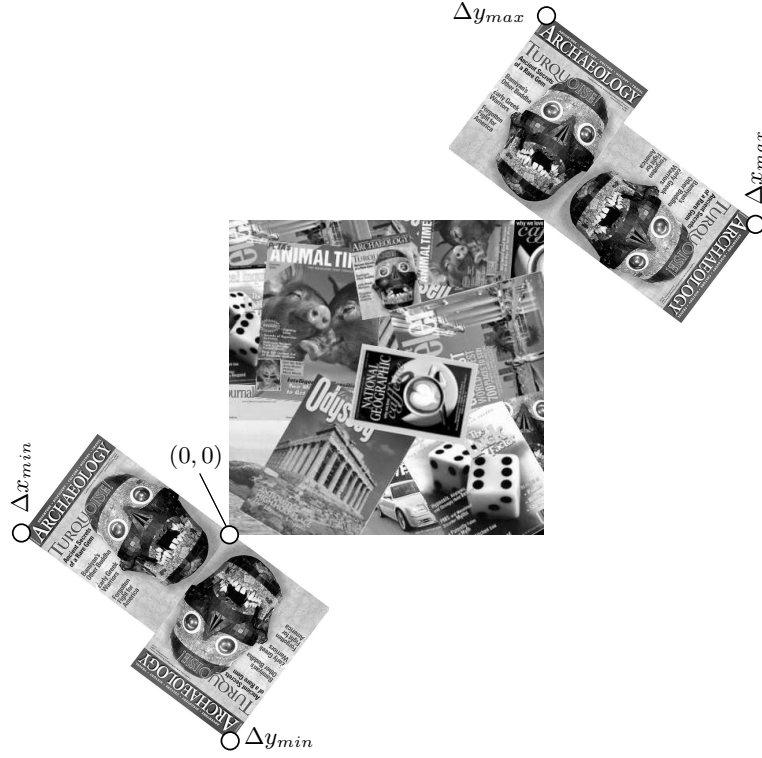


Figure 7.5: The spatial boundaries in pose space are $(-\lambda_{max}\sqrt{xdim_o^2 + ydim_o^2} < \Delta x < xdim_s + \lambda_{max}\sqrt{xdim_o^2 + ydim_o^2})$ and $(-\lambda_{max}\sqrt{xdim_o^2 + ydim_o^2} < \Delta y < ydim_s + \lambda_{max}\sqrt{xdim_o^2 + ydim_o^2})$. These are the farthest possible translations given by a match, as illustrated. Where λ_{max} is the maximum scaling factor considered and where $(xdim_o, ydim_o)$ and $(xdim_s, ydim_s)$ are respectively the object and scene dimensions.

of the k -medians algorithm and the result with the best log-likelihood is chosen. This is illustrated in the bottom of the flowchart (Figure 7.2).

The expectation maximization algorithm returns the location μ_c for each cluster, the covariance matrix Σ_c and the weight of the cluster ω_c . At this stage clusters with a very low weight could be discarded by setting a threshold on the cluster weight.

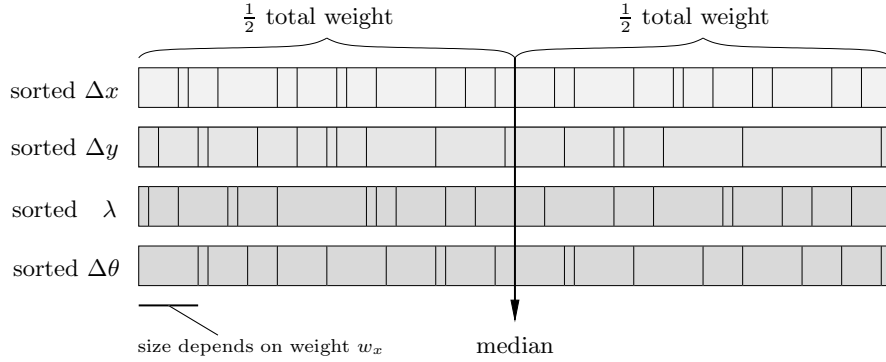


Figure 7.6: The weighted median is calculated as follows. The coordinates Δx , Δy , λ and $\Delta\theta$ are treated separately. The coordinates are then sorted on their values and the weights w_x are taken into account. The median is the value where half of the weight is on the lower side and half of the weight is on the higher side.

7.5 Retrieving the Pose

The final step in the object retrieval algorithm is to extract the pose of the object in the scene from the found clusters. If the only transformation the object has undergone in the scene is a scale-Euclidean transformation, then the cluster centers μ_c are enough to describe this transformation as they contain the translation, scaling and rotation of the object in the scene $(\Delta x, \Delta y, \lambda, \Delta\theta)$ (Figure 7.7).

However in most applications the object can undergo different types of transformations. In this section affine transformations and perspective changes of planar objects will be considered as possible transformations. Transformations that occur due to 3-dimensional rotations of non-planar objects are not considered, but they can up to some extent be modeled by affine or perspective transformations.

Noise and transformations other than scale-Euclidean transformations cause the clusters to be more spread out. The EM algorithm finds these clusters as Gaussians located at μ_c and with a covariance matrix Σ_c . How spread out the cluster is is described by its covariance matrix. By using the Mahalanobis distance $d(\mathbf{x}, \mu_c) = \sqrt{(\mathbf{x} - \mu_c)^T \Sigma_c^{-1} (\mathbf{x} - \mu_c)}$ points are selected that have a distance smaller than a threshold to the cluster center. From these cluster points not the pose coordinates themselves are taken, but the object feature information $(x_o, y_o, \sigma_o, \theta_o)$ and the scene feature information $(x_s, y_s, \sigma_s, \theta_s)$ which were used to construct the pose coordinates $(\Delta x, \Delta y, \lambda, \Delta\theta)$ (Equation 7.3). With this information it will be possible to retrieve a pose of the object that is more sophisticated



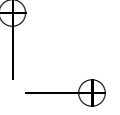
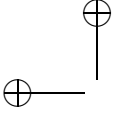
Figure 7.7: The found cluster centers μ_1 and μ_2 in pose space describe the location and pose of the query object in the scene image. Where $(\Delta x_1, \Delta y_1, \lambda_1, \Delta \theta_1) = (31, 175, 0.66, 5^\circ)$ and $(\Delta x_2, \Delta y_2, \lambda_2, \Delta \theta_2) = (255, 310, 0.40, -85^\circ)$. The circles in the image represent the matched features. The size of the circles represents the scale of the interest points.

than simply the rotation, scaling and translation. It is for example possible to retrieve the affine transformation of an object as follows.

The affine transformation of an object point (x^o, y^o) to an image point (x^s, y^s) can be written as:

$$\begin{bmatrix} x^s \\ y^s \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x^o \\ y^o \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (7.16)$$

where the object translation is (t_x, t_y) and the affine rotation, scale and stretch are represented by the m_i parameters. Since the goal is to solve for the transformation parameters, Equation (7.16) is rewritten to gather the unknowns into a column



vector:

$$\begin{bmatrix} x^o & y^o & 0 & 0 & 1 & 0 \\ 0 & 0 & x^o & y^o & 0 & 1 \\ & & \dots & & & \\ & & \dots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x^s \\ y^s \\ \vdots \end{bmatrix} \quad (7.17)$$

This equation shows a single match, but any number of matched pairs can be added, where each matched pair contributes two extra rows to the first and last matrix. At least three matched pairs are needed to provide a solution.

This linear system can be written as $\mathbf{A}\mathbf{p} = \mathbf{b}$ and the least-squares solution for the parameters $\mathbf{p} = [m_1, m_2, m_3, m_4, t_x, t_y]^T$ can be determined by solving

$$\mathbf{p} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b}. \quad (7.18)$$

Equation (7.18) minimizes the sum of squares of the distances from the projected object locations to the corresponding scene locations.

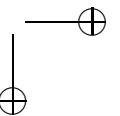
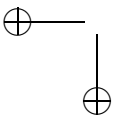
For perspective changes of planar objects a similar linear system can be written as

$$\begin{bmatrix} 1 & x^o & y^o & 0 & 0 & 0 & -x^s x^o & -x^s y^o \\ 0 & 0 & 0 & 1 & x^o & y^o & -y^s x^o & -y^s y^o \\ & & & \dots & & & & \\ & & & \dots & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \end{bmatrix} = \begin{bmatrix} x^s \\ y^s \\ \vdots \end{bmatrix}. \quad (7.19)$$

By using Equation (7.18) the least-squares solution for the unknown parameters $\mathbf{p} = [m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8]^T$ can be found. A minimum of four point matches is needed to find a solution.

7.6 Retrieval examples

In this section some retrieval examples are show. A set of magazine covers is used as objects for the retrieval task. The images of the query objects are grayscale images of 376×500 pixels (Figure 7.8). For the retrieval task three different scene images are used (Figure 7.9), the first image is a synthetic image, the other two images are real photographs of the magazine covers put on a desk.



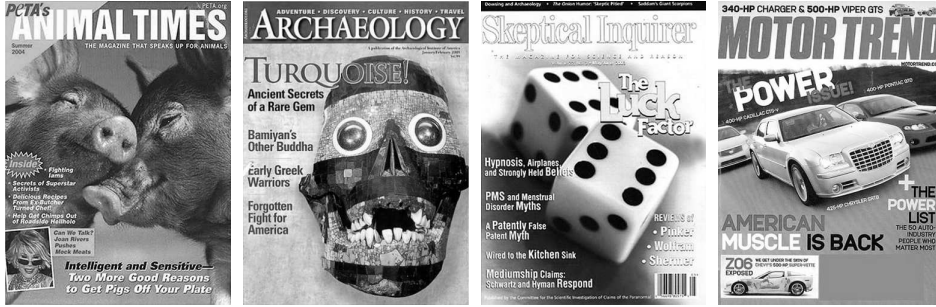


Figure 7.8: The query object images are grayscale images of 376×500 pixels.



Figure 7.9: All scene images are grayscale images. Scene 1 is a synthetic image containing only scale-Euclidean transformations of the objects, the size is 400×400 pixels. Scene 2 is a real photograph of a scene containing the magazine covers, the image size is 700×700 pixels. The image contains only perspective transformations. Scene 3 also is a real photograph of a scene containing the magazine covers, the image size is 700×700 pixels. The image contains magazine covers that are slightly bent over other objects.



Figure 7.10: Some objects retrieved from scene 2, the circles represent the interest points on which the recovered pose is based. The size of the circle represents the scale of the interest point. The pose is calculated using Equation (7.19).



Figure 7.11: Some objects retrieved from scene 2 and scene 3, the circles represent the interest points on which the recovered pose is based. The size of the circle represents the scale of the interest point. The pose is calculated using Equation (7.19). In the first image it is shown that the algorithm works even under big amounts of occlusion. In the second image the magazine is matched even though large parts of the cover are bent.



Figure 7.12: The pose of the ‘Motor Trend’ magazine cover cannot be correctly calculated in scene 2, by using Equation (7.19). The lack of points in the corners of the magazine make it impossible to accurately describe the correct perspective pose of the object. The scale-Euclidean information contained in the cluster center μ_c however is still accurate as shown in the bottom figure.

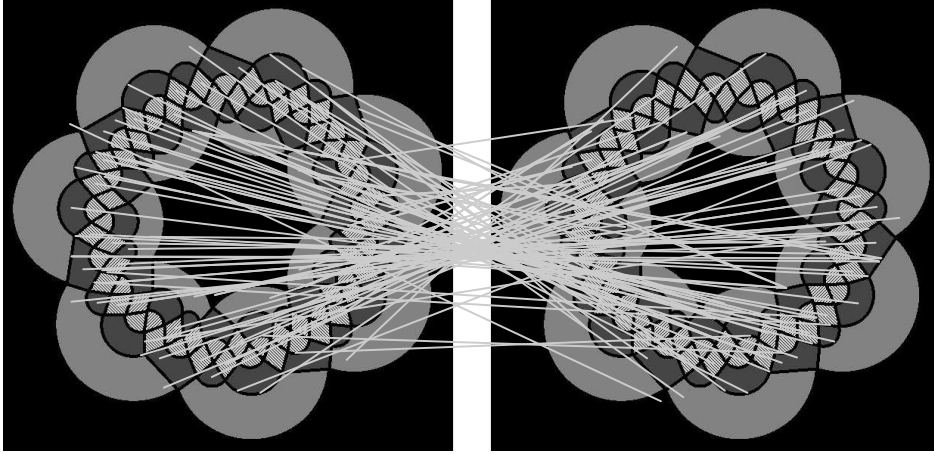
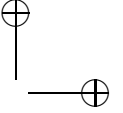
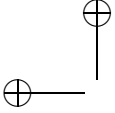


Figure 7.13: This figure illustrates the problems that occur when matching symmetric objects. The lines show matching interest points (lines connecting identical interest points in the images are not shown). Seven different pose space clusters are found in this case. This is not a shortcoming of the algorithm but a logical consequence of the symmetry of the object.

7.7 Summary and Conclusion

In this chapter an algorithm was proposed that retrieves the location and pose of a query object contained in a scene image. The algorithm uses the interest points and descriptors of the object image and compares them to the interest points and descriptors of the scene image. Matches between these features yield points in pose space. By means of clustering in this pose space and solving for perspective transformations based on the points contained in the clusters, the pose of the object is retrieved.

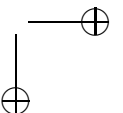
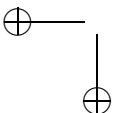
For clustering the expectation maximization algorithm is used. The pose space is modeled as a mixture of Gaussians corresponding to the clusters and a uniform distribution that models the outliers. The EM algorithm is used to obtain the most likely parameters of the mixture model. The advantage of this method is that the properties of the clusters are known after running the EM algorithm. The size, location and shape are described by the parameters found by the algorithm. This makes it possible to accurately select points contained in the cluster and to say something about the certainty of the cluster, e.g. a cluster with a high weight and small size is very likely to be accurate and most relevant. From the features contained in the pose space clusters, the affine or projective pose of an object can

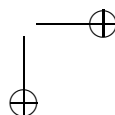
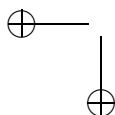
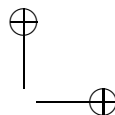
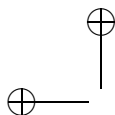


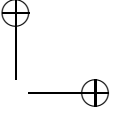
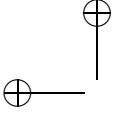
be obtained by finding the least-squares solution of a linear system.

One could attempt to solve for a smooth deformation with many parameters estimated from the object/scene feature map. This would make it possible to retrieve deformations like the bending of a magazine cover as illustrated in the third image of Figure 7.9. This however requires a large amount of correctly matched features and little false-positive matches.

The experiments show that the algorithm performs well in most cases even when a large part of the object is occluded in the scene image. When however too much of the object is occluded a perspective transformation can not be retrieved accurately. In theory four matches should be enough to find a solution for the transformation, however a number of matches sufficiently spread in the object has to be found to make an accurate estimation of the pose. The scale-Euclidean transformation obtained from the pose space cluster center, is often still reasonably accurate in this case.

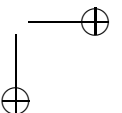
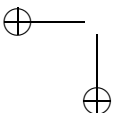


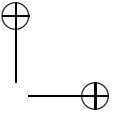
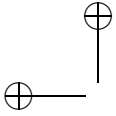




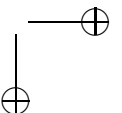
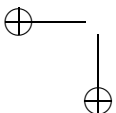
Bibliography

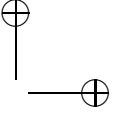
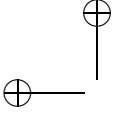
- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, February 1993.
- [2] M. A. Almsick and B. M. ter Haar Romeny. Mathvisiontools. <http://www.mathvisiontools.net>, 2006.
- [3] V. I. Arnold. *Catastrophe Theory*. Springer-Verlag, Berlin, third, revised and expanded edition, 1992.
- [4] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 714–725, 1987.
- [5] E. Balmachnova, L. M. J. Florack, B. Platel, F. M. W. Kanters, and B. M. ter Haar Romeny. Stability of top-points in scale space. In Kimmel et al. [62], pages 62–72.
- [6] E. Balmachnova, L. M. J. Florack, B. Platel, F. M. W. Kanters, and B. M. ter Haar Romeny. Stability of top-points in scale space. In *Proceedings of the 5th international conference on Scale Space Methods in Computer Vision*, pages 62–72, Hofgeismar, Germany, 2005.
- [7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
- [8] J. Blom. *Topological and Geometrical Aspects of Image Structure*. PhD thesis, University of Utrecht, Department of Medical and Physiological Physics, Utrecht, The Netherlands, 1992.
- [9] J. Blom, B. M. ter Haar Romeny, A. Bel, and J. J. Koenderink. Spatial derivatives and the propagation of noise in Gaussian scale-space. *Journal of Visual Communication and Image Representation*, 4(1):1–13, March 1993.
- [10] M. Brown and D. Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference*, pages 656–665, Cardiff, UK, 2002.
- [11] M. Brown and D. G. Lowe. Recognising panoramas. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1218, Washington, DC, USA, 2003. IEEE Computer Society.



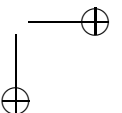
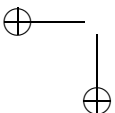


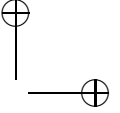
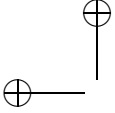
- [12] S. D. Cohen and L. J. Guibas. The Earth Mover's Distance under transformation sets. In *Proceedings of the 7th International Conference on Computer Vision*, pages 1076–1083, Kerkyra, Greece, 1999.
- [13] J. Damon. Local Morse theory for solutions to the heat equation and Gaussian blurring. *Journal of Differential Equations*, 115(2):368–401, January 1995.
- [14] J. Damon. Local Morse theory for Gaussian blurred functions. In Sparring et al. [128], chapter 11, pages 147–163.
- [15] M. Fatih Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many matching of scale-space feature hierarchies using metric embedding. In *Proceedings, Scale Space Methods in Computer Vision, 4th International Conference*, pages 17–32, June 2003.
- [16] M. Fatih Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many feature matching using spherical coding of directed graphs. In *Proceedings, 8th European Conference on Computer Vision*, pages 332–335, May 2004.
- [17] M. Fatih Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson. Object recognition as many-to-many feature matching. *International Journal of Computer Vision*, 69(2):203–222, 2006.
- [18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:185–197, 1977.
- [19] R. Duits, L. M. J. Florack, J. de Graaf, and B. ter Haar Romeny. On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, 20(3):267–298, 2004.
- [20] D. Eberly. A differential geometric approach to anisotropic diffusion. In B. M. ter Haar Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, volume 1 of *Computational Imaging and Vision Series*, pages 371–392. Kluwer Academic Publishers, Dordrecht, 1994.
- [21] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, Madison, Wisconsin, June 2003.
- [22] L. M. J. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Department of Medicine, Utrecht, The Netherlands, November 10 1993. (cum laude).



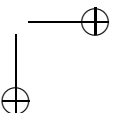
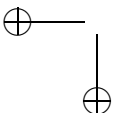


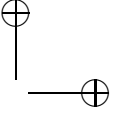
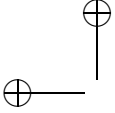
- [23] L. M. J. Florack. Detection of critical points and top-points in scale-space. In P. Johansen, editor, *Proceedings fra Den Femte Danske Konference om Mønstergenkendelse og Billedanalyse*, pages 73–81, August 1996. DIKU Tech. Rep. Nr. 96/22.
- [24] L. M. J. Florack. *Image Structure*, volume 10 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [25] L. M. J. Florack. A geometric model for cortical magnification. In Lee et al. [81], pages 574–583.
- [26] L. M. J. Florack. Visuele perceptie en digitale beeldverwerking. *Nieuw Archief voor Wiskunde*, 3(1):34–41, March 2002. (in Dutch).
- [27] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and second order invariants. In Johansen and Olsen [51], pages 22–29. Selected papers from the 7th Scandinavian Conference on Image Analysis.
- [28] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, July/August 1992.
- [29] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Cartesian differential invariants in scale-space. *Journal of Mathematical Imaging and Vision*, 3(4):327–348, November 1993.
- [30] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4(2):171–187, 1994.
- [31] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Linear scale-space. *Journal of Mathematical Imaging and Vision*, 4(4):325–351, 1994.
- [32] L. M. J. Florack, B. J. Janssen, F. M. W. Kanters, and R. Duits. Towards a new paradigm for motion extraction. In Springer-Verlag, editor, *International Conference on Image Analysis and Recognition (ICIAR 2006)*, pages 743–754, Povia de Verzim, Portugal, September 2006.
- [33] L. M. J. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–79, February 2000.
- [34] R. Gilmore. *Catastrophe Theory for Scientists and Engineers*. Dover Publications, Inc., New York, 1993. Originally published by John Wiley & Sons, New York, 1981.



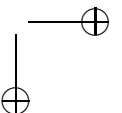
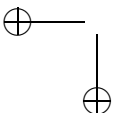


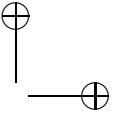
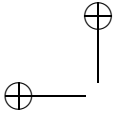
- [35] L. D. Griffin and A. C. F. Colchester. Superficial and deep structure in linear diffusion scale space: Isophotes, critical points and separatrices. *Image and Vision Computing*, 13(7):543–557, September 1995.
- [36] L. D. Griffin and M. Lillholm, editors. *Scale-Space Methods in Computer Vision: Proceedings of the Fourth International Conference, Scale-Space 2003, Isle of Skye, UK*, volume 2695 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, June 2003.
- [37] B. M. ter Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis: Multi-Scale Computer Vision Theory and Applications, written in Mathematics*, volume 27 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
- [38] B. M. ter Haar Romeny and L. M. J. Florack. Front-end vision: A multiscale geometry engine. In Lee et al. [81], pages 297–307.
- [39] B. M. ter Haar Romeny, L. M. J. Florack, J. J. Koenderink, and M. A. Viergever, editors. *Scale-Space Theory in Computer Vision: Proceedings of the First International Conference, Scale-Space'97, Utrecht, The Netherlands*, volume 1252 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, July 1997.
- [40] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [41] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Prentice Hall, 2005.
- [42] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, 1986.
- [43] P. V. C. Hough. Methods and means for recognizing complex patterns. United States Patent 3,069,654, 1962.
- [44] D. H. Hubel. *Eye, Brain and Vision*, volume 22 of *Scientific American Library*. Scientific American Press, New York, 1988.
- [45] T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrical Laboratory*, 26:368–388, 1962. (in Japanese).
- [46] Staal J., S. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. Detection of critical structures in scale space. In *SCALE-SPACE '99: Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, pages 105–116, London, UK, 1999. Springer-Verlag.



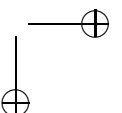
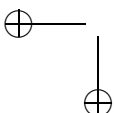


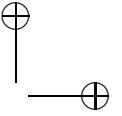
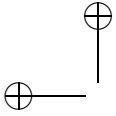
- [47] A. Bel J. Blom, B. M. ter Haar Romeny and J. J. Koenderink. Derivatives and the propagation of noise in gaussian scale space. *Journal of Visual Communication and Image Representation*, 4(1):1–13, 1993.
- [48] B. J. Janssen, L. M. J. Florack, R. Duits, and B. M. ter Haar Romeny. Optic flow from multi-scale dynamic anchor point attributes. In Springer-Verlag, editor, *International Conference on Image Analysis and Recognition (ICIAR 2006)*, pages 767–779, Povoia de Verzim, Portugal, September 2006.
- [49] P. Johansen. On the classification of toppoints in scale space. *Journal of Mathematical Imaging and Vision*, 4(1):57–67, 1994.
- [50] P. Johansen. Local analysis of image scale space. In Sporring et al. [128], chapter 10, pages 139–146.
- [51] P. Johansen and S. Olsen, editors. *Theory & Applications of Image Analysis*, volume 2 of *Series in Machine Perception and Artificial Intelligence*. World Scientific, Singapore, 1992. Selected papers from the 7th Scandinavian Conference on Image Analysis.
- [52] P. Johansen, S. Skelboe, K. Grue, and J. D. Andersen. Representing signals by their top points in scale-space. In *Proceedings of the 8th International Conference on Pattern Recognition (Paris, France, October 1986)*, pages 215–217. IEEE Computer Society Press, 1986.
- [53] S. N. Kalitzin. Topological numbers and singularities. In Sporring et al. [128], chapter 13, pages 181–189.
- [54] S. N. Kalitzin, B. M. ter Haar Romeny, A. H. Salden, P. F. M. Nacken, and M. A. Viergever. Topological numbers and singularities in scalar images: Scale-space evolution properties. *Journal of Mathematical Imaging and Vision*, 9(3), November 1998.
- [55] F. M. W. Kanters. Scalespaceviz: Software for visualizing α -scale spaces. <http://www.bmi2.bmt.tue.nl/image-analysis/people/FKanters>, 2004.
- [56] F. M. W. Kanters, L. M. J. Florack, R. Duits, and B. Platel. α -scale spaces in practice. *Pattern Recognition and Image Analysis*, 15(1):208–211, 2005.
- [57] F. M. W. Kanters, L. M. J. Florack, B. Platel, and B. M. ter Haar Romeny. Image reconstruction from multiscale critical points. In Griffin and Lillholm [36], pages 464–478.
- [58] F. M. W. Kanters, B. Platel, L. M. J. Florack, and B. M. ter Haar Romeny. Content based image retrieval using multiscale top points. In Griffin and Lillholm [36], pages 33–43.



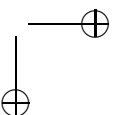
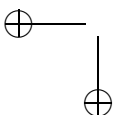


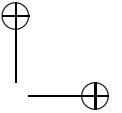
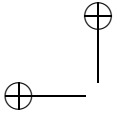
- [59] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR*, volume 2, pages 506–513, Washington, DC, USA, 2004.
- [60] M. Kerckhove, editor. *Scale-Space and Morphology in Computer Vision: Proceedings of the Third International Conference, Scale-Space 2001, Vancouver, Canada*, volume 2106 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, July 2001.
- [61] Y. Keselman, A. Shokoufandeh, M. F. Demirci, and S. Dickinson. Many-to-many graph matching via low-distortion embedding. In *Proceedings, Computer Vision and Pattern Recognition*, pages 850–857, 2003.
- [62] R. Kimmel, N. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision: Proceedings of the Fifth International Conference, Scale-Space 2005, Hofgeismar, Germany*, volume 3459 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, April 2005.
- [63] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [64] J. J. Koenderink. The structure of the visual field. In W. Güttinger and G. Dangelmayr, editors, *The Physics of Structure Formation: Theory and Simulation. Proceedings of an International Symposium*, Tübingen, Germany, October 27–November 2 1986. Springer-Verlag.
- [65] J. J. Koenderink. A hitherto unnoticed singularity of scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1222–1224, November 1989.
- [66] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, 1990.
- [67] J. J. Koenderink and A. J. van Doorn. Dynamic shape. *Biological Cybernetics*, 53:383–396, 1986.
- [68] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
- [69] J. J. Koenderink and A. J. van Doorn. The structure of two-dimensional scalar fields with applications to vision. *Biological Cybernetics*, 33:151–158, 1979.
- [70] J. J. Koenderink and A. J. van Doorn. Operational significance of receptive field assemblies. *Biological Cybernetics*, 58:163–171, 1988.
- [71] J. J. Koenderink and A. J. van Doorn. Receptive field families. *Biological Cybernetics*, 63:291–298, 1990.



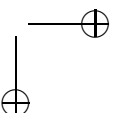
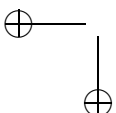


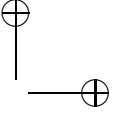
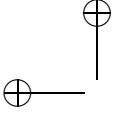
- [72] J. J. Koenderink and A. J. van Doorn. Receptive field taxonomy. In R. Eckmiller, editor, *Advanced Neural Computers*, pages 295–301. Elsevier Science Publishers B.V. (North-Holland), 1990.
- [73] A. Kuijper. *The Deep Structure of Gaussian Scale Space Images*. PhD thesis, University of Utrecht, Department of Computer Science, Utrecht, The Netherlands, June 17 2002.
- [74] A. Kuijper and L. M. J. Florack. Calculations on critical points under Gaussian blurring. In Nielsen et al. [103], pages 318–329.
- [75] A. Kuijper and L. M. J. Florack. Hierarchical pre-segmentation without prior knowledge. In *Proceedings of the 8th International Conference on Computer Vision (Vancouver, Canada, July 9–12, 2001)*, pages 487–493. IEEE Computer Society Press, 2001.
- [76] A. Kuijper and L. M. J. Florack. The hierarchical structure of images. *IEEE Transactions on Image Processing*, 12(9):1067–1079, 2003.
- [77] A. Kuijper and L. M. J. Florack. Exploiting deep structure. In Olsen et al. [107], pages 169–180.
- [78] A. Kuijper, L. M. J. Florack, and M. A. Viergever. Scale space hierarchy. *Journal of Mathematical Imaging and Vision*, 18(2):169–189, March 2003.
- [79] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 319–324, 2003.
- [80] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005.
- [81] S.-W. Lee, H. H. Bülthoff, and T. Poggio, editors. *Biologically Motivated Computer Vision*, volume 1811 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, May 2000.
- [82] L. M. Lifshitz and S. M. Pizer. A multi-resolution hierarchical approach to image segmentation based on intensity extrema. In *Information Processing in Medical Imaging Meeting*, pages 107–130, The Netherlands, 1987.
- [83] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–541, 1990.
- [84] T. Lindeberg. On the behaviour in scale-space of local extrema and blobs. In Johansen and Olsen [51], pages 38–47. Selected papers from the 7th Scandinavian Conference on Image Analysis.



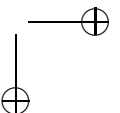
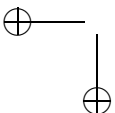


- [85] T. Lindeberg. Scale-space behaviour of local extrema and blobs. *Journal of Mathematical Imaging and Vision*, 1(1):65–99, March 1992.
- [86] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270, 1994.
- [87] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
- [88] T. Lindeberg. On the axiomatic foundations of linear scale-space. In Sporring et al. [128], chapter 6, pages 75–97.
- [89] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
- [90] T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *Proceedings of the 7th International Conference on Computer Vision*, pages 456–462, Kerkyra, Greece, 1999.
- [91] M. Loog, J. J. Duistermaat, and L. M. J. Florack. On the behavior of spatial critical points under Gaussian blurring. a folklore theorem and scale-space constraints. In Kerckhove [60], pages 183–192.
- [92] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
- [93] D. G. Lowe. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image. United States Patent 6,711,293, March 2000.
- [94] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [95] K. Mikolajczyk. Detection of local features invariant to affine transformations. Ph.D. thesis, Institut National Polytechnique de Grenoble, France, 2002.
- [96] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1792–1799, Washington, DC, USA, 2005. IEEE Computer Society.
- [97] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 257–263, Madison, WI, USA, 2003. IEEE Computer Society.

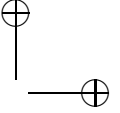
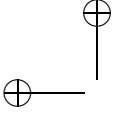




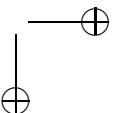
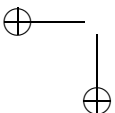
- [98] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.
- [99] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [100] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.
- [101] M. Nielsen. From paradigm to algorithms in computer vision. Master’s thesis, Department of Computer Science, University of Copenhagen, March 1995.
- [102] M. Nielsen, L. Florack, and R. Deriche. Regularization and scale space. Technical Report INRIA-RR-2352, INRIA Sophia-Antipolis, France, September 1994.
- [103] M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors. *Scale-Space Theories in Computer Vision: Proceedings of the Second International Conference, Scale-Space’99, Corfu, Greece*, volume 1682 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, September 1999.
- [104] M. Nielsen and M. Lillholm. What do features tell about images? In Kerckhove [60], pages 39–50.
- [105] A. Okabe and B. Boots. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. New York, 1992.
- [106] O. F. Olsen. Tree edit distances from singularity theory. In *Proceedings of the 5th international conference on Scale Space Methods in Computer Vision*, pages 316–326, Hofgeismar, Germany, 2005.
- [107] O. F. Olsen, L.M.J. Florack, and A. Kuijper, editors. *Deep Structure, Singularities and Computer Vision*, volume 3753 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.
- [108] O. F. Olsen and M. Nielsen. Multiscale gradient magnitude watershed segmentation. In *ICIAP ’97: Proceedings of the 9th International Conference on Image Analysis and Processing-Volume I*, pages 6–13, London, UK, 1997. Springer-Verlag.
- [109] L. G. Parratt. *Probability and experimental errors in science*. John Wiley and SONS, inc., 1961.
- [110] E. J. Pauwels, L. J. Van Gool, P. Fiddelaers, and T. Moons. An extended class of scale-invariant and recursive scale space filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):691–701, July 1995.

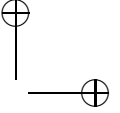
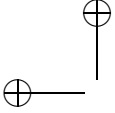


- [111] B. Platel, L. M. J. Florack, F. M. W. Kanters, and E. G. Balmachnova. Using multiscale top points in image matching. In *Proceedings of the 11th International Conference on Image Processing (Singapore, October 24–27, 2004)*, pages 389–392. IEEE, 2004.
- [112] T. Poston and I. N. Stewart. *Catastrophe Theory and its Applications*. Pitman, London, 1978.
- [113] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, New York, NY, 1985.
- [114] S. Rachev. The Monge-Kantorovich mass transference problem and its stochastic applications. *Theory of Probability and Applications*, 29:647–676, 1985.
- [115] V. Roth and B. Ommer. Exploiting low-level image segmentation for object recognition. In *Pattern Recognition–DAGM 2006*, volume 4174 of *LNCS*. Springer, 2006.
- [116] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover’s Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [117] P. T. Saunders. *An Introduction to Catastrophe Theory*. Cambridge University Press, Cambridge, 1980.
- [118] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or ”how do i organize my holiday snaps?”. In *ECCV ’02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431, London, UK, 2002. Springer-Verlag.
- [119] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
- [120] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 226–231, Lausanne, Switzerland, 2002.
- [121] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision*, pages 755–762, 2001.
- [122] A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral encoding of topological structure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–497, Fort Collins, CO, June 1999.

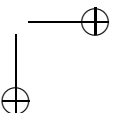
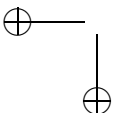


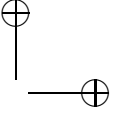
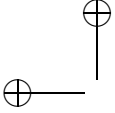
- [123] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.
- [124] A. Simmons, S. R. Arridge, P. S. Tofts, and G. J. Barker. Application of the extremum stack to neurological MRI. *IEEE Transactions on Medical Imaging*, 17(3):371–382, June 1998.
- [125] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1470, Washington, DC, USA, 2003. IEEE Computer Society.
- [126] K. Somchaipeng, K. Erleben, and J. Sporring. A multi-scale singularity bounding volume hierarchy. In *The 13-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG*, pages 179–186, 2005.
- [127] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Multi-scale singularity trees: Soft-linked scale-space hierarchies,. In *Proceedings of the 5th international conference on Scale Space Methods in Computer Vision*, pages 97 – 106, Hofgeismar, Germany, 2005.
- [128] J. Sporring, M. Nielsen, L. M. J. Florack, and P. Johansen, editors. *Gaussian Scale-Space Theory*, volume 8 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [129] R. Thom. *Stabilité Structurelle et Morphogénèse*. Benjamin, Paris, 1972.
- [130] R. Thom. *Structural Stability and Morphogenesis (translated by D. H. Fowler)*. Benjamin-Addison Wesley, New York, 1975.
- [131] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004.
- [132] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [133] J. A. Weickert, S. Ishikawa, and A. Imiya. On the history of Gaussian scale-space axiomatics. In Sporring et al. [128], chapter 4, pages 45–59.
- [134] J. A. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10(3):237–252, May 1999.





- [135] J. J. Wijk and W. A. A. Nuij. A model for smooth viewing and navigation of large 2d information spaces. *IEEE-TVCG*, 10(4):447–458, 2004.
- [136] The Free Encyclopedia Wikipedia. Expectation-maximization algorithm. http://en.wikipedia.org/wiki/Expectation-maximization_algorithm, 2006.
- [137] A. P. Witkin. Scale-space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.
- [138] J. Xiao and M. Shah. Two-frame wide baseline matching. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 603, Washington, DC, USA, 2003. IEEE Computer Society.
- [139] R. A. Young. The Gaussian derivative model for machine vision: Visual cortex simulation. *Journal of the Optical Society of America*, July 1986.
- [140] R. A. Young. The Gaussian derivative model for machine vision: I. retinal mechanisms. *Spatial Vision*, 2(4):273–293, 1987.
- [141] A. L. Yuille and T. A. Poggio. Scaling theorems for zero-crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25, 1986.





Samenvatting

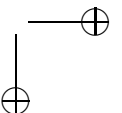
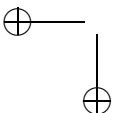
In dit proefschrift ligt de nadruk op het automatisch vergelijken van digitale beelden. De resultaten van deze studie zijn toepasbaar in verschillende beeldverwerkingsgebieden zoals het vergelijken van stereobeelden, objectherkenning, textuurherkenning, videobeeldanalyse, het automatisch combineren van panoramafoto's, objecttypeherkenning en het vinden van objectlocatie en pose. Door de jaren heen is er een groot aantal algoritmes ontwikkeld voor deze taken en sommige van deze algoritmes werken zeer goed in een specifiek beeldverwerkingsgebied. Helaas is de tendens in deze methoden een steeds grotere afhankelijkheid van willekeurige parameters en drempelwaarden. Deze parameters vertegenwoordigen vaak geen fysische grootheden waardoor het moeilijk is zinvolle waarden aan deze parameters toe te kennen. Vaak wordt dit probleem 'opgelost' door uitvoerige training waarbij alle mogelijke parameterwaarden binnen een bepaald gebied worden geprobeerd. De parameters die de beste resultaten opleveren voor de trainingset worden gebruikt voor het uiteindelijke algoritme. Dit resulteert in algoritmen die goed werken voor de beeldmodaliteit waarvoor ze getraind zijn, maar zodra de inputbeelden teveel afwijken van de trainingset werkt het algoritme niet meer.

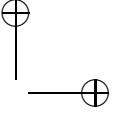
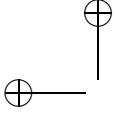
Het doel van het onderzoek beschreven in dit proefschrift is het ontwikkelen van een generieke methode voor beeldvergelijking. Deze methode dient te werken ongeacht welk type inputbeeld wordt gebruikt. Het algoritme moet zo min mogelijk parameters bevatten en de parameters die in het algoritme voorkomen moeten een fysische betekenis hebben.

Om tot dit algoritme te komen zijn methoden, met hun oorsprong in scale-space theorie, fysica en wiskunde, bekeken. Deze methoden vormen de bouwstenen van een nieuw algoritme. Het nieuwe algoritme minimaliseert de hoeveelheid parameters en drempelwaarden. Drempelwaarden die overblijven hebben een fysische betekenis en kunnen daarom gemakkelijk worden ingesteld voor de uit te voeren taak.

Hoofdstuk 1 van dit proefschrift introduceert het probleem en de relevantie ervan. Ook wordt het Europese 'Deep Structure, Singularities and Computer Vision' (DSSCV) consortium besproken waarbinnen dit promotieonderzoek heeft plaats gevonden.

Hoofdstuk 2 behandelt de principes van scale-space theorie en de hieraan gerelateerde 'deep structure of images'. Voor een algemeen beeldanalyse-algoritme is het van essentieel belang dat het algoritme onafhankelijk is van schaal. Objecten en details in beelden zijn slechts over een bepaald schaalbereik betekenisvol.



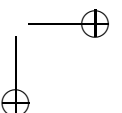
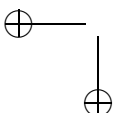


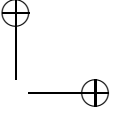
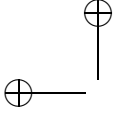
Omdat het niet vooraf bekend is welk schaalgebied bekeken moet worden, zullen alle schalen als even belangrijk moeten worden beschouwd. Essentiële operaties zoals het vervagen en het nemen van afgeleiden van beelden op willekeurige schalen worden besproken en voorbeelden worden gegeven. Een overzicht van catastrofetheorie wordt gegeven voor het scale-space geval. Deze catastrofetheorie bestudeert hoe kritieke punten bewegen als de schaalparameter verandert. De analyse van kritieke punten is essentieel voor elk van de voorgestelde algoritmen in dit proefschrift.

Hoofdstuk 3 beschrijft een aantal methoden voor het genereren van een hiërarchische structuur uit de ‘deep structure’ van een beeld. De methoden genereren een generieke segmentatie die geen voorkennis nodig heeft van het beeld. Om problemen in de bestaande algoritmes aan te pakken wordt een nieuwe methode voorgesteld. In de projectomschrijving van het DSSCV-consortium werd verondersteld dat deze hiërarchische beeldrepresentaties gebruikt zouden kunnen worden voor het vergelijken van beelden. In dit hoofdstuk wordt echter een aantal redenen gegeven waarom het vooralsnog onhaalbaar is om een dergelijke representatie te gebruiken voor vergelijkingstaken als objectherkenning.

Hoofdstuk 4 stelt een nieuwe beeldrepresentatie voor die niet kampt met de problemen die de hiërarchische representaties van hoofdstuk 3 met zich mee brengen. Het algoritme dat wordt voorgesteld vangt de scale-space-structuur van top-punten, voortkomend uit catastrofetheorie, in een graaf. Deze nieuwe representatie maakt het mogelijk om krachtige graafvergelijkingsalgoritmen te gebruiken om beelden met elkaar te vergelijken. De hiërarchische en graafgebaseerde beeldrepresentaties gedragen zich onvoorspelbaar wanneer het beeld onderhevig is aan ruis, belichtingsverandering of niet-Euclidische transformaties zoals affine of projectieve transformaties. Vanwege dit onvoorspelbare gedrag wordt in de volgende hoofdstukken een ander type beeldvergelijkingsalgoritme besproken. Deze laatste hoofdstukken wijden uit over een ‘interest point’ en ‘descriptor’ gebaseerde aanpak voor objectherkenning. Dit houdt in dat kenmerkende punten worden gevonden in beelden en dat in deze punten omgeving-beschrijvende informatie wordt berekend. Deze methode is minder gevoelig voor ruis, belichtingsverandering en niet-Euclidische transformaties en is gemakkelijker te gebruiken en aan te passen.

Hoofdstuk 5 beschrijft karakteristieke punten in de scale-space-representatie van beelden, die geschikt zijn voor beeldvergelijking. Deze kenmerkende punten dienen zo invariant mogelijk te zijn onder veranderingen in het beeld. Wanneer de taak bijvoorbeeld is om een object te lokaliseren in een scènebeeld, dan is het belangrijk dat de configuratie van de karakteristieke punten van het object ook in dezelfde configuratie voorkomt in het scènebeeld. De configuratie van de karakteristieke punten dient hetzelfde te blijven, zelfs als het object in het scènebeeld is gerooteerd, geschaald of onderhevig is aan andere belichting of ruis. In dit

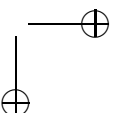
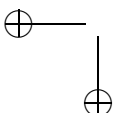


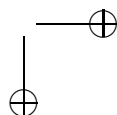
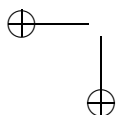
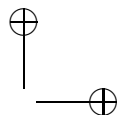
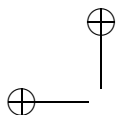


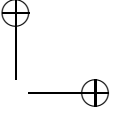
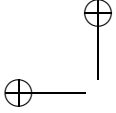
hoofdstuk worden de populaire Harris-, Harris-Laplace en SIFT-karakteristieke punten besproken en vergeleken met de in dit proefschrift voorgestelde karakteristieke punten: de top-punten. De bestaande methoden zijn afhankelijk van een groot aantal willekeurig te kiezen parameters en drempelwaarden. Deze parameters moeten worden gevonden middels een lang trainingstraject. Dit traject moet iedere keer opnieuw doorlopen worden wanneer een ander type beeld wordt beschouwd. De geïntroduceerde top-punten zijn niet afhankelijk van instelbare parameters en leveren zeer concurrerende resultaten op in de uitgevoerde experimenten. De lokalisatie van top-punten hoeft niet erg nauwkeurig te zijn omdat het mogelijk is om de positie van een top-punt te verfijnen door gebruik te maken van lokale differentiaalstructuur van het beeld. Dit maakt het mogelijk om snel top-punten te detecteren zonder de exacte locatie te verliezen. Een ander voordeel van de top-punten is de mogelijkheid om het gedrag van de punten onder ruis analytisch te beschrijven. Hierdoor kunnen onstabiele top-punten worden gevonden en worden verwijderd voordat de rest van het algoritme wordt doorlopen.

Hoofdstuk 6 behandelt zogenaamde ‘descriptors’, dit zijn kenmerken die de omgeving van de karakteristieke punten beschrijven. Voor het beschrijven van een beeld of voor objectherkenning is een set van karakteristieke punten niet voldoende. Specifieke informatie moet worden toegekend aan elk punt. In dit hoofdstuk wordt de populaire SIFT descriptor besproken en vergeleken met de meer mathematisch gefundeerde differentiaalvarianten. Deze invarianten zijn gebaseerd op schaal-Euclidische invariante combinaties van afgeleiden. Dit betekent dat rotatie, schaling en translatie geen invloed hebben op de descriptor. Deze descriptor was veroordeeld tot slecht presterend in een overzichtsartikel, maar in dit hoofdstuk wordt aangetoond dat wanneer er een juiste afstandsmaat wordt gebruikt voor de 6 waardige differentiaalvariante descriptor, deze de 128 waarde tellende SIFT descriptor overtreft in verschillende omstandigheden.

Hoofdstuk 7 presenteert de laatste stap in het op interest points en descriptors gebaseerde beeldvergelijkingsalgoritme. In dit hoofdstuk wordt specifiek ingegaan op de taak om de locatie en pose van een object te vinden uit een scènebeeld waarin dit object voorkomt. Gegeven een beeld van een te vinden object, zal het algoritme proberen om de locatie en de pose van het object te achterhalen uit het scènebeeld. Om dit te doen wordt een puntenset in een 4-dimensionele ruimte geconstrueerd. Deze puntenset wordt gegeven door de bij elkaar passende punten, gevonden door het vergelijken van interest points en descriptors van het objectbeeld en het scènebeeld. De set beschrijft de translatie, rotatie en schaling die het gezochte object heeft ondergaan in het scènebeeld. In de 4-dimensionale ruimte worden één of meerdere clusters gevonden afhankelijk van hoe vaak het object voorkomt in de scène. Uit de gegevens die aanwezig zijn in de clusters kan de locatie en pose van het object in de scène worden geschat. Met een additionele stap kan de affine of projectieve transformatie van het object worden achterhaald.







Dankwoord

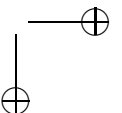
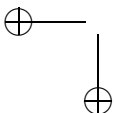
Het schrijven van een proefschrift is voornamelijk een solitaire bezigheid, maar het onderzoek en de resultaten die zijn beschreven in dit proefschrift zouden onmogelijk zijn geweest zonder interactie met vele anderen.

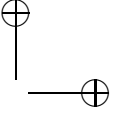
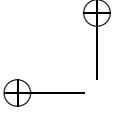
Op de eerste plaats wil ik natuurlijk mijn promotor Bart ter Haar Romeny bedanken. Zijn oneindige enthousiasme en brede interesse zijn de belangrijkste redenen geweest waarom ik mijn mastersproject begonnen ben in de Biomedische Beeldanalyse groep en hier ook gebleven ben om dit onderzoek te vervolgen met een promotie.

Mijn copromotor Luc Florack is onmisbaar geweest bij de wetenschappelijke kant van het onderzoek. Alle inhoudelijke discussies en bijeenkomsten waren uitermate nuttig en hebben vaak geleid tot nieuwe ideeën en ontwikkelingen. Vaak ook kwamen tekortkomingen in bestaande algoritmen aan het licht die vroegen om een nette wiskundige aanpak. Daarnaast zijn de uitgebreide correcties en suggesties erg welkom geweest bij het schrijven van artikelen en dit proefschrift.

Ook wil ik Frans Kanters bedanken. De vele internationale reizen die we samen binnen het Europese project hebben gemaakt hebben zeker bijgedragen aan een onvergetelijke tijd. Ook het door Frans ontwikkelde programma ScaleSpaceViz is verantwoordelijk voor een groot deel van de mooie visualisaties in dit proefschrift waaronder de afbeelding op de omslag. Een andere collega die ik ook zeer dankbaar ben is Evgeniya Balmashnova. Met haar wiskundig inzicht heeft ze me meerdere malen geholpen en haar afleidingen voor de verschillende stabiliteits- en afstandsmaten hebben bijzonder veel bijgedragen aan dit proefschrift. Ook Remco Duits verdient een vermelding, voor zijn briljante wiskundige inzichten en het eindeloze geduld om ze uit te leggen. Om mijn dank aan mijn overige kamergenoten Erik Franken en Bart Janssen uit te spreken gebruik ik graag hun eigen woord(en): “Jooooow”! Natuurlijk wil ik alle andere AiO’s, stafleden en studenten in de biomedische beeldverwerkingsgroep ook bedanken voor hun bijdrage aan het wetenschappelijke klimaat en de goede sfeer, zowel op het werk als daarbuiten.

I would also like to thank Ali Shokoufandeh for his great collaboration, hospitality and friendship. Our joint work has lead to Chapter 4 of this thesis. I am also grateful that he accepted the invitation to be one of my committee members. A special thanks also goes out to Ali’s students: Fatih Demirci for his friendship and co-work on the graph based image matching algorithms, Jeff Abrahamson for his hospitality, and to Trip Denton and John Novatnack for their help collaboration.





Let me also thank Sven Dickinson for helping us to form ideas for the graph based algorithms, for fruitful discussions and for inviting me and Frans to come to Toronto. Sven has always been willing to lend a helping hand.

Mads Nielssen also deserves a special place in this acknowledgement, for not only has he initiated the European DSSCV project that made my PhD possible, but he was also so kind to accept the invitation to be on my PhD committee.

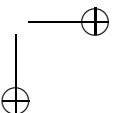
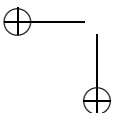
Ook Klaas Nicolaij ben ik erg dankbaar. Ondanks het feit dat mijn onderzoek buiten zijn directe onderzoeksgebied valt heeft hij de tijd genomen om mijn proefschrift door te lezen en heeft hij plaats willen nemen in mijn commissie.

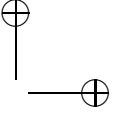
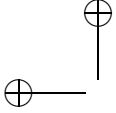
In the last year of my studies I spent three months working at the ETH in Zürich in Joachim Buhmann's Machine Learning lab. This work resulted in Chapter 7 of this thesis. I am very grateful for the friendly support I received in these months from Prof. Buhmann and his students.

Buiten de werksfeer zijn er ook mensen die ik speciaal wil bedanken. Op de eerste plaats mijn ouders, die mij met onvoorwaardelijke liefde en wijsheid hebben grootgebracht en mij hebben gesteund in al mijn keuzes. Mijn zus Eefke voor de gezelligheid en het niet al te vaak vragen "Wat doe je nu eigenlijk precies?". Adrienne, mijn partner ben ik ook heel erg dankbaar. Naast haar liefde heeft ze mij de mogelijkheid gegeven om mijn academische carrière voort te kunnen zetten in Nederland door te verhuizen vanuit Zürich.

Zonder twijfel hebben mijn vrienden ook bijgedragen aan een leuke tijd. Speciaal wil ik Luuk Barten noemen die een groot deel van mijn promotietijd mijn huisgenoot was en de lekkerste gerechten klaar maakte. Ook Bart van den Oever wil ik bedanken voor de leuke tijden en de vrijdag-frietdagen. Natuurlijk ook veel dank aan mijn andere vrienden en alle anderen die ik vergeten ben hier te noemen.

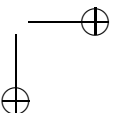
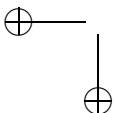
Mijn promotieonderzoek vond plaats binnen het DSSCV project ondersteund door het IST Programma van de Europese Unie (IST-2001-35443).

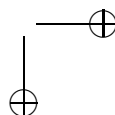
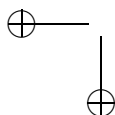
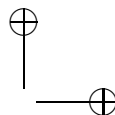
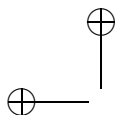


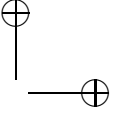
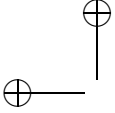


Curriculum Vitae

Bram Platel was born on the 6th of Februari in 's-Hertogenbosch, The Netherlands. He graduated from the VWO at the Jacob Roelandslyceum in Boxtel in 1996. In the same year he started studying Mechanical Engineering at the Technische Universiteit Eindhoven. After completing the first year foundation course he switched to the brand new Biomedical Engineering study from which he graduated (and received the M.Sc. degree cum laude) in 2002. His graduation project involved research in multiscale hierarchical segmentation of images, which was carried out within the Biomedical Image Analysis group at the Technische Universiteit Eindhoven. This work laid the foundation of his PhD-studies which were carried out within the Deep Structure, Singularities and Computer Vision project funded by the European Union.

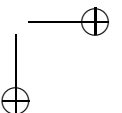
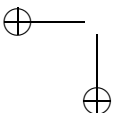






Publications

- **Top-Points as Interest Points Image Matching,**
B. Platel, E. Balmachnova, L.M.J. Florack and B.M. ter Haar Romeny
“9th European Conference on Computer Vision”, Graz, Austria, May 7-13, 2006.
- **Using Top-Points as Interest Points For Image Matching,**
B. Platel, E. Balmachnova, L.M.J. Florack and B.M. ter Haar Romeny
“1st International Workshop on Deep Structure, Singularities and Computer Vision”, The Netherlands, June 2005.
- **On Image Reconstruction from Multiscale Top-Points,**
F.M.W. Kanters, M. Lillholm, R. Duits, B.J. Janssen, B. Platel, L.M.J. Florack and B.M. ter Haar Romeny
“Scale Space Methods in Computer Vision. Proceedings of the 5th international conference on Scale Space 2005”, Germany, April 2005, pp 431–442.
- **Stability of Top-Points in Scale Space,**
E. Balmachnova, L.M.J. Florack, B. Platel and F.M.W. Kanters
“Scale Space Methods in Computer Vision. Proceedings of the 5th international conference on Scale Space 2005”, Germany, April 2005, pp 62–72.
- **Discrete Representation of Top Points via Scale Space Tessellation,**
B. Platel, M. Fatih Demirci, A. Shokoufandeh, L.M.J. Florack, F.M.W. Kanters, B.M. ter Haar Romeny, and S.J. Dickinson
“Scale Space Methods in Computer Vision. Proceedings of the 5th international conference on Scale Space 2005”, Germany, April 2005, pp 73–84.
- **Using Multiscale Top Points in Image Matching**
B. Platel, F.M.W. Kanters, L.M.J. Florack and E.G. Balmachnova
“Proceedings of the 11th International Conference on Image Processing”, Singapore, October 2004.
- **Alpha Scale Space Kernels in Practice**
F.M.W. Kanters, L.M.J. Florack, R. Duits, B. Platel
“7th International Conference on Pattern Recognition and Image Analysis”, St. Petersburg, Russian Federation, 2004, pp 260–263.
- **Using Multiscale Top Points in Image Matching**
B. Platel, L.M.J. Florack, F.M.W. Kanters, E. Balmachnova
“Advanced School for Computing and Imaging - ASCI 2004 10th Annual Conference”, Port Zélande, The Netherlands, June, 2004, Proceedings.



- **α Scale Spaces on a Bounded Domain**
R. Duits, M. Felsberg, L.M.J. Florack and B. Platel
“Scale Space Methods in Computer Vision. Proceedings of the 4th int. conference Scale Space 2003”, Isle of Skye, UK, June 2003.
- **Content Based Image Retrieval using Multiscale Top Points**
F.M.W. Kanthers, B. Platel, L.M.J. Florack and B.M. ter Haar Romeny
“Scale Space Methods in Computer Vision. Proceedings of the 4th int. conference Scale Space 2003”, Isle of Skye, UK, June 2003, pp 33–43.
- **Image Reconstruction from Multiscale Critical Points**
F.M.W. Kanthers, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny
“Scale Space Methods in Computer Vision. Proceedings of the 4th int. conference Scale Space 2003”, Isle of Skye, UK, June 2003, pp 464–478.
- **Multiscale Hierarchical Segmentation**
B. Platel, L.M.J. Florack, F.M.W. Kanthers and B.M. ter Haar Romeny
“Advanced School for Computing and Imaging - ASCI 2003
9th Annual Conference, Heijen”, The Netherlands, June 4-6, 2003, Proceedings, p 161.
- **Validation of Tissue Modelization and Classification Techniques in T1-Weighted MR Brain Images**
M. Bach Cuadra, B. Platel, E. Solanas, T. Butz, and J.-Ph. Thiran
“Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, 5th International Conference”, Tokyo, Japan, September 25-28, 2002, Proceedings, Part I, p 290.