



ROYAL INSTITUTE OF TECHNOLOGY
STOCKHOLM, SWEDEN

STOCKHOLM
UNIVERSITY



DEPARTMENT OF NUMERICAL ANALYSIS AND COMPUTING SCIENCE
TRITA-NA-P9108 • ISSN 1101-2250 • ISRN KTH/NAP-91/8-SE • CVAP84

Discrete Scale-Space Theory and the Scale-Space Primal Sketch

Tony Lindeberg

Dissertation, May 1991

Computational Vision and Active Perception Laboratory (CVAP)



NADA (Numerisk Analys och Datalogi)
KTH
100 44 Stockholm

Department of Numerical Analysis
and Computing Science
Royal Institute of Technology
S-100 44 Stockholm, Sweden

Discrete Scale-Space Theory and the Scale-Space Primal Sketch

Tony Lindeberg

DISSERTATION, MAY 91

TRITA-NA-P9108 • ISSN 1101-2250 • ISRN KTH/NAP--91/8--SE • CVAP84

Akademisk avhandling för
teknisk doktorsexamen vid
Kungl. Tekniska Högskolan

Maj 1991

©Tony Lindeberg 1991

ISRN KTH/NA/P - 91/8 - SE
ISSN 1101-2250 : TRITA-NA-P9108
Norstedts Tryckeri, Stockholm

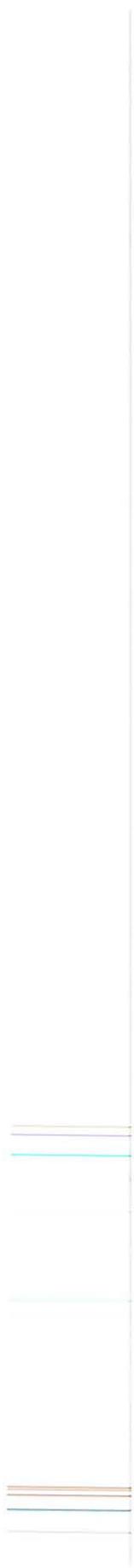
Abstract

This thesis, within the subfield of computer science known as computer vision, deals with the use of scale-space analysis in early low-level processing of visual information. The main contributions comprise the following five subjects:

- *The formulation of a scale-space theory for discrete signals. Previously, the scale-space concept has been expressed for continuous signals only. We propose that the canonical way to construct a scale-space for discrete signals is by convolution with a kernel called the discrete analogue of the Gaussian kernel, or equivalently by solving a semi-discretized version of the diffusion equation. Both the one-dimensional and two-dimensional cases are covered. An extensive analysis of discrete smoothing kernels is carried out for one-dimensional signals and the discrete scale-space properties of the most common discretizations to the continuous theory are analysed.*
- *A representation, called the scale-space primal sketch, which gives a formal description of the hierarchical relations between structures at different levels of scale. It is aimed at making information in the scale-space representation explicit. We give a theory for its construction and an algorithm for computing it.*
- *A theory for extracting significant image structures and determining the scales of these structures from this representation in a solely bottom-up data-driven way.*
- *Examples demonstrating how such qualitative information extracted from the scale-space primal sketch can be used for guiding and simplifying other early visual processes. Applications are given to edge detection, histogram analysis and classification based on local features. Among other possible applications one can mention perceptual grouping, texture analysis, stereo matching, model matching and motion.*
- *A detailed theoretical analysis of the evolution properties of critical points and blobs in scale-space, comprising drift velocity estimates under scale-space smoothing, a classification of the possible types of generic events at bifurcation situations and estimates of how the number of local extrema in a signal can be expected to decrease as function of the scale parameter. For two-dimensional signals the generic bifurcation events are annihilations and creations of extremum-saddle point pairs. Interpreted in terms of blobs, these transitions correspond to annihilations, merges, splits and creations.*

Experiments on different types of real imagery demonstrate that the proposed theory gives perceptually intuitive results.

Keywords: *computer vision, low-level processing, scale-space, diffusion, Gaussian filtering, discrete smoothing, primal sketch, segmentation, descriptive elements, scale detection, image structure, focus-of-attention, tuning low-level processing, blob detection, edge detection, edge focusing, histogram analysis, junction classification, perceptual grouping, texture analysis, critical points, classification of blob events, bifurcations, drift velocity, density of local extrema, multi-scale representation, digital signal processing*



Sammanfattning

Denna avhandling, inom det delområde av datalogin som går under namnet datorseende, behandlar användningen av skalrumsanalys i de första stegen av tidig lågnivåbearbetning av visuella bilddata. De huvudsakliga bidragen omfattar:

- *Formulering av en skalrumsteori för diskreta signaler. Tidigare har skalrumsbegreppet uttryckts enbart för kontinuerliga data. Vi föreslår att det bara finns ett (kanoniskt) sätt att definiera ett skalrum för diskreta signaler, nämligen genom att falla med den diskreta motsvarigheten till Gausskärnan eller ekvivalent sett genom att lösa en semi-diskretiserad version av diffusionsekvationen. Både det endimensionella och det tvådimensionella fallet behandlas. En omfattande analys av diskreta faltningsskärnor med utjämningssegenskaper genomförs för endimensionella signaler och de diskreta skalrumsegenskaperna hos de vanligaste diskretiseringarna av den kontinuerliga teorin analyseras.*
- *En representation vid namn skalrumsskissen, vilken ger en formell beskrivning av de hierarkiska relationer som finns mellan strukturer på olika skalor i en bild. Den är avsedd att lyfta fram egenskaper i skalrumsrepresentationen så att de blir explicita. Vi ger en teori för dess konstruktion och en algoritm för att bygga upp den beräkningsmässigt.*
- *En teori för att extrahera signifikanta bildstrukturer och bestämma deras skalnivåer från denna representation på ett helt datadrivet sätt.*
- *Exempel som visar hur sådan kvalitativ information extraherad från skalrumsskissen kan användas för att vägleda och förenkla andra tidiga visuella processer. Tillämpningar ges mot kantdetektion, histogramanalys och klassificering baserad på lokala egenskaper. Bland ytterligare möjliga tillämpningar kan nämnas perceptuell gruppering, texturanalys, stereomatchning, modellmatchning samt rörelse.*
- *En detaljerad analys av hur lokala extrempunkter och blobbar kan förväntas bete sig i skalrummet, innefattande uppskattningar av deras drifhastighet, en klassificering av vilka typer av händelser som är möjliga vid bifurkationspunkter samt uppskattningar av hur antalet lokala extrempunkter i en signal kan förväntas minska som funktion av skalparametern. För tvådimensionella signaler utgörs de generiska bifurkationshändelserna av par bestående av en lokal extrempunkt och en sadelpunkt som försvinner eller skapas då skalparametern ökar. Uttryckt i termer av extrempunktsregioner svarar de möjliga övergångarna mot annihilationer, sammanslagningar, splittringar och skapanden.*

Experiment på olika typer av verkliga bilddata visar att den föreslagna teorin ger perceptuellt sett intuitiva och rimliga resultat.

Nyckelord: datorseende, lågnivåbearbetning, skalrum, diffusion, gaussfiltering, diskret utjämning, primärskiss, segmentering, deskriptiva element, skaldetektion, bildstruktur, fokusering av uppmärksamheten, styrning av lågnivåbearbetning, blobdetektion, kantdetektion, kantfokusering, histogramanalys, förgreningspunktsklassificering, perceptuell gruppering, texturanalys, kritiska punkter, klassificering av blobhändelser, bifurkationer, drifhastighet, densitet av lokal extrempunkt, multiskalrepresentation, digital signalbehandling

Acknowledgments

First, I would very much like to thank Prof. Jan-Olof Eklundh for introducing me into this inspiring field, for excellent supervision based on his broad experience, for providing an stimulating and cheerful research environment to work in, his encouraging support and his way to always find time for discussions.

I am very grateful to Kjell Brunnström for the enjoyable collaboration with the work on junction classification and many interesting discussions, Harald Winroth and Jonas Gårding for sharing their time and knowledge on any matter from mathematics and philosophy to programming styles and efficient coding, Dr. Fredrik Bergholm for many discussions about scale-space as well as for valuable comments on several manuscript drafts, Dr. Stefan Carlsson for useful discussions about smoothing transformations serving as a large source of inspiration to the discrete scale-space theory, Ambjörn Naeve and Dr. Lars Svensson for always sharing their intuition and insights into many aspects of mathematics, Dr. Richard Weiss for enjoyable discussions and cooperation, Matti Rendahl for maintaining a highly reliable systems environment in our laboratory “Bild- och Grafiklaboratoriet” and for always providing wizardly help when needed as well as Anna Thorsson for arranging almost anything you could ask for.

I would also like to thank my other colleagues at the Computational Vision and Active Perception Laboratory, CVAP, for their friendship and for the help they have given me in so many ways: Magnus Andersson, Ann Bengtson, Dr. Demetre Betsis, Kiyo Chinzei, Dr. Meng-Xiang Li, Dr. Niklas Nordström, Göran Olofsson, Kourosh Pahlavan, Eva Setterlind and Wei Zhang.

I would like to express my gratitude to the staff members, teachers and researchers at the Royal Institute of Technology with whom I have had many interesting discussions and got valuable advice from during the development of this work. In particular I would like to mention Prof. Germund Dahlquist, Prof. Heinz-Otto Kreiss, Dr. Bengt Lindberg, and Dr. Jesper Ooppelstrup at the Department of Numerical Analysis and Computing Science and Dr. Michael Benedicts, Prof. Anders Björner, Prof. Lars Holst, Dr. Torbjörn Kolsrud and Dr. Johan Philip at the Department of Mathematics. I am also very grateful to Thure Valdsoo at the Department of Aeronautics for providing the interesting aerosol problem to work with as well as to Dr. Anders Liljeborg at the Department of Physics IV for useful help with the laboratory equipment at the “Bild- och Grafiklaboratoriet”.

In preparing this manuscript I would like to thank Birgitta Krasse for redrawing many of the figures, Roswitha Graham for designing the cover and arranging the printing as well as Jan-Michael Rynning for valuable help enabling photo-setting of images.

This work was made possible by a graduate position, “exzellenstjänst”, provided by the Royal Institute of Technology together with financial support from the National Swedish Board for Technical Development, STU. This support is gratefully acknowledged.

All implementations have been made within the Candela and CanApp programming environment for image analysis developed at the Computational Vision and Active Perception Laboratory, CVAP. This manuscript has been typeset with L^AT_EX.

The thesis is based on the following papers:

Journal Articles:

- [1] Lindeberg T.P. (1990) "Scale-Space for Discrete Signals", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-12, No. 3, pp234-254.
- [2] Lindeberg T.P., Eklundh J.O. (1991) "On the Computation of a Scale-Space Primal Sketch", *Journal of Visual Communication and Image Representation*, Vol. 2, No. 1, pp55-78.
- [3] Lindeberg T.P., Eklundh J.O. (1990) "The Scale-Space Primal Sketch: Construction and Experiments", To appear in *Image and Vision Computing*.
- [4] Brunnström K., Eklundh J.O., Lindeberg T.P. (1990) "On Scale and Resolution in Active Analysis of Local Image Structure", *Image and Vision Computing*, Vol. 8, No. 4, pp289-296.
- [5] Lindeberg T.P., Eklundh J.O. (1990) "Analysis of Aerosol Images Using the Scale-Space Primal Sketch", To appear in *Machine Vision and Applications*.

International Conferences:

- [6] Lindeberg T.P., Eklundh J.O. (1990) "Scale Detection and Region Extraction from a Scale-Space Primal Sketch", *Proceedings of the 3:rd International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp416-426.
- [7] Brunnström K., Eklundh J.O., Lindeberg T.P. (1990) "On Scale and Resolution in the Analysis of Local Image Structure", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-27, pp3-12.

Regional Conferences:

- [8] Lindeberg T.P. (1989) "Scale-Space for Discrete Images", *Proceedings of the 6:th Scandinavian Conference on Image Analysis*, Oulu, Finland, June 19-22, pp1098-1107.
- [9] Lindeberg T.P., Eklundh J.O. (1990) "Construction of a Scale-Space Primal Sketch", *Proceedings of the BMVC90 — British Machine Vision Conference*, Oxford, Great Britain, September 24-27, pp97-102.
- [10] Lindeberg T.P. (1991) "On the Behaviour in Scale-Space of Local Extrema and Blobs", To appear in *The 7:th Scandinavian Conference on Image Analysis*, Aalborg, Denmark, August 13-16.

Technical Reports: (With material not covered by the previously listed papers).

- [11] Lindeberg T.P., Eklundh J.O. (1990) "Guiding Early Visual Processing with a Scale-Space Primal Sketch", *Technical Report TRITA-NA-P9013, CVAP-72*, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, *Submitted*.
- [12] Lindeberg T.P. (1991) "On the Behaviour in Scale-Space of Local Extrema and Blobs", *Submitted*.
- [13] Lindeberg T.P. (1988) "On the Construction of a Scale-Space for Discrete Images", *Technical Report TRITA-NA-P8808, CVAP-55*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.
- [14] Lindeberg T.P. (1991) "Two-Dimensional Discrete Scale-Space Formulation", *Technical Report*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

Contents

1	Introduction and Overview	1
1.1	Goal	2
1.2	The Nature of the Problem	2
1.2.1	Ill-posedness	3
1.2.2	Grouping	3
1.2.3	Operator Size	4
1.2.4	Scale	5
1.3	Scale-Space Representation	6
1.3.1	Non-Creation of New Structure	7
1.3.2	Other Multi-Scale Approaches	8
1.3.3	Multi-Scale v.s. Multi-Resolution	8
1.3.4	Theoretical Scale-Space Properties	9
1.4	Philosophies and Ideas behind the Approach	9
1.4.1	Making Information Explicit	9
1.4.2	Scale and Segmentation	10
1.4.3	Detection of Image Structure	11
1.4.4	Computational Issues	12
1.4.5	Consistency over Scales	12
1.5	Relations to Traditional Numerical Analysis	12
1.5.1	Modelling, Simulation and Inverse Problem	12
1.5.2	Scale and Resolution	13
1.5.3	Interpreting the Results	13
1.5.4	Approximation and Regularization	14
1.5.5	Principles behind the Work	14
1.6	Organization of the Presentation	14
1.6.1	Part I: Scale-Space Theory for Discrete Signals	15
1.6.2	Part II: Theory of the Scale-Space Primal Sketch	17
1.6.3	Part III: Applications of the Scale-Space Primal Sketch	18
I	Discrete Scale-Space Theory	21
2	Scale-Space for 1-D Discrete Signals	23
2.1	Introduction	23
2.2	Scale-Space Axioms	24
2.3	Properties of Scale-Space Kernels	26
2.3.1	Positivity and Unimodality in the Spatial Domain	26

2.3.2	Generalized Binomial Kernels	27
2.3.3	Positivity and Unimodality in the Fourier Domain	29
2.3.4	Kernels with Three Non-Zero Elements	34
2.4	Kernel Classification	35
2.4.1	Background	35
2.4.2	Classification of Discrete Scale-Space Kernels	38
2.5	Axiomatic Scale-Space Construction	40
2.5.1	Discrete Scale-Space with Discrete Scale Parameter	40
2.5.2	Discrete Scale-Space with Continuous Scale Parameter	41
2.5.3	Equivalent Formulation for Continuous Signals	44
2.6	Discrete Scale-Space Properties of Some Numerical Approximations of the Continuous Scale-Space Theory	47
2.6.1	Sampled Gaussian Kernel	47
2.6.2	Discretized Diffusion Equation	49
2.6.3	Integrated Gaussian Kernel	52
2.7	Summary and Discussion	53
2.7.1	Ideal Low-Pass Filters and Block Average Filters	53
2.7.2	Positivity and Unimodality is Necessary but not Sufficient	54
2.7.3	Recursive Filters	54
2.8	Conclusion: Scale-Space for 1-D Discrete Signals	54
3	Scale-Space for 2-D Discrete Signals	55
3.1	From One to Two Dimensions	55
3.2	Selecting Two-Dimensional Scale-Space Axioms	56
3.2.1	Basic Definitions	58
3.3	Axiomatic 2D Discrete Scale-Space Construction	58
3.3.1	Definitions	59
3.3.2	Necessity	61
3.3.3	Sufficiency	62
3.3.4	Equivalent One-Dimensional Formulation	63
3.4	Parameter Determination	64
3.4.1	Derivation of the Fourier Transform	64
3.4.2	Unimodality in the Fourier Domain	65
3.4.3	Separability	66
3.4.4	Discrete Iteration	67
3.4.5	Spatial Isotropy	67
3.4.6	Remaining Degree of Freedom	69
3.5	2D Summary and Discussion	69
3.6	Possible Extensions	69
3.6.1	Anisotropic Smoothing	69
3.6.2	Higher Dimensions	70
3.6.3	Finite Data	71
3.6.4	Other Types of Grids	72
3.6.5	Further Work	72

4	Implementational Implications and Conclusions	73
4.1	Discrete Definitions of “Derivatives”	73
4.1.1	The Laplacian of the Gaussian	73
4.1.2	The Gradient of the Gaussian	75
4.1.3	Normalization	76
4.1.4	Summary	77
4.2	Kernel Graphs	77
4.3	Implementing Scale-Space Smoothing	77
4.3.1	Truncation and Filter Coefficient Generation	78
4.3.2	Application to the N -jet	79
4.4	Summary and Discussion	79
4.5	Conclusions: Scale-Space for Discrete Signals	79
4.6	Philosophy	80
II	The Scale-Space Primal Sketch: Theory	89
5	Definition of the Representation	91
5.1	Grey-Level Blobs	91
5.1.1	Definition of Grey-Level Blob	92
5.1.2	Mathematical Definition	92
5.1.3	Properties	94
5.1.4	Relations Between Bright and Dark Blobs	94
5.2	Motivations for a Multi-Scale Hierarchy	96
5.2.1	Blob Detection and Scale-Space Smoothing	96
5.3	Scale-Space Blobs—Linking Grey-Level Blobs across Scales	97
5.4	Grey-Level Blob Extraction: Experimental Results	98
5.4.1	Remarks	101
5.5	Measuring Significant Image Structure	102
5.5.1	Measuring Scale-Space Lifetime	102
5.5.2	Transformation of the Scale Parameter: Effective Scale	103
5.5.3	Transformation of the Blob Volumes	106
5.5.4	Resulting Representation — The Scale-Space Primal Sketch	109
6	Evolution Properties in Scale-Space: Drift Velocities and Bifurcation Events	113
6.1	Trajectories of Critical Points in Scale-Space	115
6.1.1	Interpretation: Drift Velocity Estimates	118
6.1.2	Interpretation: Extremal Paths	120
6.1.3	Formal Definition of Scale-Space Blob	121
6.2	Behaviour Near Singularities: Classification	124
6.2.1	Background	124
6.2.2	Application to Scale-Space Representation: One-Parameter Families	127
6.2.3	Interpretations	128
6.2.4	Classification of Blob Events at Bifurcations in Scale-Space	130
6.3	Behaviour Near Singularities: Examples in 1D	133
6.3.1	Third Order Taylor Expansion of the Scale-Space Embedding	133
6.3.2	Evolution Properties of Local Extrema, Grey-Level Blobs and Scale-Space Blobs in 1D Continuous Scale-Space	136

6.4	Behaviour Near Singularities: Examples in Two Dimensions	138
6.4.1	Fold	139
6.4.2	Cusp	142
6.4.3	Elliptic Umbilic	147
6.4.4	Summary	149
6.5	Density of Local Extrema as Function of Scale	149
6.5.1	Continuous Analysis	150
6.5.2	Discrete Analysis	152
6.5.3	Comparisons Between the Continuous and Discrete Results	155
6.5.4	Extension to Two Dimensions	156
6.6	Summary	156
7	Computing the Representation	159
7.1	Grey-Level Blob Detection	159
7.1.1	The One-Dimensional Case	159
7.1.2	Grey-Level Blob Invariants	160
7.1.3	Generic v.s. Non-Generic Signals	161
7.1.4	Sequential Implementation	162
7.2	Linking Grey-Level Blobs into Scale-Space Blobs	163
7.2.1	Blob-Blob Matching	164
7.2.2	Extremum-Blob Matching	165
7.2.3	Registering Bifurcations in Scale-Space	166
7.2.4	Delimiting the Refinement Depth	170
7.2.5	Scale Levels and Computation of the Refinement Scale	170
7.2.6	Basic Blob Linking Algorithm	170
7.2.7	Continuous Scale Parameter v.s. Fixed Scale Sampling	172
7.2.8	Blob Linking v.s. Extremum Linking	172
7.3	Computing the Scale-Space Blob Volumes	172
7.4	Data Structure	173
7.5	Possible Improvements of the Algorithm	173
7.5.1	Local Refinements	173
7.5.2	Drift Velocity Estimates	173
7.5.3	Approximate Description	173
7.5.4	Subsampling at Coarser Scales	174
7.5.5	Other Normalization Methods	174
III	The Scale-Space Primal Sketch: Applications	175
8	Detecting Image Structures	177
8.1	Detecting Significant Image Structures and Their Scales	177
8.2	Motivation for the Assumptions	178
8.2.1	Transformational Invariance: Structure	178
8.2.2	Stability in Scale-Space: Perceptual Salience	178
8.2.3	Reduction of the Representation: Abstraction	179
8.3	Basic Extraction Method for Image Structures	179
8.4	Experimental Results	179
8.5	Further Treatment of the Generated Blob Hypotheses	184

8.6	About the Selection of Appropriate Scale	184
8.6.1	Relations between Appropriate Scale and Object Size	185
8.6.2	Partial Ordering	185
8.6.3	Several Instances of a Region	185
8.6.4	Number of Layers: Complexity	186
8.7	Additional Experiments	186
9	Guiding Early Visual Processes	199
9.1	Application to Edge Detection	199
9.1.1	Edge Detection at a Proper Scale	200
9.1.2	Matching Blobs to Edges	201
9.1.3	Edge Focusing	206
9.1.4	Experimental Results	207
9.1.5	Alternative: Individual Treatment of the Blob Hypotheses	207
9.1.6	Conclusions	213
9.2	Application to Histogram Analysis	214
9.2.1	Experimental Results: Histogram-Based Colour Segmentation	214
9.2.2	Sensitivity to Quantization Effects	216
9.3	Application to Junction Classification	216
9.3.1	Background: Classifying Junctions by Active Focusing	217
9.3.2	Setting Window Size from Blob Information	217
9.3.3	Computing Window Size from Blob Size	222
9.3.4	Experimental Results	222
9.3.5	Detection of Candidate Junction Points Initiated by the Scale-Space Primal Sketch	223
9.3.6	Summary and Discussion	226
9.4	Example: Analysis of Aerosol Images	226
9.4.1	Experimental Results	227
9.4.2	Conclusions	229
9.5	Other Possible Applications	231
9.5.1	Texture Analysis	231
9.5.2	Perceptual Grouping	233
9.5.3	Matching	233
10	Summary and Discussion	235
10.1	The Scale-Space Primal Sketch	235
10.1.1	Qualitative Properties	235
10.1.2	Extraction of Structure — Transformational Invariance	236
10.2	Scale-Space Experiences	236
10.2.1	Suppression of Local Extrema due to Noise	236
10.2.2	Stable Scale is a Local Property	236
10.2.3	Stable Scale is a Multi-Valued Function	237
10.2.4	Decreasing Amplitude of Feature Points	237
10.3	Relations to Previous Work	238
10.4	Conclusions	238
IV	Appendix	241

A	Technical Details and More Examples	243
A.1	From Chapter 2	243
A.1.1	Unimodality of the Fourier Transform in the Non-Circulant Case . . .	243
A.1.2	Positivity and Unimodality are Necessary but not Sufficient	244
A.2	From Chapter 3	247
A.2.1	Separated Convolution with $T(n; t)$ Satisfies the Diffusion Equation .	247
A.2.2	Equivalent 1-D Formulation of the 2-D Discrete Scale-Space	247
A.2.3	Derivation of the MacLaurin Expansion of the Fourier Transform . . .	248
A.3	From Chapter 4	249
A.3.1	The l_1 Norms of the Difference between Various Discrete Implemen- tations of the Scale-Space Theory	249
A.4	From Chapter 5	251
A.5	From Chapter 6	251
A.5.1	Polynomials Satisfying the Diffusion Equation.	251
A.5.2	Investigation about the Roots to $4x^3 + 12tx + v = 0$	251
A.5.3	Detailed Investigation of the Singularity Set for the Elliptic Umbilic .	254
A.5.4	Derivation of $p_d(t)$ in the Discrete Case	259
A.5.5	Asymptotic Expression for $p_d(t)$ at Fine Scales	261
A.5.6	Asymptotic Expression for $p_d(t)$ at Coarse Scales	263
A.6	From Chapter 7	264
A.6.1	Algorithmic Performance	264
A.6.2	Bifurcation Statistics	264
A.6.3	Data Structure	265
A.7	Test Images	266

Chapter 1

Introduction and Overview

Computer vision deals with the problem of deriving meaningful interpretations or descriptions from visual data. What should be meant by meaningful is, of course, strongly dependent on the goal of the analysis, i.e., the underlying purpose why we want to make use of visual information. One reason can be that of machine vision — the desire to provide machines and robots with visual abilities. Other common applications concern image processing, where one can mention image enhancement, visualization and analysis of medical data as well as remote sensing, data compression and the design of visual aids etc. A more theoretical reason why computer vision is studied is the tremendously inspiring challenge of trying to understand the workings of biological visual systems, which accomplish their tasks in such a reliable way essential for the survival of most living creatures.

Some of the most basic questions that still remain to be answered concern what type of information in images is relevant for accomplishing different tasks, how this information is extracted from the sensory data and how such features can be related to properties of environment. Then, what is vision? To the question “What does it mean to see?” Marr [Mar82] answers:

... vision is the *process* of discovering from images what is present in the world and where it is

emphasizing that vision is an information-processing task. He also stresses that the issue of *internal representation* of information is of outmost importance. Only by representation can information be captured and made available to decision processes. The purpose of a representation is to make certain aspects of the information content *explicit*, that is, immediately accessible without any need for additional processing.

There have been different opinions in the computer vision community about how a visual system should be constructed. A long debate concerned the choice between bottom-up and top-down based reasoning. It has been argued by many authors that a visual system should be constructed in a modular way with different levels of processing. At the simplest level of abstraction three layers can be distinguished, denoted low-level, intermediate level and high-level. Although also other types of design strategies have been proposed such as active vision, see Bajscy [Baj85], and “labyrinthic design”, see Aloimonos [Alo90], implying that it is probably not as easy to clearly separate out different processing levels as would be needed for a dogmatic interpretation of the three-layer description, the need for some kind of early low-level processing and representation for providing a sparse but rich set of primitives for other processing modules still remains highly motivated.

This thesis deals with the use of a certain type of approach, *scale-space representation*, for analysing data at the lowest levels in such a chain of information processing. The aim is to operate directly on the raw pixels values without any type of pre-processing. The suggested methodology can be said to be intended as a first confrontation between the reasoning process and the raw image data. We will not make any specific assumptions about how higher-level processes are to operate on the output. Therefore, we believe that the approach should be applicable to a variety of reasoning strategies.

Computer vision is a cross-disciplinary field with research methodologies from several scientific disciplines such as computer science, mathematics, neurophysiology, physics and psychology. The approach taken here will be computational¹. We will develop a theory and a framework for how certain aspects of image information can be represented and analysed at the very earliest processing stages of a machine visual system.

1.1 Goal

The goal we are aiming at is a methodology, where significant structures can be extracted from an image in a solely bottom-up way, without any a priori information. We will suggest a *ranking of events* in order of significance based on *volumes* of four-dimensional objects in a *scale-space* representation where the *scale* dimension is treated as *equally important* as the spatial and grey-level coordinates. The associated *extraction* method is based on a *systematic parameter variation principle* where *locally stable states* are detected and abstractions are determined from those. We will exemplify how *qualitative scale and region information* extracted in this way can be used for *guiding the focus-of-attention* and *tuning other early visual processes* as to *simplify* their tasks. The general principle is to *adapt the low-level processing to the local structure* of an image. The leading idea of the thesis is to construct a framework in which these operations can be formalized.

1.2 The Nature of the Problem

When given an image as obtained from a standard camera device, say a digitized video signal or a scanned photograph, all information is in principle in the pixel values represented as a matrix of numerical data. If this information is presented to a human observer with the pixel values coded as grey-level intensities, then he or she will in general have no problems in perceiving and interpreting what the image represents.

However, if the same pattern of grey-level values is coded as decimal digits, or as a three-dimensional diagram with the grey-level values drawn as a function of the image coordinates, the problem is no longer as easy for biological vision. A person not familiar with the field often underestimates the difficulties in designing algorithms for interpreting data on this numerical form. The problem with the matrix representation of the image is that the information is only *implicit* in the data.

¹Although there are neuropsychological indications for the existence of processing at multiple scales in biological vision systems, we will not make any claims that the methodology to be proposed here in any way describes how processing is done in human perception. We will rather be concerned with what visual information can be extracted by a computer. When biological vision is discussed it will be mainly as a source of inspiration.

1.2.1 Ill-posedness

The task of a visual processing system can be said to be to *extract* meaningful information about the outside world from a set of pixel values that are the result of light measurements from a physical scene.

In principle this vision problem is impossible to solve if it is stated as a pure mathematical problem. Given a data set of grey-level values, there will always be an infinite number of scenes that could have given rise to the same result. To realize that this is the case, consider for instance a photograph on a paper or a slide projected onto a screen. We easily interpret such light distributions on flat surfaces as corresponding to three-dimensional objects with perceived depth. From this viewpoint the vision problem is ill-posed² in the sense of Hadamard, since it does not have any unique solution. A rigorous person without plenty of unspoiled optimism would probably take this as a very good motivation to study some other field of science where the pre-requisites could be more clearly stated and better suited for formal analysis. Nevertheless, despite this, the human visual system as well as other biological vision systems are capable of coping with the indeterminacy. Moreover, since vision is generally regarded as our most important sense one can believe that there must be some properties in the image data reaching the retina that make the visual perception³ possible.

1.2.2 Grouping

A main purpose of the low-level processing modules is to provide a reasonable set of primitives that can be used for further processing or reasoning modules. A fundamental problem in this context concerns which points in the image are related to each other and correspond to objects in the scene, i.e., which pixels in the image belong together and form meaningful entities. This is the problem of primitive grouping or perceptual organization. Before any such grouping operations have been performed, the matrix of grey-level values is, from the viewpoint of interpretation, in principle only an unstructured data set of numerical values.

The grouping problem has been extensively studied in psychology, especially by the Gestaltists [Kof35], and in computer vision, see e.g. Lowe [Low85] or Ahuja and Tuceryan [Ahu89] for an overview, and it seems to be generally agreed upon that the existence of active grouping processes in the human perception can be regarded as established. Witkin and Tenenbaum [Wit83b] discuss this property:

People are able to perceive structures in images, apart from the perception of three-dimensionality, and apart from the recognition of familiar objects. We impose organization on data ... even when we have no idea what it is we are organizing. What is remarkable is the degree to which such naively perceived structure survives more or less intact once a semantic context is established: the naive observer often sees essentially the same thing as an expert does. ... It is almost as if the visual system has some basis for guessing *what* is important without knowing *why*.

²For a mathematical problem to be regarded as *well-posed*, Hadamard stated three criteria: (i) a solution should exist (ii) the solution should be unique and (iii) the solution should depend continuously on the input data. A well-posed problem is not necessarily *well-conditioned*.

³Of course, experiences and expectations are generally believed to play a very important role in the perception process. However, also that information must be related to the incoming image data in some way. Moreover, the experiences must have been acquired (learned) in some way, at least partially based on visual data

Although the gestalt school of psychology formulated rules as those of proximity, similarity, closure, continuation, symmetry and familiarity, we still have no satisfactory understanding of how these mechanisms operate from a quantitative point of view.

1.2.3 Operator Size

In order to be able to derive any information from an image it is necessary to operate on the data with some operators. This leads to two simple but very fundamental problems: One has to determine where to apply the operator and which operator size to use.

To illustrate this problem consider the task of detecting edges. It is generally agreed upon that this type of image features represents important information, since edges in the image often correspond to discontinuities in depth, surface orientation, reflectance properties or illumination in the physical world. A standard way of extracting edges from an image is by gradient computation followed by some type of post-processing step where “high values” should be separated from “low values”, e.g. by detection of local maxima or by thresholding on the gradient magnitude. For simplicity consider the one-dimensional case and assume that the gradient is computed with a central difference operator. More sophisticated approaches exist, but they will face similar problems. The selection of step size leads to a well-known trade-off question: A small step size will give a small truncation error but the noise sensitivity might be severe. Conversely, a large step size reduces the noise sensitivity at the cost of an increased truncation error. In the worst case we may even miss the interesting slope and get meaningless results if the difference quotient is formed over a larger distance than the object to be considered in the image, see Figure 1.1 for an illustration. Therefore, only a certain interval of step sizes can be appropriate for extracting the main slope of the signal that we perceive when looking at that figure. Note also that this slope may in fact be interpreted as due to noise (or some other phenomena that should be neglected) if it is a part superimposed onto some coarser scale structure (not visible here).

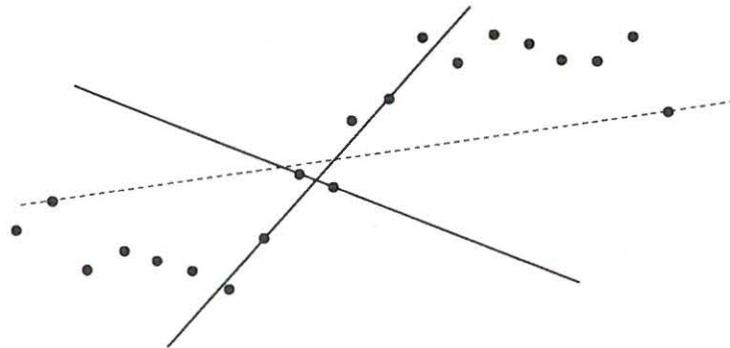


Figure 1.1: *Illustration of the basic scale problem involved when computing the gradients that are to form the basis for edge detection. The lines show the effects of computing derivative approximations from noisy data (here represented as a set of dots) using a central difference operator with varying step size. Note that a if the step size is selected too small then the noise sensitivity can be severe. A larger step size on the other hand reduces the noise sensitivity at the cost of an increased truncation error. In the worst case we may even miss the interesting slope and get meaningless results if the difference quotient is formed over a larger distance than the object to be considered in the image.*

1.2.4 Scale

The problem falls back on the basic scale problem, namely that that objects in the world and details in an image only exist and make sense over a limited range of scale⁴. A typical example is the concept of a branch of a tree which makes sense only on the scale say from a few centimeters to at most a few meters. It is meaningless to discuss the tree concept at the nanometer or the kilometer level. At those levels of scale it is more relevant to talk about the molecules, which form the leaves of the tree, or the forest, in which the tree grows. Similarly, it is meaningful to talk about a cloud only at a coarse scale. At finer scales it is more appropriate to talk about the individual droplets, which in turn consist of water molecules, atoms etc.

For a finite image only structures within a certain range of scales can be resolved and registered. This interval is delimited by the *inner scale* corresponding to the sampling density, that is, the resolution of the image and the *outer scale* corresponding to the size of the image.

These properties indicate that if one aims at describing the structure of an image, the scale concept is of crucial importance. A methodology that has been suggested to deal with this issue is by representing signals at multiple scales. Moreover, since in general no particular levels of scale can be pre-supposed without strong a priori knowledge, it is natural that all levels of scale have to be considered. The main idea of creating a *multi-scale representation* of a signal is by generating a whole family of derived signals where the fine-scale information is successively suppressed. Then a mechanism, which systematically simplifies the data and removes the finer scale details or the high-frequency information, is required. This operation, which will be termed *scale-space smoothing*, must be available at any level of scale.

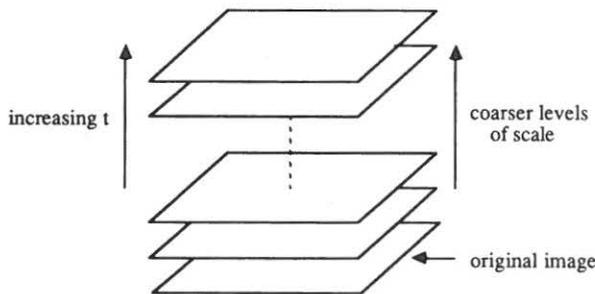


Figure 1.2: A multi-scale representation of a signal is an ordered set of derived signals intended to represent the original signal at various levels of scale.

Why should one represent a signal at multiple scales or different levels of resolution when all information is anyway in the original data? The reason for this is that we would like

⁴An important philosophical question in this context concerns if this property should be attributed to the actual physical objects themselves or just to our subjective way of perceiving and categorizing them. For instance, a table made out of wood certainly has a fine-scale texture with underlying fibral and molecular structures that we usually suppress when dealing with it for everyday purposes. Obviously such finer scale properties will always be there but anyway we almost always automatically disregard those. One may speculate that the organization at multiple scales may in fact be just one of our ways of simplifying our extremely complicated environment into a hierarchical structure as to be able to cope with it. However, even if this standpoint would be the “true” one, it could still be the way that there are properties in image data that make such hierarchical organization suitable and also, possibly, efficient.

to explicitly cope with the scale aspect. Another aim is to simplify further processing by removing unnecessary and disturbing details, such that the later stage processing tasks can be simplified.

1.3 Scale-Space Representation

A methodology proposed by Witkin [Wit83a] and Koenderink, van Doorn [Koe84] to obtain such a multi-scale representation is by embedding the original signal into a one-parameter family of derived signals, the *scale-space*, where the parameter⁵ t , denoted *scale parameter*, is intended to somehow describe the current level of scale⁶. Let us briefly review the procedure as it is formulated for one-dimensional continuous signals: Given a signal $f : R \rightarrow R$ a function⁷ $L : R \times R_+ \rightarrow R$ is defined by $L(\cdot; 0) = f(\cdot)$ and convolution with the Gaussian kernel $g : R \times R_+ \setminus \{0\} \rightarrow R$

$$L(x; t) = \int_{\xi=-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} f(x - \xi) d\xi \quad (1.1)$$

if $t > 0$. Equivalently, the family can be regarded as generated by the diffusion equation

$$\frac{\partial L}{\partial t} = \frac{1}{2} \frac{\partial^2 L}{\partial x^2} \quad (1.2)$$

with initial condition $L(\cdot; 0) = f(\cdot)$. For a two-dimensional signal $f : R^2 \rightarrow R$ the scale-space $L : R^2 \times R_+ \rightarrow R$ is given by convolution with the two-dimensional Gaussian kernel

$$L(x, y; t) = \int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} \frac{1}{2\pi t} e^{-(\xi^2+\eta^2)/2t} f(x - \xi, y - \eta) d\xi d\eta \quad (1.3)$$

or equivalently as the solution to the two-dimensional diffusion equation

$$\frac{\partial L}{\partial t} = \frac{1}{2} \left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \right) \quad (1.4)$$

where the initial conditions are of the same type as in the one-dimensional case. Similar ideas can be applied to higher dimensions. However, since the amount of generated data in general increases rapidly with the number of dimensions, we will here be restricted to one-dimensional and two-dimensional signals, since these cases (currently) have the highest relevance to computer vision applications.

At first glance the task of designing a multi-scale signal representation could be regarded as rather arbitrary. Would it suffice to carry out just *any* type of “smoothing operation”? This is, however, not the case. A crucial problem is that the transformation from a fine

⁵The parameter t used in this presentation corresponds to σ^2 , where σ is the standard deviation of the Gaussian kernel.

⁶We have not yet formally defined what we mean by scale. In principle there should be a correspondence between the scale parameter and a characteristic length of a characteristic object in the scale-space representation at that scale. However, so far the scale parameter should be interpreted only as an *abstract* scale parameter implying a weak ordering property of objects of different size without any direct mapping from its actual value to the size of features in a signal represented at that scale. Later in Section 8.6 we will specify further what types of relations between the size of image features and the actual value of the scale parameter that this definition leads to.

⁷ R_+ denotes the set of real non-negative numbers.

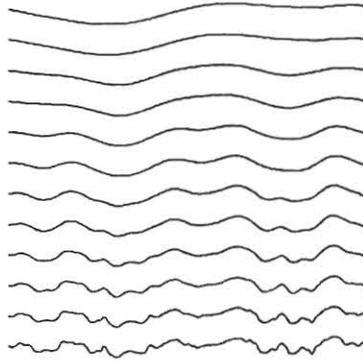


Figure 1.3: The main idea with a scale-space representation of a signal is to generate a family of derived signals where the fine-scale information is successively suppressed. This figure shows a signal that has been successively smoothed by convolution with the Gaussian kernel. (Adapted from Witkin (1983)).

scale to a coarse scale really can be regarded as a simplification such that fine scale features disappear *monotonically*. If new artificial structures could be created at coarser scales, not corresponding to important regions in the finer scale representations of the signal, then it would be impossible to determine whether a feature at a coarse scale corresponded to a simplification of some coarse scale structure from the original image or if it were just an accidental phenomenon, say an amplification of the noise, *created by the smoothing method* — not the data. Therefore, it is of outmost importance that artifacts are not introduced by the smoothing transformation when going from a finer to a coarser scale.

1.3.1 Non-Creation of New Structure

Then, what should one mean by structure? When Witkin [Wit83a] coined the term scale-space of a one-dimensional signal, he observed that the number of zero-crossings in the second derivative of the signal decreased monotonically with scale and took that as a basic characteristic of the representation. In fact this property holds for derivatives of arbitrary order and also implies that the number of local extrema in any derivative of the signal cannot increase with increasing scale. From this viewpoint convolution with a Gaussian kernel can really be regarded as possessing a strong smoothing property.

Later, when Koenderink and van Doorn [Koe84] extended the scale-space concept to two-dimensional signals they introduced the notion of *causality*, which means that new level curves must not be created when the scale parameter is increased. In other words, it should always be possible to trace a grey-level value existing at a certain level of scale to a similar grey-level at any finer level of scale. The reverse statement does of course not need to be true. Combined with *homogeneity* and *isotropy* constraints, which essentially mean that all spatial points and all scale levels should be handled in a similar manner, it was shown that these criteria necessarily and sufficiently lead to a formulation in terms of the diffusion equation, both in one and two dimensions. A similar result, although based on slightly different assumptions, was given by Yuille and Poggio [Yui86] regarding the zero-crossings of the Laplacian. Yet another proof was provided by Babaud et al. [Bab86] who showed that natural constraints on a one-dimensional smoothing kernel necessarily implied that the

kernel had to be a Gaussian.

To summarize, it has been established that within the class of convolution transformations (which means that the blurring is given by shift-invariant linear filtering), the only reasonable way of embedding a signal into one-parameter multi-scale family of representations is by the diffusion equation or equivalently by convolution with the Gaussian kernel. It can by now be regarded as generally agreed upon that this formulation describes the canonical way to construct a multi-scale signal representation.

1.3.2 Other Multi-Scale Approaches

The idea of representing signals at multiple scales is not new⁸. Early work in this direction has been performed by e.g. Rosenfeld and his co-workers, see e.g. [Ros71, Ros84], and Klinger [Kli71] about the representation of signals at different levels of resolution⁹, in particular using *pyramids*. A pyramid is a set of successively smoothed *and* sub-sampled representations of the original signal, organized in such a way that the number of pixels decreases with a constant factor (usually either 2 or 4) from one layer to the next.

These ideas have been developed further by e.g. Burt and Adelson [Bur83], Crowley and his co-workers [Cro84a, Cro84b, Cro87] and others, see e.g. Meer et al. [Mee87] and Cantoni and Levialdi [Can86]. Marr [Mar76] and Marr, Hildreth [Mar80] made use of difference of Gaussians (DOG), which are approximations to the Laplacian of the Gaussian, at different scales. Recently, a concept of anisotropic diffusion has been proposed by Perona and Malik [Per90] and been developed further by Nordström [Nor90].

Among other types of representations involving multiple scales one can mention the Gabor functions [Gab46] as well as the wavelet theory, see e.g. Strömberg [Str83] and Meyer [Mey88], which has been applied to image analysis by Mallat [Mal88, Mal89]. Multi-grid methods, see e.g. Hackbush [Hac85], are receiving a growing interest in numerical analysis together with techniques based on hierarchical basis functions for finite element spaces, see Yserentant [Yse86] and Szeliski [Sze90]. Another interesting early work was done by Ehrich and Lai [Ehr78]. They did not directly rely on multiple scales, but a different type of hierarchical signal representation based on the inclusion of extremal regions into each other.

Multi-scale representations of *curves* have been studied by e.g. Bengtsson and Eklundh [Ben86, Ben90], who define a sequence of polygons approximating the original data with varying accuracy, Mokhtarian and Mackworth [Mok86, Mac88, Mok88], who smooth the coordinate functions of a parameterized curve, Lowe [Low88] who suggests a way to compensate for the shrinking problems in that type of smoothing, and Kimia et al. [Kim90], who use a reaction-diffusion approach.

1.3.3 Multi-Scale v.s. Multi-Resolution

The main difference between a multi-scale and a multi-resolution representation is that a multi-scale representation is defined by smoothing, where one in principle uses the same number of grid points at all levels of scale, while in a multi-resolution representation the

⁸One of the most important contributions with Witkin's and Koenderink and van Doorns' scale-space formulation was the systematic way to *relate* and *interconnect* representations at different scales.

⁹The scale and the resolution concepts are sometimes used interchangeably in the vision literature and their precise meanings are not always clear. Worth emphasizing in this context is therefore that with resolution in this thesis we mean just the spatial density of grid points used in the sampling of the image. Scale on the other hands stands for the characteristic length over which variations in the image take place and/or the operator size used for processing the image data.

main objective is to reduce the number of grid points from one layer to the next. In order to reduce the aliasing problems some pre-filtering must be performed before the sub-sampling step is carried out. Different operators have been proposed for this task, see e.g. Burt and Adelson [Bur83], Crowley et al. [Cro84a, Cro84b] or Meer et al. [Mee87].

Hence, a multi-resolution representation will be efficient in the sense that the number of grid points will be rapidly decreased, while a scale-space representation will get more and more redundant as the scale increases. A wavelet representation is in fact non-redundant while a scale-space representation can be said to be maximally redundant. On the other hand in a scale-space representation, the representations at all levels of scale are immediately accessible without any need for further computations. The task of operating on the data will be successively simplified, since a feature existing at a coarse scale will in general correspond to a larger number of grid points than a feature at a fine scale. In pyramid representations, however, this relation remains unchanged — there is a fixed relation between the scale parameter and the resolution. Moreover, in contrast to the pyramids and the wavelets, the scale-space representation is invariant to translations in space.

Another important property with the scale-space representation is that the behaviour of structure across scales can be analytically described with a simple formalism. By definition it is given as the solution to the diffusion equation, which means that features at different scales can be related to each other in a precise manner. Moreover, the pyramid representations imply a fixed sampling step in scale or resolution that cannot be decreased, while the scale-space concept possesses a *continuous* scale parameter. Therefore, one can expect the task of following or tracking features across scales to be easier in a multi-scale than in a multi-resolution representation, since refinements of the scale sampling can be performed whenever required. Finally, it is sometimes argued that the pyramid representations undersample the signals along the scale direction.

1.3.4 Theoretical Scale-Space Properties

There have been thorough investigations about the theoretical properties of this representation. As was described above, the fundamental property of non introducing new “artificial” structure has been given different formulations by different authors. The behaviour of structures under this type of smoothing has been analysed by Koenderink and van Doorn [Koe86]. Other studies concerning edges have been made by Bergholm [Ber87] and Clark [Cla88]. Hummel [Hum86] investigated the information content in the zero-crossings of the Laplacian. These properties together make the scale-space representation special and one should therefore be careful of not using the term “scale-space” for other possible types of multi-scale-like representations, like those that can be obtained, e.g. by varying regularization parameters and error criteria in optimization methods, unless similar theoretical properties can be proved.

1.4 Philosophies and Ideas behind the Approach

1.4.1 Making Information Explicit

This scale-space theory constitutes a well-founded framework for handling structures at different scales. However, the information in the scale-space embedding is only *implicit* in the grey-level values. The smoothed images in the raw scale-space representation contain no

explicit information about the features in them or the relations between features at different levels of scale.

One of the main goals of this thesis is to present such an explicit representation, called the *scale-space primal sketch*, and to demonstrate that it enables *extraction of significant image structures* in such a way that the output can be used for *guiding later stage processes in early vision* as to simplify their tasks. We shall treat intensity images, the grey-level landscape, and the objects will be blobs, that is, bright regions on dark backgrounds or vice versa. However, the theory applies to any bounded function and is therefore useful in many tasks occurring in computer vision, like the study of level curves and spatial derivatives in general, depth maps, colour etc, and also of histograms and for point clustering and grouping in one or several variables.

1.4.2 Scale and Segmentation

Many methods in image analysis implicitly assume that the problems of scale detection and initial segmentation have been solved. Models based on spatial derivatives ultimately rely upon the computation of difference approximations, which means that they will face similar scale problems as were described in the discussion about edge detection from gradient data in Section 1.2.4. Although we will here be concerned mainly with static imagery, the same type of problems arise also when dealing with image sequences. In other words, when computing derivatives from measured data we in general always fall back to a basic scale problem, namely that of selecting a filter mask size¹⁰ for the approximation.

A commonly used technique to improve the results obtained in computer vision and other branches of applied numerical analysis is by pre-processing the input data with some amount of smoothing and/or careful tuning of the operator size or some other parameters. In some situations the output result might depend strongly on these processing steps. For some algorithms these so called tuning parameters can be estimated, in other cases they are set manually. A robust image analysis method, intended to work in an autonomous robot situation, must however be able to make such decisions. How should this be done? We contend that these problems are in many situations nothing but disguised scale problems.

More generally, in order to be able to apply a refined mathematical model like a differential equation or some kind of deformable template it is necessary to have some kind of qualitative initial information, i.e., a domain where the differential equation is (assumed to be) valid or an initial region for application of the raw deformable template. Examples can be obtained from many "shape from X" methods, which in general assume that they are applied to a domain in the image where the underlying assumptions are satisfied. A commonly used assumption is that of smoothness implying that the region in the image, to which the model is applied to, must correspond to, say, one physical object or one facet of a surface etc. How should we select such regions *automatically*? Many methods cannot be used unless this non-trivial part of the problem has been solved.

How do we detect appropriate scales and regions of interest when there is no a priori information available? In other words, how to detect the scale of an object and where to search for it before knowing what kind of object we are studying and before knowing where it is located. This problem arises implicitly in many kinds of processes, e.g. dealing with texture, contours etc. It all seems to boil down to an impossible chicken-in-the-egg problem.

¹⁰Observe that it is not so much the actual size of the filter mask that is important but rather the *characteristic length* over which the difference approximation is computed.

The solution of the pre-attentive recognition problem requires the solution of the scale and region problems and vice versa. However, in this work we will show that such *scale and region determination actually can be performed computationally from raw image data by early low-level processing*. The basic tool for the analysis will be the scale-space theory.

We argue that once scale information is available and once we have extracted “regions of interest” the remaining processing tasks can be much simpler. We will support this claim by experiments on edge detection and classification based on local features.

1.4.3 Detection of Image Structure

The main features arising in the scale-space representation are smooth regions which are brighter or darker than their background and stand out from the surrounding. We will call them blobs (a precise definition will be given later). The purpose of the suggested scale-space primal sketch representation is to make these blobs as well as their relations across scales explicit. The idea is also that this representation should reflect the intrinsic shape of the grey-level landscape — not be an effect of some externally chosen criteria or tuning parameters. The theory should in a bottom-up fashion allow for a data-driven detection of significant structures, their relations and the scales at which they occur. We will, indeed, experimentally show that the proposed representation gives intuitively reasonable results, in which salient structures are (coarsely) segmented out. Hence, this representation can serve as a guide to subsequent, more finely tuned processing, that requires knowledge about where and at which scales structure occurs. In this respect it can serve as a mechanism for focus-of-attention.

As one application demonstrating the predictive power of our method we have integrated the output from the scale-space primal sketch with an algorithm known as edge focusing, see Bergholm [Ber87]. We let the extracted scale level and region information serve as to initiate an edge focusing procedure starting at an *adaptively determined local scale* determined from a relevant scale of a significant blob. The experiment shows that, at a proper level of scale, edges can be detected without thresholding, however at the cost of possibly poor localization. But, the localization can be considerably improved using the edge focusing method, which tracks the safely detected edges at coarse levels of scale through scale-space to corresponding and better localized edges at finer levels of scale.

Since the proposed representation tries to capture *all* the important structure with a small set of primitives, it bears some similarity to Marr’s primal sketch, even though fewer primitives are used. However, the central issue here is to explicitly represent also the scale at which different events occur. In this respect our work addresses problems similar to those studied by Bischof and Caelli [Bis88]. They try to parse the scale-space by defining a measure of stability. However, their work focuses on zero-crossings of the Laplacian and is therefore less general than our approach. Moreover, they overlook the fact that in measuring significance or stability of structures we must treat the scale parameter properly. The behaviour of structure over scale will be analyzed to give the basis of such measurements.

Of course, several other representations of the grey-level landscape have been proposed without relying on scale-space theory. Let us also note that Pizer and his co-workers, [Lif87, Piz87], indeed, have performed studies of the behaviour of local extrema in scale-space. However, we will defer discussing the relations to these representations until we have described our own methodology.

1.4.4 Computational Issues

The idea of scale-space representation of images, suggested by Witkin [Wit83a] has, in particular, been developed by Koenderink and van Doorn [Koe84, Koe86]. Our work is aimed at complementing this work by considering the *computational aspects* and by adding means of *making significant structures and scales explicit*. It turns out that several problems have to be solved to this end. One basic problem is how to measure significant and insignificant behaviour over scale. This involves questions about "the amount of structure in an image without structure" and the interference between salient structure and the inner and outer scales. An observation in this context is that noise can survive for a long time during scale-space blurring. In measuring significance we need an appropriate scaling of the scale parameter. It turns out that these problems touch upon general issues about the appearance of structure in images. We will show that they can be given well-founded solutions and that the theoretical framework can be robustly implemented in a rigorous manner.

1.4.5 Consistency over Scales

The main idea with our approach is to *link* features at different levels of scales in scale-space into (four-dimensional) higher order objects, called scale-space blobs, and to extract significant image features based on the appearance and lifetime of the higher order objects in scale-space. We argue that *significant image features must be stable with respect to variations in scale*. Another important point with our work is that we treat *the scale parameter as equally important as the spatial and grey-level coordinates*. This is directly reflected in the fact that *the primitives in our representation are objects having extent not only in space and grey-level, but also in scale*.

1.5 Relations to Traditional Numerical Analysis

In principle we are to derive information from image data by operating on it with certain operators. An obvious question is then why this could not be seen as an ordinary standard problem in numerical analysis and be solved with standard numerical techniques? Let us point out several reasons to why the problem is hard.

1.5.1 Modelling, Simulation and Inverse Problem

Traditional numerical analysis is often concerned with the *simulation* of mathematical or physical models, for example formulated as discrete approximations to continuous differential equations, which are rather good descriptions of the underlying reality. The problems are usually well-defined, the models can often be treated as exact and the errors involved in these types of computations are mainly due to discretization and round-off errors.

In computer vision we have a different situation. Given a signal, the task is to analyse and extract information from it. We are trying to solve an *inverse problem* where the noise level is generally substantially higher¹¹ and the modelling¹². aspect is still open. With

¹¹A rule of thumb sometimes used in this context is that when derivatives of order higher than two are computed from raw image data, then the amplitude of the amplified noise will often be of the same order of magnitude as the derivative of the signal, or be even higher.

¹²The *geometry* of image formation is quite simple and well understood, but our knowledge about the complicated *physical phenomena* (comprising reflections etc) and how to model those from a computational viewpoint is still rather vague. In addition, we have the problem of *representing* the enormous variety of

a precise model of the illumination situation as well as the reflectance properties of the surfaces in the environment one could conceive solving for the surface geometry based on the physical light characteristics. This is the subject of e.g. shape from shading. However, it is well-known that this problem of *reconstructing the world* is extremely hard, to a large extent because it is very difficult to formulate an accurate and useful physical model for the image formation process, but also because such a model would require a lot of additional a priori knowledge in order to be computationally tractable. Although further attempts to explore the situation in more detail are being made, see e.g. Forsyth and Zissermann [For89a, For89b] or Nayar et al. [Nay90], most shape from shading and similar algorithms still rely on very restrictive simplifying assumptions.

1.5.2 Scale and Resolution

Other aspects are those of scale and resolution. In numerical analysis the accuracy can often be increased by a refinement in the grid sampling. The selection of a larger grid size is often mainly motivated by efficiency reasons, since one is simulating exact equations. In computer vision algorithms the number of grid points used for *resolving structures* in a given image is sometimes very low, something that we believe makes a difficult problem even more difficult. This restriction can be however be relaxed in an active vision situation as will be developed in Section 9.3. A more serious problem is that of scale. In most standard numerical problems the inner scale is zero, which means that the smaller grid size that is being used, the higher will the accuracy be in the computations (compare again with the example in Section 1.2.3). In easy problems, the solutions asked for often contain variations taking place on essentially one single scale.

Problems having solutions with variations on different scales are more complicated and require more advanced algorithms for their solution. Examples can be obtained from *computer fluid dynamics*, where turbulence and very thin boundary layers are known to lead to very hard numerical problems. These fine-scale phenomena cannot always be fully resolved by the discrete approximations, and in fact some type of (sometimes artificial) smoothing (dissipative terms) is often required. When the fine-scale phenomena are not properly dealt with, they can interfere with and disturb the coarse-scale phenomena that usually are the ones of interest in e.g. design applications. Moreover, the occurrences of *discontinuities* in the solutions, which are also very frequent in image data, are known to complicate the situation.

The idea with the scale-space representation is to separate out information at different scales. Note however that we are confronted with a very difficult problem, since in general we have just very little or no a priori knowledge at all about what types of structures we are studying or at what scales they occur.

1.5.3 Interpreting the Results

If we apply some operator all over an image we will hopefully get reasonable answers in those regions where the underlying assumptions for the method are valid (provided that the operator size has been appropriately tuned). However we will also get *false alarms* in regions where the assumptions are not satisfied. One could say that such a uniform application of an operator enforces an answer in every point even though any well-defined

different situations that can occur in the real world as well the question of how *cognitive* aspects should be incorporated into the process.

answer does not exist. In general it is very hard to distinguish, just from the output of such an operation, which responses can be trusted as correct and which ones should be rejected. Plain thresholding on the magnitude of the response is usually not sufficient. Therefore, a conservative strategy is to rather aim at deriving a sparse set of safe and reliable cues at the risk of “missing” a few ones that possibly should be included than to try to compute “every” feature at the risk of including a large number of false responses. This is the motivation to why we would like to determine in advance where to apply¹³ refined operations.

1.5.4 Approximation and Regularization

It is sometimes argued that the main aims of approximation theory have already been accomplished. Nevertheless, one is confronted with serious problems when applying this theory to irregular and noisy measurement data as those obtained from images. Some of the most basic problems concern how one should determine a region in space appropriate for fitting a model to the data and how one should tune the associated parameters, such as the filter weights. An approach that has been used extensively in computer vision during the last decade is regularization. This technique has been applied to a variety of reconstruction problems, see e.g. Terzopoulos et al. [Ter86, Ter87, Ter88, Kas87, Wit87], Pentland [Pen88, Pen90], Blake and Zisserman [Bla87] and Aloimonos and Schulmann [Alo89]. The basic methodology is to define some functional, which is a weighted combination of different error criteria, and then try to compute the function within some restricted space that minimizes it. These methods often contain a large number of parameters but the theory gives little or no information about how they should be set without manual intervention, although attempts have been made to learn them from examples [Alo89]. In addition we have a verification problem, since the algorithm is forced to always find a solution within the given space. How does one determine whether that function resembles the answer we actually want (to the original problem). To summarize, both these types of methods require a careful setting of their associated parameters as well as the regions in space to which they should be applied.

1.5.5 Principles behind the Work

A basic intention with the work presented here is to pre-process the data and to derive context information from it in such a way that the output from these types of operations can be well-defined. Although we do not claim that we have solved these problems and even though further complications may appear on the way to the solution, we believe that the framework to be developed here represents a significant step towards posing the questions in a context where standard numerical techniques could be readily applied and give useful answers.

1.6 Organization of the Presentation

The thesis deals with the very fundamental problems that are associated with the use of scale-space analysis in early low-level processing of visual data. More specifically some of the main questions we will address are:

¹³This is a problem arising mainly in an initialization phase of a reasoning process. In a situation where the time aspect is present, this problem should be simplified, since then the context knowledge could be used for predictions to the future. It is generally argued that problems become easier once the boot-strapping step has been performed.

- How should the scale-space model be implemented computationally? The scale-space theory has been formulated for continuous signals while realistic signals are discrete.
- Can the scale-space representation be used for extracting information? How should that be done?
- The scale-space representation in itself contains no information about preferred scales. In fact, without any a priori scale information all levels of scale must be treated similarly. Is it possible to determine a sparse set of appropriate scales for processing the information?
- How can the scale-space concept interact and cooperate with other processing modules?
- What can happen in scale-space? What is the behaviour of structure in scale-space? How will features evolve under scale-space smoothing? What types of events can take place?

The presentation is divided into three parts. We start by developing a *scale-space theory for discrete signals*. Then we present a representation called *the scale-space primal sketch*, which is a formal representation of structures in scale-space at multiple scales aimed at making information in the scale-space representation explicit. Finally we demonstrate how this representation can be integrated with other visual modules. We illustrate how qualitative scale and region information extracted from the scale-space primal sketch can be used for *guiding other low-level processes and simplifying their tasks*. We will now, in the form of a long abstract, give a brief overview of some of the main results to be presented in each one of the different parts.

1.6.1 Part I: Scale-Space Theory for Discrete Signals

We start by formulating a scale-space theory for *discrete* signals. In one dimension it is possible to *completely characterize* which linear transformations on the form

$$f_{out}(x) = \sum_{n=-\infty}^{\infty} K(n)f(x-n) \quad (1.5)$$

can be regarded as smoothing transformations. An exhaustive treatment is given, answering the following two main questions:

1. Which linear transformations on that form remove structure in the sense that the number of local extrema does not increase?
2. How should one create a multi-scale family of representations with the property that a signal at a coarser level of scale never contains more structure than a signal at a finer level of scale?

Qualitative properties of the relevant kernels are derived. We show that they necessarily have to be non-negative and unimodal both in the spatial and the frequency domains. It is also shown that all such kernels with finite support can be derived from *generalized binomial kernels* having a generating function of the form

$$\varphi_K(z) = Cz^k \prod_{i=1}^N (1 + \alpha_i z) \quad (k, N \in \mathbb{Z}) \quad (1.6)$$

We propose that there is only one reasonable way to define a scale-space for one-dimensional discrete signals comprising a *continuous* scale parameter

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t) f(x - n) \quad (1.7)$$

namely by discrete convolution with the family of kernels called *the discrete analog of the Gaussian kernel*

$$T(n; t) = e^{-t} I_n(t) \quad (1.8)$$

where I_n are the modified Bessel functions of integer order. Similar arguments applied in the continuous case *uniquely* lead to the Gaussian kernel. Some obvious discretizations of the continuous scale-space theory are discussed in view of the results presented. We show that the scale-space family *equivalently* is given by the solution to the semi-discretized *diffusion* equation:

$$\frac{\partial L}{\partial t} = \frac{1}{2} \nabla_3^2 L \quad (1.9)$$

with initial condition $L(\cdot; t) = f(\cdot)$. The commonly adapted technique with a sampled Gaussian can lead to undesirable effects, since scale-space violations might occur in the corresponding representation. The result exemplifies the fact that properties derived in the continuous case might be violated after discretization.

A two-dimensional theory, showing how the scale-space should be constructed for images, is given based on the requirement that *local extrema must not be enhanced* when the scale parameter is increased continuously. We show that this requirement, combined with linear shift-invariant smoothing and uniform treatment of all scale levels, by necessity and sufficiency implies that the scale-space representation has to satisfy the equation

$$\frac{\partial L}{\partial t} = \frac{C}{2} \left((1 - \gamma) \nabla_5^2 L + \gamma \nabla_x^2 L \right) \quad (1.10)$$

for some $C > 0$ and $\gamma \in [0, 1]$, where ∇_5^2 and ∇_x^2 denote the well-known discrete five-point and cross-operators approximating the continuous Laplacian. In the separable case, corresponding to $\gamma = 0$, the resulting scale-space representation can be computed by separated convolution with the kernel $T(n; t)$.

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m; t) T(n; t) f(x - m, y - n) \quad (1.11)$$

We outline how a discrete version of the N-jet representation, see Koenderink and van Doorn [Koe87], with derivatives computed from the scale-space representation at different scales, can be defined. The presented discrete theory has computational advantages compared to a scale-space implementation based on the sampled Gaussian, for example in the sense that discrete approximations to derivatives can be computed *directly* from the scale-space representations at different scales, without any need for repeating the smoothing operation. The main reason for this is that the discrete nature of the implementation has been taken into account already in the theoretical formulation of the scale-space representation, which means that the involved operators will *commute*.

1.6.2 Part II: Theory of the Scale-Space Primal Sketch

We present a multi-scale representation of grey-level shape, called *scale-space primal sketch*, which makes explicit information in scale-space as well as the relations between certain features at different levels of scale. The representation is based on blobs, that are regions either brighter or darker than the background. We give a formal definition of what we mean by a *grey-level blob*, which is a local extremum with extent in a grey-level image at a certain scale, and a *scale-space blob*, which is a set of grey-level blobs linked across scales. The extent of a scale-space blob in the scale direction is delimited by *bifurcations* between critical points, or equivalently, by bifurcations between blobs. These events also define *hierarchical relations* between scale-space blobs at different scales. The scale-space primal sketch can be interpreted as a tree-like data structure with the scale-space blobs as vertices and the bifurcation events as arcs between those.

The representation is obtained in a completely bottom-up data-driven manner, without relying on any specific parameters or error criteria. We treat grey-level images, but the approach is valid for any bounded function, and can therefore be used for deriving properties of e.g. spatial derivatives.

As to enable comparisons of significance between structures at different scales we need to transform the coordinate axes in the scale-space representation in such a way that structures at different scales will be treated in a uniform manner. We show that natural requirements on a transformed scale parameter, *effective scale* τ , imply that there is in principle only one reasonable way to define it, namely by

$$\tau(t) = \log \left(\frac{p_0}{p(t)} \right) \quad (1.12)$$

where $p(t)$ is the expected density of local extrema at scale t in the scale-space representation of a reference signal and p_0 is a constant. From estimates of how the density of local extrema can be expected to vary with scale we show that for continuous signals this function will be a logarithm, while for discrete signals it will be approximately logarithmic at coarse scales and approximately linear at fine scales. It turns out that the volumes of the grey-level blobs must be transformed in a similar manner. That normalization is based on simulation results accumulated from the evolution properties of grey-level blobs extracted from random noise signals.

We investigate the theoretical properties of the representation by applying elementary techniques from real analysis, singularity theory and statistics to derive analytical results for the behaviour in scale-space of critical points and related entities.

The implicit function theorem for can be used for describing the general nature of the trajectories that the critical points will form when the scale parameter in scale-space is changed. We derive estimates for the drift velocity of critical points and straight edges. For critical points we have

$$\frac{dr}{dt} = -\frac{1}{2}(\mathcal{H}L)^{-1}\nabla^2(\nabla L) \quad (1.13)$$

where $\mathcal{H}L$ denotes the Hessian matrix and ∇L the gradient vector. For a straight edge the drift velocity in the normal direction to the edge is

$$\frac{dr}{dt} = -\frac{1}{2} \frac{\frac{\partial^4 L}{\partial n^4}}{\frac{\partial^3 L}{\partial n^3}} \quad (1.14)$$

These expressions show that the drift velocity momentarily may tend to infinity. Generically, this occurs at bifurcation situations only.

We analyse the qualitative behaviour of critical points in bifurcation situations and classify what types of blob events are possible. In one dimension the generic bifurcation events for critical points are annihilations of pairs consisting of one local maximum and one local minimum. In the two-dimensional case, pairs consisting of a local extremum and a saddle point can be both annihilated and created with increasing scale. Interpreted in terms of blobs these events correspond to, in one dimension *annihilations* and *merges*, while in two dimensions the list of possible blob events comprises *annihilations*, *merges*, *splits* and *creations*. A set of illustrative examples is presented, demonstrating how the blobs behave in characteristic situations.

We describe an algorithm for actually computing the representation. It is based on detection of *grey-level* blobs at different levels of scale. On that output data an *adaptive* scale sampling algorithm operates and performs the actual linking of the grey-level blobs into *scale-space blobs* as well as the registration of the bifurcations and the blob events.

1.6.3 Part III: Applications of the Scale-Space Primal Sketch

We develop a framework for how the scale-space primal sketch can be used for *extracting* significant image structures and their scales and how this type of qualitative information in turn can be used for *guiding* other early visual modules and *simplifying* their tasks. From measurements of stability and significance in scale-space the representation gives a qualitative description of the image structure with information about approximate location and extent as well as appropriate scale for important regions in the image — allowing for detection of stable scales and regions of interest.

In other words, it generates safe segmentation cues and can hence be seen as guiding the focus-of-attention and preceding further processing, which can then be properly tuned. We argue that once such scale and region information is available many other processing tasks can become much simpler. The extraction method is based on the *assumption* that:

- Significant blobs in scale-space correspond to important structures in the image.

The actual *ranking of events* in order of significance is based on the *volumes of the scale-space blobs* in the *four-dimensional* scale-space given by the space, grey-level and scale coordinates. A scale-space blob in general exists over some interval in scale. As *appropriate scale* for such a blob we take the scale where the blob response is as its highest, that is the scale level for which the three-dimensional grey-level blob volume, treated as function of the scale parameter, assumes its maximum. Two important principles behind this approach are that:

- We *link* related features at different level of scale in scale-space and *treat the scale parameter as equally important* as the spatial and grey-level coordinates. This is directly reflected in the fact that the primitives of our representation are objects having extent not only in space and grey-level but also in scale.
- We subject the image to a *systematic parameter variation* in order to detect important image structures by registering *locally stable states* and determining abstractions from those.

An important aspect here is that *stable scale is a local property* associated with objects, not with entire images. Previous methods often face serious problems when assuming the existence of a global stable scale. It is usually impossible to detect such a globally valid scale, since the size of objects and also their distance to the camera will in general vary substantially over an image.

Applications are given to edge detection, histogram analysis and junction classification demonstrating how the proposed method can be used for guiding various sub tasks in early visual processing.

- When integrating the scale-space primal sketch with edge detection we *detect edges* at a *locally adapted scale* determined from a significant scale-space blob. This will simplify the detection problem at the cost of possibly poor localization. In fact we do not do any thresholding on gradient magnitude. Then the localization can be improved by following the edges to finer scales with a method known as edge focusing.
- For histogram analysis we use the scale-space primal sketch for automatic peak detection.
- We also describe how the scale-space primal sketch can be used for providing context information necessary for an *active focusing procedure* aimed at *classifying junctions*. We show how a range of window sizes can be set from the blob information as well as how the blobs can serve to guide the focus-of-attention.

Finally, we briefly outline how the scale-space primal sketch can be applied to other visual tasks such as texture analysis, perceptual grouping and matching problems. Experiments on real imagery demonstrate that the proposed theory gives perceptually intuitive results.

Part I

Discrete Scale-Space Theory

Chapter 2

Scale-Space for 1-D Discrete Signals

2.1 Introduction

The scale-space theory has been developed and well-established for continuous signals and images. However, it does not tell us anything about how the implementation should be performed computationally for real-life problems, i.e. discrete signals and images. In principle, we believe that there are two approaches possible.

- Apply the results obtained from the continuous scale-space theory by discretizing the occurring equations. For instance the convolution integral (1.1) can be approximated by a sum using customary numerical methods. Or, the diffusion equation (1.2) can be discretized in space with the ordinary five-point Laplace operator forming a set of coupled ordinary differential equations, which can be further discretized in scale. If the numerical methods are chosen with caution, we will certainly get reasonable approximations to the continuous numerical values. But we are not guaranteed that the original scale-space conditions, however formulated in a discrete situation, will be preserved.
- Define a genuinely discrete theory by postulating suitable axioms.

The goal with the first part of this thesis is to develop the second item and to address the formulation of a scale-space theory for discrete images. We will start with a one-dimensional signal analysis. In this case it is possible to characterize exactly which kernels can be regarded as smoothing kernels and a complete and exhaustive treatment will be given. One among many questions which are answered is the following: If one performs repeated averaging, does one then get scale-space behaviour? We will also present a family of kernels, which are the discrete analog of the Gaussian family of kernels. The set of arguments, which in the discrete case uniquely leads to this family of kernels, do in the continuous case uniquely lead to the Gaussian family of kernels.

The structure of the two-dimensional problem is more complex, since it is difficult to formulate what should be meant by preservation of structure in this case. However, by slight modification of the arguments used in the one-dimensional case, we will give an answer to how the scale-space for two-dimensional discrete signals should be constructed. In the separable case it reduces to separated convolution with the presented one-dimensional discrete analog of the Gaussian kernel. The representation obtained in this way has computational advantages compared to the commonly adopted approach, where the scale-space is based on different versions of the sampled Gaussian kernel. One of many spin-off products which

come up naturally is a well-conditioned and efficient method to calculate (a discrete analog of) the Laplacian of the Gaussian. It is well-known that the implementation of the Laplacian of the Gaussian has lead to computational problems, see e.g. Grimson and Hildreth [Gri85].

The theory developed in this presentation does also have the attractive property that it is linked to the continuous theory through a discretized version of the diffusion equation. This means that continuous results may be transferred to the discrete implementation provided that the discretization is done correctly. However, the important point with the scale-space concept to be outlined here is that the properties we want from a scale-space hold not only in the ideal theory but also in the discretization¹, since the discrete nature of the problem has been taken into account already in the theoretical formulation of the scale-space representation. Therefore, we believe that the suggested way to implement the scale-space theory really describes the proper way to do it.

The presentation is organized as follows: In Section 2.2 we define what we mean by a scale-space representation and a one-dimensional discrete scale-space kernel. Then in a straightforward and constructive manner Section 2.3 illustrates some qualitative properties that must be possessed by scale-space kernels. A complete characterization as well as an explicit expression for the generating function of all discrete scale-space kernels are given in Section 2.4. Section 2.5 develops the concept of a discrete scale-space with a *continuous* scale parameter. The formulation is equivalent to the previous scale-space formulation, which in the continuous case leads to the Gaussian kernel. Section 2.6 discusses discrete scale-space properties of some obvious discretizations of the convolution integral and the diffusion equation. Section 3.1 describes some problems which occur due to the more complicated topology in two dimensions. In Chapter 3 we develop the scale-space for two-dimensional discrete images. Then in Chapter 4 we compare the discrete scale-space representation with the commonly used approach, where the scale-space implementation is based on various versions of the sampled Gaussian kernel. The numerical implementation of the discrete scale-space is treated in Section 4.3. Finally, Section 4.5 gives a brief summary of the main results.

The results presented should have implications for image analysis as well as other disciplines of digital signal processing.

2.2 Scale-Space Axioms

By a scale-space we mean a family of derived signals intended to represent the original signal at various levels of scale. Each member of the family should be associated with a value of a scale parameter intended to somehow describe the current level of scale. The scale parameter, here denoted by t , may be either discrete ($t \in Z_+$) or continuous ($t \in R_+$) and we obtain two different types of discrete scale-spaces — discrete signals with a discrete scale parameter and discrete signals with a continuous scale parameter. However, in both cases we start from the following basic assumptions:

¹In a practical implementation we are of course faced with truncation and rounding errors due to finite precision. But the idea with this approach is that we hope to improve our algorithms by including at least the discretization effects already in the theory. In ordinary numerical analysis for simulation of physical phenomena it is almost always possible reduce these effects by increasing the density of mesh points, if the current grid is not fine enough to give a prescribed accuracy in the result. However, in computer vision we are often locked to some fixed maximal resolution, beyond which additional image data are not available.

- Every representation should be generated by a linear and shift-invariant transformation of the original signal. Therefore, the smoothing operator can be expressed as a convolution operator.
- An increasing value of the scale parameter t should correspond to coarser levels of scale and signals with less structure. Particularly, $t = 0$ should represent to the original signal.
- All signals should be real-valued functions $: Z \rightarrow R$ defined on the same infinite grid; in other words no pyramid representations will be used.

The essential requirement is that a signal at a coarser level of scale should contain less structure than a signal at a finer level of scale. If one regards the number of local extrema as one measure of the amount of structure it is thus necessary that the number of local extrema in space does not increase as we go from a finer to a coarser level of scale. It can be shown that the family of functions generated by convolution with the Gaussian kernel possesses this property in the continuous case. We state it as the basic axiom for our one-dimensional analysis and define:

Definition 2.1 (Discrete scale-space kernel (1D))

A one-dimensional discrete kernel $K : Z \rightarrow R$ is called a scale-space kernel if for all signals $f_{in} : Z \rightarrow R$ the number of local extrema in the convolved signal $f_{out} = K * f_{in}$ does not exceed the number of local extrema in the original signal.

A minor complication is involved in this statement. If either f_{in} or f_{out} would happen to have a plateau the question must be raised about how many local extrema the plateau should be counted as. At this moment we will not go into the details of those peculiar cases. We count a plateau as one local maximum (minimum) if there are strictly smaller (larger) values bounding it both at the left and at the right, see Fig 2.1. An accurate treatment will be given in Section 2.4.



Figure 2.1: Examples illustrating the definition of local extremum. (a) A local maximum (generic case). (b) A plateau counted as one local maximum. (c) A plateau not counted as a local extremum.

An important observation to be made is that this definition equivalently can be expressed in terms of zero-crossings just by replacing the string “local extrema” with “zero-crossings”. The result follows from the facts that a local extremum in a discrete function f is equivalent to a zero-crossing in its first difference Δf , defined by $(\Delta f)(x) = f(x + 1) - f(x)$, and that the difference operator commutes with the convolution operator.

However, the stated definition has further consequences. It means that the number of local extrema (zero-crossings) in any n :th order difference of the convolved signal cannot be greater than the number of local extrema (zero-crossings) in the n :th order difference of the original signal. Actually, the result can be generalized to arbitrary linear operators.

Proposition 2.1 (General smoothing property of discrete scale-space kernels)

Let $K : Z \rightarrow R$ be a discrete scale-space kernel and \mathcal{L} an operator (from the space of real-valued discrete functions to itself), which commutes with K . Then for any $f : Z \rightarrow R$ (such that the involved quantities exist) the number of local extrema (zero-crossings) in $\mathcal{L}(K * f)$ cannot exceed the number of local extrema (zero-crossings) in $\mathcal{L}(f)$.

Proof: Let $g = \mathcal{L}(f)$. As K is a scale-space kernel the number of local extrema (zero-crossings) in $K * g$ cannot be larger than the number of local extrema (zero-crossings) in g . Since, K and \mathcal{L} commute $K * g = K * \mathcal{L}(f) = \mathcal{L}(K * f)$ and the result follows. \square

This shows that not only the function, but also all its “derivatives” will become smoother. Accordingly, convolution with a discrete scale-space kernel can really be regarded as a smoothing operation.

To realize that the number of local extrema or zero-crossings can increase even in a rather uncomplicated situation consider the input signal

$$f_{in}(x) = \begin{cases} -3 & \text{if } n = 0 \\ 2 & \text{if } n = \pm 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

and convolve it with the kernels $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{1}{2}, \frac{1}{2})$ and $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. The results are shown in Figure 2.2 (b), (c) and (d) respectively. As we see, both the number of local extrema and the number of zero-crossings have increased for the first kernel, but not for the two latter ones. Thus, an operator which naively can be apprehended as a smoothing operator, might actually give a less smooth result. Further, it can really matter if one averages over three instead of two points and how the averaging is performed.

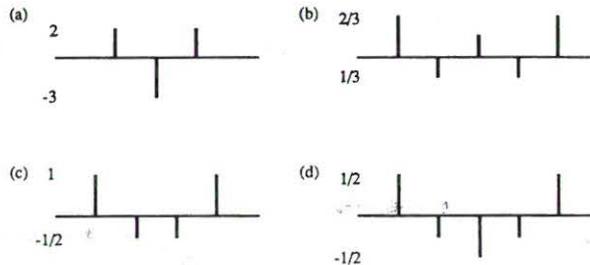


Figure 2.2: (a) Input signal. (b) Convolved with $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (c) Convolved with $(\frac{1}{2}, \frac{1}{2})$. (d) Convolved with $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$.

In order to get familiar with the consequences of the definition we will illustrate what this scale-space property means. We start by pointing out a few general qualitative requirements of a scale-space kernel that are necessarily induced by the given axiom. We will also show that the two latter kernels indeed are discrete scale-space kernels.

2.3 Properties of Scale-Space Kernels

2.3.1 Positivity and Unimodality in the Spatial Domain

By considering the impulse response it is possible to draw some qualitative conclusions about the properties of a discrete scale-space kernel. Let the input function be the discrete delta

function

$$f_{in}(x) = \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

Then, the output signal will be identical to the kernel

$$f_{out}(x) = (K * \delta)(x) = K(x) \quad (2.3)$$

$\delta(x)$ has exactly one local maximum and no zero-crossings. Therefore in order to be a scale-space kernel K must not have more than one extremum and no zero-crossings. Thus,

Proposition 2.2 (Positivity)

All coefficients of a scale-space kernel must have the same sign.

Proposition 2.3 (Unimodality)

The coefficient sequence of a scale-space kernel $\{K(n)\}_{n=-\infty}^{\infty}$ must be unimodal².

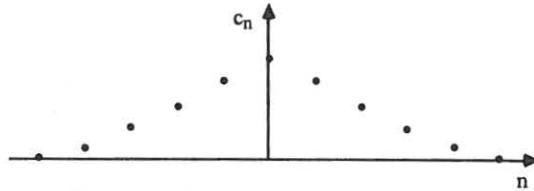


Figure 2.3: The filter coefficient sequence $K(n)_{n=-\infty}^{\infty}$ of a discrete scale-space kernel must be positive and unimodal.

Without loss of generality we can therefore restrict the rest of the treatment to positive sequences where all $K(n) \geq 0$.

It seems reasonable to require³ that $K \in l_1$, i.e. that $\sum_{n=-\infty}^{\infty} |K(n)|$ is finite. If f_{in} is bounded and $K \in l_1$ then the convolution is well-defined and the Fourier transform of the filter coefficient sequence exists. This requirement also allows us to normalize the coefficients such that $\sum_{n=-\infty}^{\infty} K(n) = 1$. Particularly, the filter coefficients $K(n)$ must then tend to zero as n goes to infinity.

2.3.2 Generalized Binomial Kernels

Consider a two-kernel with only two non-zero filter coefficients:

$$K^{(2)}(n) = \begin{cases} p & \text{if } n = 0 \\ q & \text{if } n = -1 \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

Assume that $p \geq 0$, $q \geq 0$ and $p + q = 1$.

It is easy to verify that the number of zero-crossings (local extrema) in $f_{out} = K^{(2)} * f_{in}$ cannot exceed the number of zero crossings (local extrema) in f_{in} . This result follows from

²A real sequence is said to be unimodal if it is first ascending (descending) and then descending (ascending).

³Some regularity requirement must be imposed on the input signal as well. Throughout our following considerations we will stick to one general convention. If nothing else is explicitly mentioned we assume that f_{in} is sufficiently regular such that the involved quantities exist and are well-defined.

the fact that convolution of f_{in} with $K^{(2)}$ is equivalent to the formation a weighted average of the sequence $\{f_{in}(x)\}_{x=-\infty}^{\infty}$, see Figure 2.4. The values of the output signal can be constructed geometrically and will fall on straight lines connecting the values of the input signal. The offset along the x -axis is determined by the ratio $q/(p+q)$. It is obvious that no additional zero-crossings can be introduced by this transformation. Thus, a kernel on the form (2.4) is a discrete scale-space kernel.

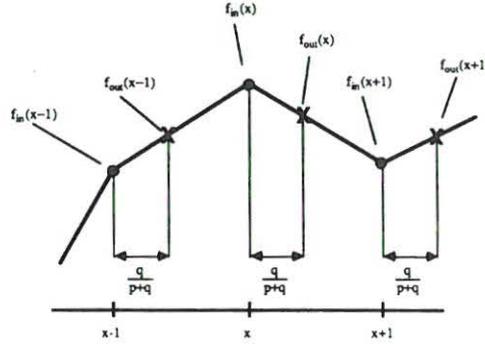


Figure 2.4: To convolve a signal f_{in} with a two-kernel $K^{(2)}(n)$ is equivalent to to form a weighted average of the sequence $\{f_{in}(x)\}_{x=-\infty}^{\infty}$. It is obvious that no new zero-crossings can be introduced by this transformation.

Directly from the definition of a scale-space kernels it follows that if two kernels K_a and K_b are scale-space kernels then also $K_a * K_b$ is a scale-space kernel.

Lemma 2.4 (Repeated application of scale-space kernels)

If two kernels K_a and K_b are scale-space kernels then also $K_a * K_b$ is a scale-space kernel

Repeated application of this result yields:

Proposition 2.5 (Repeated averaging leads to scale-space kernels)

All kernels K on the form $*_{i=1}^n K_i^{(2)}$, with $K_i^{(2)}$ according to (2.4), are discrete scale-space kernels.

The filter coefficients generated in this way can be regarded as a kind of generalized binomial coefficients. The ordinary binomial coefficients are obtained, except for a scaling-factor, as a special case if all p_i and q_i are equal. Another formulation of Proposition 2.5 in terms of generating functions is also possible.

Proposition 2.6 (Generating function of generalized binomial kernels)

All kernels with the generating function $\varphi_K(z) = \sum_{n=-\infty}^{\infty} K(n)z^n$ on the form

$$\varphi_K(z) = C z^k \prod_{i=1}^N (p_i + q_i z) \quad (2.5)$$

where $p_i > 0$, $q_i > 0$ and $k \in Z$ are discrete scale-space kernels.

Proof: The generating function of a kernel on the form (2.4) is $\varphi_{K_i^{(2)}}(z) = p_i + q_i z$. As convolution in the spatial domain corresponds to multiplication of generating functions Proposition 2.5 gives that

$$\varphi_K(z) = \varphi_{K_1^{(2)}}(z) \varphi_{K_2^{(2)}}(z) \dots \varphi_{K_N^{(2)}}(z) \quad (2.6)$$

is the generating function of a scale-space kernel. A constant scaling-factor C or a translation $\varphi_{\text{transl}}(z) = z^k$ cannot affect the number of local extrema. Therefore these factors can be multiplied onto $\varphi_K(z)$ without changing the scale-space properties. \square

Another way to express this result is as follows:

Proposition 2.7 (Sufficient criterion for scale-space kernels)

Let $c_{-m}, \dots, c_{-1}, c_0, c_1, \dots, c_n$ be the coefficients of a discrete kernel with finite support. Then a sufficient condition for the kernel to be a scale-space kernel is that all roots of the generating function

$$\varphi(z) = c_{-m}z^{-m} + \dots + c_{-1}z^{-1} + c_0 + c_1z + \dots + c_nz^n \quad (2.7)$$

are real and non-positive.

Proof: Let $k = -m$, $N = n + m$ in (2.5). If all roots of $\varphi(z)$ are real and negative then (2.5) in Proposition 2.6 must be the factorization of (2.7). \square

2.3.3 Positivity and Unimodality in the Fourier Domain

The Fourier transform of a symmetric sequence on the form (2.5) has some interesting properties. The most general generating function of such a sequence can be written as

$$\varphi_K(z) = c \prod_{\nu=1}^N (p_\nu + q_\nu z)(p_\nu + q_\nu z^{-1}) \quad (2.8)$$

Consider one factor $(p_\nu + q_\nu z)(p_\nu + q_\nu z^{-1})$. Its Fourier transform is

$$\psi_K(\theta) = \sum_{n=-\infty}^{\infty} K(n)e^{-in\theta} = \varphi_K(e^{i\theta}) = (p_\nu + q_\nu e^{i\theta})(p_\nu + q_\nu e^{-i\theta}) = p_\nu^2 + q_\nu^2 + 2p_\nu q_\nu \cos \theta \quad (2.9)$$

On the interval $[-\pi, \pi]$ this function is non-negative. It assumes its maximum value $(p_\nu + q_\nu)^2$ for $\theta = 0$ and its minimum value $(p_\nu - q_\nu)^2$ for $\theta = \pm\pi$. $\psi_K(\theta)$ is monotonically increasing on $[-\pi, 0]$ and monotonically decreasing on $[0, \pi]$, in other words unimodal. It is easy to show

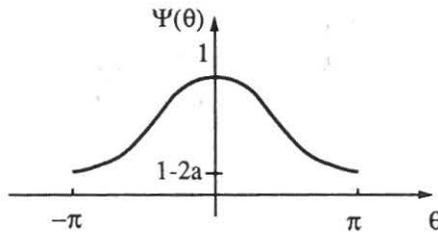


Figure 2.5: The Fourier transform of a (normalized) symmetric three-kernel with the coefficients $(a/2, 1 - a, a/2)$ is $\psi(\theta) = 1 - a(1 - \cos \theta)$. If $0 \leq a \leq 1/2$ this function is non-negative and unimodal on the interval $[-\pi, \pi]$. In the special case $a = 1/2$ the Fourier transform tends to zero at the end points of the interval.

that any finite product of non-negative increasing (decreasing) functions is also increasing (decreasing). Consequently, the Fourier transform of a symmetric kernel on the form (2.5) is non-negative and unimodal on the interval $[-\pi, \pi]$. In this section we will derive results showing that the Fourier transform of any symmetric scale-space kernel must possess these properties. The proofs, which sometimes are of a rather technical nature, can be skipped by the hasty reader without loss of continuity.

2.3.3.1 No Real Negative Eigenvalues of the Convolution Matrix

If the convolution transformation $f_{out} = K * f_{in}$ is represented on matrix form $\mathbf{f}_{out} = C\mathbf{f}_{in}$ a matrix with constant values along the diagonals $C_{i,j} = K(i - j)$ appears. Such a matrix is called a Toeplitz matrix. If this matrix has a real and negative eigenvalue then the corresponding kernel cannot be a scale-space kernel.

Proposition 2.8 (No real negative eigenvalues of the convolution matrix)

Let $K : Z \rightarrow R$ be a discrete kernel with finite support and filter coefficients $c_n = K(n)$. If for some dimension N the $N \times N$ convolution matrix

$$C^{(N)} = \begin{pmatrix} c_0 & c_{-1} & \cdots & c_{2-N} & c_{1-N} \\ c_1 & c_0 & c_{-1} & \cdots & c_{2-N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{N-2} & \cdots & c_1 & c_0 & c_{-1} \\ c_{N-1} & c_{N-2} & \cdots & c_1 & c_0 \end{pmatrix} \quad (2.10)$$

has a negative eigenvalue with a corresponding real eigenvector then K cannot be a scale-space kernel. Particularly, if the kernel is symmetric then all eigenvalues must be real and non-negative.

Proof: Because of Proposition 2.2 it is sufficient to study kernels having only non-negative filter coefficients. Assume that $C^{(N)}$ has a real negative eigenvalue for some dimension N and a corresponding real eigenvector \mathbf{v} . Let I_N be the index set $1..N$. Create an input signal f_{in} , which is equal to the components of \mathbf{v} for $x \in I_N$ and zero otherwise. Convolve this signal with the kernel. Then for $x \in I_N$ the values of $K * f_{in}$ will be equal to the corresponding components of $C^{(N)}\mathbf{v}$ (see Figure 2.6). As \mathbf{v} is an eigenvector with a negative

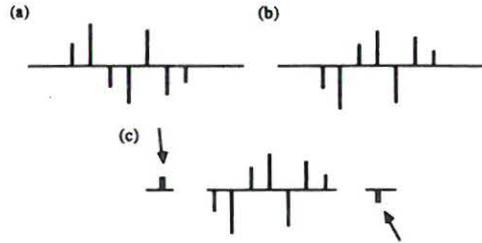


Figure 2.6: (a) The eigenvector \mathbf{v} . (b) The components of $C^{(N)}\mathbf{v}$ having indices $1..N$. (c) The components of $K * f_{in}$.

eigenvalue the components of $C^{(N)}\mathbf{v}$ and \mathbf{v} have opposite signs. This means that \mathbf{v} , $C^{(N)}\mathbf{v}$ and $K * f_{in}$ all have the same number of internal zero-crossings provided that we observe only the components in I_N .

The reversal of these components and the positivity of the filter coefficients guarantee that at least one additional zero-crossing will occur in the output signal. Let α denote the index of the first non-zero component of f_{in} . If $f_{in}(\alpha)$ is positive (negative) then due to the negative eigenvalue $K * f_{in}(\alpha)$ will be negative (positive). Since the filter coefficients are non-negative the first non-zero component of $K * f_{in}$ (at position β) will have the same sign as $f_{in}(\alpha)$, i.e. positive (negative). Consequently, we have found at least one additional zero-crossing in $K * f_{in}$ between these two positions (α and β). The same argument can be carried out at other end point producing another scale-space violation. This shows that K cannot be a scale-space kernel. \square

2.3.3.2 Positivity in the Frequency Domain

The eigenvalues of a Toeplitz matrix are closely related to the the Fourier transform of the corresponding sequence of coefficients, see e.g. Grenander [Gre58] or Gray [Gra72]. A theorem by Toeplitz [Toe11] relates the eigenvalues⁴ of an infinite Toeplitz matrix C with elements $C_{i,j} = c_{i-j}$ to the the values of the generating function associated with the sequence of filter weights. Assume that $\varphi(z) = \sum_{n=-\infty}^{\infty} c_n z^n$ is convergent in the ring $r < |z| < R$, where $0 < r < 1 < R$. Then the eigenvalues of C coincide with the set of complex values that $\varphi(z)$ assumes on the unit circle $|z| = 1$. This property allows us to derive an interesting corollary from Proposition 2.8.

Proposition 2.9 (Non-negative Fourier transform)

The Fourier transform $\psi_K(\theta) = \sum_{n=-\infty}^{\infty} K(n)e^{-in\theta}$ of a symmetric discrete scale-space kernel K with finite support is non-negative.

Proof: Let $\lambda_1^{(N)}$ denote the smallest eigenvalue of the convolution matrix of dimension N and let m denote the minimum value⁵ the Fourier transform ψ_K assumes on $[-\pi, \pi]$. As a consequence of a theorem by Grenander [Gre58] Section 5.2 p65 about the asymptotic distribution of eigenvalues of a finite Toeplitz matrix it follows that

$$\lim_{N \rightarrow \infty} \lambda_1^{(N)} = m \quad \lambda_1^{(N)} \geq m \quad (2.11)$$

If m is strictly negative then as $\lim_{N \rightarrow \infty} \lambda_1^{(N)} = m$ it follows that $\lambda_1^{(N)}$ will be negative for some sufficiently large N . According to Proposition 2.8 the kernel cannot be a scale-space kernel. \square

2.3.3.3 Unimodality in the Frequency Domain

If a linear transformation is to be regarded as a smoothing transformation it turns out to be necessary that the low frequency components are not suppressed more than the high frequency components. This means that the Fourier transform must not increase when the absolute value of the frequency increases. The occurring unimodality property is easiest to establish for circular convolution. In that case the convolution matrix becomes circulant⁶, which means that its eigenvalues and eigenvectors can be determined analytically.

Proposition 2.10 (Unimodal Fourier transform; wrap-around)

Let $\{c_n\}_{n=-\infty}^{\infty}$ be the filter coefficients of a symmetric discrete kernel with $c_n = 0$ if $|n| > N$. For all integers $M \geq N$ it is required that the transformation given by multiplication with the $(2M + 1) \times (2M + 1)$ symmetric circulant matrix $C_C^{(M)}$ (2.12), defined by $(C_C^{(M)})_{i,j} = c_{i-j}$ ($i, j = 0..M$) and circulant extension, should be a scale-space transformation. Then, necessarily the Fourier transform $\psi(\theta) = \sum_{n=-\infty}^{\infty} c_n e^{-in\theta}$ must be unimodal on $[-\pi, \pi]$.

⁴ λ is said to be an eigenvalue of an infinite matrix C if the matrix $C - \lambda I$ has no bounded inverse. I denotes the unit matrix.

⁵Due to the symmetry of the kernel, $\psi_K(\theta)$ assumes only real values. The minimum value exists, since $\psi(\theta)$ is a continuous function and the interval $[-\pi, \pi]$ is compact.

⁶In a circulant matrix each row is a circular shift of the previous row except for the first row which is a circular shift of the last row.

$$C_C^{(M)} = \begin{pmatrix} c_0 & c_1 & \cdots & c_N & & c_N & \cdots & c_1 \\ c_1 & c_0 & c_1 & & c_N & & \ddots & \vdots \\ \vdots & & \ddots & & & \ddots & & c_N \\ c_N & & & \cdot & & \ddots & & \\ & c_N & & \cdot & & & & c_N \\ & & \ddots & & \cdot & & & c_N \\ c_N & & \ddots & & \cdot & & \ddots & \vdots \\ \vdots & \ddots & & c_N & & & c_0 & c_1 \\ c_1 & \cdots & c_N & & c_N & \cdots & c_1 & c_0 \end{pmatrix} \quad (2.12)$$

Proof: The core in the proof is to show that if a kernel has a non-unimodal Fourier transform then there exists some low frequency component that disappears faster than some other high frequency component. By considering a signal which is a superposition of two such components will show that repeated application of the convolution operator will eventually lead to an increase in the number of local extrema when the low frequency component has died out and the high frequency component dominates, see also Figure 2.7.

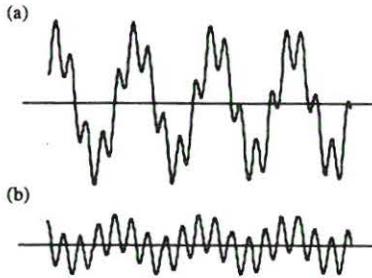


Figure 2.7: (a) Input signal consisting of a low frequency component of high amplitude and a high frequency component of low amplitude. (b) In the output signal the low frequency component has been suppressed while the high frequency component remains unchanged. As we see, additional zero-crossings have been introduced.

We will introduce a temporary definition. If \mathbf{x} is a vector of length L let $V(\mathbf{x})$ denote the number of zero-crossings in the sequence of components $x_1, x_2, \dots, x_L, x_1$. By verification one shows that the eigenvalues λ_m and eigenvectors \mathbf{v}_m of $C_C^{(M)}$ are

$$\lambda_m = \sum_{n=-M}^M c_n e^{-\frac{2\pi i m n}{2M+1}} = \sum_{n=-N}^N c_n e^{-\frac{2\pi i m n}{2M+1}} \quad (m = -M..0..M) \quad (2.13)$$

$$(\mathbf{v}_m)_k = \sin\left(\frac{2\pi m k}{2M+1}\right) \quad (m = -M..-1, k = -M..0..M) \quad (2.14)$$

$$(\mathbf{v}_m)_k = \cos\left(\frac{2\pi m k}{2M+1}\right) \quad (m = 0..M, k = -M..0..M)$$

We note that $V(\mathbf{v}_m)$ increases as $|m|$ increases. Further, the eigenvalues $\lambda_m = \psi\left(\frac{2\pi m}{2M+1}\right)$ of $C_C^{(M)}$ are uniformly sampled values of the Fourier transform and a larger value of $|m|$ corresponds to a larger absolute value of the argument ψ .

Now, assume that the Fourier transform is not unimodal. (Without loss of generality we can presuppose that ψ is non-negative on $[-\pi, \pi]$, because otherwise, according to Proposition 2.9, the kernel cannot be a scale-space kernel.) Then, as ψ is a continuous function of θ it is possible to find some sufficiently large integer \tilde{M} such that there exist $\theta_\alpha = \frac{2\pi\alpha}{2\tilde{M}+1}$ and $\theta_\beta = \frac{2\pi\beta}{2\tilde{M}+1}$ satisfying $\psi(\theta_\beta) > \psi(\theta_\alpha)$ for some integers $\beta > \alpha$ in $[0, \tilde{M}]$.

To summarize, $C_C^{(\tilde{M})}$ has eigenvalues $\lambda_\beta > \lambda_\alpha$ and corresponding eigenvectors with $V(\mathbf{v}_\beta) > V(\mathbf{v}_\alpha)$. We will show that this situation leads to a scale-space violation. The scale-space properties are not affected by a scaling factor. Therefore, we can equivalently study $B = \frac{1}{\lambda_\beta} C_C^{(\tilde{M})}$. For both eigenvectors we define the smallest and largest absolute values $v^{(absmin)}$ and $v^{(absmax)}$ by

$$v^{(absmin)} = \min_{k=1..N} |v_k| \quad ; \quad v^{(absmax)} = \max_{k=1..N} |v_k| \quad (2.15)$$

Let $\mathbf{x} = c\mathbf{v}_\alpha + \mathbf{v}_\beta$ where c is chosen large enough such that $V(\mathbf{x}) = V(\mathbf{v}_\alpha)$. This can always be achieved if $|c| v_\alpha^{(absmin)} > v_\beta^{(absmax)}$, since then the components of \mathbf{x} and \mathbf{v}_α will have pairwise same signs. ($v_\alpha^{(absmin)}$ will be strictly positive as all components of \mathbf{v}_α are non-zero.) Then consider $B\mathbf{x} = \frac{1}{\lambda_\beta}(c\lambda_\alpha\mathbf{v}_\alpha + \lambda_\beta\mathbf{v}_\beta)$ and study

$$B^k \mathbf{x} = c \left(\frac{\lambda_\alpha}{\lambda_\beta} \right)^k \mathbf{v}_\alpha + \mathbf{v}_\beta \quad (2.16)$$

For a fixed value of c we can always find a sufficiently large value of k such that $V(B^k \mathbf{x}) = V(\mathbf{v}_\beta)$. In a similar manner to above one verifies that the condition $|c| \left| \frac{\lambda_\alpha}{\lambda_\beta} \right|^k v_\alpha^{(absmax)} < v_\beta^{(absmin)}$ suffices. Consequently, $V(B^k \mathbf{x}) > V(\mathbf{x})$ which shows that the transformation induced by B^k is not a scale-space transformation. Therefore, B cannot be a scale-space kernel since at least one scale-space violation must have occurred in the series of k successive transformations. \square

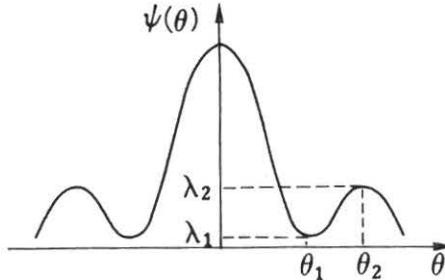


Figure 2.8: If the Fourier transform is not unimodal on $[-\pi, \pi]$, i.e. if there exist $\theta_2 > \theta_1$ in $[0, \pi]$ such that $\psi(\theta_2) > \psi(\theta_1)$ then the corresponding transformation cannot be a scale-space transformation.

The result can be extended to comprise non-circular convolution as well. The idea behind the proof is to construct an input signal consisting of several periods of the signal leading to a scale-space violation in the proof of Proposition 2.10. Then, the convolution effect on the “interior” periods will be identical to effect on one period by circular convolution. If the signal consists of a sufficient number of periods the boundary effects will be negligible

The necessity of this property can also be shown directly from the positivity and unimodality properties in the spatial and Fourier domains. Observe that the usual binomial kernel with the coefficients $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ is actually a boundary case.

At this moment one could ask one-self if these results can be generalized to hold for kernels with arbitrary numbers of non-zero filter coefficients. I.e. if all discrete scale-space kernels with finite support have a generating function on the form (2.5). This question will be answered in the next section.

2.4 Kernel Classification

Until now we have postulated an axiom in terms of local extrema or equivalently zero-crossings and investigated some of its consequences for signal transformations expressed as linear convolution with a shift-invariant kernel. We have seen that the sequence of filter coefficients must be positive and unimodal and that its sum should be convergent. For symmetric kernels the Fourier transform must be positive and unimodal on $[-\pi, \pi]$.

In this section we will perform a complete characterization of the scale-space kernels. We have studied the literature and seen that several interesting results can be derived from the theory of total positivity. The proofs of the important theorems are sometimes of a rather complicated nature for a reader with an engineering background. We will not burden the presentation with them but give a brief background to the theory and a few summarizing results without proof.

The pioneer in the subject of variation-diminishing transforms was I.J. Schoenberg. He studied the subject in a series of papers from 1930 to 1953 [Sch30, Sch48, Sch53]. Later the theory of total positivity has been covered in a monumental monograph by Karlin [Kar68]. A recent paper by Ando [And87] reviews the field using skew-symmetric vector products and Schur complements of matrices as major tools. The questions issued in this treatment constitute a new application of these not too well-known but very powerful results.

2.4.1 Background

Consider first a general linear transformation of discrete signal where the kernel does not need to be shift-invariant.

$$f_{out}(x) = \sum_{y=-\infty}^{\infty} K(x, y) f_{in}(y) \quad (x \in Z) \quad (2.20)$$

Two notions of sign changes in vectors will be used, see e.g. Karlin [Kar68] or Ando [And87]. Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ be a vector of n real numbers. We denote by $V^-(\mathbf{x})$ the (minimum) number of sign changes obtained in the sequence x_1, x_2, \dots, x_n if all zero terms are deleted and by $V^+(\mathbf{x})$ the maximum number of sign changes possible in the sequence x_1, x_2, \dots, x_n if each zero value is allowed to be replaced by either $+1$ or -1 . We use a special convention saying that the number of sign changes in the null vector is -1 .

The interesting sequences and kernels will be defined in terms of minors of the transformation matrix. Given a kernel $K : X \times Y \rightarrow R$ we form minors of arbitrary order r by selections of $x_1 < x_2 < \dots < x_r$ from X and of $y_1 < y_2 < \dots < y_r$ from Y . The determinant of the resulting matrix with components $\{K(x_i, y_j)\}_{i,j=1..r}$ is called "a minor of order r "

and denoted by

$$K \begin{pmatrix} x_1, x_2, \dots, x_r \\ y_1, y_2, \dots, y_r \end{pmatrix} = \begin{vmatrix} K(x_1, y_1) & K(x_1, y_2) & \cdot & \cdot & K(x_1, y_r) \\ K(x_2, y_1) & K(x_2, y_2) & \cdot & \cdot & K(x_2, y_r) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ K(x_r, y_1) & K(x_r, y_2) & \cdot & \cdot & K(x_r, y_r) \end{vmatrix} \quad (2.21)$$

A basic concept when dealing with variation-diminishing properties is sign-regularity:

Sign-regularity:

A discrete kernel $K : Z \times Z \rightarrow R$ is said to be sign-regular (SR_∞) if all its r -order minors have same sign for every order r from 1 through ∞ , i.e. if there exists a sequence of constants $\varepsilon_1, \varepsilon_2, \dots$ each $+1$ or -1 such that

$$\varepsilon_r K \begin{pmatrix} x_1, x_2, \dots, x_r \\ y_1, y_2, \dots, y_r \end{pmatrix} \geq 0 \quad (2.22)$$

for all choices of $x_1 < x_2 < \dots < x_r$ and $y_1 < y_2 < \dots < y_r$ from Z .

In other words, sign-regularity means that it is impossible to find two minors of same order having opposite signs. If strict inequality holds for all r then K is said to be strictly sign-regular (SSR_∞). General linear transformations (not necessarily shift-invariant) possessing variation-diminishing properties in the sense that they never increase the number of sign changes in a vector, can be fully characterized in terms of sign-regularity.

Classification of general variation-diminishing transformations I:

Let A be an $n \times m$ real matrix with $n \geq m$. Then the linear map A from R^m to R^n diminishes variations in sign in the sense that

$$V^+(Ax) \leq V^-(x) \quad \text{for all } x \in R^m \ x \neq 0 \quad (2.23)$$

if and only if A is strictly sign-regular (SSR_∞).

The original proof of this powerful theorem, forming the foundation of the theory for variation-diminishing transforms, can be found in Schoenberg [Sch53]. Ando [And87] derives it using skew-symmetric vector products. Another formulation is possible [And87] if A is known to be of full rank.

Classification of general variation-diminishing transformations II:

Let A be an $n \times m$ real matrix of rank m . Then

$$V^-(Ax) \leq V^-(x) \quad (2.24)$$

holds for all $x \in R^m$ ($x \neq 0$) if and only if A is sign-regular (SR_∞).

We note that the condition (2.24) is equivalent to the formulation we expressed in Definition 2.1. Consequently, sign-regularity and full rank are the necessary and sufficient conditions for a kernel to be a potential scale-space kernel. A narrower class of transformations is obtained if all minors are required to be non-negative, see e.g. Karlin [Kar68].

Total positivity:

A discrete kernel $K : Z \times Z \rightarrow R$ is said to be totally positive (TP_∞) if all its minors are nonnegative; i.e. if

$$K \begin{pmatrix} x_1, x_2, \dots, x_p \\ y_1, y_2, \dots, y_p \end{pmatrix} \geq 0 \quad (2.25)$$

$$x_1 < x_2 < \dots < x_p; \quad y_1 < y_2 < \dots < y_p; \quad p = 1, 2, \dots, \infty$$

An important subclass of totally positive kernels appears if the discrete kernel is required to be shift-invariant i.e. if $K(x, y)$ can be written as $k(x - y) = c_{x-y}$.

Pólya frequency sequence:

A sequence $\{c_n\}_{n=-\infty}^{\infty}$ is said to be a Pólya frequency sequence if all minors of the infinite Toeplitz matrix

$$C = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_0 & c_{-1} & c_{-2} & \cdot & \cdot \\ \cdot & \cdot & c_1 & c_0 & c_{-1} & \cdot & \cdot \\ \cdot & \cdot & c_2 & c_1 & c_0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad (2.26)$$

are non-negative.

The importance of the Pólya frequency sequences becomes apparent when we require that the generating function converges, which for instance holds if the sum of the filter coefficients is convergent.

Normalized Pólya frequency sequence:

A Pólya frequency sequence $\{c_n\}_{n=-\infty}^{\infty}$ having a generating function $\varphi(z) = \sum_{n=-\infty}^{\infty} c_n z^n$ which converges in an annulus $r < |z| < R$ ($0 < r < 1 < R$) such that $\varphi(z) \neq 0$ is called a normalized Pólya frequency sequence.

According to a theorem by Schoenberg [Sch48] sign-regularity combined with the Toeplitz structure implies total positivity. Consequently,

Classification of variation-diminishing convolution transformations:

The convolution transformation

$$f_{out}(x) = \sum_{n=-\infty}^{\infty} c_n f_{in}(x - n)$$

is variation-diminishing i.e.

$$V^-(f_{out}) \leq V^-(f_{in})$$

holds for all f_{in} if and only if the sequence of filter coefficients $\{c_n\}_{n=-\infty}^{\infty}$ is a normalized Pólya frequency sequence.

In other words, every shift-invariant discrete scale-space kernel corresponds to a normalized Pólya frequency sequence.

There exists a remarkably explicit characterization theorem for the generating function of a PF_∞ -sequence. It has been proved in several steps by Edrei and Schoenberg, see [Sch53] or [Kar68].

Classification of Pólya frequency sequences:

An infinite sequence $\{c_n\}_{n=-\infty}^{\infty}$ is a Pólya frequency sequence if and only if its generating function $\varphi(z) = \sum_{n=-\infty}^{\infty} c_n z^n$ is of the form

$$\varphi(z) = c z^k e^{(q_{-1}z^{-1} + q_1 z)} \prod_{i=1}^{\infty} \frac{(1 + \alpha_i z)(1 + \delta_i z^{-1})}{(1 - \beta_i z)(1 - \gamma_i z^{-1})} \quad (2.27)$$

$$c > 0; \quad k \in \mathbb{Z} \quad q_{-1}, q_1, \alpha_i, \beta_i, \gamma_i, \delta_i \geq 0; \quad \sum_{i=1}^{\infty} (\alpha_i + \beta_i + \gamma_i + \delta_i) < \infty$$

The sequence $\{c_n\}_{n=-\infty}^{\infty}$ is normalized if and only if it in addition holds that $\beta_i < 1$ and $\gamma_i = 1$, see [Kar68].

2.4.2 Classification of Discrete Scale-Space Kernels

The results from the previous section allow us to completely classify which kernels are scale-space kernels. To summarize, we can state two criteria; one in terms of minors of the convolution matrix and one in terms of the generating function of the convolution kernel.

Theorem 2.14 (Classification of discrete scale-space kernels I)

A discrete kernel $K : \mathbb{Z} \rightarrow \mathbb{R}$ is a scale-space kernel if and only if the corresponding sequence of filter coefficients $\{K(n)\}_{n=-\infty}^{\infty}$ is a normalized Pólya frequency sequence, i.e. if all minors of the infinite matrix

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & K(0) & K(-1) & K(-2) & \cdot & \cdot \\ \cdot & \cdot & K(1) & K(0) & K(-1) & \cdot & \cdot \\ \cdot & \cdot & K(2) & K(1) & K(0) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad (2.28)$$

are non-negative.

Theorem 2.15 (Classification of discrete scale-space kernels II)

An discrete kernel $K : \mathbb{Z} \rightarrow \mathbb{R}$ is a discrete scale-space kernel if and only if its generating function $\varphi_K(z) = \sum_{n=-\infty}^{\infty} K(n)z^n$ is of the form

$$\varphi_K(z) = c z^k e^{(q_{-1}z^{-1} + q_1 z)} \prod_{i=1}^{\infty} \frac{(1 + \alpha_i z)(1 + \delta_i z^{-1})}{(1 - \beta_i z)(1 - \gamma_i z^{-1})} \quad (2.29)$$

$$c > 0; \quad k \in \mathbb{Z}; \quad q_{-1}, q_1, \alpha_i, \beta_i, \gamma_i, \delta_i \geq 0$$

$$\beta_i, \gamma_i < 1; \quad \sum_{i=1}^{\infty} (\alpha_i + \beta_i + \gamma_i + \delta_i) < \infty$$

Note that we get the Fourier transform of the kernel by replacing z by $e^{-i\theta}$.

The product structure of this expression corresponds to the previously mentioned property that if K_a and K_b are scale-space kernels then also $K_a * K_b$ is a scale-space kernel. The meanings of the leading factors C and z^k are just rescaling and translation. In $(1 + \alpha_i z)$ and $(1 + \delta_i z^{-1})$ we recognize rewritten versions of the generating functions of two-kernels. The factors in the denominator are Taylor expansions of geometric series, which correspond to moving average processes of the forms $f_{out}(x) = f_{in}(x) + \beta_i f_{out}(x - 1)$ and $f_{out}(x) = f_{in}(x) + \gamma_i f_{out}(x + 1)$. The exponential factor describes infinitesimal smoothing. Its interpretation will become clearer in the next section, when we derive the discrete scale-space with a continuous scale parameter. To conclude, this classification implies that

Corollary 2.16 (Primitive discrete smoothing transformations)

For discrete signals $Z \rightarrow R$ there are exactly five primitive types of linear and shift-invariant smoothing transformations, of which the last two ones are trivial:

- *two-point weighted average or generalized binomial smoothing*

$$f_{out}(x) = f_{in}(x) + \alpha_i f_{in}(x - 1) \quad (\alpha \geq 0) \quad (2.30)$$

$$f_{out}(x) = f_{in}(x) + \delta_i f_{in}(x + 1) \quad (\delta \geq 0) \quad (2.31)$$

- *moving average or first order recursive filtering*

$$f_{out}(x) = f_{in}(x) + \beta_i f_{out}(x - 1) \quad (\beta \geq 0) \quad (2.32)$$

$$f_{out}(x) = f_{in}(x) + \gamma_i f_{out}(x + 1) \quad (\gamma \geq 0) \quad (2.33)$$

- *infinitesimal smoothing or diffusion smoothing, see Section 2.5.2 for further explanation.*
- *rescaling*
- *translation*

Moreover we have that

Corollary 2.17 (Decomposition property of scale-space kernels)

A convolution transformation is a smoothing transformation with discrete scale-space properties if and only if it can be decomposed into primitive transformations, which are all smoothing transformations possessing scale-space properties.

This means that the inverse statement of Lemma 2.4 is true and that once a non-smoothing transformation has been performed, that step it is impossible to fully compensate for by further smoothing. Of course, one could in general expect that such further smoothing leads to a signal with a smaller number of local extrema. However, there will always exist some signals for which this is not possible.

For kernels with finite support q_{-1}, q_1, β_i and γ_i must be zero and the infinite product must be replaced with a finite one. Then, the generating function will be reduced to $\varphi_K(z) = c z^k \prod_{i=1}^N (1 + \alpha_i z)(1 + \delta_i z^{-1})$, for some finite N , which except for rescaling and translation is the generating function of the class of generalized binomial kernels in Proposition 2.5 and 2.6. Hence,

Theorem 2.18 (Classification of discrete scale-space kernels with finite support)
The kernels on the form $\ast_{i=1}^n K_i^{(2)}$, with $K_i^{(2)}$ according to (2.4), are (except for rescaling and translation) the only discrete scale-space kernels with finite support.

An immediate consequence of this result is that *convolution with a finite scale-space kernel can be decomposed into convolution with kernels having two strictly positive consecutive filter coefficients.* This gives further emphasis to the statement that the generalized binomial kernels are, except for a trivial translation, the only discrete scale-space kernels with finite support. In the symmetric case the generating function can be further reduced to $\varphi_K(z) = c \prod_{i=1}^N (1 + \alpha_i z)(1 + \alpha_i z^{-1})$, which shows that

Corollary 2.19 (Symmetric discrete scale-space kernels with finite support)
Every symmetric discrete scale-space kernel can be decomposed into convolutions with symmetric three-kernels of type

$$(a_i, b_i, a_i) \quad \text{where } b_i \geq 2a_i > 0 \quad (2.34)$$

In other words, every symmetric discrete scale-space kernel with finite support has a Fourier transform of the form

$$\psi_K(\theta) = \prod_{i=1}^N (b_i + 2a_i \cos(\theta)) \quad (2.35)$$

The representation (2.29), which gives a catalogue of all one-dimensional discrete smoothing kernels, can sometimes be very convenient for further analysis. For example, starting from (2.29) it is almost trivial to show that the Fourier transform of a symmetric discrete scale-space kernel is unimodal and non-negative on the interval $[-\pi, \pi]$. Due to the symmetry we have $q_{-1} = q_1$, $\alpha_\nu = \delta_\nu$ and $\beta_\nu = \gamma_\nu$. As a first step one replaces z with $e^{-i\theta}$ (which gives the Fourier transform) and shows that each one of the factors $e^{(q_{-1}z^{-1} + q_1z)}$, $(1 + \alpha_\nu z)(1 + \delta_\nu z^{-1})$ and $((1 - \beta_\nu z)(1 - \gamma_\nu z^{-1}))^{-1}$ is a non-negative and unimodal function of θ on $[-\pi, \pi]$. The remaining details are left to the reader.

2.5 Axiomatic Scale-Space Construction

2.5.1 Discrete Scale-Space with Discrete Scale Parameter

With the classification result from the previous section in mind an apparent way to get a multi-scale representation of a discrete signal f is by defining a set of discrete functions L_i ($i = 0..n$) where $L_0 = f$ and each coarser level is calculated by convolution from the previous one $L_i = K_{i \leftarrow i-1} \ast L_{i-1}$ ($i = 1..n$). The kernels $K_{i \leftarrow i-1}$ should be appropriately selected scale-space kernels corresponding to suitable amounts of blurring. The scale-space condition for each kernel guarantees that signals at coarser levels of scale (larger value of i) do not contain more structure than signals at finer levels of scale. This leads to a so-called sampled scale-space with a *discrete* scale parameter. Combined with a sub-sampling operator it provides a possible theoretical basis for the pyramid representations⁷. However, one problem arises. How should one select the kernels/scale-levels *a priori* in order to achieve a sufficiently dense sampling in scale?

⁷Note that when dealing with pyramids there are other problems arising due to the lower number of grid points, aliasing and the fixed scale sampling that might influence the design criteria. Those issues are not covered by this treatment.

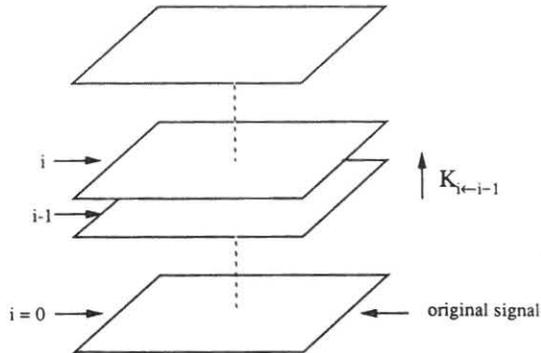


Figure 2.9: Given the classification of discrete scale-space kernels it is straightforward, at least in principle, to construct a scale-space representation associated with a discrete scale parameter: Start from the original signal and select a set of kernels $K_{i+1 \leftarrow i}$, each one describing the transformation from a scale level i to the next coarser level $i + 1$, where every such kernel should be a discrete scale-space kernel. Then one is guaranteed that any coarser level of scale j does not contain more local extrema any a finer level of scale i provided that $j \geq i$. However, there is still one problem that needs to be solved. How should the kernels be selected in order to achieve an appropriate sampling in scale?

2.5.2 Discrete Scale-Space with Continuous Scale Parameter

The goal in this section is to tie together scale-space kernels corresponding to different degrees of smoothing in a systematic manner such that a *continuous* scale parameter can be introduced. The concept of a continuous scale parameter is of considerable importance, since we will no longer be locked to fixed pre-determined discrete levels of scale. It allows us to defocus signals with an arbitrary amount of blurring, which will certainly make it easier to locate and trace events in scale-space. Of course, it is impracticable to generate the representations at all levels of scale in a real implementation. However, the important idea is that, in contrast to the pyramid representations where the scale levels are fixed, with a continuous scale parameter the scale-space representation at *any* level of scale can be calculated if desired.

We will not consider the question about how to choose a suitable set of scale levels in a practical case. Imagine for instance that we want to trace events, like local extrema, zero-crossings, edges [Ber87] or convex and concave regions, as the blurring proceeds in scale-space. In order to analyze scale-space behaviour, the continuum of multi-scale representations must be sampled at some levels of scale. It is certainly a non-trivial problem to make an appropriate selection of these levels, and it seems plausible that the sampling rate along the scale direction should depend upon the signal under study. If in some scale interval the representation varies relatively smoothly we should be able to allow a larger scale step than if it were strongly varying. We will thus be lead to methods that automatically regulate the scale step, based on the local structure of the signal as function of the spatial and scale coordinates, compare also with the drift velocity estimates in Chapter 6.1.1 and the linking algorithm across scales in Chapter 7.2. The point with a scale-space with a continuous scale parameter is that it provides a theoretical framework for the development of such algorithms, in which the scale steps can be varied arbitrarily. We do not need to select any set of scale levels *in advance*, but can leave the decision open to the actual situation.

We start from the axioms given in Section 2 and postulate that the scale-space should

be generated by convolution with a one-parameter family of kernels, i.e. $L(x; 0) = f(x)$ and

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t)f(x - n) \quad (t > 0) \quad (2.36)$$

This form of the smoothing formula reflects the requirements about linear shift-invariant smoothing and a continuous scale parameter. The amount of structure in a signal must not increase with scale. This means that for any $t_2 > t_1$ the number of local extrema in $L(x; t_2)$ must not exceed the number of local extrema in $L(x; t_1)$. Particularly, by setting t_1 to zero we realize that each $T(\cdot; t)$ must be a scale-space kernel.

In order to simplify the analysis a semi-group requirement $T(\cdot; s) * T(\cdot; t) = T(\cdot; s + t)$ is imposed on the family of kernels. This property makes it possible to calculate the representation $L(\cdot; t_2)$ at a coarser level t_2 from the representation $L(\cdot; t_1)$ at a finer level t_1 ($t_2 > t_1$) by convolution with a kernel *from the one-parameter family*. In summary,

$$\begin{aligned} L(\cdot; t_2) &= \{\text{definition}\} = T(\cdot; t_2) * f = \{\text{semi-group}\} = \\ &= (T(\cdot; t_2 - t_1) * T(\cdot; t_1)) * f = \{\text{associativity}\} = \\ &= T(\cdot; t_2 - t_1) * (T(\cdot; t_1) * f) = \{\text{definition}\} = T(\cdot; t_2 - t_1) * L(\cdot; t_1) \end{aligned} \quad (2.37)$$

As each $T(\cdot; t)$ is required to be a scale-space kernel, the semi-group property ensures that the scale-space property holds between any two levels of scale. It also means that all scale levels will be treated in a uniform manner.

We will show below that the conditions mentioned, combined with a normalization criterion $\sum_{n=-\infty}^{\infty} T(n; t) = 1$ and a symmetry constraint $T(-n; t) = T(n; t)$, determine the family of kernels up to a positive scaling parameter⁸ α . One obtains,

$$T(n; t) = e^{-\alpha t} I_n(\alpha t) \quad (2.38)$$

where I_n are the modified Bessel functions of integer order. These functions with real arguments are except for an alternating sign sequence equal to the ordinary Bessel functions J_n of integer order with purely imaginary arguments.

$$I_n(t) = I_{-n}(t) = (-1)^n J_n(it) \quad (n \geq 0, t > 0) \quad (2.39)$$

Theorem 2.20 (Scale-space for discrete signals; Necessity and sufficiency)

Given any one-dimensional signal $f : Z \rightarrow R$ let $L : Z \times R_+ \rightarrow R$ be a one-parameter family of functions defined by $L(x; 0) = f(x)$ ($x \in Z$) and

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t)f(x - n) \quad (2.40)$$

*($x \in Z, t > 0$), where $T : Z \times R_+ \rightarrow R$ is a one-parameter family of symmetric functions satisfying the semi-group property $T(\cdot; s) * T(\cdot; t) = T(\cdot; s + t)$ and the normalization criterion $\sum_{n=-\infty}^{\infty} T(n; t) = 1$. For all signals f it is required that if $t_2 > t_1$ then the number of local extrema (zero-crossings) in $L(x; t_2)$ must not exceed the number of local extrema (zero-crossings) in $L(x; t_1)$. Then necessarily (and sufficiently),*

$$T(n; t) = e^{-\alpha t} I_n(\alpha t) \quad (2.41)$$

for some non-negative real α , where I_n are the modified Bessel functions of integer order.

⁸For simplicity, the parameter α , which only affects the scaling of the scale parameter, will be set to 1 after the end of this section.

Proof: As mentioned above, every kernel $T(n; t)$ must be a scale-space kernel. A theorem by Karlin [Kar68] states that the only semi-group of normalized Pólya frequency sequences has a generating function of the form

$$\varphi(z) = e^{t(az^{-1}+bz)} \quad (2.42)$$

where $t > 0$ and $a, b \geq 0$. This result, which forms the basis of the proof can be easily understood from Theorem 2.15. If a family $h(\cdot; t)$ possesses the semi-group property $h(\cdot; s) * h(\cdot; t) = h(\cdot; s+t)$ then its generating function must necessarily obey the relation $\varphi_{h(\cdot; s)}\varphi_{h(\cdot; t)} = \varphi_{h(\cdot; s+t)}$ for all non-negative s and t . This excludes the factors z^k , $(1 + \alpha_i z)$, $(1 + \delta_i z^{-1})$, $(1 - \beta_i z)$ and $(1 - \gamma_i z^{-1})$ from (2.27). What remains are the constant and the exponential factors. The argument of the exponential factor must also be linear in t in order to fulfill the adding property of the scale parameters of the kernels under convolution.

Due to the symmetry the generating function must satisfy $\varphi_h(z^{-1}) = \varphi_h(z)$, which in our case leads to $a = b$. For simplicity, let $a = b = \frac{\alpha}{2}$, and we get the generating function for the modified Bessel functions of integer order, see Abramowitz and Stegun [Abr64] (9.6.33).

$$\varphi_I(z) = e^{\frac{\alpha t}{2}(z^{-1}+z)} = \sum_{n=-\infty}^{\infty} I_n(\alpha t) z^n \quad (2.43)$$

We obtain a normalized kernel if we let $T : Z \times R_+ \rightarrow R$ be defined by $T(n; t) = e^{-\alpha t} I_n(\alpha t)$. Set z to 1 in the generating function $e^{\frac{\alpha t}{2}(z^{-1}+z)} = \sum_{n=-\infty}^{\infty} I_n(\alpha t) z^n$. Then it follows that $\sum_{n=-\infty}^{\infty} I_n(\alpha t) = e^{\alpha t}$, which means that $\sum_{n=-\infty}^{\infty} T(n; t) = 1$. The semi-group property is trivially preserved after normalization. \square

This theorem, which is one of the main results of this chapter, provides us with an explicit controlled method to preserve structure in the spatial domain as we let a discrete signal erode by smoothing it to coarser level of scales. The kernel $T(n; t) = e^{-\alpha t} I_n(\alpha t)$ possesses similar properties in the discrete case as those who make the ordinary Gaussian kernel special in the continuous case. Therefore it is natural to refer to it as *the discrete analog of the Gaussian kernel*, see also Norman [Nor60].

Definition 2.2 (Discrete analog of the Gaussian kernel)

The kernel $T : Z \times R_+ \rightarrow R$ given by $T(n; t) = e^{-\alpha t} I_n(\alpha t)$ is called the discrete analog of the Gaussian kernel, or shorter, the discrete Gaussian.

2.5.2.1 Properties of the Discrete Analog of the Gaussian Kernel

We will now point out some elementary properties of this kernel. In the special case $t = 0$ it holds that

$$I_n(0) = I_{-n}(0) = \delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (n \geq 0) \quad (2.44)$$

which means that $T(\cdot; 0) = \delta(\cdot)$ and the convolution expression (2.40) with T according to (2.41) is valid for $t = 0$ as well. Observe that when the scale parameter tends to zero the continuous Gaussian kernel tends to the continuous delta function while the discrete analog of the Gaussian kernel instead tends to the discrete delta function.

For large t on the other hand it holds that the discrete analog of the Gaussian kernel approaches the continuous Gaussian. This can be understood by studying an asymptotic

expression for the modified Bessel functions for large t , see Abramowitz and Stegun [Abr64] (9.7.1).

$$I_n(t) = \frac{e^t}{\sqrt{2\pi t}} \left(1 - \frac{4n^2 - 1}{8t} + O\left(\frac{1}{t^2}\right) \right) \quad (2.45)$$

which shows that

$$T(n; t) - g(n; t) = e^{-t} I_n(t) - \frac{1}{\sqrt{2\pi t}} e^{-\frac{n^2}{2t}} = \frac{1}{\sqrt{2\pi t}} \left(\frac{1}{8t} + O\left(\frac{1}{t^2}\right) \right) \quad (2.46)$$

If the relation (2.43) is multiplied with the factor e^{-t} and if z is replaced with $e^{-i\theta}$ one gets the analytical expression for the Fourier transform of $T(n; t)$.

Proposition 2.21 (Fourier transform of the discrete Gaussian kernel)

The Fourier transform of the kernel $T(n; t) = e^{-\alpha t} I_n(\alpha t)$ is

$$\psi_T(\theta) = \sum_{n=-\infty}^{\infty} T(n; t) e^{-in\theta} = e^{\alpha t (\cos\theta - 1)} \quad (2.47)$$

For completeness, we give the variance of this kernel as well

Proposition 2.22 (Variance of the discrete analog of the Gaussian kernel)

The variance of the kernel $T(n; t) = e^{-\alpha t} I_n(\alpha t)$ is

$$\sum_{n=-\infty}^{\infty} n^2 T(n; t) = t \quad (2.48)$$

Proof: This can be easily shown from a recurrence relation for the modified Bessel functions,

$$I_{n-1}(t) - I_{n+1}(t) = \frac{2n}{t} I_n(t) \quad (2.49)$$

see e.g. Abramowitz and Stegun [Abr64], and the normalization condition. We have

$$\sum_{n=-\infty}^{\infty} n^2 T(n; t) = \sum_{n=-\infty}^{\infty} n^2 e^{-t} I_n(t) = \sum_{n=-\infty}^{\infty} n^2 e^{-t} \frac{t}{2n} (I_{n-1}(t) - I_{n+1}(t)) = \quad (2.50)$$

$$\begin{aligned} \frac{te^{-t}}{2} \left(\sum_{n=-\infty}^{\infty} n I_{n-1}(t) - \sum_{n=-\infty}^{\infty} n I_{n+1}(t) \right) &= \frac{te^{-t}}{2} \left(\sum_{m=-\infty}^{\infty} (m+1) I_m(t) - \sum_{m=-\infty}^{\infty} (m-1) I_m(t) \right) = \\ \frac{te^{-t}}{2} \sum_{m=-\infty}^{\infty} (m+1 - m+1) I_m(t) &= te^{-t} \sum_{m=-\infty}^{\infty} I_m(t) = t \sum_{m=-\infty}^{\infty} T(m; t) = t \quad (2.51) \end{aligned}$$

□

Compare with the variance of the continuous Gaussian kernel, which is $\sigma^2 = t$. All moments of odd order are of course zero due to symmetry.

2.5.3 Equivalent Formulation for Continuous Signals

If similar arguments are applied in the continuous case we obtain the Gaussian kernel. In order to give a background to the analysis, we will first briefly review some important theorems from the theory of variation-diminishing convolution transformations for continuous signals. Then we will use those results to give a new and equivalent formulation of the scale-space for continuous signals.

2.5.3.1 Background

Let $S^-(f)$ denotes the number of sign changes in a function f defined by

$$S^-(f) = \sup V^-(f(t_1), f(t_2), \dots, f(t_m)) \quad (2.52)$$

where the supremum is extended over all sets $t_1 < t_2 < \dots < t_m$ ($t_i \in R$), m is arbitrary but finite and $V^-(\mathbf{x})$ denotes the number of sign changes in a vector \mathbf{x} defined in Section 2.4.1. The transformation

$$f_{out}(\eta) = \int_{\xi=-\infty}^{\infty} f_{in}(\eta - \xi) dG(\xi) \quad (2.53)$$

where G is a distribution function, is said to be variation-diminishing if

$$S^-(f_{out}) \leq S^-(f_{in}) \quad (2.54)$$

holds for all continuous and bounded f_{in} . The continuous correspondence to Pólya frequency sequences is called Pólya frequency functions. Also this concept is defined in terms of total positivity and shift invariance, see e.g. Karlin [Kar68].

Total positivity (continuous case):

A continuous kernel $K(x, y) : R \times R \rightarrow R$ is said to be totally positive (TP_∞) if all minors, of every order r from 1, 2 to infinity, are non-negative, i.e. if there for all choices of $x_1 < x_2 < \dots < x_r$ and $y_1 < y_2 < \dots < y_r$ from R holds that

$$K \begin{pmatrix} x_1, x_2, \dots, x_r \\ y_1, y_2, \dots, y_r \end{pmatrix} \geq 0 \quad (2.55)$$

$$x_1 < x_2 < \dots < x_r; \quad y_1 < y_2 < \dots < y_r; \quad r = 1, 2, \dots, \infty$$

Pólya frequency functions:

A function $k : R \rightarrow R$ is said to be a Pólya frequency function if the function $K : R \times R \rightarrow R$ given by $K(x, y) = k(x - y)$ is totally positive.

The variation-diminishing property of continuous convolution transformations on the form (2.53) can be completely characterized in terms of Pólya frequency functions. The following results are due to Schoenberg [Sch50], see also Hirschmann and Widder [Hir55] or Karlin [Kar68].

Classification of continuous variation-diminishing transformations I:

The transformation (2.53) is variation-diminishing if and only if G is either, up to a sign change, a cumulative Pólya frequency function

$$G(t) = \epsilon \int_{u=-\infty}^t k(u) du \quad (2.56)$$

where $\epsilon = \pm 1$ and $k(u)$ is a Pólya frequency function, or else G is a step function with only one jump.

Classification of continuous variation-diminishing transformations II:

The transformation (2.53) is variation-diminishing if and only if G has a bilateral Laplace-Stieltjes transform of the form

$$\int_{\xi=-\infty}^{\infty} e^{-s\xi} dG(\xi) = C e^{\gamma s^2 + \delta s} \prod_{i=1}^{\infty} \frac{e^{a_i s}}{1 + a_i s} \quad (-c < \operatorname{Re}(s) < c) \quad (2.57)$$

for some $c > 0$, where $C \neq 0$, $\gamma \geq 0$, δ and a_i are real and $\sum_{i=1}^{\infty} a_i^2$ is convergent.

Interpreted in the spatial domain, these results imply that for continuous signals there are four primitive types of linear and shift-invariant smoothing transformations; convolution with the *Gaussian kernel*,

$$h(\xi) = e^{-\gamma\xi^2} \quad (2.58)$$

convolution with the *truncated exponential functions*,

$$h(\xi) = \begin{cases} e^{-|\lambda|\xi} & \xi \geq 0 \\ 0 & \xi < 0 \end{cases} \quad h(\xi) = \begin{cases} e^{|\lambda|\xi} & \xi \leq 0 \\ 0 & \xi > 0 \end{cases} \quad (2.59)$$

as well as trivial *translation* and *rescaling*. Moreover, it means that a shift-invariant linear transformation is a smoothing operation if and only if it can be decomposed into these primitive operations.

2.5.3.2 Continuous Scale-Space with Continuous Scale Parameter

These results show that the Pólya frequency functions are the natural functions to start from when defining a scale-space representation for continuous signals, or equivalently, that *the Pólya frequency functions are the continuous scale-space kernels*. If again a semi-group requirement and a symmetry constraint are imposed on these kernels the Gaussian kernel will remain as the only candidate.

Theorem 2.23 (Scale-space for continuous signals; Necessity and sufficiency)

Given any one-dimensional continuous signal $f : R \rightarrow R$ let $L : R \times R_+ \rightarrow R$ be a one-parameter family of functions defined by $L(\cdot; 0) = f(\cdot)$ and

$$L(x; t) = \int_{\xi=-\infty}^{\infty} g(\xi; t)f(x - \xi)d\xi \quad (2.60)$$

($x \in R, t > 0$), where $g : R \times R_+ \rightarrow R$ is a one-parameter family of symmetric functions satisfying the semi-group property $g(\cdot; s) * g(\cdot; t) = g(\cdot; s + t)$ and the normalization criterion $\int_{\xi=-\infty}^{\infty} g(\xi; t)d\xi = 1$. For all signals f it is required that if $t_2 > t_1$ then the number of local extrema⁹ (zero-crossings) in $L(x; t_2)$ must not exceed the number of local extrema (zero-crossings) in $L(x; t_1)$. Suppose also that $g(\xi; t)$ is Borel-measurable as a function of t . Then necessarily (and sufficiently),

$$g(\xi; t) = (2\pi\alpha t)^{-1/2} \exp(-\xi^2/2\alpha t) \quad (2.61)$$

for some non-negative real α .

Proof: According to the above treatment every kernel $g(\cdot; t)$ must be a continuous scale-space kernel, that is a Pólya frequency function. A theorem by Karlin [Kar68] shows that these conditions uniquely define the Gaussian family of kernels.

Classification of semi-groups of Pólya frequency functions:

Let $g : X \times R_+ \rightarrow R$ denote a one-parameter family of Pólya frequency functions integrable on the real axis and fulfilling the semi-group property

$$g(\cdot; t_1) * g(\cdot; t_2) = g(\cdot; t_1 + t_2) \quad (2.62)$$

⁹In the continuous case, the variation-diminishing property is normally expressed in terms of zero-crossings. Thus, this formulation is valid only if the differentiation operator commutes with the convolution operator. If problems occur we prefer to base the discussion on zero-crossings instead.

Suppose also that $g(x; t)$ is Borel-measurable as a function of t . Then, necessarily

$$g(x; t) = \frac{1}{\sqrt{2\pi\alpha t}} e^{-(x-\delta t)^2/2\alpha t} \quad -\infty < x < \infty; \quad t > 0 \quad \delta \in R \quad (2.63)$$

Because of the symmetry constraint the constant δ must be zero. The constant α only affects the scaling of the scale parameter. Hence, it can be set to one without loss of generality. \square

Consequently, this theorem provides a new formulation of the one-dimensional scale-space theory for continuous signals, leading to the same result as the work by Koenderink and van Doorn [Koe84] and Babaud et.al. [Bab86], as well as further support for the firm belief that Theorem 2.20 states the canonical way to define a scale-space for discrete signals. The assumption of Borel-measurability means no important restriction. It is well-known that all continuous functions are Borel-measurable.

2.6 Discrete Scale-Space Properties of Some Numerical Approximations of the Continuous Scale-Space Theory

In this section we will consider some numerical approximations, which are close at hand for the convolution integral (1.1) and the diffusion equation (1.2). Using the classification results derived in previous sections we will investigate if the occurring transformations possess scale-space properties in the discrete sense. One aim is to analyze the previously commonly adapted approach where the filter coefficients are set to sampled values of the Gaussian kernel. We show that some undesired effects occur, mainly due to the fact that the semi-group property does not hold after discretization. We also show that the transformation obtained by convolution with the presented discrete analog of the Gaussian kernel is equivalent to the solution of a discretized version of the diffusion equation. This result as well as some other interconnections between the scale-space formulations for continuous and discrete signals provide further motivation for the selection of T as the canonical discrete scale-space kernel. The rendering is of necessity somewhat technical and the details can be skipped by the hasty reader without loss of continuity.

2.6.1 Sampled Gaussian Kernel

Maybe the most obvious way to approximate the convolution integral

$$L(x; t) = \int_{\xi=-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} f(x - \xi) d\xi \quad (x \in R, t > 0) \quad (2.64)$$

numerically is by the rectangle rule of integration. Provided that no truncation of the infinite integration interval is performed this leads to the approximation formula.

$$\tilde{L}(x; t) = \sum_{n=-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-n^2/2t} f(x - n) \quad (2.65)$$

i.e. discrete convolution with the sampled Gaussian kernel. We will show that this representation might lead to undesirable effects. From the definitions of PF_{∞} -functions and PF_{∞} -sequences in terms of minors it is clear that

Lemma 2.24 (A sampled PF_∞ -function is a PF_∞ -sequence)

Uniform sampling of a continuous scale-space kernel gives a discrete scale-space kernel.

Therefore, since the Gaussian kernel is a PF_∞ function it follows that the transformation from the zero level $L(\cdot; 0)$ to a higher level never increases the number of local extrema (zero-crossings). However, we will show below that the transformation from an arbitrary low level $\tilde{L}(\cdot; t_1)$ to an arbitrary higher level $\tilde{L}(\cdot; t_2)$ is in general *not* a scale-space transformation. Thus, we are not always guaranteed that the amount of structure will decrease monotonically with scale. More precisely,

Proposition 2.25 (Scale-space properties of the sampled Gaussian kernel)

The transformation from a low level $t_1 \geq 0$ to an arbitrary higher level $t_2 > t_1$ in the representation (2.65), generated by discrete convolution with the sampled Gaussian kernel, is a scale-space transformation if and only if either t_1 is zero or the ratio t_2/t_1 is an odd integer.

Proof: Assume that we construct the “scale-space” for a discrete signal by convolution with the sampled Gaussian kernel, i.e. given a discrete signal $f : Z \rightarrow R$ we define the family of functions $\tilde{L} : Z \times R_+ \rightarrow R$ by $\tilde{L}(x; 0) = f(x)$ ($x \in Z$) and

$$\tilde{L}(x; t) = \sum_{n=-\infty}^{\infty} g(n; t) f(x - n) \quad (x \in Z, t > 0) \quad (2.66)$$

where

$$g(n; t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{n^2}{2t}} \quad (n \in Z, t > 0) \quad (2.67)$$

We will make use of an expression for the generating function for the discrete kernel corresponding to the sampled Gaussian. For simplicity we let $q_t = e^{-\frac{1}{2t}}$. One can show, see e.g. Mumford [Mum83], that

$$\varphi_t(z) = \sum_{n=-\infty}^{\infty} g(n; t) z^n = \frac{1}{\sqrt{2\pi t}} \sum_{n=-\infty}^{\infty} q_t^{n^2} z^n = C_t \prod_{n=0}^{\infty} (1 + q_t^{2n+1} z)(1 + q_t^{2n+1} z^{-1}) \quad (2.68)$$

where

$$C_t = \frac{1}{\sqrt{2\pi t}} \prod_{n=1}^{\infty} (1 - q_t^{2n}) \quad (2.69)$$

Comparison with the complete characterization of the generating function of a discrete scale-space kernel (16) in Theorem 2 shows that the sampled Gaussian kernel is a discrete scale-space kernel. This constitutes another proof of the property that for any signal f the number of local extrema in $\tilde{L}(x; t)$ ($t > 0$) does not exceed the number of local extrema in f . However, we will now show that this scale-space property does *not* hold between two arbitrary levels.

Let t_1 and t_2 be two levels ($t_2 > t_1 > 0$) of the representation (2.66) and let φ_{in} be the generating function of the input signal. Then the generating functions of $\tilde{L}(x; t_1)$ and $\tilde{L}(x; t_2)$ are

$$\varphi_{\tilde{L}_1}(z) = \varphi_{t_1}(z) \varphi_{in}(z) \quad \varphi_{\tilde{L}_2}(z) = \varphi_{t_2}(z) \varphi_{in}(z) \quad (2.70)$$

Let φ_{diff} describe the transformation from $\tilde{L}(x; t_1)$ to $\tilde{L}(x; t_2)$. Thus,

$$\varphi_{\tilde{L}_2}(z) = \varphi_{diff}(z) \varphi_{\tilde{L}_1}(z) \quad (2.71)$$

Combination of (2.70), (2.71) and (2.69) gives

$$\varphi_{diff}(z) = \frac{\varphi_{\tilde{L}_2}(z)}{\varphi_{\tilde{L}_1}(z)} = \frac{C_{t_2}}{C_{t_1}} \cdot \frac{\prod_{m=0}^{\infty} (1 + q_{t_2}^{2m+1} z)(1 + q_{t_2}^{2m+1} z^{-1})}{\prod_{n=0}^{\infty} (1 + q_{t_1}^{2n+1} z)(1 + q_{t_1}^{2n+1} z^{-1})} \quad (2.72)$$

According to the complete characterization of scale-space kernels it follows that the corresponding kernel is a scale-space kernel if and only if (2.72) can be written on the form (2.27). Then, for each factor $(1 + q_{t_1}^{2n+1} z^{\pm 1})$ in the denominator there must exist a corresponding factor in the numerator $(1 + q_{t_2}^{2m+1} z^{\pm 1})$, i.e for each n there must exist an m such that

$$q_{t_1}^{2n+1} = q_{t_2}^{2m+1} \quad (2.73)$$

Insertion of $q_{t_i} = e^{-\frac{1}{2t_i}}$ and reduction gives the necessary and sufficient requirement

$$2m = \frac{t_2}{t_1}(2n + 1) - 1 \quad (2.74)$$

It is clear that this relation cannot hold for all $n \in \mathbb{Z}$ if t_1 and t_2 are chosen arbitrarily. The transformation from $\tilde{L}(x; t_1)$ to $\tilde{L}(x; t_2)$ ($t_2 > t_1$) is a scale-space transformation if and only if the ratio $\frac{t_2}{t_1}$ is an odd integer. \square

The result constitutes an example of the fact that properties derived in the continuous case might be violated after discretization. The main reason why the scale-space property fails to hold between two arbitrary levels is because *the semi-group property of the Gaussian kernel is not preserved after discretization*¹⁰.

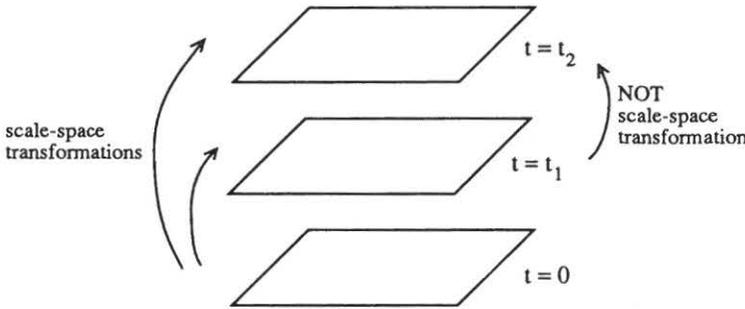


Figure 2.10: In the “scale-space representation” produced by discrete convolution with the sampled Gaussian kernel the transformation from the zero level to any coarser level of scale is always a scale-space transformation. However, the transformation between two arbitrary levels is in general not a scale-space transformation.

2.6.2 Discretized Diffusion Equation

The scale-space family generated by (2.36) and (2.38) can be interpreted in terms of a discretized version of the diffusion equation.

¹⁰This means that if a representation at a level $t_2 > 0$ is computed via an intermediate level t_1 ($0 < t_1 < t_2$) by application of the approximation formula (2.66) in two steps, the computation does yield the same result as if it would have been computed directly from the original signal.

Theorem 2.26 (Diffusion formulation of the scale-space for discrete signals)

Given a discrete signal $f : Z \rightarrow R$ in l_1 let $L : Z \times R_+ \rightarrow R$ be the discrete scale-space representation given by

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t)f(x - n) \quad (2.75)$$

where $T : Z \times R_+ \rightarrow R$ is the discrete analog of the Gaussian kernel. Then L is the solution of the system of ordinary differential equations

$$\frac{\partial L(x; t)}{\partial t} = \frac{1}{2}(L(x + 1; t) - 2L(x; t) + L(x - 1; t)) \quad (x \in Z) \quad (2.76)$$

with initial conditions $L(x; 0) = f(x)$, i.e. the system of differential equations obtained if the diffusion equation (1.2) is discretized in space but solved analytically in time.

Proof: From the relation

$$2I'_n(t) = I_{n-1}(t) + I_{n+1}(t) \quad (2.77)$$

for modified Bessel functions, see e.g Abramowitz and Stegun [Abr64], one easily shows that $T(n; t) = e^{-t}I_n(t)$ satisfies:

$$\frac{\partial T(n; t)}{\partial t} = \frac{\partial}{\partial t} (e^{-t}I_n(t)) = e^{-t}I'_n(t) - e^{-t}I_n(t) = \quad (2.78)$$

$$e^{-t} \frac{I_{n-1}(t) + I_{n+1}(t)}{2} - e^{-t}I_n(t) = \frac{1}{2}(T(n - 1; t) - 2T(n; t) + T(n + 1; t)) \quad (2.79)$$

which in turn means that

$$\frac{\partial L(x; t)}{\partial t} = \frac{\partial}{\partial t} \sum_{n=-\infty}^{\infty} T(n; t)f(x - n) = \sum_{n=-\infty}^{\infty} \frac{\partial T(n; t)}{\partial t} f(x - n) = \quad (2.80)$$

$$\sum_{n=-\infty}^{\infty} \frac{1}{2}(T(n - 1; t) - 2T(n; t) + T(n + 1; t))f(x - n) =$$

$$\frac{1}{2}(L(x - 1; t) - 2L(x; t) + L(x + 1; t))$$

The regularity condition on f justifies the change of order between differentiation and infinite summation. \square

This provides another motivation for the selection of $T(n; t) = e^{-t}I_n(t)$ as the canonical discrete scale-space kernel. If (2.76) is further discretized in scale using Eulers method we obtain the iteration formula

$$L_i^{k+1} = \frac{\Delta t}{2}L_{i+1}^k + (1 - \Delta t)L_i^k + \frac{\Delta t}{2}L_{i-1}^k \quad (2.81)$$

where the subscripts denote the spatial coordinates and the superscripts the iteration indices. Equivalently one iteration with this formula can be described as discrete convolution with the three-kernel

$$\left(\frac{\Delta t}{2}, 1 - \Delta t, \frac{\Delta t}{2} \right) \quad (2.82)$$

Proposition 2.12 states that this kernel is a scale-space kernel if and only if

$$\Delta t \leq \frac{1}{2} \quad (2.83)$$

which is a stronger condition on Δt than induced by the stability criterion for Euler's forward method, see e.g. Strang [Str86]. From Corollary 2.19 we have that all symmetric scale-space kernels with finite support can be derived from kernels of this latter form. Hence, they provide a possible set of primitive kernels for the scale-space with a discrete scale parameter discussed in Section 2.5.1.

Proposition 2.27 (Diffusion equation and discrete scale-space kernels)

All symmetric discrete scale-space kernels with finite support arise from repeated application of the discretization of the diffusion equation (2.81), using if necessary different $\Delta t_k \in [0, \frac{1}{2}]$.

In many applications the scale step in multi-scale representations with discrete scale parameter has selected such that $\Delta t = \frac{1}{2}$. Note, however, that for any $0 \leq \Delta t \leq \frac{1}{2}$ the kernel given by (2.82) is a discrete scale-space kernel. Hence, it enables a finer sampling in scale also for the scale-space with discrete scale parameter.

It is not too difficult to derive the analytical solution to the system of scale-continuous equations (2.76). Assume that we want to compute the scale-space representation for a fixed value of t . We can use the discretization (2.81) with n steps in the scale-direction such that the step size $\Delta t = t/n$ satisfies (2.83). As each iteration step consists of a linear convolution the final solution can equivalently be obtained by convolution with the composed kernel $K_{composed} = \ast_{i=1}^n K_{step}$. Let us derive an asymptotic expression for its generating function. The generating function for the transformation corresponding to one iteration with the formula (2.81) is

$$\varphi_{step}(z) = \frac{\Delta t}{2}z^{-1} + (1 - \Delta t) + \frac{\Delta t}{2}z \tag{2.84}$$

and the generating function of the composed kernel describing the transformation from the scale zero to scale t is

$$\varphi_{composed,n}(z) = (\varphi_{step}(z))^n = \left(\frac{\Delta t}{2}z^{-1} + (1 - \Delta t) + \frac{\Delta t}{2}z\right)^n \tag{2.85}$$

which can be written as

$$\varphi_{composed,n}(z) = \left(1 + \frac{t}{n}\left(\frac{z^{-1}}{2} - 1 + \frac{z^1}{2}\right)\right)^n \tag{2.86}$$

after substitution of $\frac{t}{n}$ for Δt . Since $\lim_{n \rightarrow \infty} (1 + \frac{\alpha_n}{n})^n = e^\alpha$ if $\lim_{n \rightarrow \infty} \alpha_n = \alpha$ it follows that

$$\lim_{n \rightarrow \infty} \varphi_{composed,n}(z) = e^{-\frac{t}{2}(1-z^{-1})} e^{-\frac{t}{2}(1-z)} = e^{-t} e^{\frac{t}{2}(z^{-1}+z)} \tag{2.87}$$

We recognize the generating function of the family of discrete kernels we arrived at when we constructed the discrete scale-space in Section 2.5.2. e^{-t} is the normalization factor. Consequently, this provides a more constructive proof of the property that the transformation obtained by convolution with the discrete analog of the Gaussian is equivalent¹¹ to the analytical solution of the system of equations obtained by discretizing the diffusion equation on a fixed equidistant grid in space.

Proposition 2.28 (Repeated averaging and the diffusion equation)

The discrete scale-space generated by convolution with the discrete Gaussian kernel (2.41)

¹¹The conclusion is valid only if the solution to the discretization (2.81) converges to the solution of the continuous equations (2.76) when $\Delta t \rightarrow 0$. This does for instance hold if $f \in l_1$ or $f \in l_2$.

or equivalently by the semi-discretized version of the diffusion equation (2.76) describes the limit case of repeated iteration of the recurrence relation (2.81) as the scale step tends to zero.

This is not surprising bearing Theorem 2.23 in mind. The essence of this treatment is that when one applies the scale-space theory to discrete signals one should discretize what is necessary, namely along the spatial coordinate. The continuous scale parameter can be left untouched.

2.6.3 Integrated Gaussian Kernel

Another way of discretizing the convolution integral (2.64) is by integrating the continuous Gaussian kernel over each pixel support region. This method can be regarded as giving “a more true approximation”¹² than the method with the sampled Gaussian, especially at fine scales (compare also with Chapter 4). The resulting approximation formula corresponds to discrete convolution with the kernel given by

$$c_i = \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} d\xi \quad (2.88)$$

This choice of filter coefficients is equivalent to the continuous formulation (2.64) if we let the continuous signal f be a piecewise constant function, which is equal to the discrete pixel value over each pixel support region. Another possibility is to let f in (2.64) be defined by linear interpolation between the discrete values, which leads to

$$c_i = \int_{i-1}^i (\xi - i + 1) \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} d\xi + \int_i^{i+1} (i - \xi + 1) \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} d\xi \quad (2.89)$$

According to a theorem by Karlin [Kar68] it holds that that a kernel, given by the difference operator applied to uniformly sampled values of an integrated Pólya frequency function, is a Pólya frequency sequence.

Uniform sampling of integrated PF_∞ functions:

Let $f(x)$ be a PF_∞ sequence and form

$$g(x) = \int_{\xi=-\infty}^{\infty} f(\xi) d\xi \quad (2.90)$$

Then $(\Delta g)(n) = g(n+1) - g(n)$ constitutes a PF_∞ sequence.

This means that the transformation from the original signal ($t = 0$) to an arbitrary level of scale ($t_1 > 0$) is always a scale-space transformation. However, we cannot expect any

¹²This issue actually comes down to philosophical questions in the image formation process. What do the recorded pixel values actually represent? Often they are implicitly without notice regarded as sampled values of the underlying physical light intensity in the real world. In reality this is certainly not true, but under that assumption the formula (2.65) should be a proper discretization. (Except for the fact that the grid is not dense enough to resolve the rapid variations in the integrand.) Presumably, a more correct statement is that the pixel values should be regarded as the result of first applying a continuous convolution operator to the physical light intensity and then as a second step sampling that output uniformly. The integration formula defined by (2.88) is an example of the latter model. In that case the kernel function is assumed to be one within the whole pixel support region and zero outside. Probably, a bell-shaped kernel would be more realistic.

semigroup property to hold exactly and will probably arrive at similar scale-space problems as with the sampled Gaussian kernel when considering transformations between arbitrary scale levels. We leave it as an open problem to judge whether the second kernel (2.89) is a scale-space kernel or not.

Proposition 2.29 (Scale-space properties of the integrated Gaussian kernel)

The transformation from the zero level to a coarser level in the representation generated by discrete convolution with the integrated Gaussian kernel, given by (2.88), is a discrete scale-space transformation.

2.7 Summary and Discussion

The aim of this treatment has been to investigate the discrete aspects of the one-dimensional scale-space theory. We have studied linear and shift-invariant transformations and stated a requirement on kernels saying that the number of local extrema in a convolved signal must not exceed the number of local extrema in the original signal. As an immediate consequence we saw that the coefficient sequence must be non-negative and unimodal. For symmetric kernels the same requirements hold for the Fourier transform. We saw that the interesting kernels could be completely classified in terms of total positivity — all shift-invariant discrete scale-space kernels are equivalent to normalized Pólya frequency sequences. The generating function of such a sequence/kernel possesses a very simple characterization, implying that there are only three non-trivial types of primitive smoothing transformations; repeated averaging, recursive smoothing and diffusion smoothing.

Then we introduced a continuous scale parameter and showed that the only reasonable way to define a scale-space for discrete signals is by convolution with the one-parameter family of kernels $T(n; t) = e^{-t}I_n(t)$, where I_n are the modified Bessel functions of integer order. When similar arguments were applied in the continuous case we were uniquely led to the Gaussian kernel. The kernel T does also have the attractive property that it is equivalent to the limit case of a certain discretization of the diffusion equation. The idea of a continuous scale parameter even for discrete signals is of considerable importance, since it permits arbitrary degrees of smoothing, i.e. we are no longer restricted to specific predetermined levels of scale. Due to the semi-group property, the scale-space condition holds between any two levels of representation. We showed that the commonly used technique, where the “scale-space” is constructed by convolution with the sampled Gaussian kernel, might lead to undesirable effects, since in general the transformation from an arbitrary fine level to a randomly selected coarser level is not a scale-space transformation.

Let us finally point out some other aspects of the presented theory that have not been mentioned elsewhere.

2.7.1 Ideal Low-Pass Filters and Block Average Filters

The unimodality requirement on discrete scale-space kernels implies that an “ideal low-pass filter” is not a smoothing kernel in this sense because of the ringing phenomena in the spatial domain. This means that the first pre-filtering step that is often carried out in digital signal processing in order to guarantee band-limited signals actually violates the scale-space conditions. Neither does a block average filter possess scale-space properties, unless its width is either 1 or 2. This can be easily understood from the ringing phenomena and the non-negative values introduced in the frequency domain.

2.7.2 Positivity and Unimodality is Necessary but not Sufficient

Note that the positivity and unimodality requirements for discrete scale-space kernels are necessary but not sufficient requirements. In other words, there exist kernels, which are non-negative and unimodal both in the spatial and the frequency domain but are not discrete scale-space kernels. This can be easily shown, for instance by considering a symmetric five-kernel having a generating function with only complex roots, see Appendix A.1.2.

Observation 2.30 (Positivity and unimodality not sufficient)

The positivity and unimodality requirements in the spatial and the frequency domain are necessary but not sufficient conditions for a one-dimensional discrete kernel $Z \rightarrow R$ to be a discrete scale-space kernel.

2.7.3 Recursive Filters

According to the classification of discrete scale-space kernels, it follows that the recursive filters suggested by Deriche [Der87a, Der87b] possess discrete scale-space properties if and only if they can be implemented as a series of first order smoothing filters, i.e., if and only if their generating function

$$H_{a,b}(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{n-1} z^{-(n-1)}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (2.91)$$

can be factorized to the form

$$\varphi_K(z) = c \prod_{k=1}^n \frac{1 + \delta_k z^{-1}}{1 - \gamma_k z^{-1}} \quad (2.92)$$

where $c > 0$, $\gamma_k, \delta_k \geq 0$ and $\gamma_k < 1$, compare with (2.29).

2.8 Conclusion: Scale-Space for 1-D Discrete Signals

The results from this one-dimensional treatment seem to point all in the same direction. *The natural way to apply the scale-space theory to discrete signals is apparently by discretizing the diffusion equation, not the convolution integral.*

Chapter 3

Scale-Space for 2-D Discrete Signals

3.1 From One to Two Dimensions

The extension of the previous one-dimensional theory to two and higher dimensions is not obvious, since it is possible to show that there do not exist any non-trivial kernels on R^2 or Z^2 with the property that they never introduce new local extrema. Lifshitz and Pizer [Lif87] present an illuminating counter-example:

Imagine a two-dimensional image function consisting of two hills, one of them somewhat higher than the other one, see Fig. 3.1. Assume that they are smooth, wide, rather bell-shaped surfaces situated some distance apart clearly separated by a deep valley running between them. Connect the two tops by a narrow sloping ridge without any local extrema, so that the top point of the lower hill no longer is a local maximum. Let this configuration be the input image. When the operator corresponding to the diffusion equation is applied to the geometry, the ridge will erode much faster than the hills. After a while it has eroded so much that the lower hill appears as a local maximum again. Thus, a new local extremum has been created.

The same argument can be carried out in the discrete case. Of course, we have to consider connectivity when we define what we mean by local extrema. But this question is only of a formal nature. Given an arbitrary non-trivial convolution kernel it is always possible to create a counter-example where the number of local extrema can increase, provided that the peaks are located sufficiently wide apart and the valley between them is sufficiently deep. Therefore, it is not clear what we should mean with a scale-space property in two space dimensions. We cannot generalize the formulation in terms of zero-crossings of the Laplacian either. From the counter-example it is apparent that a level curve may split into two during erosion. Consequently, we cannot expect to find a nontrivial kernel never increasing the number of zero-crossing curves either¹.

Anyway, we should not be too surprised. In some sense the decomposition of the scene is intuitively quite reasonable. The narrow ridge is a fine-scale phenomenon and should therefore disappear before the coarse-scale peaks. In this case it is rather the measure on structure than the smoothing method that is the decisive factor.

The property that new local extrema can be created by linear smoothing seems inherent and inescapable in two and higher dimensions. Also other types of features, which are possible candidates for being “measures of structure”, like zero-crossings, convex and concave

¹However, new zero-crossings of the Laplacian, not arising from splits of previously existing zero-crossings of the Laplacian, cannot be created due to the causality property.

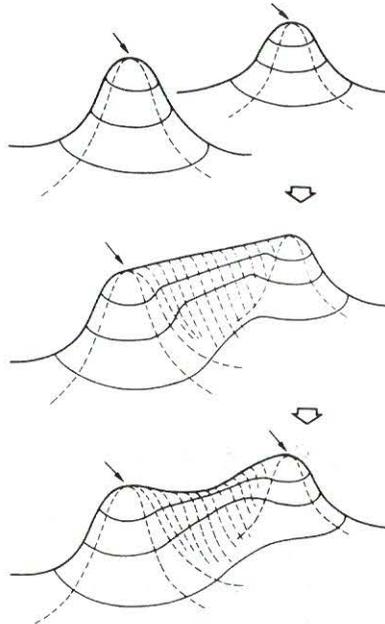


Figure 3.1: *New local extrema can be created by the diffusion equation in the two-dimensional case*

regions etc. may be created, see e.g. Yuille [Yui88].

Therefore, when extending the theory to higher dimensions, we should not be too locked to the previously given definition of a discrete scale-space kernel. In one dimension the number of local extrema is a natural measure of structure on which a theory can be founded — in two dimensions obviously not. Instead the previously given treatment should be understood in a wider sense as a characterization of which one-dimensional convolution transformations can be regarded as smoothing transformations.

Is it true that the discrete analogue of the Gaussian kernel used as a separated kernel is the natural discrete kernel in two dimensions? If one, due to computational considerations, wants to use separable discrete kernels, one could, of course, heuristically argue that the kernel should at least have a good performance in one dimension. Another indication in that direction is obtained if one studies a discretized version of the two-dimensional diffusion equation. In Appendix A.2.1 it is shown that separated convolution with the one-dimensional discrete analogue of the Gaussian kernel describes the solution of the system of ordinary differential equations, which appears if the diffusion equation is discretized in space but not in time (scale).

In this chapter we will develop a two-dimensional theory based on somewhat modified axioms, which however in one dimension turns out to give the same result as the previous formulation. In a special case, the resulting scale-space representation can be reduced to the representation given by separated convolution with the discrete analogue of the Gaussian kernel.

3.2 Selecting Two-Dimensional Scale-Space Axioms

From the discussion in the previous section it is clear that the one-dimensional treatment cannot be generalized directly to higher dimensions. However, an important point about

the study we have performed, is that we have acquired a deep understanding on what one-dimensional linear transformations can be regarded as smoothing transformations. We have also shown that the only reasonable way to convert the one-dimensional scale-space theory from continuous signals to discrete signals is by discretization of the diffusion equation.

Koenderink, van Doorn [Koe84] derive the two-dimensional scale-space for continuous images from three assumptions — causality, homogeneity and isotropy. The leading idea is that every grey-level at a coarse level of scale should be possible to trace from the same grey-level at a finer level of scale. In other words, no new grey-level surfaces² should be created in the scale-space representation when the scale parameter increases, see Fig. 3.2. Using differential geometry they show that these requirements uniquely lead to the diffusion equation, or equivalently to convolution with the Gaussian kernel.

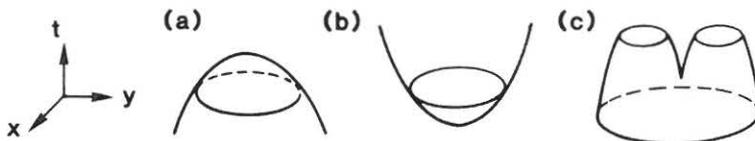


Figure 3.2: Grey-level surfaces $L(x, y; t) = z_0$. (a) Causal (and generic) grey-level surface. (b) Non-causal (and impossible) grey-level surface. (c) Grey-level surface corresponding to the example in Fig. 4 where one grey-level curve splits into two.

It is of course impossible to apply these ideas directly, since there do not exist any direct correspondences to level curves and differential geometry in the discrete case. However, an alternative way to express the previous ideas is by requiring that *if for some scale level t_0 a point (x_0, y_0) is a local maximum for the scale-space representation at that level (regarded as a function of the space coordinates only) then its value must not increase when the scale parameter increases*. Analogously, if a point is a local minimum then its value must not decrease when the scale parameter increases.

It is clear that this formulation is equivalent to the formulation in terms of grey-levels for continuous images, since if the grey-level value at a local maximum (minimum) would increase (decrease) a new grey-level curve would be created. Inversely, if a new grey-level curve is created then some local maximum must have increased or some local minimum must have decreased.

An intuitive description of this requirement is that it prevents local extrema from being enhanced and from “popping up out of nowhere” when the scale parameter increases. As we have seen earlier, we can never prevent the number of local extrema from being increased. However the idea is that those creation events should be “few”.

In the next section we will show that this condition combined with a continuous scale parameter means a strong restriction on the smoothing method also in the discrete case, and we will again obtain a discretized version of the diffusion equation. In a special case the resulting scale-space representation will be reduced to the family of functions generated by separated convolution with the discrete analogue of the Gaussian kernel, $T(n; t)$.

²By a grey-level surface we mean an iso-surface in scale-space i.e. a connected set of points $(x, y; t) \in \mathbb{R}^2 \times \mathbb{R}$ such that $L(x, y; t) = z_0$ for some grey-level value z_0 .

3.2.1 Basic Definitions

Before getting into the detailed scale-space formulation we will need to make a few definitions. The eight-neighbours of a point $(x, y) \in Z^2$ will be denoted $N_8(x, y)$. If the central point is included as well we will use the notation $N_8^+(x, y)$. The notion of extremum points will be as follows:

Definition 3.1 (Discrete local maximum)

A point (x, y) is said to be a (weak) local maximum point for a function $g : Z^2 \rightarrow R$ if $g(x, y) \geq g(\xi, \eta)$ for all $(\xi, \eta) \in N_8(x, y)$.

Definition 3.2 (Discrete local minimum)

A point (x, y) is said to be a (weak) local minimum point for a function $g : Z^2 \rightarrow R$ if $g(x, y) \leq g(\xi, \eta)$ for all $(\xi, \eta) \in N_8(x, y)$.

It is also useful here to introduce two common discrete operators, approximating the two-dimensional Laplace operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, namely the five-point operator ∇_5^2 and the cross operator ∇_{\times}^2 , defined by³:

$$(\nabla_5^2 f)(x, y) = f(x-1, y) + f(x+1, y) + f(x, y-1) + f(x, y+1) - 4f(x, y) \quad (3.1)$$

$$(\nabla_{\times}^2 f)(x, y) = \frac{1}{2}(f(x-1, y-1) + f(x-1, y+1) + f(x+1, y-1) + f(x+1, y+1) - 4f(x, y)) \quad (3.2)$$

The corresponding one-dimensional operator is the three-point operator, ∇_3^2 , given by

$$(\nabla_3^2 f)(x) = f(x-1) - 2f(x, y) + f(x+1) \quad (3.3)$$

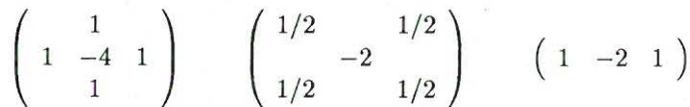


Figure 3.3: Computational molecules for (a) the five-point operator ∇_5^2 (b) the cross operator ∇_{\times}^2 and (c) the three-point operator ∇_3^2 . (Throughout this treatment we use a unit step size.)

3.3 Axiomatic 2D Discrete Scale-Space Construction

When we construct the scale-space for two-dimensional discrete images we follow the ideas from the one-dimensional case, see Section 2.5.2. We start by postulating that the scale-space should be generated by convolution with a one-parameter of kernels, i.e. $L(x, y; 0) = f(x, y)$ and

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m, n; t) f(x-m, y-n) \quad (t > 0) \quad (3.4)$$

As mentioned earlier, this form on the smoothing formula corresponds to the requirements about linear shift-invariant smoothing and a continuous scale parameter. We want both

³In our considerations the step size h is set to 1.

coordinate directions to be processed identically. Therefore all kernels should be symmetric. We will also impose a semi-group condition on the family T . This means that all scale levels will be treated similarly, i.e. the smoothing operation does not depend on the scale value, and that the transformation from a lower scale level to a higher scale level will be given by convolution with a kernel from the family, compare with (2.37).

The smoothing criterion will be the requirement about local extrema indicated in the previous section. It is convenient to express it as a condition on the derivative of the scale-space family with respect to the scale parameter. In order to ensure a proper statement of this condition, where differentiability is guaranteed, we will need to state a series of preliminary definitions leading to the desired scale-space formulation.

3.3.1 Definitions

We start by summarizing the basic properties we would like a family of kernels to satisfy in order to be a candidate family for the generation of a scale-space representation.

Definition 3.3 (Pre-scale-space family of kernels)

A one-parameter family of kernels $T : Z^2 \times R_+ \rightarrow R$ is said to be a pre-scale-space family of kernels if it satisfies

- $T(\cdot, \cdot; 0) = \delta(\cdot, \cdot)$
- the semi-group property $T(\cdot, \cdot; s) * T(\cdot, \cdot; t) = T(\cdot, \cdot; s + t)$
- the symmetry properties⁴ $T(-x, y; t) = T(x, y; t)$ and $T(y, x; t) = T(x, y; t)$ for all $(x, y) \in Z^2$.
- the continuity requirement $\|T(\cdot, \cdot; t) - \delta(\cdot, \cdot)\|_1 \rightarrow 0$ when $t \downarrow 0$

Definition 3.4 (Pre-scale-space representation)

Let $f : Z^2 \rightarrow R$ be a discrete signal and $T : Z^2 \times R_+ \rightarrow R$ a pre-scale-space family of kernels. Then the one-parameter family of signals $L : Z^2 \times R_+ \rightarrow R$ given by

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m, n; t) f(x - m, y - n) \quad (3.5)$$

is said to be the pre-scale-space representation of f generated by T .

Provided that the input signal f is sufficiently regular, these conditions on the family of kernels T guarantee that the representation L is differentiable and satisfies a system of linear differential equations.

Lemma 3.1 (A pre-scale-space representation is differentiable)

Let $L : Z^2 \times R_+ \rightarrow R$ be the pre-scale-space representation of a signal $f : Z^2 \rightarrow R$ in l_1 . Then L satisfies the differential equation

$$\frac{\partial L}{\partial t} = \mathcal{A}L \quad (3.6)$$

for some linear and shift-invariant operator \mathcal{A} .

⁴ $T(x, -y; t) = T(x, y; t)$ is implied from the two other properties.

Proof: If f is sufficiently regular, e.g. if $f \in l_1$, we can define a family of operators $\{\mathcal{T}_t, t > 0\}$, here from l_1 to l_1 , by $\mathcal{T}_t f = T(\cdot, \cdot; t) * f$. Due to the conditions imposed on the kernels it will satisfy the relation

$$\lim_{t \rightarrow t_0} \|(\mathcal{T}_t - \mathcal{T}_{t_0})f\|_1 = \lim_{t \rightarrow t_0} \|(\mathcal{T}_{t-t_0} - \mathcal{I})(\mathcal{T}_{t_0}f)\|_1 = 0 \quad (3.7)$$

where \mathcal{I} is the identity operator. Such a family is called a strongly continuous semigroup of operators, see Hille and Philips [Hil57] p58-59. A semi-group is often characterized by its infinitesimal generator \mathcal{A} defined by,

$$\mathcal{A}f = \lim_{h \downarrow 0} \frac{\mathcal{T}_h f - f}{h} \quad (3.8)$$

The set of elements f for which \mathcal{A} exists is denoted $\mathcal{D}(\mathcal{A})$. This set is not empty and it never reduces to the zero element. Actually, it is even dense in l_1 , see Hille and Philips [Hil57] p307. If this operator exists we obtain

$$\begin{aligned} \lim_{h \downarrow 0} \frac{L(\cdot, \cdot; t+h) - L(\cdot, \cdot; t)}{h} &= \lim_{h \downarrow 0} \frac{\mathcal{T}_{t+h} f - \mathcal{T}_t f}{h} = \\ \lim_{h \downarrow 0} \frac{\mathcal{T}_h(\mathcal{T}_t f) - (\mathcal{T}_t f)}{h} &= \mathcal{A}(\mathcal{T}_t f) = \mathcal{A}L(\cdot, \cdot; t) \end{aligned} \quad (3.9)$$

According to a Theorem by Hille and Phillips [Hil57] p308 strong continuity implies that $\frac{\partial}{\partial t}(\mathcal{T}_t f) = \mathcal{A}\mathcal{T}_t f = \mathcal{T}_t \mathcal{A}f$ for all $f \in \mathcal{D}(\mathcal{A})$. Hence, the scale-space family L must obey the differential equation

$$\frac{\partial L}{\partial t} = \mathcal{A}L \quad (3.10)$$

for some linear operator \mathcal{A} . Since L is generated from f by a convolution operation it follows that \mathcal{A} must be shift-invariant. \square

This property allows us to formulate the previously indicated scale-space property in terms of derivatives of the scale-space representation with respect to the scale parameter. In every local maximum point we require the grey-level value not to increase and in every local minimum point the value not to decrease.

Definition 3.5 (Pre-scale-space property: Non-enhancement of local extrema)
A differentiable one-parameter family of signals $L : Z^2 \times R_+ \rightarrow R$ is said to possess pre-scale-space properties, or equivalently not to enhance local extrema, if for every value of the scale parameter $t_0 \in R_+$ it holds that if $(x_0, y_0) \in Z^2$ is a local extremum point for the mapping $(x, y) \mapsto L(x, y; t_0)$ then the derivative of L with respect to t in this point satisfies

$$\frac{\partial L}{\partial t}(x_0, y_0; t_0) \leq 0 \quad \text{if } (x_0, y_0) \text{ is a local maximum point} \quad (3.11)$$

$$\frac{\partial L}{\partial t}(x_0, y_0; t_0) \geq 0 \quad \text{if } (x_0, y_0) \text{ is a local minimum point} \quad (3.12)$$

Now we can state that a pre-scale-space family of kernels is a scale-space family of kernels if it satisfies this property for any input signal.

Definition 3.6 (Scale-space family of kernels)
A one-parameter family of pre-scale-space kernels $T : Z^2 \times R_+ \rightarrow R$ is said to be a scale-space family of kernels if for any signal $f : Z^2 \rightarrow R \in l_1$ the pre-scale-space representation of f generated by T possesses pre-scale-space properties, i.e. if for any signal local extrema are never enhanced.

Definition 3.7 (Scale-space representation)

A pre-scale-space representation $L : Z^2 \times R_+ \rightarrow R$ of a signal $f : Z^2 \rightarrow R$ generated by a family of kernels $T : Z^2 \times R_+ \rightarrow R$, which are scale-space kernels, is said to be a scale-space representation of f .

We will now develop how these requirements strongly restrict the possible class of kernels and scale-space representations.

3.3.2 Necessity

We start by showing that these conditions necessarily imply that the family L satisfies a semi-discretized version of the diffusion equation.

Theorem 3.2 (Scale-space for 2-D discrete signals: Necessity)

A scale-space representation $L : Z^2 \times R_+ \rightarrow R$ of a signal $f : Z^2 \rightarrow R$ satisfies the differential equation

$$\frac{\partial L}{\partial t} = \alpha \nabla_5^2 L + \beta \nabla_x^2 L \tag{3.13}$$

with initial condition $L(\cdot, \cdot; 0) = f(\cdot, \cdot)$ for some constants $\alpha \geq 0$ and $\beta \geq 0$.

Proof: The proof consists of two parts. The first step has already been established in Lemma 3.1, where we showed that the requirements on the kernels imply that the family L obeys a linear differential equation. Because of the shift invariance $\mathcal{A}L$ can be written

$$(\mathcal{A}L)(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{m,n} L(x - m, y - n; t) \tag{3.14}$$

In the second step we construct counterexamples from various simple test functions in order to delimit the class of possible operators.

The extremum point conditions (3.11), (3.12) (combined with Definitions 3.6-3.7) mean that \mathcal{A} must be local, i.e. that $a_{m,n} = 0$ if $|m| > 1$ or $|n| > 1$. This is easily understood by studying the following counterexample: First, assume that $a_{\tilde{m},\tilde{n}} > 0$ where either $|\tilde{m}| > 1$ or $|\tilde{n}| > 1$ (or both), and define a function $f_1 : Z^2 \rightarrow R$ by

$$f_1(x, y) = \begin{cases} \varepsilon > 0 & \text{if } (x, y) = (0, 0) \\ 0 & \text{if } (x, y) \in N_8(0, 0) \\ 1 & \text{if } (x, y) = (\tilde{m}, \tilde{n}) \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

Obviously, $(0, 0)$ is a local maximum point for f_1 . From (3.6) and (3.14) we get that $\frac{\partial L}{\partial t}(0, 0; 0) = \varepsilon a_{0,0} + a_{\tilde{m},\tilde{n}}$. It is clear that this value can be positive provided that ε has been chosen small enough. Hence, L cannot satisfy (3.11). In a similar manner one shows that also $a_{\tilde{m},\tilde{n}} < 0$ leads (let $\varepsilon < 0$) to a violation against the extremum point condition (3.12). Consequently, $a_{\tilde{m},\tilde{n}}$ must be zero if either of $|\tilde{m}|$ or $|\tilde{n}|$ is larger than one. Thereby, (3.6) will be reduced to

$$\frac{\partial L}{\partial t}(x, y; t) = \sum_{(m,n) \in N_8^+(0,0)} a_{m,n} L(x - m, y - n; t) \tag{3.16}$$

where $N_8^+(0,0)$ denotes the set of eight-neighbours to the origin including the origin. Due to the symmetry conditions, opposite coefficients must be equal, i.e. $a_{-m,n} = a_{m,n}$ and $a_{n,m} = a_{m,n}$. Thus, (3.16) can be written

$$\frac{\partial L}{\partial t} = \begin{pmatrix} a & b & a \\ b & c & b \\ a & b & a \end{pmatrix} L \quad (3.17)$$

for some a , b and c . Then, consider the function

$$f_2(x, y) = \begin{cases} 1 & \text{if } (x, y) \in N_8^+(0, 0) \\ 0 & \text{otherwise} \end{cases} \quad (3.18)$$

With the given (weak) definitions of local extremum points it is clear that $(0,0)$ is both a local maximum point and a local minimum point. Hence $\frac{\partial L}{\partial t}(0,0; 0)$ must be zero and we obtain the relation $4a+4b+c=0$. This means that (3.17) can be split into two components.

$$\frac{\partial L}{\partial t} = \begin{pmatrix} a & b & a \\ b & c & b \\ a & b & a \end{pmatrix} L = \alpha \begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix} L + \beta \begin{pmatrix} 1/2 & & 1/2 \\ & -2 & \\ 1/2 & & 1/2 \end{pmatrix} L \quad (3.19)$$

provided that $\alpha = b$ and $\beta = 2a$. The condition $4a+4b+c=0$ is trivially satisfied. Finally, by considering the test function

$$f_3(x, y) = \begin{cases} \epsilon > 0 & \text{if } (x, y) = (0, 0) \\ 1 & \text{if } (x, y) = (\tilde{m}, \tilde{n}) \\ 0 & \text{otherwise} \end{cases} \quad (3.20)$$

for some (\tilde{m}, \tilde{n}) in $N_8(0,0)$ one easily realizes that $a_{m,n}$ must be non-negative if $(m, n) \in N_8(0,0)$. This shows that $\alpha \geq 0$ and $\beta \geq 0$ in the differential equation. The initial condition follows directly from the definition of pre-scale-space kernel. \square

3.3.3 Sufficiency

The reverse statement of Theorem 3.2 is also true. This sufficiency is much easier to establish:

Theorem 3.3 (Scale-space for 2-D discrete signals: Sufficiency)

Let $f : Z^2 \rightarrow R$ be a discrete signal in l_1 and let $L : Z^2 \times R_+ \rightarrow R$ be the representation generated by the solution to differential equation

$$\frac{\partial L}{\partial t} = \alpha \nabla_5^2 L + \beta \nabla_{\times}^2 L \quad (3.21)$$

with initial condition $L(\cdot, \cdot; 0) = f(\cdot, \cdot)$ for some fixed $\alpha \geq 0$. Then L is a scale-space representation of f .

Proof: It follows almost trivially that L possesses pre-scale-space properties, i.e. that L does not enhance local extrema, if we rewrite the differential equation on the form

$$\begin{aligned} \frac{\partial L}{\partial t}(x, y; t) = & \alpha[L(x, y-1; t) - L(x, y; t)] + \\ & \alpha[L(x, y+1; t) - L(x, y; t)] + \\ & \alpha[L(x-1, y; t) - L(x, y; t)] + \\ & \alpha[L(x+1, y; t) - L(x, y; t)] + \\ & \frac{1}{2}\beta[L(x-1, y-1; t) - L(x, y; t)] + \\ & \frac{1}{2}\beta[L(x+1, y-1; t) - L(x, y; t)] + \\ & \frac{1}{2}\beta[L(x-1, y+1; t) - L(x, y; t)] + \\ & \frac{1}{2}\beta[L(x+1, y+1; t) - L(x, y; t)] \end{aligned} \quad (3.22)$$

If for some scale level t a point (x, y) is a local maximum point then all differences (within brackets) are non-positive, which means that $\frac{\partial L}{\partial t}(x, y; t) \leq 0$ provided that $\alpha \geq 0$ and $\beta \geq 0$. Similarly, if a point is a local minimum point then the differences are all non-negative and $\frac{\partial L}{\partial t}(x, y; t) \geq 0$.

What remains to verify is that L actually satisfies the requirements for being a pre-scale-space representation. Since L is generated by a linear differential equation it follows that L can be written as the convolution of f with some kernel T , i.e. $L(\cdot, \cdot; t) = T(\cdot, \cdot; t) * f$. The requirements on pre-scale-space kernels can be shown to hold by letting the input signal f be the discrete delta function. The semi-group property of the kernels follows from the fact that the coefficients α and β are constant and the solution at a time $s+t$ hence can be computed from the solution at an earlier time s by letting the time increase by t . The symmetry properties of the kernel are obvious from the symmetry of the differential equation. The continuity at the origin follows directly from the differentiability. \square

These results show that a one-parameter family of signals is a scale-space representation if and only if it satisfies the differential equation (3.13).

3.3.4 Equivalent One-Dimensional Formulation

From the proofs it is apparent that if similar arguments are applied in the one-dimensional case, we are uniquely lead to the one-dimensional scale-space concept developed earlier in Theorem 2.20 and Theorem 2.26. To summarize,

Theorem 3.4 (Scale-space for 1-D discrete signals: Necessity)

A scale-space representation $L : Z \times R_+ \rightarrow R$ of a signal $f : Z \rightarrow R$ satisfies the differential equation

$$\frac{\partial L}{\partial t} = \alpha \nabla_3^2 L \quad (3.23)$$

with initial condition $L(\cdot; 0) = f(\cdot)$ for some constant $\alpha \geq 0$.

Theorem 3.5 (Scale-space for 1-D discrete signals: Sufficiency)

Let $f : Z \rightarrow R$ be a discrete signal in l_1 and let $L : Z \times R_+ \rightarrow R$ be the representation generated by the solution to differential equation

$$\frac{\partial L}{\partial t} = \alpha \nabla_3^2 L \quad (3.24)$$

with initial condition $L(\cdot; 0) = f(\cdot)$ for some fixed $\alpha \geq 0$ and $\beta \geq 0$. Then L is a scale-space representation of f .

For completeness the corresponding relevant definitions are given in Appendix A.2.2. These results show that, *combined with the requirements about a continuous scale parameter and semi-group structure, the condition about suppression of local extrema is in one dimension equivalent to the condition about decreasing number of local extrema.*

Consequently, also this formulation in terms of local extrema has lead to a discretized version of the diffusion equation. But here in the two-dimensional case there is apparently another degree of freedom left in the class of possible smoothing operators, since a linear combination of the two common discrete Laplacian operators ∇_y^2 and ∇_x^2 is admitted on the right hand side of the the differential equation. The effects of combining those in different ways will be illuminated in the next section.

3.4 Parameter Determination

If (3.13) is rewritten on the form

$$\frac{\partial L}{\partial t} = C \left((1 - \gamma) \nabla_y^2 L + \gamma \nabla_x^2 L \right) = C \nabla_\gamma^2 L \quad (3.25)$$

one realizes that the interpretation of the parameter C is just a trivial rescaling of the scale parameter. Thus, without loss of generality⁵ we may set C to $\frac{1}{2}$ in order to get the same scaling constant as in the one-dimensional case (2.76). What is left to investigate is how the remaining degree of freedom in the parameter $\gamma \in [0, 1]$ affects the scale-space representation.

If $\gamma = 1$ then a undesirable situation appears. Since the cross-operator only links diagonal points, the system of ordinary differential equations given by (3.25) can then be split into two *uncoupled* systems, one operating on the points with even coordinate sum $x + y$ and the other operating on the points with odd coordinate sum. It is clear that this is really an unwanted behaviour, since then even after a substantial amount of “blurring”, for certain types of input signals the “smoothed” grey-level landscape may still have a rather saw-toothed shape.

3.4.1 Derivation of the Fourier Transform

Further arguments showing that γ must not be too large can be obtained if one studies the Fourier transform of the corresponding scale-space family of kernels. Using a methodology similar to the derivation of the generating function of the solution to the one-dimensional semi-discretized diffusion equation in Section 2.6.2, we can derive the generating function of the kernel describing the transformation from the original image to the scale-space representation at a certain scale, which in turn gives us the Fourier transform.

Proposition 3.6 (Fourier transform of the discrete scale-space)

Let $L : Z^2 \times R_+ \rightarrow R$ be the scale-space representation of a discrete signal $f : Z^2 \rightarrow R$ generated by the differential equation (3.25) with initial condition $L(\cdot, \cdot; 0) = f(\cdot, \cdot)$. Assume that $f \in l_1$. Then the generating function of the kernel describing the transformation from

⁵The case when $C = 0$ is obviously not interesting since then all scale-space representations would be equal.

the original signal to the representation at a certain scale t is given by

$$\varphi_T(z, w) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m, n; t) z^m w^n = e^{-(2-\gamma)+\frac{(1-\gamma)}{2}(z^{-1}+z+w^{-1}+w)+\frac{\gamma}{4}(z^{-1}w^{-1}+z^{-1}w+zw^{-1}+zw)} \quad (3.26)$$

Its Fourier transform is

$$\psi_T(u, v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m, n; t) e^{-i(mu+nv)} = e^{-(2-\gamma)t + (1-\gamma)(\cos u + \cos v)t + (\gamma \cos u \cos v)t} \quad (3.27)$$

Proof: If (3.25) is discretized further is scale using Euler's explicit method with scale step Δt , we get an iteration formula of the form

$$\begin{aligned} L_{i,j}^{k+1} &= (1 - (2 - \gamma)\Delta t) L_{i,j}^k + \\ &\quad \frac{(1 - \gamma)\Delta t}{2} (L_{i-1,j}^k + L_{i+1,j}^k + L_{i,j-1}^k + L_{i,j+1}^k) + \\ &\quad \frac{\gamma\Delta t}{4} (L_{i-1,j-1}^k + L_{i-1,j+1}^k + L_{i+1,j-1}^k + L_{i+1,j+1}^k) \end{aligned} \quad (3.28)$$

where the subscripts i and j denote the spatial coordinates x and y respectively and the superscript k denotes the iteration index. The generating function describing one iteration with this transformation is

$$\begin{aligned} \varphi_{step}(z, w) &= (1 - (2 - \gamma)\Delta t) + \\ &\quad \frac{(1 - \gamma)\Delta t}{2} (z^{-1} + z + w^{-1} + w) + \\ &\quad \frac{\gamma\Delta t}{4} (z^{-1}w^{-1} + z^{-1}w + zw^{-1} + zw) \end{aligned} \quad (3.29)$$

Assume that we want to compute the scale-space representation at a scale level t using n iterations with a scale step $\Delta t = \frac{t}{n}$. Then the generating function describing the composed transformation can be written

$$\begin{aligned} \varphi_{composed,n}(z, w) &= (\varphi_{step}(z, w))^n = \\ &= \left(1 + \frac{t}{n}(-(2 - \gamma) + \frac{(1-\gamma)}{2}(z^{-1} + z + w^{-1} + w) + \frac{\gamma}{4}(z^{-1}w^{-1} + z^{-1}w + zw^{-1} + zw))\right)^n \end{aligned} \quad (3.30)$$

after substitution of Δt for $\frac{t}{n}$. Since $\lim_{n \rightarrow \infty} (1 + \frac{\alpha n}{n})^n = e^\alpha$ if $\lim_{n \rightarrow \infty} \alpha_n = \alpha$ it follows that

$$\lim_{n \rightarrow \infty} \varphi_{composed,n}(z) = e^{-(2-\gamma)+\frac{(1-\gamma)}{2}(z^{-1}+z+w^{-1}+w)+\frac{\gamma}{4}(z^{-1}w^{-1}+z^{-1}w+zw^{-1}+zw)} \quad (3.31)$$

provided that the discretization (3.28) converges to the actual solution of (3.25). From this expression the Fourier transform is directly obtained by replacing z with e^{-iu} and w with e^{-iv} . \square

3.4.2 Unimodality in the Fourier Domain

It is easy to verify that the Fourier transform is unimodal if and only if $\gamma \leq \frac{1}{2}$.

Proposition 3.7 (Unimodality of the Fourier transform)

The Fourier transform (3.27) of the kernel describing the transformation from the original signal to the smoothed representation at a coarser level of scale is unimodal if and only if $\gamma \leq \frac{1}{2}$.

Proof: We would like the Fourier transform to decrease with $|u|$ and $|v|$ for all u and v in $[-\pi, \pi]$. Differentiation of (3.27) gives

$$\frac{\partial \psi}{\partial u} = -\psi(u, v) \sin u (1 - \gamma(1 + \cos v)) t \quad (3.32)$$

$$\frac{\partial \psi}{\partial v} = -\psi(u, v) \sin v (1 - \gamma(1 + \cos u)) t \quad (3.33)$$

The partial derivatives $\frac{\partial \psi}{\partial u}$ and $\frac{\partial \psi}{\partial v}$ have opposite signs to the variables u and v respectively if and only if the factors $(1 - \gamma(1 + \cos v))$ and $(1 - \gamma(1 + \cos u))$ are non-negative for all u and v , i.e. if and only if $\gamma \leq \frac{1}{2}$. If this condition is satisfied then any directional derivative in a direction away from the origin will be negative. \square

3.4.3 Separability

The transformation kernel is separable if and only if its Fourier transform is separable, i.e. if and only if $\psi_T(u, v)$ can be written on the form $U_T(u)V_T(v)$ for some functions U_T and V_T . From (3.27) we realize that this separation is possible if and only if $\gamma = 0$. Hence,

Proposition 3.8 (Separability of the 2-D discrete scale-space)

The convolution kernel associated with the scale-space representation defined by $L(x, y; t) = f(x, y)$ and

$$\frac{\partial L}{\partial t} = \frac{1}{2} \left((1 - \gamma) \nabla_y^2 L + \gamma \nabla_x^2 L \right) \quad (3.34)$$

is separable if and only if $\gamma = 0$. Then L is given by

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} T(m; t) \sum_{n=-\infty}^{\infty} T(n; t) f(x - m, y - n) \quad (t > 0) \quad (3.35)$$

where $T(n; t) = e^{-t} I_n(t)$ and I_n are the modified Bessel functions of integer order.

Proof: The Fourier transform $\psi_T(u, v)$ can be written on the form $U_T(u)V_T(v)$ for some functions U_T and V_T if and only if the term with $\cos u \cos v$ can be eliminated from the argument of the exponential function, i.e. if and only if γ is zero. In that case the Fourier transform reduces to

$$\psi_T(u, v) = e^{(-2 + \cos u + \cos v)t} = e^{(-1 + \cos u)t} e^{(-1 + \cos v)t} \quad (3.36)$$

which corresponds to separated smoothing with the one-dimensional discrete analogue of the Gaussian kernel, first along one coordinate direction and then along the other one. It can also be verified directly that (3.35) satisfies (3.34) by differentiating the kernel $T(m; t)T(n; t)$ with respect to t and then carrying out similar calculations as in the proof of Theorem 2.26, see Appendix A.2.1. \square

In other words, in the separable case the resulting two-dimensional discrete scale-space corresponds to repeated application of the one-dimensional scale-space concept along each coordinate direction.

3.4.4 Discrete Iteration

If as indicated in the proof of Proposition 3.6 the differential equation (3.25) is further discretized in scale using Euler's explicit method with scale step Δt , we get an iteration kernel with the coefficients.

$$\begin{pmatrix} \frac{\gamma\Delta t}{4} & \frac{(1-\gamma)\Delta t}{2} & \frac{\gamma\Delta t}{4} \\ \frac{(1-\gamma)\Delta t}{2} & 1 - (2-\gamma)\Delta t & \frac{(1-\gamma)\Delta t}{2} \\ \frac{\gamma\Delta t}{4} & \frac{(1-\gamma)\Delta t}{2} & \frac{\gamma\Delta t}{4} \end{pmatrix} \quad (3.37)$$

Clearly, this kernel is unimodal if and only if $\gamma \leq \frac{2}{3}$. One can show that it is separable if and only if $\gamma = \Delta t$, see below. In that case the corresponding one-dimensional kernel is a discrete scale-space kernel in the sense given in Definition 2.1 if and only if $\Delta t \leq \frac{1}{2}$, see (2.83). This gives a further indication that γ should not exceed $\frac{1}{2}$.

Observation 3.9 (Separability of the iteration kernel)

The iteration kernel (3.37), corresponding to discrete forward iteration with Euler's explicit method, is separable if and only if $\gamma = \Delta t$. In that case, the corresponding one-dimensional kernel is a discrete scale-space kernel if and only if $0 \leq \gamma \leq 1/2$.

Proof: Since the kernel is symmetric and the coefficients sum to one, the kernel is separable if and only if it can be written as a kernel $(a, 1 - 2a, a)$ convolved with itself, i.e. if and only if there exists an $a \geq 0$ such that

$$a^2 = \frac{\gamma\Delta t}{4} \quad (3.38)$$

$$a(1 - a) = \frac{(1 - \gamma)\Delta t}{2} \quad (3.39)$$

$$(1 - a)^2 = 1 - (2 - \gamma)\Delta t \quad (3.40)$$

The first equation has one non-negative root $a = \frac{\sqrt{\gamma\Delta t}}{2}$. Insertion into the second equation gives two conditions on Δt , either $\Delta t = 0$ or $\Delta t = \gamma$. One verifies that these roots satisfy the third equation. \square

It is worth mentioning, that if the extremum definitions, Definition 3.1 and Definition 3.2, would have been based on four-neighbours instead of eight-neighbours then $\gamma = 0$ would have appeared as a necessary condition in Theorem 3.2.

3.4.5 Spatial Isotropy

Another aspect which might affect the selection of γ is spatial isotropy. It is not clear that rotational invariance is a primary quality to be aimed at in the discrete case, since we are anyway locked to a fixed square grid. It is also far from obvious what should be meant by spatial isotropy in a discrete situation. Possibly, it is better to talk about the lack of spatial isotropy, spatial anisotropy or rotational asymmetry. However, since the Fourier transform is a continuous function of u and v , one can regard its variation as a function of the polar angle, given a fixed value of the radius, as one measure of this property⁶. If one expresses $\psi_T(u, v)$ in polar coordinates $u = \omega \cos \phi$, $v = \omega \sin \phi$ and examines the resulting expression,

$$\psi_T(\omega \cos \phi, \omega \sin \phi) = e^{h(\omega \cos \phi, \omega \sin \phi)t} \quad (3.41)$$

⁶This measure describes how much the amplitude of a sampled planar sine wave $e^{i\vec{\omega} \cdot \vec{x}}$ is suppressed as a function of the propagation direction $\vec{\omega}$.

where

$$\begin{aligned}
 h(\omega \cos \phi, \omega \sin \phi) = & -(2 - \gamma) + \\
 & (1 - \gamma)(\cos(\omega \cos \phi) + \cos(\omega \sin \phi)) + \\
 & \gamma \cos(\omega \cos \phi) \cos(\omega \sin \phi)
 \end{aligned} \tag{3.42}$$

one realizes that the value of γ , which gives the smallest angular variation for a fixed value of ω , depends on ω . Hence, with this formulation, the “rotational invariance” is scale dependent.

If $\gamma = \frac{1}{3}$ we get the nine-point operator ∇_9^2 , see Figure 3.4 and e.g Dahlquist [Dah74]. As we will see later it is not difficult to show that for large spatial scales, this value of γ gives the “most” isotropic second order approximation of the continuous Laplacian operator at the cost of a non-separable convolution kernel. But if we use a non-zero value of γ , it should be noted that the discrete scale-space representation can always be computed efficiently in the Fourier domain, using (3.27).

$$\begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 2/3 & -10/3 & 2/3 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

Figure 3.4: Computational molecule for the nine-point operator $h^2 \nabla_9^2$ corresponding to $\gamma = \frac{1}{3}$. At coarse scales this value of γ gives the spatially least anisotropic approximation of the continuous diffusion equation.

Proposition 3.10 (Rotational invariance in the Fourier domain)

The value of γ that gives the least rotational invariance for large scale phenomena in the solution to the differential equation (3.25) is $\gamma = \frac{1}{3}$.

Proof: Express $\psi_T(u, v)$ on the form $\psi_T(u, v) = e^{h(u,v)t}$ and introduce polar coordinates (ω, ϕ) by

$$\begin{cases} u = \omega \cos \phi \\ v = \omega \sin \phi \end{cases} \tag{3.43}$$

Then the Taylor expansion of h for small values of ω is, see Appendix A.2.3,

$$h(\omega \cos \phi, \omega \sin \phi) = -\frac{1}{2}\omega^2 + \frac{1}{24}(1 + (6\gamma - 2) \cos^2 \phi \sin^2 \phi)\omega^4 + O(\omega^6) \tag{3.44}$$

where the $O(\omega^6)$ term depends on both ϕ and γ . Observe that if $\gamma = \frac{1}{3}$ then the ϕ -dependence decreases with ω as ω^6 instead of as ω^4 . \square

This means that $\gamma = \frac{1}{3}$ asymptotically, i.e. with increasing spatial scale, gives the most isotropic smoothing effect on coarse scale events. The reason why we desire spatial isotropy at coarse scales rather than at fine scales is because the grid effects become smaller for coarse scale phenomena, which in turn makes it more meaningful to talk about rotational invariance. In this context it should be noted that, if we use a non-zero value of γ , the discrete scale-space representation can always be calculated efficiently in the Fourier domain using (3.27).

3.4.6 Remaining Degree of Freedom

We leave the question about definite selection of γ open. However, from a computational point of view it seems very plausible that $\gamma = 0$ should not be a too bad choice. As we will see in the next chapter, the closed-form expressions for some derived quantities will also become simple in this case. A possible disadvantage with that approach is that it emphasizes the x - and y -directions as being special directions.

3.5 2D Summary and Discussion

The proper way to apply the scale-space theory to two-dimensional discrete images is apparently by discretization of the diffusion equation. Starting from a requirement that local extrema must not be enhanced when the continuous scale parameter is increased we have shown that a necessary and sufficient condition for a family of derived representations to be a scale-space family is that it satisfies the differential equation

$$\frac{\partial L}{\partial t} = \frac{C}{2} \left((1 - \gamma) \nabla_y^2 L + \gamma \nabla_x^2 L \right) \quad (3.45)$$

for some real constants C and γ where $\gamma \in [0, 1]$. Our recommendation is that γ should not exceed $\frac{1}{2}$. $\gamma = 0$ gives a separable convolution kernel, while $\gamma = \frac{1}{3}$ leads to a spatially more isotropic smoothing effect on coarse scale objects. In the separable case the scale-space representation can be calculated by separated convolution with the presented one-dimensional discrete analogue of the Gaussian kernel, $T(n; t)$.

3.6 Possible Extensions

The treatment so far has been restricted to one- and two-dimensional signals defined on infinite and uniformly sampled square grids using uniform smoothing of all grid points, because this is the natural special case we have been interested in when dealing with image data generated from standard camera devices. However, there is nothing in principle that prevents those restrictions from being removed.

3.6.1 Anisotropic Smoothing

In a recent paper, Perona and Malik [Per90] propose the use of anisotropic smoothing. The motivation behind their approach is to try to avoid or to reduce the shape distortions introduced by scale-space smoothing across object boundaries, particularly with application to edges. The way they suggest to prevent this from happening is by modifying the diffusion coefficients such as to favour intraregion smoothing to interregion smoothing.

Using the maximum principle they show that the resulting anisotropic scale-space representation satisfies a similar suppression property for local extrema as was the basis for Koenderink and van Doorns [Koe84] continuous scale-space formulation and this two-dimensional discrete treatment. From the proofs of Theorem 3.2 and Theorem 3.3 it is obvious that the ideas behind the discrete scale-space concept can be easily extended to such anisotropic diffusion if we let the coefficients of the linear operator \mathcal{A} vary with both the scale parameter, the grey-level values and the spatial coordinates. However, when introducing an anisotropic diffusion equation we have to sacrifice the convolution form of smoothing as well the semi-group property. Therefore, when proving the necessity of the representation a certain form

of the smoothing formula may have to be assumed, e.g. of the form (3.6) where \mathcal{A} depends on the scale parameter and is no longer shift invariant.

In this work we have throughout made use of uniform smoothing all over the image at the possible cost of such smoothing across “object boundaries”. The motivation behind this choice is that we are mainly interested in using the scale-space representation for actually *detecting* image structures. Therefore, we would like to introduce as few commitments as possible into the process. The approach we instead have adopted is to first detect candidate regions of interest. Then, once such a candidate has been detected as a region in an image, its localization can be improved in various ways. For an example, compare with the integration of blob detection and edge detection in Section 9.1.

Modifying the diffusion coefficient requires some kind of a priori information concerning which structures in the image should be smoothed and which structures should not. In Peronas and Maliks case there is a tuning function, giving the diffusion coefficient as function of the gradient magnitude, that needs to be determined.

There is also another aspect of the approach we find somewhat dubious. When the scale parameter t tends to infinity the solution to the anisotropic diffusion equation will tend to a function that is not constant, but contains various sharp edges. Hence, the choice of the tuning function in the method somehow implies an implicit assumption about a “final segmentation” of the image. It is not clear that such a concept exists or can be defined rigorously.

3.6.2 Higher Dimensions

In Section 3.3.4 we showed that the ideas behind the two-dimensional scale-space concept could be directly applied to one-dimensional signals. Similarly, they can be extended to arbitrary n -dimensional discrete signals $Z^n \rightarrow R$, although the amount of data generated in a practical application may increase dramatically with the number of dimensions. Analogies to the definitions given in Section 3.3.1 can be obtained almost directly just by replacement of Z^2 by Z^n . All we have to take care of is that the symmetry condition in Definition 3.3 is stated properly. One can require that

$$T(-x_1, x_2, \dots, x_n; t) = T(x_1, x_2, \dots, x_n; t) \quad (3.46)$$

$$T(P_k^n(x_1, x_2, \dots, x_n); t) = T(x_1, x_2, \dots, x_n; t) \quad (3.47)$$

hold for all $(x_1, \dots, x_n) \in Z^n$, all $t \in R_+$ and for all possible permutations P_k^n of n elements.

The proof of Lemma 3.1 is independent of the number of dimensions. In the analogies of Theorem 3.2 and Theorem 3.3 we will have to replace the operator $\alpha \nabla_x^2 + \beta \nabla_x^2$ with a corresponding n -dimensional discrete operator ∇_{nD}^2 approximating the continuous Laplacian. Only the coefficients in ∇_{nD}^2 corresponding to the nearest neighbours of a point can be non-zero because of the locality requirement induced by the non-enhancement of local extrema. Moreover, by studying a piecewise constant signal one verifies that the sum of the coefficients must be zero. When the symmetry constraints have been applied to this operator there will be n remaining parameters left to be determined. One of those can be removed since it will only affect the scaling of the scale parameter. In the separable case the corresponding scale-space representation can be computed by separated convolution with the one-dimensional discrete analogue of the Gaussian kernel along each coordinate direction.

3.6.3 Finite Data

A practical problem that always arises when implementing linear filtering concerns what to do with those pixels near the image boundary for which a part of the filter mask stretches outside the current image. In this treatment we have throughout assumed the signals to be defined for all the points in an infinite square grid and not gone into the complications that occur due to the boundary effects if the signal function is defined only for a finite subset of the integers.

The most conservative outlook is, of course, to regard the output as undefined as soon as a computation requires some image data outside the available domain. In the case with scale-space smoothing this approach would, however, lead to a rapidly decreasing image size, something hardly desirable, since the peripheral coefficients decrease towards zero very rapidly and the (untruncated) convolution masks actually have infinite support. A variety of ad hoc methods have been used/proposed to solve for this problem; zero value extension, periodic wrap-around, mirroring at the boundaries, subtraction of the steady-state component, solving the diffusion equation with adiabatic boundary conditions etc. However, we believe that neither of these techniques can give a desired results in all situations. The result depends too much on how the image behaves near the boundary. For some very simple cases it might be enough do an ad hoc extension. But this requires some kind of a priori information about what can be expected to be in the scene.

There is no getting away from the fact that all finite images have boundaries and that problems arise if one tries to analyze objects near them. By necessity, the peripheral image values of a smoothed finite image will be less reliable than the central ones. Instead we think that if one really needs accurate values near the boundary of an image then one should instead try to acquire additional image data such that the convolution operation becomes well-defined. This can be easily achieved within the active vision paradigm simply by moving the camera such that image values become available in a sufficiently large neighbourhood of the object of interest. We think that the task of analysing an object manifesting itself at a certain scale requires input data in a region around the object. The width of this frame depends both on the current level of scale and the prescribed accuracy of the analysis.

If one because of computational efficiency and simplicity uses the extension approach there is one aspect we would like to emphasize. If one wants the semi-group property to hold exactly between arbitrary scale levels (except for numerical rounding and truncation errors) it is necessary that the representations at all scales are generated *directly* from the original extended signal using the approximation (4.20). If cascade smoothing is used then the truncation of the intermediate representations at the boundaries implies that the semi-group property will be violated unless the size of the intermediate representations is increased.

Of course, a genuinely finite approach is also possible. In this presentation we have chosen not to develop the subject, since the associated problems are somehow artificial and difficult to handle in a consistent manner, although in the one-dimensional case the concepts of sign-regularity and semi-groups of totally positive matrices in principle provide possible tools for handling this issue.

One way of avoiding both the infiniteness and the boundary problems could be by using a spherical camera. Then, the ordinary planar camera geometry would appear as an approximate description for foveal vision, i.e. small solid angles in the central field of vision.

3.6.4 Other Types of Grids

Neither is the assumption of a square grid any necessary restriction. The same type of treatment can be carried out on e.g. a hexagonal grid with the semi-group property preserved, and also in principle on a grid corresponding to non-uniform spatial sampling provided that the diffusion coefficients are modified accordingly. In the latter case some a priori form of the smoothing formula may have to be assumed when proving the necessity of the representation. An interesting case to consider might actually be the non-uniformly sampled spherical camera.

3.6.5 Further Work

Let us finally point out that there is one main issue that we have not considered in this treatment, namely *scale dependent spatial sampling*. This issue is certainly of importance in order to improve the computational efficiency both when computing the representation and for algorithms working on the data. The scale-space concept outlined here uses the same spatial resolution at all levels of scale. The pyramid representations on the other hand imply a fixed relation between scale and resolution as well as a fixed scale step that one cannot go below. In fact, the scale is given directly by the resolution.

Since the smoothed images at coarser scales will get more and more redundant it seems plausible that some kind of subsampling could be done at the coarser scales without very much loss of information. It would be interesting to carry out an analysis about how much information would be lost by such an operation and regarding to which extent the a subsampling operator could be introduced in this representation, anyway maintaining the continuous scale parameter and without introducing any severe discontinuities along the scale direction that could lead be a potential source to numerical difficulties for algorithms working on the output from the representation.

Chapter 4

Implementational Implications and Conclusions

In this chapter we will first describe some computational and implementational implications of the presented discrete scale-space theory and then conclude this overall treatment by summarizing the main results.

4.1 Discrete Definitions of “Derivatives”

The scale-space representation obtained from the discrete theory has some implementational advantages compared to the commonly adapted approach, where the scale-space implementation is based on different versions of the sampled Gaussian kernel. It allows for discrete definitions of the derivatives of the Gaussian arising in the N-jet representation suggested by Koenderink and van Doorn [Koe87].

4.1.1 The Laplacian of the Gaussian

Consider for instance the computation of the Laplacian of the Gaussian $\nabla^2 G$ of an image f . It is well-known that $\nabla^2 G$ is not a separable kernel — a clear disadvantage in terms of computational efficiency, unless the convolutions are carried out in the frequency domain. It is also known that the straightforward implementation consisting of smoothing with the sampled Gaussian kernel followed by application of a discrete Laplacian gives unsatisfactory results, since the values obtained in this way deviate too much from the sampled values of $\nabla^2 G$. A common approach to circumvent this problem has been by the calculation of difference of Gaussians (DOG) instead, see e.g. Marr and Hildreth [Mar80]. However, this method will only give approximate results, and the selection of the scale-step Δt leads to a numerical trade-off between cancellation of digits and accuracy in the representation. It also requires the computation of two smoothed representations instead of one.

4.1.1.1 Approximations based on the Continuous Theory

To summarize, we have that there are several possible ways to get a discrete approximation of the Laplacian of the Gaussian of a signal; *discrete convolution with the sampled Gaussian kernel*

$$(\nabla^2 L)(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} (\nabla^2 G)(m, n; t) f(x - m, y - n) \quad (x, y) \in Z^2 \quad (4.1)$$

corresponding to the application of the rectangle rule of integration to the convolution integral

$$(\nabla^2 L)(x, y; t) = \int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} (\nabla^2 G)(\xi, \eta; t) f(x - \xi, y - \eta) d\xi d\eta \quad (x, y) \in R^2 \quad (4.2)$$

discrete convolution with the sampled Gaussian kernel followed by the discrete Laplacian

$$(\tilde{\nabla}^2 L)(x, y; t) = \nabla_{\gamma}^2 \left(\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} G(m, n; t) f(x - m, y - n) \right) \quad (x, y) \in Z^2 \quad (4.3)$$

corresponding to a two-step discrete approximation of at first the convolution integral

$$L(x, y; t) = \int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} G(\xi, \eta; t) f(x - \xi, y - \eta) d\xi d\eta \quad (x, y) \in R^2 \quad (4.4)$$

using the rectangle rule of integration and then the Laplacian operator applied to that result

$$(\nabla^2 L)(x, y; t) = \nabla^2 \left(\int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} G(\xi, \eta; t) f(x - \xi, y - \eta) d\xi d\eta \right) \quad (x, y) \in R^2 \quad (4.5)$$

as well as *difference of sampled Gaussians*

$$(\tilde{\nabla}^2 L)(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{2(G(m, n; t + \Delta t_1) - G(m, n; t - \Delta t_2))}{\Delta t_1 + \Delta t_2} f(x - m, y - n) \quad (x, y) \in \mathbb{Z}^2 \quad (4.6)$$

corresponding to discretization of the derivative with respect to the scale parameter in the diffusion equation

$$(\nabla^2 L)(x, y; t) = 2 \frac{\partial L}{\partial t}(x, y; t) \approx \frac{L(x, y; t + \Delta t_1) - L(x, y; t - \Delta t_2)}{\Delta t_1 + \Delta t_2} \quad (x, y) \in R^2 \quad (4.7)$$

where L in turn is given by the convolution integral (4.4). Of course, the sampled Gaussian kernel may in all these cases be replaced by the integrated Gaussian kernel in order to yield “more true approximations” at fine scales, compare also with Section 2.6.3.

In the continuous case the various expressions (4.2), (4.5) and (4.7) will all be equivalent when Δt_1 and Δt_2 tend to zero provided that the signal f is sufficiently regular. However, the different discrete approximations (4.1), (4.3) and (4.6) will *not* give the same but different output results, not even in the limit case. The main reason why these expressions are no longer equivalent is because the operators involved, which commute in the continuous case, do not commute after discretization.

4.1.1.2 Discrete Definition of the Laplacian of the Gaussian

The discrete scale-space concept outlined in the previous chapters allows for a discrete definition of the Laplacian of the Gaussian of an image, for which the discrete analogies of (4.2), (4.5) and (4.7) are all maintained equal. From the diffusion equation (3.25) we have that

$$\frac{\partial L}{\partial t} = \frac{1}{2}((1 - \gamma)\nabla_5^2 L + \gamma\nabla_{\times}^2 L) = \frac{1}{2}\nabla_{\gamma}^2(T * f) = \frac{1}{2}T * (\nabla_{\gamma}^2 f) = \frac{1}{2}(\nabla_{\gamma}^2 T) * f \quad (4.8)$$

In this discrete case ∇_{γ}^2 commutes with the smoothing operator and we can compute the discrete analogue of the Laplacian of the Gaussian of an image in several ways. We take the output of those *equivalent* operations as the discrete definition of this concept. When implementing this operation we have several possibilities to compute the output:

- in two sweeps — a smoothing step with the discrete analogue of the Gaussian kernel followed by the application of the discrete Laplacian operator ∇_γ^2 .
- a discrete Laplacian step applied to the original signal followed by smoothing.
- by computation of the Laplacian of the smoothing kernel as a first step and then by convolving the signal with that kernel.
- as the limit case of differences of discrete Gaussians, compare with (4.7).

Note again that *all methods give exactly the same result, since the (discrete) smoothing operator commutes with the (discrete) Laplacian*, provided that the same value of γ is used in all discrete Laplacian operators. With the first method the amount of computational work required to compute the discrete analogue of the Laplacian of the Gaussian of an image is, if $\gamma = 0$, just one separable two-dimensional smoothing step followed by an efficient application of the discrete Laplacian. The second method can be slightly advantageous for algorithms where only the Laplacian of the Gaussian is required, since then the Laplacian step needs to be carried out just once. The third method destroys the separability¹ and should probably be avoided. The first method is really the one to prefer in situations where both the smoothed image and its spatial derivatives are required.

4.1.2 The Gradient of the Gaussian

The discrete scale-space does also provide a convenient formulation of gradient calculations. For simplicity, consider the separable case when $\gamma = 0$. Then a one-dimensional analysis is sufficient. Let δ_x denote the well-known central difference operator in the x -direction defined by

$$(\delta_x f)(x, y) = \frac{1}{2}(f(x+1, y) - f(x-1, y)) \quad (4.9)$$

Similarly to the previous case, $\delta_x L$ can be computed either by application of δ_x on the smoothed image, the original image or on the smoothing kernel. The effect of this gradient calculation is given by the effect δ_x has on the one-dimensional kernel applied in the x -direction. From a recurrence relation for the modified Bessel functions (4.22) we get an explicit analytical expression for $(\delta_x T)(x; t)$, namely

$$(\delta_x T)(x; t) = \frac{1}{2}e^{-t}(I_{x+1}(t) - I_{x-1}(t)) = \frac{1}{2}e^{-t}\left(-\frac{2x}{t}I_x(t)\right) = -\frac{x}{t}T(x; t) \quad (4.10)$$

Note the similarity with the derivative of the continuous Gaussian kernel

$$\left(\frac{\partial}{\partial x}G\right)(x; t) = -\frac{x}{t}G(x; t) \quad (4.11)$$

4.1.2.1 Approximations of the Continuous Equations

If one instead would have used the approach based on the sampled Gaussian kernel it is clear that convolution with the sampled x -gradient of the Gaussian

$$(\delta_x \tilde{L})(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{\partial G}{\partial x}(m, n; t) f(x-m, y-n) \quad (x, y) \in Z^2 \quad (4.12)$$

¹As mentioned earlier, the convolution kernel is separable if and only if $\gamma = 0$

corresponding to the rectangle rule of integration approximation applied to the integral

$$\left(\frac{\partial}{\partial x}L\right)(x, y; t) = \int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} \left(\frac{\partial G}{\partial x}\right)(\xi, \eta; t) f(x - \xi, y - \eta) d\xi d\eta \quad (x, y) \in R^2 \quad (4.13)$$

would not have given the same result as application of δ_x on the “scale-space representation” generated by smoothing with the sampled Gaussian kernel

$$(\delta_x \tilde{L})(x, y; t) = \delta_x \left(\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} G(m, n; t) f(x - m, y - n) \right) \quad (x, y) \in Z^2 \quad (4.14)$$

corresponding to a two-step discrete approximation of at first the convolution integral (4.4) using the rectangle rule of integration and then the x -gradient operator applied to that result

$$\left(\frac{\partial}{\partial x}L\right)(x, y; t) = \frac{\partial}{\partial x} \left(\int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} G(\xi, \eta; t) f(x - \xi, y - \eta) d\xi d\eta \right) \quad (x, y) \in R^2 \quad (4.15)$$

Gradient approximations of type suggested in (4.9), although based on binomial kernels in pyramids, have been used e.g. by Crowley [Cro87].

Second order derivatives can be obtained either by application of δ_x twice or by using the well-known discrete approximation to the second derivative ∇_3^2 . The first approach is advantageous in the sense that the discrete analogue to a derivative of any order can be obtained by repeated application of the δ_x operator. The second approach gives a higher accuracy and also preserves the coupling between the second order spatial derivative with the first order derivative with respect to the scale parameter as required in the diffusion equation. Higher order “discrete derivatives” can be formed by combinations of these operators.

4.1.3 Normalization

Another disadvantage with the sampled Gaussian kernel appears for small values of t . Then, as the continuous Gaussian kernel tends towards the continuous delta function when t tends to zero, the central coefficient may get very dominant. Even though the integral of the continuous kernel is normalized to one, the central peak can drive the sum of the filter coefficients to a value substantially greater than one². This negative effect at fine scales is further amplified when derivatives of the Gaussian and/or when difference operators are applied to the smoothed grey-level images. Such problems do not occur with the discrete analogue of the Gaussian kernel, since this kernel tends to the discrete delta function as t tends to zero and the filter coefficients always sum up to one.

$$\sum_{n=-\infty}^{\infty} T(n; t) = 1 \quad (4.16)$$

Other normalization conditions that are trivially satisfied are

$$\sum_{n=0}^{\infty} (\delta_x T)(n; t) = -T(0; t) \quad (4.17)$$

²It has been suggested that this effect should be compensated for by renormalization of the filter coefficient sequence. But this operation does not solve the major problem since the mutual relation between the coefficients remains unchanged anyway. It only leads to a rescaling of the output image. The problem with the sampled Gaussian kernel for small values of t is rather that it appears as having a smaller t -value than it should.

$$\sum_{n=-\infty}^{\infty} (\nabla_3^2 T)(n; t) = 0 \quad (4.18)$$

4.1.4 Summary

The effects mentioned in this section are all due to the difference between continuous theory and discrete implementation. As indicated above, *the main reason why they arise is because the involved operators, which commute in the continuous case, do not commute when the discretization operator is involved*, compare also with violated semi-group property discussed in Section 2.6.1. With the discrete scale-space theory presented in this treatment we feel that we have accomplished a structured way to eliminate this kind of problems.

4.2 Kernel Graphs

In order to illustrate the difference between the discrete analogue of the Gaussian kernel and the continuous Gaussian kernel we have drawn their graphs at a few levels of scale together with corresponding results for the first and second order derivatives and differences, see Figures 4.1-4.5. For comparison the sampled Gaussian kernel and the integrated Gaussian kernel have been shown next to these graphs. As we see, the difference between the two kernels is largest at fine levels of scale and becomes smaller as the kernels approach each other at coarser levels of scale.

4.3 Implementing Scale-Space Smoothing

According to the definition of the scale-space for discrete signals, the representation of a one-dimensional signal f at a scale-level t is given by,

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t) f(x - n) \quad (x \in Z, t > 0) \quad (4.19)$$

where $T(n; t) = e^{-t} I_n(t)$. When this transformation is to be implemented computationally there are a few numerical problems that must be considered:

- The infinite convolution sum must be replaced with a finite one.
- Normally, the modified Bessel functions are not available as standard library routines. Therefore, we must design an algorithm to generate the required filter coefficients $T(n; t)$ for a given value of t .
- A realistic signal is finite, but a finite approximation of (4.19) might need additional values.

In this section we will discuss the first two items. We will not go into the complications, which arise from finite signals. Instead we assume that f is defined for all those integers, where signal values are required for our algorithms.

4.3.1 Truncation and Filter Coefficient Generation

A reasonable approach to approximate (4.19) is to truncate³ the infinite sum for some sufficiently large value of N ,

$$L(x; t) \approx \sum_{n=-N}^N T(n; t) f(x - n) \quad (x \in Z, t > 0) \quad (4.20)$$

chosen such that the absolute error in L due to truncation does not exceed a given error bound $\tilde{\epsilon}$. If we assume that f is bounded ($|f(x)| \leq M$) we get the sufficient condition

$$2 \sum_{n=N+1}^{\infty} T(n; t) \leq \frac{\tilde{\epsilon}}{M} = \epsilon_{trunc} \quad (4.21)$$

An easy way to generate the filter coefficients is to use the recurrence relation, see Abramowitz [Abr64] (9.6.26),

$$I_{n-1}(t) - I_{n+1}(t) = \frac{2n}{t} I_n(t) \quad (4.22)$$

One can use Miller's algorithm, see e.g. Press et al. [Pre86] p142, and start the recurrence with an arbitrary seed $I_{N_{start}} = 1$ and $I_{N_{start}+1} = 0$ for a sufficiently large start index N_{start} . As n decreases the iterates obtained from (4.22) will successively approach the correct solution. The sequence of iterates can be normalized if $I_0(t)$ is computed by a separate routine. Once a sufficient number of filter coefficients has been computed, it is easy to determine how many that are actually needed from the condition $\sum_{n=-N}^N T(n; t) \geq 1 - 2\epsilon_{trunc}$. A more detailed investigation as well as an algorithm generating the filter coefficients $T(n; t)$ can be found in [Lin88] Section 5 and Appendix A.3.

Another possibility is of course to start from the expression for the Fourier transform (2.47) and perform the convolutions in the frequency domain instead. At coarse scales this method will be computationally far more efficient than convolutions carried out in the spatial domain. Then also, the truncation error in the convolution integral can be expected to be substantially reduced, since the only truncation that occurs in the frequency domain is because of the finite size of the actual image subject to the Fourier transformations. However, some precautions may have to be taken in order to reduce the wrap-around effects, for instance by extending the signal before the fast Fourier transform is carried out.

In the separable case when $\gamma = 0$ the two-dimensional scale-space smoothing can be implemented by application of the one-dimensional smoothing formula along each coordinate direction. For square filter masks the truncation error ϵ_{2D} in the two-dimensional case is related to the truncation error ϵ_{1D} in the one-dimensional case by

$$1 - \epsilon_{2D} = (1 - \epsilon_{1D})^2 \quad (4.23)$$

³Observe that by truncating the infinite kernel we actually violate the scale-space conditions and can no longer assume the scale-space property to hold exactly between two scale levels. Actually, we are not even guaranteed that the truncated convolution kernel is a discrete scale-space kernel. One possible approach to reduce this problem might be by trying to find the generalized binomial kernel of given size that in some sense is the closest approximation to the infinite support discrete analogue of the Gaussian kernel. Also the possible ringing in the Fourier domain introduced by truncation in the spatial domain might cause problems, and it could possibly be better to "round off" the kernels at the tails. In order to avoid these complications we instead assume that the upper bound on the truncation error ϵ_{trunc} is selected small enough such that those effect can be readily ignored.

4.3.2 Application to the N -jet

Let us again point out that once the scale-space smoothing step has been carried out, the “discrete derivatives” can be computed directly from the smoothed grey-level images by application of their corresponding filter masks (containing just a small number of non-zero coefficients) on the smoothed data. Hence, there is no need to redo any smoothing by convolving the image with any large size filter masks derived from derivatives of the Gaussian kernel.

Note, however, that the absolute error due to truncation of the infinite discrete analogue of the Gaussian kernel increases by this operation, with a factor of 2 when computing first order differences and a factor of 8 when computing the discrete analogue of the Laplacian of the Gaussian. Therefore the truncation error should be selected small enough in the smoothing approximation.

4.4 Summary and Discussion

We have seen that the discrete scale-space representation given by discretization of the diffusion equation has computational advantages compared to the commonly used approach, where the scale-space implementation is based on various versions of the sampled Gaussian kernel. It can be expected that the difference is largest for small values of the scale parameter, when the sampled Gaussian kernel and the discrete analogue of the Gaussian kernel deviate as most. When the scale parameter increases these two kernels approach each other, see also Section 2.5.2.1, and we might expect that the difference becomes smaller. This effect can also be understood from another point of view. At coarse levels of scale the large scale phenomena dominate in the scale-space representation, which means that the grid effects become smaller, since a characteristic length in the smoothed image will be large compared to the distance between adjacent grid points. It is difficult to say generally how large the numerical effects are in an actual implementation and how seriously they affect the output result, since this is very much determined by the algorithms working on the scale-space representation and the goal of the analysis in which the scale-space part is just one of the modules. However, in Figures 4.6-4.7 we have tried to visualize how some measures of the difference between the sampled Gaussian kernel and the discrete analogue of the Gaussian kernel behave as a function of the scale parameter. Tabulated values for a few values of t are given in Appendix A.3.1. The graphs verify that the difference is largest for small values of t and show that it increases with higher order differences. Do also note the large difference between the sampled second derivative of the Gaussian kernel and the second difference of the sampled Gaussian kernel.

Finally, it should be explicitly stressed that the discrete scale-space theory is closely linked to the continuous scale-space theory through the discretization of the diffusion equation. This means that continuous results can be transferred to discrete implementation provided that the discretization is done correctly. The discussion in the previous section is intended to exemplify the technique.

4.5 Conclusions: Scale-Space for Discrete Signals

The first part of this thesis gives a basic and extensive treatment of discrete aspects of the scale-space theory. A genuinely discrete scale-space theory is developed and its connection

to the continuous scale-space theory is explained. Special attention is given to discretization effects, which occur when results from the continuous scale-space theory are to be implemented computationally. The one-dimensional problem is solved completely in an axiomatic manner. The two-dimensional problem is more complex, but we answer the question about how the two-dimensional discrete scale-space should be constructed. The main results can be summarized as follows (References to central theorems and appropriate sections of the thesis are given within parenthesis):

- The proper way to apply the scale-space theory to discrete signals is by discretization of the diffusion equation, *not* the convolution integral (Thms. 2.20, 2.26, 3.2, 3.3, Prop. 2.25, and Secs. 4.1, 2.6).
- The discrete scale-space obtained in this way can be described by convolution with the kernel $T(n; t)$, which is the discrete analogue of the Gaussian kernel (Thm. 2.20, Prop. 3.8 and Sec. 3.4).
- A scale-space implementation based on the sampled Gaussian kernel might lead to undesirable effects and computational problems, especially at fine levels of scale (Prop. 2.25 and Sec. 4.1).
- The one-dimensional discrete smoothing transformations can be characterized exactly and a complete catalogue is given (Thms. 2.14, 2.15).
- All one-dimensional discrete smoothing transformations with finite support arise from repeated averaging over two adjacent elements (Thm. 2.18 and Props. 2.5, 2.6). The kernel $T(n; t)$ describes the limit case of such an averaging process (Prop. 2.28).
- The symmetric one-dimensional discrete smoothing kernels are non-negative and unimodal, both in the spatial and the frequency domain (Props. 2.2, 2.3, 2.9, 2.11 and Sec. 2.4). These conditions are necessary but not sufficient (Obs. 2.30).

The important idea with the scale-space concept suggested in this paper is that the discrete nature of the implementation has been taken into account already in the theoretical formulation of the scale-space representation.

4.6 Philosophy

The formulation in terms of the diffusion equation appears to be a natural unification of the existing scale-space theory for continuous signals and the presented scale-space theory for discrete signals. One could say that the *primary* formulation of the scale-space theory is by the *diffusion equation*. Then,

- the Gaussian kernel appears as the fundamental solution of the *continuous* diffusion equation.
- the discrete analogue of the Gaussian kernel is the fundamental solution of the *discrete* diffusion equation.

During recent years “Gaussian smoothing” has become a wide-spread concept in the computer vision society. In view of these results one should rather say “diffusion smoothing”.

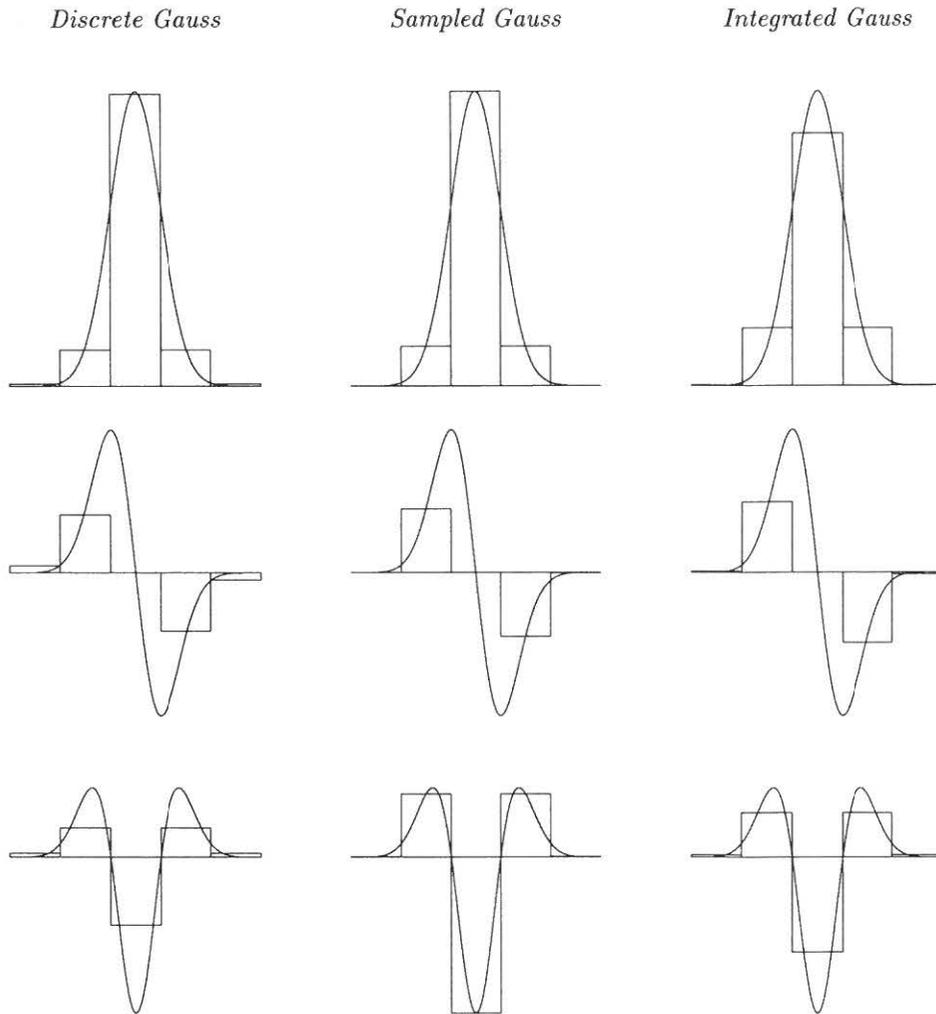


Figure 4.1: $t = 0.25$: Comparisons between the discrete analogue of the Gaussian kernel (left column), the sampled Gaussian kernel (middle column) and the integrated Gaussian kernel (right column). The upper row shows the raw smoothing kernel, the middle row the first order differences/derivatives and the lower row the second order difference/derivatives. The block diagrams indicate the discrete kernels and the smooth curve the continuous Gaussian.

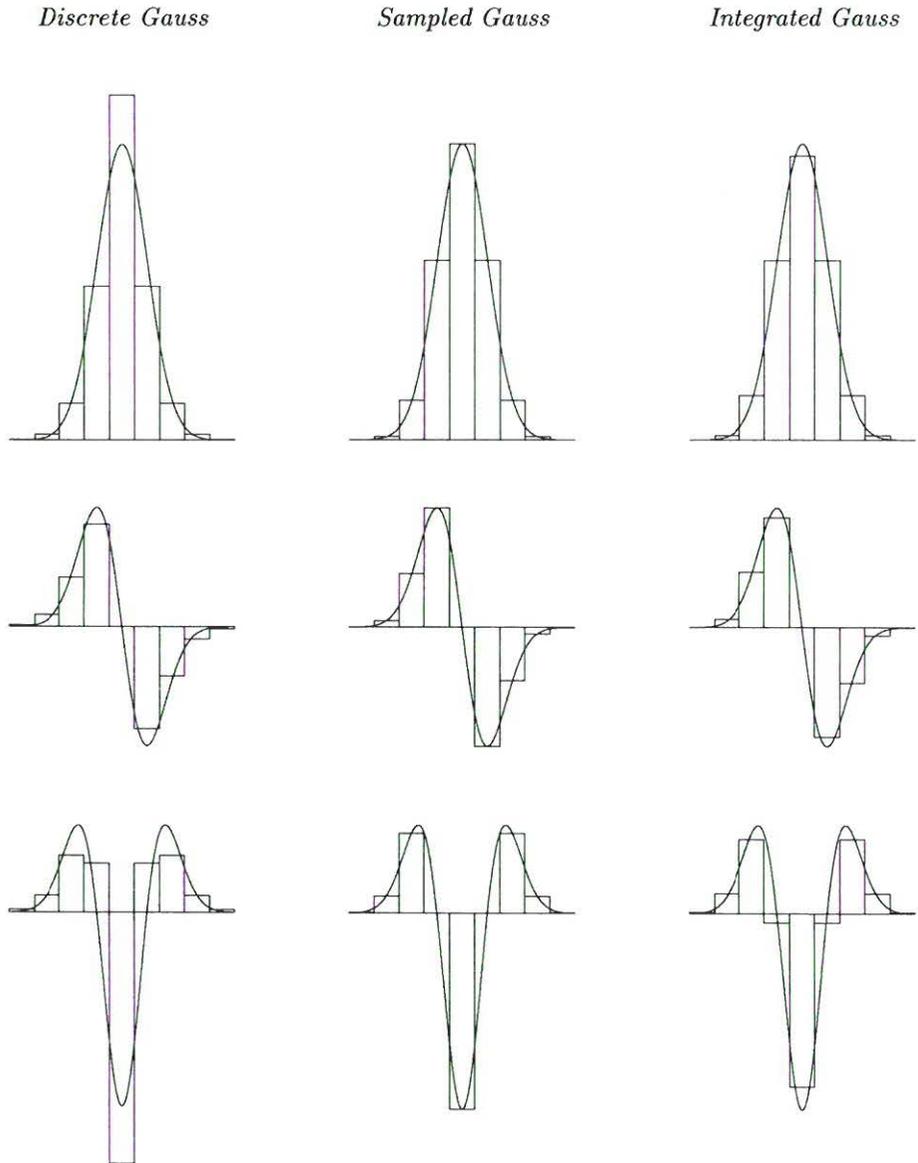


Figure 4.2: $t = 1.0$: Comparisons between the discrete analogue of the Gaussian kernel (left column), the sampled Gaussian kernel (middle column) and the integrated Gaussian kernel (right column). The upper row shows the raw smoothing kernel, the middle row the first order differences/derivatives and the lower row the second order difference/derivatives. The block diagrams indicate the discrete kernels and the smooth curve the continuous Gaussian.

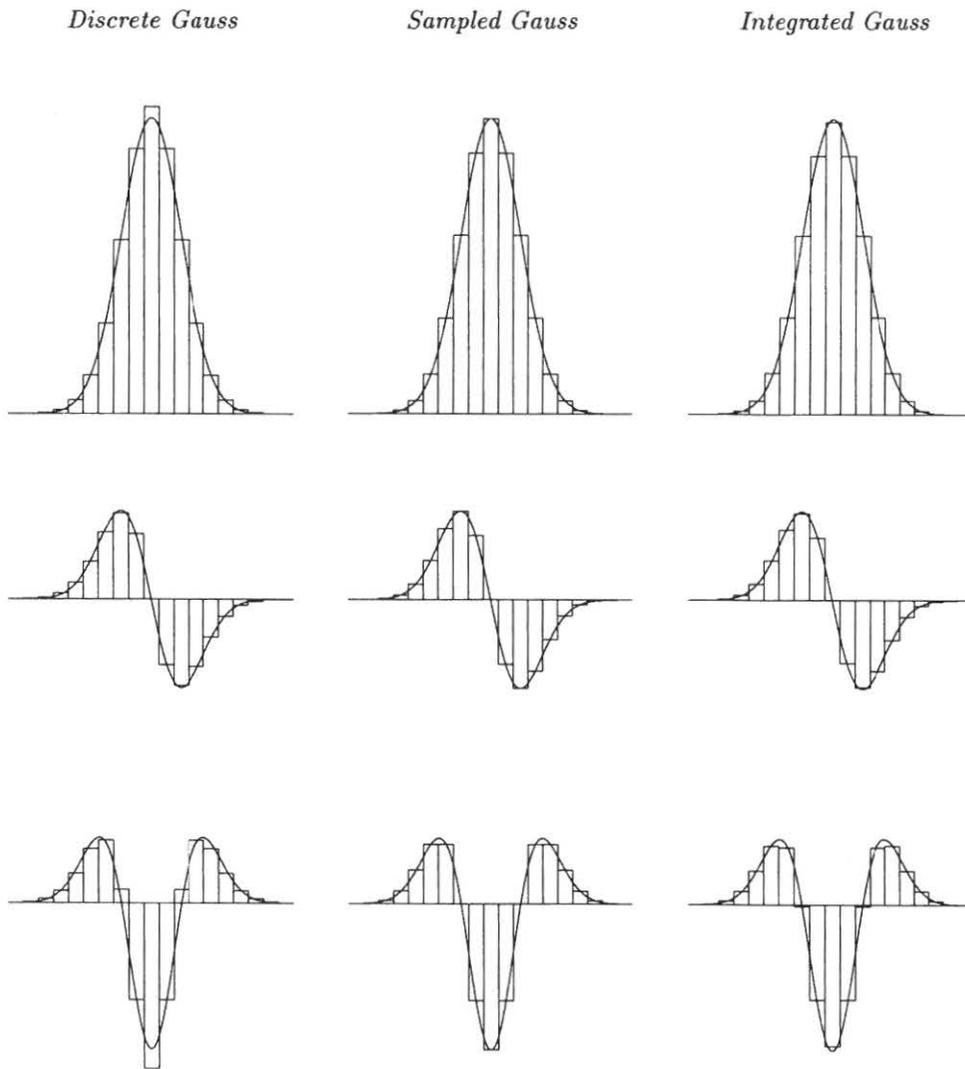


Figure 4.3: $t = 4.0$: Comparisons between the discrete analogue of the Gaussian kernel (left column), the sampled Gaussian kernel (middle column) and the integrated Gaussian kernel (right column). The upper row shows the raw smoothing kernel, the middle row the first order differences/derivatives and the lower row the second order difference/derivatives. The block diagrams indicate the discrete kernels and the smooth curve the continuous Gaussian.

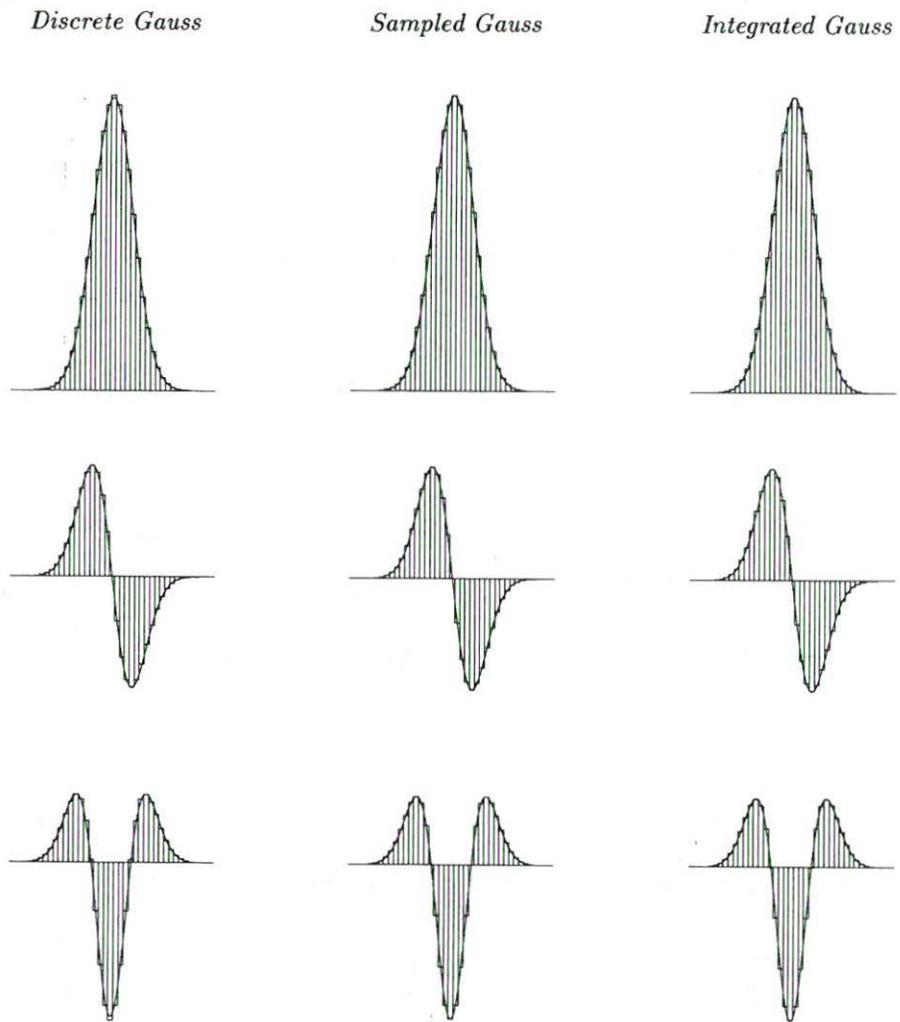


Figure 4.4: $t = 16.0$: Comparisons between the discrete analogue of the Gaussian kernel (left column), the sampled Gaussian kernel (middle column) and the integrated Gaussian kernel (right column). The upper row shows the raw smoothing kernel, the middle row the first order differences/derivatives and the lower row the second order difference/derivatives. The block diagrams indicate the discrete kernels and the smooth curve the continuous Gaussian.

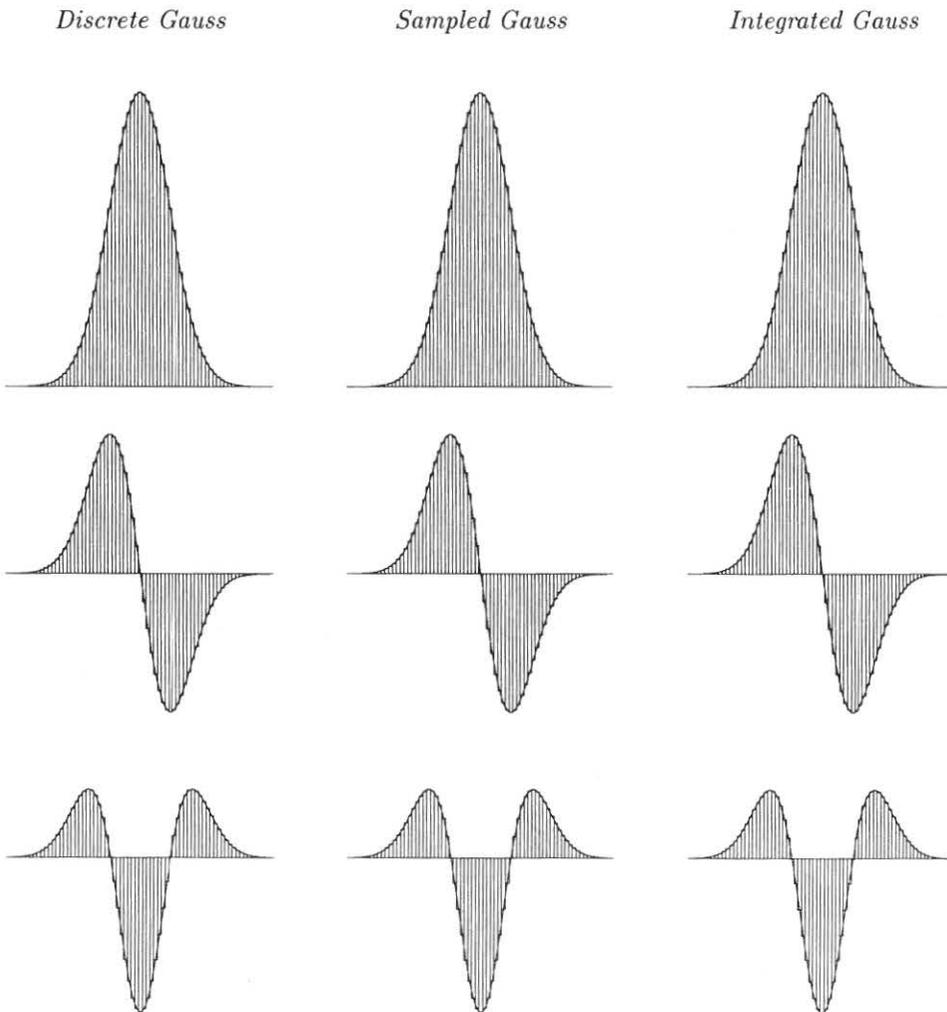


Figure 4.5: $t = 64.0$: Comparisons between the discrete analogue of the Gaussian kernel (left column), the sampled Gaussian kernel (middle column) and the integrated Gaussian kernel (right column). The upper row shows the raw smoothing kernel, the middle row the first order differences/derivatives and the lower row the second order difference/derivatives. The block diagrams indicate the discrete kernels and the smooth curve the continuous Gaussian.

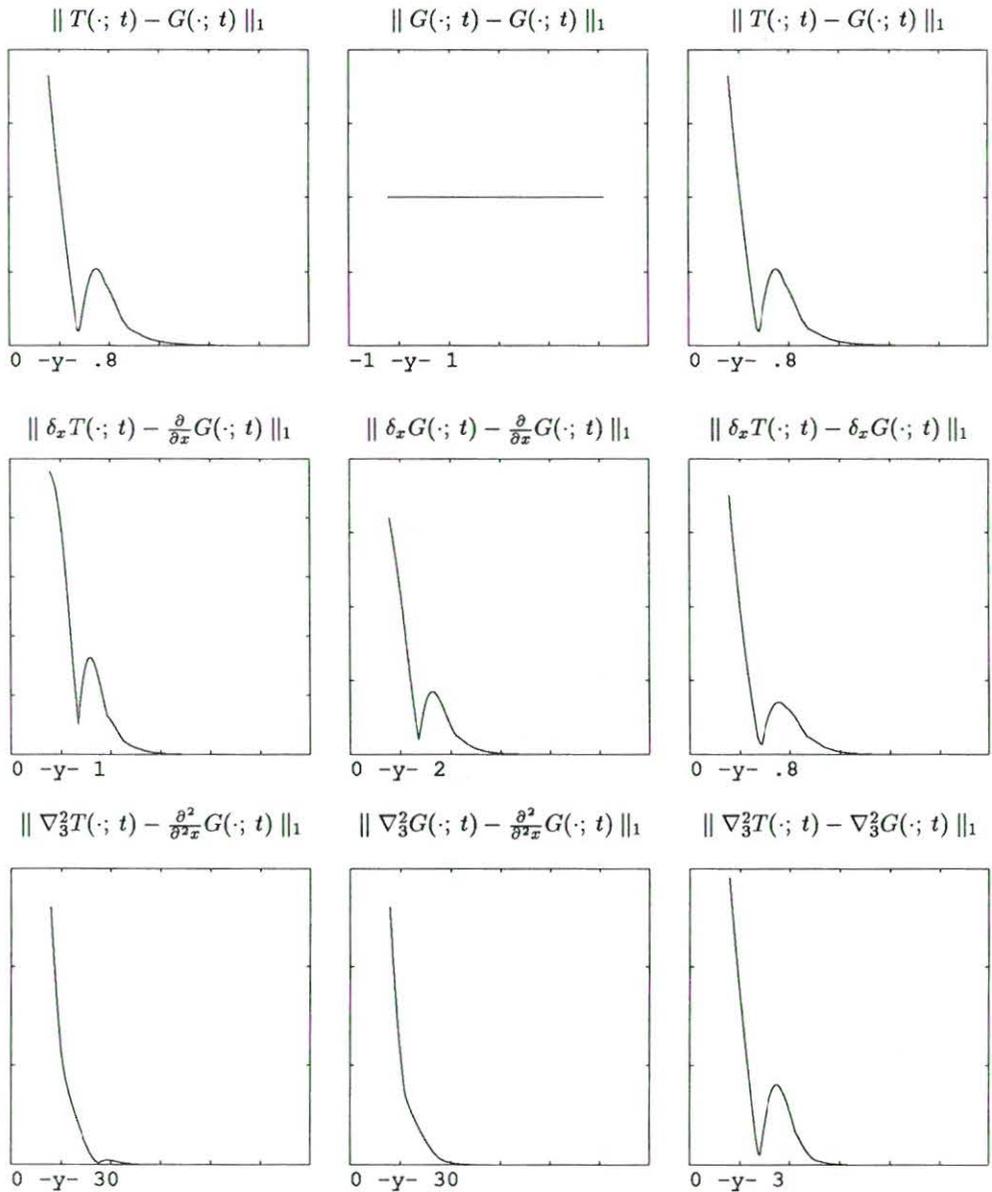


Figure 4.6: l_1 norms of some differences between the sampled Gaussian kernel $G(\cdot; t)$ and the discrete analogue of the Gaussian kernel $T(\cdot; t)$ in the one-dimensional case. The left column shows comparisons between differences of the discrete analogue of the Gaussian kernel and sampled derivatives of the Gaussian kernel, the middle column comparisons between sampled derivatives of the Gaussian kernel and differences of the sampled Gaussian kernel and finally the right column differences of the discrete analogue of the Gaussian kernel compared with differences of the sampled Gaussian kernel. The top row displays the result for zero order differences/derivatives, the middle row for first order differences/derivatives and the bottom row shows the result for second order differences and derivatives. As we see, the magnitude of the error increases with the order of the derivatives/differences. The scaling of the x-axis is logarithmic with range between t equals 0.01 and 10000. The range of the y-axis is shown below each graph.

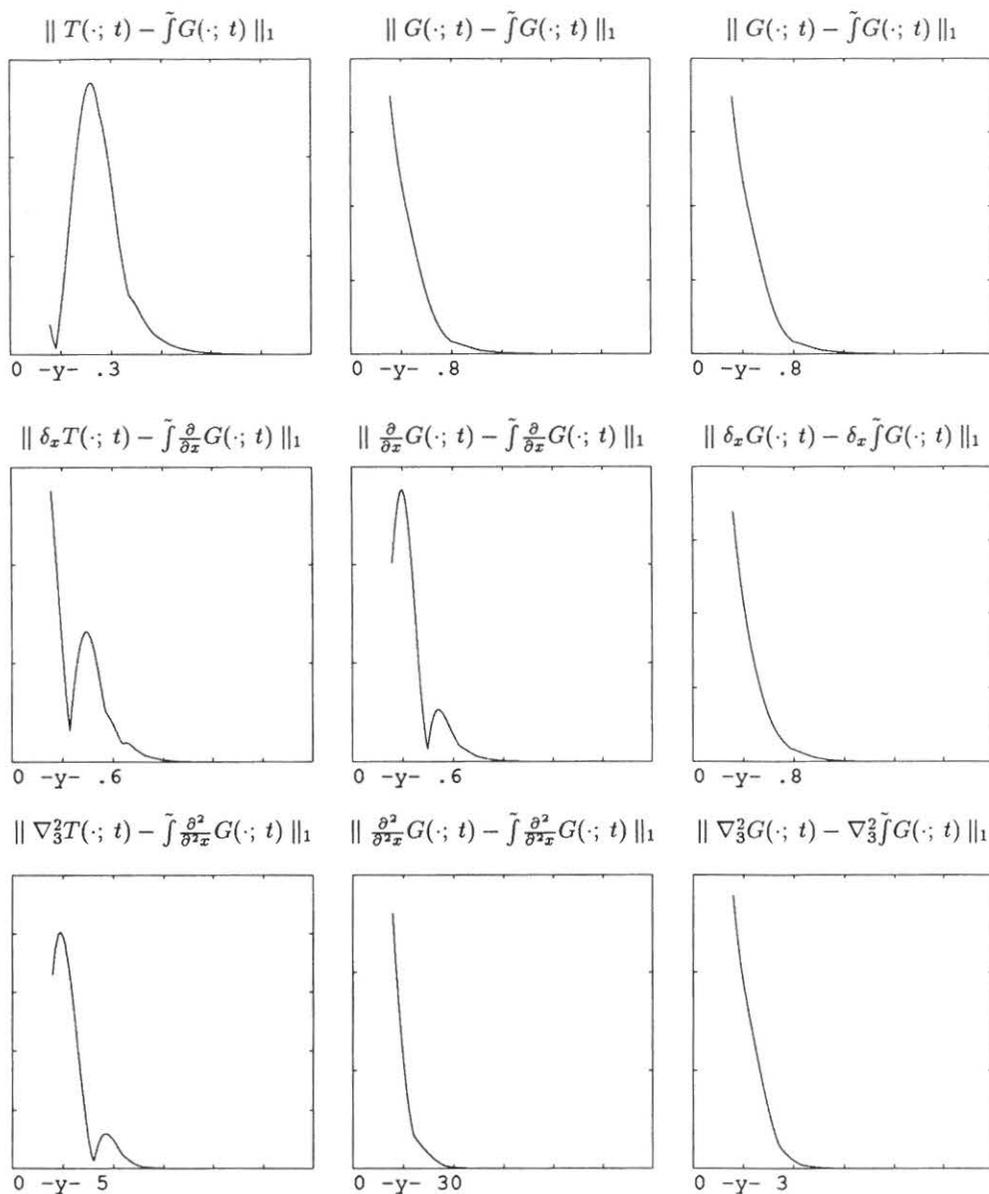


Figure 4.7: l_1 norms of some differences between the integrated Gaussian kernel $\tilde{f}G(\cdot; t)$ and the discrete analogue of the Gaussian kernel $T(\cdot; t)$ or the sampled Gaussian $G(\cdot; t)$ in the one-dimensional case. Here, the modified integration sign \tilde{f} stands for integration over each pixel support region. The left column shows comparisons between differences of the discrete analogue of the Gaussian kernel and integrated derivatives of the Gaussian kernel, the middle column comparisons between sampled derivatives of the Gaussian kernel and integrated derivatives of the sampled Gaussian kernel and finally the right column differences of the sampled Gaussian kernel compared with differences of the integrated Gaussian kernel. The top row displays the result for zero order differences/derivatives, the middle row for first order differences/derivatives and the bottom row shows the result for second order differences and derivatives. As we see, the magnitude of the error increases with the order of the derivatives/differences. The scaling of the x-axis is logarithmic with range between t equals 0.01 and 10000. The range of the y-axis is shown below each graph.

Part II

The Scale-Space Primal Sketch: Theory

Chapter 5

Definition of the Representation

The scale-space theory provides a well-founded framework for dealing with image structures, which naturally occur at different scales. According to this theory one can from a given signal generate a family of derived signals by successively removing features when moving from fine to coarse scale. In contrast to other multi-scale or multi-resolution representations, scale-space is based on a precise mathematical definition of causality, and the behaviour of structure as scale changes can be analytically described. However, the information in the scale-space embedding given by the diffusion equation is only *implicit* in the grey-level values. The smoothed images in the raw scale-space representation contain no *explicit* information about features or the relations between features at different levels of scale.

The goal of the second part of this thesis is to present a theory for constructing such an explicit representation on the basis of formal scale-space theory. This material constitutes the framework for the third part, where we will demonstrate that the suggested representation enables extraction of significant image structure and that it can serve as a guide to other processes in early vision.

We shall treat intensity images, the grey-level landscape, and the objects will therefore be blobs, that is bright regions on dark backgrounds or vice versa. However, the theory applies to any bounded function and is therefore useful in many tasks occurring in computer vision, such as the study of level curves and spatial derivatives in general, depth maps, colour etc, and also histograms and point clustering and grouping in one or several variables.

From experiments one can (visually and subjectively) observe that the main features arising in the scale-space representation seem to be blob-like, i.e., they are smooth regions either brighter or darker than the background. Especially regions which appear to stand out from the surroundings in the original image seem to be further enhanced by the scale-space smoothing. In the suggested scale-space primal sketch we will focus on this aspect of image structure with the purpose of building a formal representation to make such information in scale-space explicit. Therefore, there is a need to formalize what should be meant with a “blob”.

5.1 Grey-Level Blobs

What properties do we require from a blob definition? Intuitively, one would generally like a blob to be a connected region that is either significantly brighter or significantly darker than its neighbourhood. It should have a sufficiently large area and be stable over some sufficiently large interval in scale-space. One would also like a blob to have some kind of

natural significance measure associated with it.

It is clear that a blob should be a region associated with at least one (or possibly more) local extremum point. However, it is essential to define the spatial extent of the blob region around the extremum. Ehrich and Lai [Ehr78] considered this problem. They allowed peaks to extend to valleys, a definition that will give unintuitive results e.g. for small peaks on large slopes. Koenderink and van Doorn [Koe84] briefly touch upon the problem and our definition is related to their argument.

5.1.1 Definition of Grey-Level Blob

The blob definition we base this work on should be evident from Figure 5.1. The basic idea is to let the blob extend “until it would merge with another blob”. To intuitively illustrate this notion, consider a grey-level image at a fixed level of scale, and study the case with bright blobs on a dark background. Imagine the image function as a flooded grey-level landscape. If the water level sinks gradually, peaks will appear. At some instances two different peaks become connected. The corresponding elevation levels or grey-levels are called the *base-levels* of the blobs and are used for delimiting the spatial extent of the blobs. The *support region* of the blob is defined to consist of those points that have a grey-level exceeding the base-level and can be reached from the local maximum point without descending below the base-level of the blob.

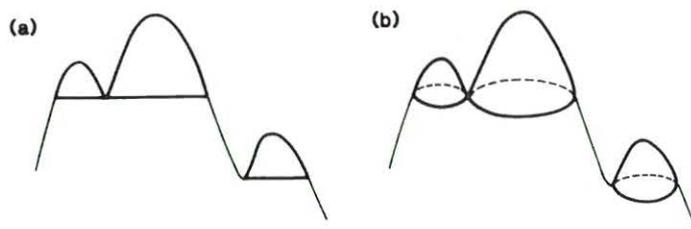


Figure 5.1: Illustration of the grey-level blob definition for (a) a one-dimensional signal and (b) a two-dimensional signal. This figure shows bright blobs on a dark background. In one dimension a bright grey-level blob is given by a pair consisting of a local maximum and a local minimum, in two dimensions generically by a pair consisting of a maximum and a saddle.

Hence, a bright blob will grow and include points having lower grey-levels until it would meet with another blob. As soon as it has got confronted with the other blob the blob region stops growing, not only in the region around the neighbour blob but also in all other directions. In this sense the blob definition can be regarded as rather *conservative*.

From this construction we may also proceed and define the *grey-level blob* as the region delimited by the grey-level surface and the base-level and the *blob contrast* as how deep one has to descend from the maximum point in order to climb another blob. Consequently, a grey-level blob is a 3D object with extent both in space and grey-level, whose size is called *grey-level blob volume* and comprises both the amplitude and the spatial extent of the blob, see Figure 5.2.

5.1.2 Mathematical Definition

To give a precise mathematical definition the concepts introduced above consider again the case with bright blobs on dark background and assume a continuous grey-level signal

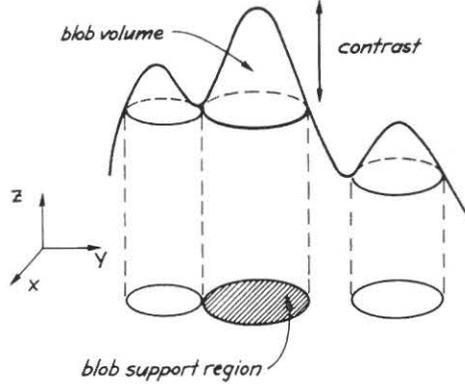


Figure 5.2: Some descriptive quantities of a grey-level blob in the two-dimensional case: volume, area, contrast. Note that the grey-level blob has extent both in space (x, y) and grey-level z .

$f : R^2 \rightarrow R$ at a fixed level of scale. Consider two local maxima, A and B . They are connected by an infinite set of paths, $P_{A,B}$. On each path, $p_{A,B}$, the grey-level function assumes a minimum. To reach another maximum from A , one must at least descend to the grey-level

$$z_{base}(A) = \sup_{B \in M} \sup_{p_{A,B} \in P_{A,B}} \min_{(\xi, \eta) \in p_{A,B}} f(\xi, \eta) \quad (5.1)$$

where M is the set of all local maxima. In the compact case to be considered later, we may replace sup with max and inf with min and write:

$$z_{base}(A) = \max_{B \in M} \max_{p_{A,B} \in P_{A,B}} \min_{(\xi, \eta) \in p_{A,B}} f(\xi, \eta) \quad (5.2)$$

$z_{base}(A)$ constitutes the previously mentioned base-level of the bright blob associated with the local maximum A . The *support region* $D_{support}(A)$ of the blob is the region

$$D_{support}(A) = \{r \in R^2 : \sup_{p_{A,r} \in P_{A,r}} \inf_{(\xi, \eta) \in p_{A,r}} f(\xi, \eta) \geq z_{base}(A)\} \quad (5.3)$$

The difference in grey-level between the extremum point and the base-level gives the *blob contrast*.

$$C_{blob}(A) = f(A) - z_{base}(A) \quad (5.4)$$

Finally the grey-level blob associated with the local maximum A is the set of points

$$G_{blob}(A) = \{(x, y, z) \in R^2 \times R : ((x, y) \in D_{support}(A)) \wedge (z_{base}(A) \leq z \leq f(x, y))\} \quad (5.5)$$

To summarize,

Definition 5.1 (Bright grey-level blob of a continuous signal (2D))

Given a continuous signal $f : R^2 \rightarrow R$ let $A \in R^2$ be a local maximum point, $z_{base}(A)$ its associated base-level as given by (5.1) and $D_{support}(A)$ its associated support region as defined in (5.3). Then the bright grey-level blob associated with A , denoted $G_{blob}(A)$, is the region

$$G_{blob}(A) = \{(x, y, z) \in R^2 \times R : ((x, y) \in D_{support}(A)) \wedge (z_{base}(A) \leq z \leq f(x, y))\} \quad (5.6)$$

It is worth stressing that with this blob concept we regard a grey-level blob as an object with extent both in space and grey-level. The definition is expressed for a two-dimensional continuous function, but applies in any number of dimensions. Similarly, it can be extended to comprise discrete signals by replacement of R^2 with Z^2 and by letting the paths $p_{A,B}$ be given by a suitable connectivity concept, e.g., eight-connectivity for a square grid. For discrete signals it is, however, because of algorithmic reasons more suitable to define the support region of the blob as those pixels that have a grey-level (strictly) *exceeding* the base-level of the blob in order to obtain grey-level blobs that are disjoint objects. In the two-dimensional discrete case we get:

Definition 5.2 (Bright grey-level blob of a discrete signal (2D))

Given a discrete signal $f : Z^2 \rightarrow R$ let A be a local maximum point, $z_{base}(A)$ its associated base-level as given by (5.1) (where the connectivity is defined based on eight-connectivity) and $D_{support}(A)$ its associated support region defined by

$$D_{support}(A) = \{r \in Z^2 : \sup_{p_{A,r} \in P_{A,r}} \inf_{(\xi,\eta) \in p_{A,r}} f(\xi, \eta) > z_{base}(A)\} \quad (5.7)$$

Then the bright grey-level blob associated with A , denoted $G_{blob}(A)$, is the region

$$G_{blob}(A) = \{(x, y, z) \in Z^2 \times R : ((x, y) \in D_{support}(A)) \wedge (z_{base}(A) \leq z \leq f(x, y))\} \quad (5.8)$$

Local minima can be treated analogously and every local minimum point will give rise to dark blob on bright background. Hence, each local extremum point will be associated with a region in two-space and a volume in three-space.

5.1.3 Properties

It can easily be verified that a blob will be connected. Moreover, in one dimension the base level of a bright blob will be attained at a local minimum point, in two dimensions generically at a saddle point, see Figure 5.1. In other words, a grey-level blob of a one-dimensional signal is generically given by a pair consisting of one local maximum and one local minimum. A grey-level blob of a two-dimensional blob is given by a similar pair of a local extremum and a saddle. Consequently, the blobs are directly determined from *geometric properties* of the grey-level landscape, namely the first order singularities of the grey-level function.

These blobs are not purely local features, as are extrema, but *regional*. In fact, this is not only because they are defined as regions. An inherent property of the stated definition is that it leads to a *competition between parts*. The presence of a nearby blob might neutralize it or reduce its size. In other words, things manifest themselves only compared to their background. These aspects reflect important principles of the approach.

We will see later that the blobs are easier to trace across scales than are local extrema. This is because they will be stable over some scale interval, a property considered important also by Bischof and Caelli [Bis88] and Koenderink and van Doorn [Koe84]. In fact, our definition is closely related to their measurement of shape based on relative densities. The grey-level function can be seen as a mass distribution in the plane.

5.1.4 Relations Between Bright and Dark Blobs

Note that this definition leads to *separate* systems for bright blobs on dark background and dark blobs on bright background. This implies that a spatial point may belong both to a

bright blob and a dark blob and that some points will also be left unclassified. Consequently, the given definition will, in contrast to, e.g., the sign of the Laplacian of the Gaussian, only attempt to make a partial (and hopefully safer) classification of the grey-level landscape.

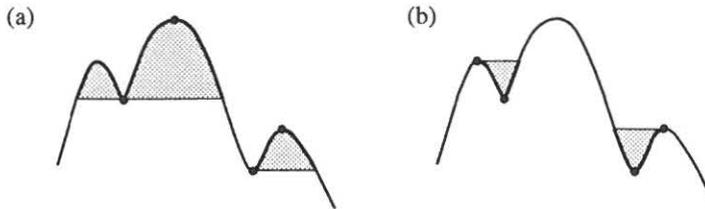


Figure 5.3: In one dimension the bright and dark blobs of a signal will be closely coupled, since an extremum point which delimits the extent of a blob will in general be the seed of a blob of the reverse polarity.

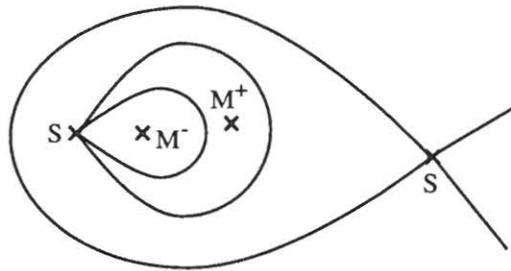


Figure 5.4: Example showing that in two dimensions some points may actually be classified as belonging to both a dark blob and a bright blob. This phenomenon can be prevented from happening if the blob definition is modified such that a blob delimits its own extent in this type of situations. Then it will be guaranteed that no points belong to both dark and bright blobs.

In one dimension the dark and bright blobs of a signal will be strongly related since a minimum point, which delimits the extent of a bright blob, can also be the seed of a dark blob. This gives a natural coupling between blobs of reverse polarity, see Figure 5.3(a). In two dimensions the situation is slightly different, since a saddle point that delimits the extent of a bright blob will in general not delimit the extent of any dark blob, unless the signal is degenerate. Therefore, in two dimensions a point will in general belong to either a dark blob or a bright blob but not both. However, for certain types of situations it may indeed happen that some points are classified as belonging to both a dark blob and a bright blob, see Figure 5.3(b), which shows a dark blob “contained” in a bright blob. If for some reasons this type of phenomenon is not desired then it can be easily prevented from happening if the blob definition is modified slightly so that a blob can be allowed to “delimit its own extent”.

Finally, let us point out that what we have defined here is a *grey-level blob* at one level of scale. When we link grey-level blobs over scales we will obtain scale-space blobs, which will be described after the next section.

5.2 Motivations for a Multi-Scale Hierarchy

The concept of a blob at a single level of scale is not powerful enough for extraction of relevant image structure. It is easy to realize that it leads to strong noise sensitivity, since two closely situated local extrema will neutralize each other, see Figure 5.5. This means that a large peak distorted by a few local extrema with low amplitude will not be detected as one unit, only the fine scale blobs will be found. This kind of problem has been considered also by Ehrich and Lai [Ehr78]. They suggest the use of a so-called relational-tree, in order to obtain the spatial relations between superimposed blobs, without developing that concept further. However, their relational-tree will still be noise sensitive, since the hierarchical relations between blobs are determined directly by the grey-levels in the valleys of the original signal.



Figure 5.5: A high-contrast large scale peak with two superimposed low-contrast fine-scale peaks will not be detected if the signal is considered at one scale only.

5.2.1 Blob Detection and Scale-Space Smoothing

To some extent the noise sensitivity in such a situation can be reduced with a suitable amount of smoothing. However, at a fine level of scale (without smoothing) it is difficult to detect that the configuration consists of two large peaks with small superimposed low-amplitude peaks. A naive observer might say that the situation can be resolved easily with thresholding, but how does one select a proper threshold automatically?

One possible way of designing a blob detector that could “handle” such a configuration and detect the underlying large peaks could be “by making the blob detector more intelligent”. Such an approach would still face a difficult and undefined question: How deep may the valley between the superimposed blobs be before they are regarded as belonging to different blobs? To avoid such dilemmas we will in this work instead take a contrary approach.

The idea with our method is to use a *simple blob definition based on distinct geometrical properties of the signal*. Then, we use *the scale-space embedding to integrate local properties into regional descriptors*, and to make the hierarchical relations between features at different levels of scale explicit. Applied to the previous example it means that the fine-scale peaks will disappear after some degree of scale-space smoothing and the underlying coarse scale peak will appear as one unit. Hence, in a multi-scale representation of the signal it will be made explicit that the configuration in Figure 5.5 consists of a coarse scale blob with two superimposed fine scale blobs.

The method to achieve this is by linking grey-level blobs over several levels of scale into higher-order objects, called scale-space blobs, which in addition to extending in space and grey-level have extent also in scale. How those objects are constructed will be described in the next section.

5.3 Scale-Space Blobs—Linking Grey-Level Blobs across Scales

In general, a grey-level blob existing at one level of scale in scale-space will correspond to a similar blob both at a finer level of scale and at a coarser level of scale. By linking together such grey-level blobs across scales we obtain four-dimensional objects, which we call *scale-space blobs*¹.

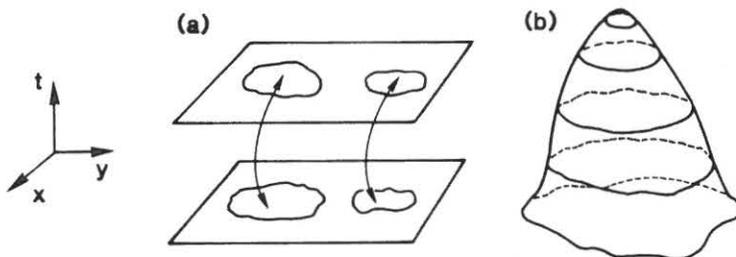


Figure 5.6: (a) By linking together similar grey-level blobs at adjacent levels of scale we obtain (b) scale-space blobs, which are objects having extent both in space, scale and grey-level. (In this figure we have omitted the grey-level coordinate. The slices illustrate the blob support regions of the grey-level blobs.)

At some levels of scale in scale-space it might be impossible to accomplish a plain link between a grey-level blob at the current level of scale to a similar grey-level blob at a coarser or finer scale — a catastrophe affecting the connectivity of the blobs has occurred. The generic situations telling how blobs may behave with scale will be classified in Chapter 6. According to that treatment four possible types of blob events may occur when the scale parameter *increases*:

- *annihilation* — a blob disappears
- *merge* — two blobs merge into one
- *split* — one blob splits into two
- *creation* — a new blob appears

The scale levels where these singularities take place are used for delimiting the extent in the *t*-direction of the scale-space blobs. Consequently, every scale-space blob will be associated with a minimum scale and a maximum scale, denoted the *appearance scale* and the *disappearance scale* respectively, see Figure 5.6. The difference² between the disappearance scale and the appearance scale yields the *scale-space lifetime* of the blob. Precise mathematical definitions of these concepts are given in Section 6.1.3.

In merge situations and split situations we regard the involved grey-level blobs existing before the bifurcation event as belonging to different scale-space blobs than the grey-level blobs existing after the bifurcation. In special configurations it may happen that a blob without a hole forms a torus, or that a torus fills in its hole. These events are also stable

¹A formal definition of this concept is given in Section 6.1.3

²It turns out that some transformation of the scale parameter is necessary in order for the difference between scale values to capture the concept of scale-space lifetime “properly”. This issue is considered in Section 5.5.

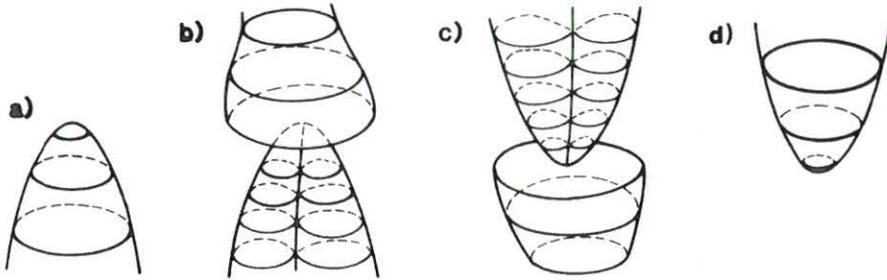


Figure 5.7: Common blob events in scale-space: (a) annihilation (b) merge (c) split (d) creation

in the sense of singularity theory, but we will not let them affect the scale-space blobs. The grey-level blobs involved in such processes will be treated as belonging to the same scale-space blob.

The scale values where these blob events occur define the previously mentioned appearance scales and disappearance scales for the involved scale-space blobs. This means that the scale-space lifetime of a scale-space blob is directly determined by the singularities in scale-space.

These objects will constitute the fundamental primitives in our proposed scale-space primal sketch. The idea with this representation is to detect the scale-space blobs in scale-space and to build a data structure that makes them as well as their relations between scales *explicit*. This implies that grey-level blobs must be detected at all levels of scale, the actual linking of grey-level blobs between scales into scale-space blobs must be performed and that the bifurcations taking place in scale-space must be registered. The computational aspects of these tasks will be briefly described in Section 7.1 and Section 7.2. However, in order to get acquainted with the blob concepts just defined, we will first present some experimental results illustrating the effects of detecting grey-level blobs at various levels of scale in scale-space.

5.4 Grey-Level Blob Extraction: Experimental Results

In Figure 5.8 we give an example with a realistic toy block image showing how (the support regions of) the extracted grey-level blobs³ behave with increasing scale together with the raw grey-level images in the scale-space representation. We see that at fine levels of scale mainly small blobs due to noise and surface texture are detected. When the scale increases the noise blobs disappear gradually, although much faster in regions near steep gradients. Notable in this context is that blobs due to noise can survive for a long time in scale-space if they are located in regions with slowly varying grey-level intensity. This observation shows that *scale-space lifetime alone cannot be used as the basis for a significance measure*, since then the significance of such blobs due to noise would be substantially overestimated⁴. At coarse levels of scale, the toy blocks appear at single blob objects. Finally, at very coarse

³Each one of these blob images has been obtained directly from the scale-space representation at a *single* level of scale. In other words, no scale linking was performed in the generation of these images. An algorithm for extracting grey-level blobs from a discrete image will be described in Section 7.1.

⁴Of course, the contrast of such noise blobs decreases, but it is far from clear that it is possible to set a threshold on objective grounds.

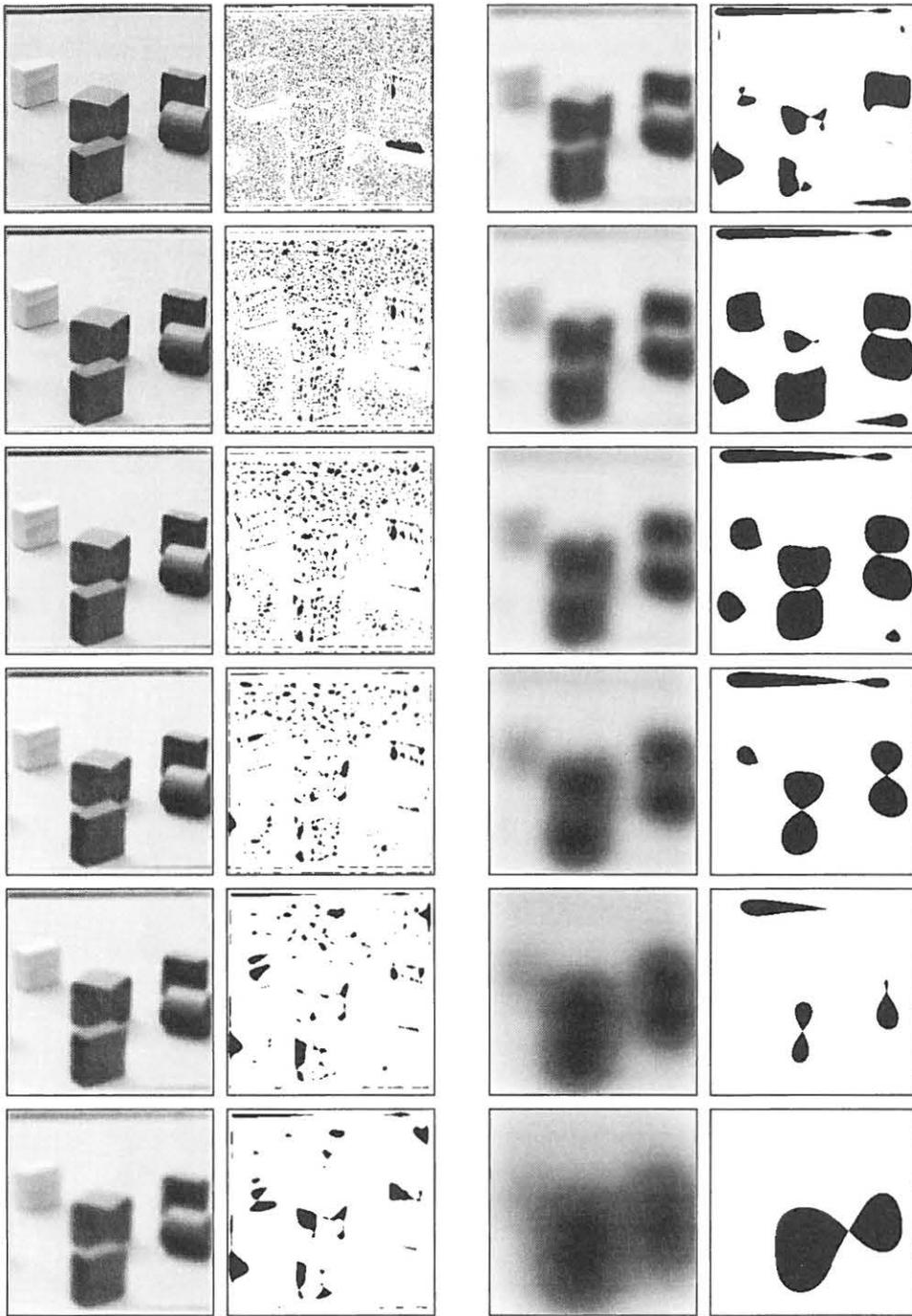


Figure 5.8: Grey-level and (dark) grey-level blob images of a toy block image at scale levels $t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 (from top left to bottom right).

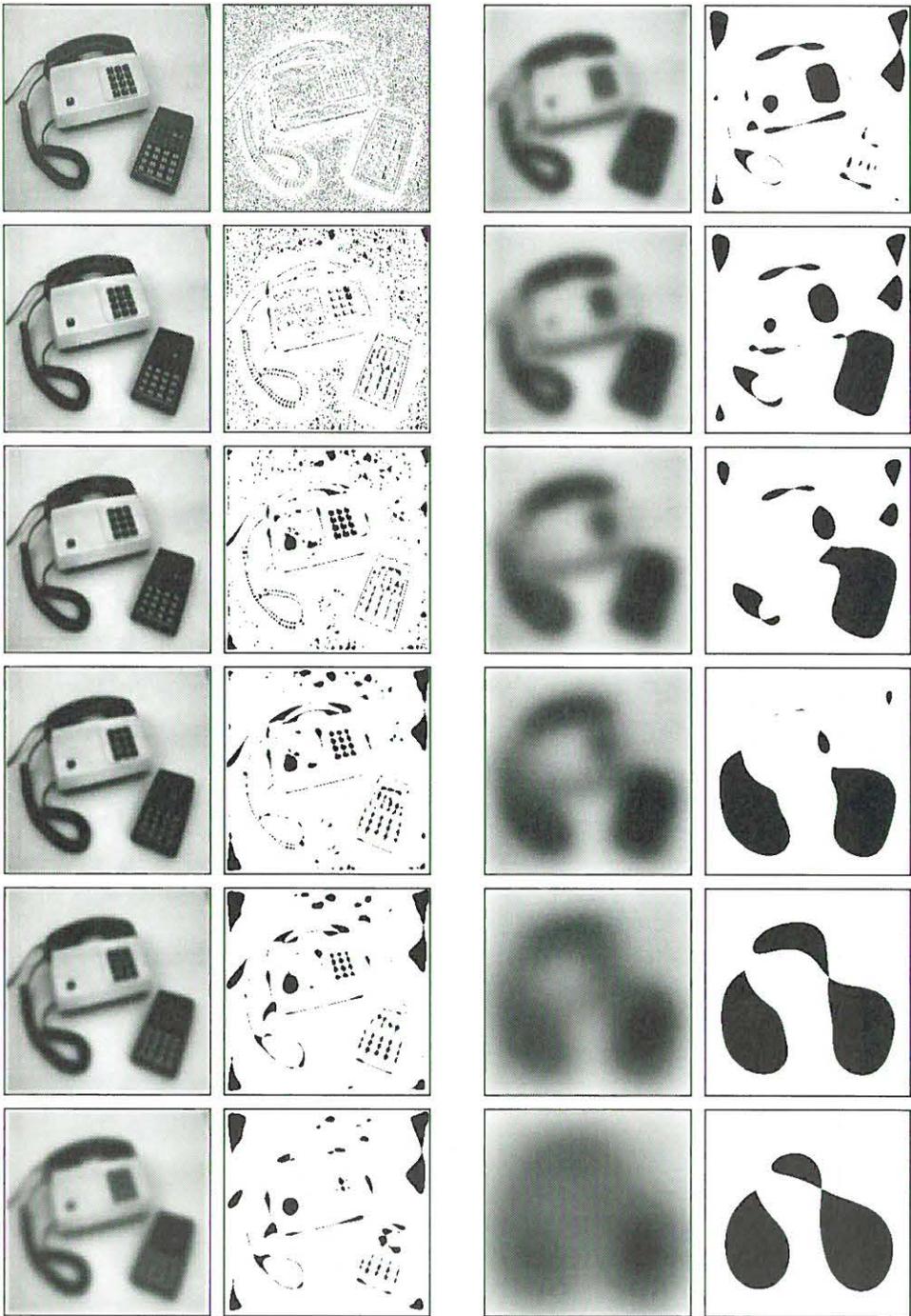


Figure 5.9: Grey-level and (dark) grey-level blob images of a telephone and calculator image at scale levels $t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 (from top left to bottom right).

levels of scale, adjacent blocks become grouped into larger entities.

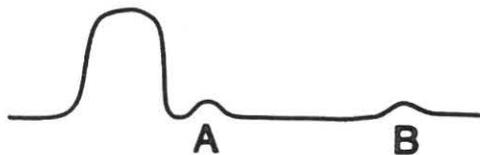


Figure 5.10: *Low-contrast blobs in regions with slowly varying intensity may have long lifetime in scale-space. However, low-contrast blobs located near steep gradients will have short lifetime. This means that blob A will disappear before blob B in this example.*

Figure 5.9 shows a similar scale-space sequence for a telephone and calculator image. In this case as well, one can notice that at the finest levels of scale only blobs due to noise are detected, and that some blobs in the background actually survive over a relatively large range of scale. Moreover a hierarchical behaviour between grey-level blobs at different scales appears again. The buttons on the telephone keyboard manifest themselves as blobs after a small amount of smoothing. At coarser levels of scale they merge into one unit (the keyboard). One can also observe that some other dark details in the image, the calculator, the cord and the receiver, appear as single blobs at coarser level of scale.

One could say that the grey-level blob concept shows an extreme degree of noise sensitivity, which can be circumvented by the scale-space smoothing. But it is certainly a far from trivial problem to determine a proper amount of smoothing *automatically*, based on existing conventional methods.

The aim with the suggested blob linking between scales is to determine which blobs in the scale-space representation can be regarded as significant, *without any prior information* about neither scale, spatial location nor the shape of the primitives⁵. As we will see later, the output from the linking procedure also enables determination of a relevant scale for each blob, i.e., a suitable amount of blurring for treating that *individual*⁶ blob.

5.4.1 Remarks

As stated earlier the idea behind this combination of grey-level blobs and scale-space smoothing is that instead of trying to design “an intelligent blob detector” to handle difficult situations as the case above with superimposed local extrema, we establish a *simple blob*

⁵Except for the previously mentioned fact that the scale-space blurring favours blob-like or Gaussian-shaped objects

⁶We emphasize the word *individual* since we believe that *stable scales (if they exist) are in general associated with objects — not with entire images*. For the toy block image one could possibly say that the scales $t = 128$ and $t = 1024$ are stable in the sense that the grey-level blob detection algorithm finds blobs all with meaningful interpretation in the smoothed grey-level images at these levels of scale. However, for more general images of moderate complexity such “stable scales” will not exist. From experiments one quickly learns that a scale level well fit for one part of an image will in general not be useful for treating other parts of the image (compare with the blob behaviour at fine scale levels in the toy block image and the overall appearance in the telephone and calculator image).

However, the assumption about a globally stable scale for an image is sometimes used implicitly in many computer vision algorithms — for instance when edge detection is performed using the same amount of smoothing all over an image. Instead we believe that better performance can be obtained if the scale levels are adapted to the local image structure, compare with the integration of the output from the scale-space primal sketch with the edge focusing method, developed in Section 9.1.

definition based on distinct topological properties of the signal. Then, we use *the scale-space embedding to integrate local properties into regional descriptors*, and to make the hierarchical relations between features at different levels of scale explicit.

Another aspect of this definition of a scale-space blob is that we treat blobs not just as entities within an image at a given level of scale, but as objects in the four-dimensional scale-space parameterized by the two spatial coordinates, the grey-level coordinate and the scale parameter. In other words, we treat *the scale parameter as equally important as the spatial and grey-level coordinates*, and *the primitives of our representation are objects having extent not only in grey-level and space, but also in scale.*

It will be demonstrated later that this is a powerful approach, particularly since our notion of scale-space blobs gives us a natural geometric measure of significance, namely the volume of the scale-space blob in the four-dimensional scale-space⁷. Before going into the experimental results, which will be presented in Section 8, we will however first describe some normalization aspects that are necessary when computing the representation.

5.5 Measuring Significant Image Structure

Since our ultimate goal of the analysis is to extract important regions in the image based on the significance of scale-space blobs in the scale-space representation, there is an absolute need for some methodology for comparing blob significance *between* different levels of scale. In other words, what we actually desire is a mechanism to judge if a blob, existing only at coarse levels of scale, can be regarded as more significant or less significant than a blob, with extent primarily at fine levels of scales.

The approach we propose is to use the volumes of the scale-space blobs defined in Section 5.2. We suggest that it is a useful quantity for such a significance measure, since it comprises both the grey-level blob volume, which is a combination of the contrast, spatial extent and lifetime of the blob in scale-space, see also Section 8.2. However, if one is to base a significance quantity on this quantity, it is of crucial importance that the scale parameter and grey-level blob volume are measured in proper units, since in principle the x , y , z and t axes could be transformed by arbitrary monotone functions.

5.5.1 Measuring Scale-Space Lifetime

Consider for instance the measurement of scale-space lifetime. According to a wide-spread paradigm the scale-space should be sampled logarithmically in scale, i.e., the ratio between successive scale values should not vary with scale. Based on this idea one could be inspired to define the scale-space lifetime as $\log t_D - \log t_A$, where t_D and t_A are disappearance and appearance scales of the scale-space blob respectively. It seems reasonable that this would give a good description at coarse levels of scale, since it is well-known that “things happen approximately logarithmically with scale”. However, such an approach would certainly lead to unreasonable results for discrete signals at fine levels of scale, since then a blob existing at $t = 0$ would be given an infinite lifetime. Similarly, one can observe that $t_D - t_A$ will not work either, since then the lifetime of blobs existing at coarse levels of scale would be substantially overestimated. Consequently, there is a need to introduce a transformed more

⁷Note that in this context we regard the scale-space not merely as a plain set of gradually smoothed (but anyway relatively disparate or just loosely connected) grey-level images, but as an *intimately* connected entity.

realistic scale parameter τ , which will be called *effective scale*, such that scale-space lifetime measured by $\tau_D - \tau_A$ gives a proper description of the behaviour in scale-space. This new scale parameter should neither favour coarse scales to fine scales nor the opposite.

In this section we will first give a formal treatment showing how the notion of “effective scale” can be defined in a precise way. We will also give experimental results showing how the major blob descriptors (volume, area, contrast) are expected to behave with scale, and explain how these results can be used to rescale the descriptors in question. Some other facts that will be illuminated are that the inner scale and the outer scale of an image really must be taken into account in an actual implementation.

5.5.2 Transformation of the Scale Parameter: Effective Scale

At first glance the problem of transforming the scale parameter might seem rather ad hoc. What properties do we want from an “effective scale parameter? Assuming that we have a measure for the amount of structure in an image then a natural requirement would be that the amount of structure, which is destroyed if the scale parameter increases with one unit, should not depend on the current scale. In other words, if we plot the amount of structure as a function of the effective scale parameter, we should expect the curve to be a straight line. However, what does one mean with the amount of structure in an image? Moreover, even if we had a definition of the measure of structure it would be possible to transform it and then the effective scale could also be transformed in a similar way, while the graph would remain a straight line. Hence, if one is to define a measure of structure one still has a transformation function to determine.

Another natural requirement is that the expected lifetime of a scale-space blob in scale-space should not vary with scale. Assuming that we know how the number of blobs depends on the scale, this condition will actually determine the transformation of the scale parameter, except for an arbitrary but unimportant affine transformation.

5.5.2.1 Definition and Derivation of Effective Scale and Effective Structure

Assume that we know how the expected number of extremum points per unit length, i.e., the density of grey-level blobs, behaves over scale. In other words, assume that we know how

$$p(t) = \{\text{the expected density of extremum points at scale } t\} \quad (5.9)$$

varies with t . Assume that the amount of structure in an image can be measured with the expected number of local extrema⁸ per unit length. What we want to define is a transformed scale τ and a transformed measure of structure m , such that the new coordinate system becomes “natural”, i.e., we want to define transformation functions h and g such that the new coordinates $\tau = h(t)$ and $m = g(p)$ capture the concepts of structure and scale in a “natural sense”. From this discussion the following requirements seem reasonable to pose on the new coordinate system:

Requirement 5.1 (Uniform decrease in the amount of structure)

The expected amount of effective structure dm , which is destroyed if the effective scale is

⁸In one dimension the number of local extrema seems to be a reasonable measure of the amount of structure, since a whole scale-space theory can be founded on this basis, see Chapter 2. In two dimensions the situation is more elaborate, since in this case the number of local extrema in an image may actually increase with scale due to blob splits. However, the expected number of local extrema, as an average over many images, will always be expected to decrease.

increased with $d\tau$, should be independent of both the current effective scale and the current amount of effective structure. In other words,

$$\frac{dm}{d\tau} = \frac{dg}{dh} = A = \text{constant} \quad (5.10)$$

Requirement 5.2 (Uniform decay intensity for local extrema)

The probability that an extremum point (or a blob) disappears after a small increment $d\tau$ in effective scale should be independent of both the effective scale τ and the current amount of structure m in the image. That is

$$\frac{\frac{dp}{d\tau}}{p} = \frac{\frac{dp}{dh}}{p} = \frac{d(\log p)}{dh} = B = \text{constant} \quad (5.11)$$

Integration of (5.10) and (5.11) gives:

$$g(h) = Ah + C \quad (5.12)$$

$$\log p = Bh + D \quad (5.13)$$

where C and D are integration constants. Reasonable boundary conditions state that if the scale t is zero then also the effective scale h should also be zero. Let p_0 denote⁹ $p(0)$ and g_0 the amount of structure at scale $t = 0$. After a few calculations it then follows that

$$g(p) = g_0 + \frac{A}{B} \log \frac{p}{p_0} \quad (5.14)$$

$$h(t) = \frac{1}{B} \log \frac{p}{p_0} \quad (5.15)$$

Since A , B and g_0 are arbitrary constants we set $A = -1$, $B = -1$ and $g_0 = 0$. Then

$$g(p) = \log \frac{p}{p_0} \quad (5.16)$$

$$h(t) = \log \frac{p_0}{p(t)} \quad (5.17)$$

Equation (5.16) describes how the measure of effective structure should be computed from the measured density of local extrema, while (5.17) gives the relation between the effective scale parameter τ and the ordinary scale parameter t . To summarize,

Theorem 5.1 (Effective scale)

Assume that we know how the expected density of local extrema p behaves as a function of scale t , let τ and m be the effective scale and the effective measure of structure given by Requirement 5.1 and Requirement 5.2. Then the transformation function h from t to τ and the transformation function g from \tilde{p} to \tilde{m} are, except for an arbitrary affine transformation, given by

$$\tilde{m} = g(\tilde{p}) = \log \frac{\tilde{p}}{\tilde{p}_0} \quad (5.18)$$

$$\tau = h(t) = \log \frac{p_0}{p(t)} \quad (5.19)$$

where $p(t)$ is the expected density of local extrema at scale t and $\tilde{p}(t)$ is the measured density of local extrema at scale t .

⁹For continuous signals it might be more convenient to set $p_0 = 1$, since $p(0)$ may be infinite and/or difficult to estimate.

This concept of effective scale, which is the natural unit for measurements of scale, will be of crucial importance in the extraction of significant blobs. What is now left to determine is how the density of extrema can be expected to behave with scale. Experimental results will be given the next section. But we will first illustrate some consequences of the stated definition.

5.5.2.2 Examples and Experimental Results

For continuous signals it is known that the number of local extrema in a signal decreases approximately as t^α with scale, where α is approximately $-\frac{1}{2}$. This relation has been discussed by other authors, see e.g. Müssigmann [Müs89], and can be motivated theoretically, at least for one-dimensional signals generated by white noise or fractal noise normal processes, see Section 6.5. Hence, we have $p(t) = \text{constant}/t^\alpha$ which means that

$$m(t) = \log p(t) = \log \text{constant} - \alpha \log t \quad (5.20)$$

and the curve giving the number of local extrema as a function of scale will be a straight line in a log-log-diagram.

For discrete signals the number of extrema will also show the same qualitative behaviour at coarse levels of scale, when the grid effects are negligible. However, at fine levels of scale the $t^{-\alpha}$ -behaviour cannot hold, since it is based on the assumption that the original signal contains equal amount of structure over all levels of scale. The discrete signal is limited by its inner scale given by the sampling density. These ideas are illustrated in Figure 5.11,

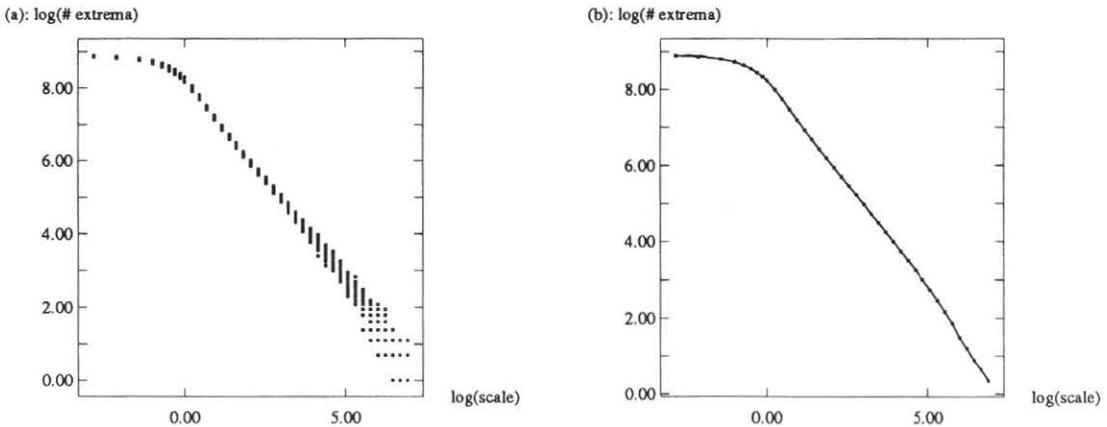


Figure 5.11: Experimental results showing the number of local extrema as function of the scale parameter t in log-log scale (a) measured values (b) accumulated mean values. The dashed line indicates the value at $t = 0$. Note that a straight-line approximation is valid only in the interior part of the scale interval. At the lower end point of the interval we have interference with the inner scale, given by the sampling density of the image, and the higher end point there is interference with the outer scale, given by the size of the image.

where we show the logarithm of the number of extrema as a function of the logarithm of the scale parameter. The left diagram shows simulated results for a large number of point noise images (see below). The right curve shows the average of these results. Note that

the straight line approximation is valid only in an interior scale interval. At fine scales we have interference with the inner scale, given by the sampling density of the image, and at coarse scales there is interference with the outer scale, given by the size of the image. A theoretical analysis for one-dimensional signals generated by white noise and fractal noise normal processes will be carried out in Section 6.5.

The notion of effective scale takes the inner scale into account and guarantees that we have a precise definition of scale-space lifetime at fine levels of scale. Combined with the notion of a scale-space for discrete signals, which takes the discrete nature of implementation into account, it gives us the necessary tool to investigate the fine scale structures.

In this presentation we have chosen not to treat the behaviour at very coarse levels of scale, since there the treatment of the image boundaries will affect the scale-space behaviour substantially. Also, if one wants to study objects at such a coarse scale that the boundary effects become important, then the problem is undefined, and one should instead try to acquire additional image data in a region around the current image, such that the scale-space smoothing becomes well-defined.

5.5.3 Transformation of the Blob Volumes

Similarly to the scale parameter, the grey-level blob volumes need to be rescaled, since the size of the grey-level blob volumes will vary substantially with scale. When the scale parameter increases in scale-space the peaks in the grey-level landscape will erode and the fine scale details will be successively removed. This means that we can expect the mean value of the grey-level blob contrasts to decrease and the mean value of the grey-level blob area to decrease, when the scale parameter increases. But, what about the mean value of the grey-level blob volumes, will it increase or decrease? Experimental results, which will be given later, show that the mean value of the blob volumes actually decreases with scale at fine scales and increases with scale at coarser scale.

Hence, if these effects are not taken into account then the significance of the coarse-scale blobs will be substantially over-estimated compared to the significance of fine-scale blob. In other words if no compensation is performed non-significant structures at coarse levels of scale may be ranked as more important than important structures at fine levels of scale.

It is clear that the blob behaviour depends very much on the image (since we actually want to use it for segmentation). Is it then possible to talk about some kind of average behaviour. It might happen that the blob behaviour varies substantially from one image to another. How should one then be able to talk about expected behaviour?

5.5.3.1 Simulation Results

A conservative approach to the problem is to study point noise images, that is images with no spatial structure, i.e., images with no simple relations between the grey-levels of different pixels. If we accumulate statistics about how blobs in such images are expected to behave with scale, we will get an estimate of how much structure the multi-scale blob detection algorithm will find in images without spatial structure. In this way we get an estimate of the extent of accidental groupings in scale-space.

We have made experiments on several point-noise images with normal distribution, rectangle distribution and exponential distribution. The results are shown in Figure 5.11 and Figure 5.12. As we see, the qualitative behaviour is not very much affected by the image

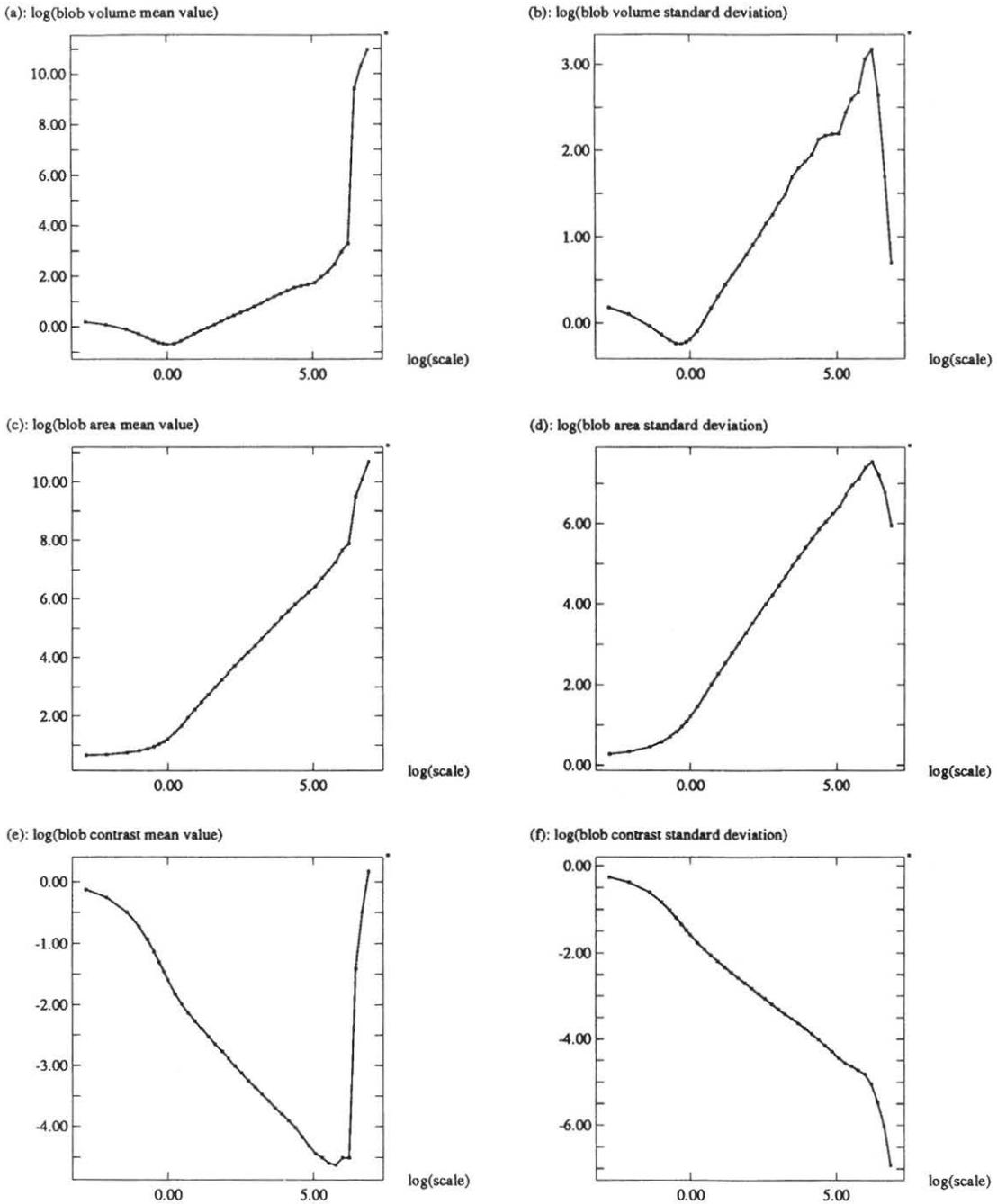


Figure 5.12: Experimental results for point noise images showing how the grey-level blob descriptors vary with scale (in log-log scale). (a) the mean value of the blob volumes, $V_m(t)$ (b) the standard deviation of the blob volumes, $V_\sigma(t)$ (c) the mean value of the blob areas, $A_m(t)$ (d) the standard deviation of the blob areas, $A_\sigma(t)$ (e) the mean value of the blob contrasts, $C_m(t)$ (f) the standard deviation of the blob contrasts, $C_\sigma(t)$. The outliers at the very coarse scales are due to interference with the outer scale of the image.

synthesis process. The number of extrema and the expected blob area turn out to be quantities very insensitive to the image synthesis method. The blob contrast and the blob volume must be rescaled in order for the curves to fit, since these latter quantities are proportional to the amplitude of the signal.

Note that in the intermediate scale interval the curves can be approximated with straight lines. This means that the blob descriptors vary with scale approximately as powers of t^α . Fits to straight lines in the interval $t \in [4, 64]$ for $256 * 256$ -images give that the exponents are approximately 0.5, 1.0 and -0.5 . Hence the blob descriptors behave approximately in the following way for large values of t .

$$V_m(t) \sim \sqrt{t} \quad V_\sigma(t) \sim \sqrt{t} \quad (5.21)$$

$$A_m(t) \sim t \quad A_\sigma(t) \sim t \quad (5.22)$$

$$C_m(t) \sim \frac{1}{\sqrt{t}} \quad C_\sigma(t) \sim \frac{1}{\sqrt{t}} \quad (5.23)$$

5.5.3.2 Effective Grey-Level Blob Volume

Based on these results we will have a basic tool to differentiate between significant and non-significant structure *across scales*. If a grey-level blob has a blob volume smaller than the expected blob volume for point noise images it can hardly be regarded as significant. On the other hand, if at some level of scale the blob volume is much larger than the expected blob volume, and if the difference in blob volume is much larger than the expected variation around the expected standard deviation then blob may be regarded as significant. A natural normalization to perform is to subtract by the mean value and divide by the standard deviation. Hence, an effective blob volume at scale t could be defined as

$$V_{eff,prel}(t) = \frac{V(t) - V_m(t)}{V_\sigma(t)} \quad (5.24)$$

where $V(t)$ is the measured grey-level blob volume at scale t , $V_m(t)$ the mean value of the grey-level blob volumes at scale t for point noise images and $V_\sigma(t)$ the standard deviation of the grey-level blob volumes at scale t for point noise images. However, note that this definition implies a few problems. Since the blob volumes depend strongly on the amplitude of the signal this quantity may be sensitive to the scaling of $V_m(t)$. Another negative aspect is that, since this quantity may assume negative values it is not suited for integration. In the current implementation we have chosen to define the *effective grey-level blob volume* in the following way, which empirically turns out to give reasonable results.

$$V_{eff}(t) = \begin{cases} 1 + V_{eff,prel} & \text{if } V_{eff,prel} \geq 0 \\ e^{V_{eff,prel}} & \text{otherwise} \end{cases} \quad (5.25)$$

Hence, the effective volume of the mean value will be one. For larger volumes it will be grow affinely with $V_{eff,prel}$. Thus, V_{eff} and $V_{eff,prel}$ will show the same qualitative behaviour for significant grey-level blobs. For smaller volumes it will decay to zero, and the qualitative difference will increase gradually as the significance decreases. However, note that we get a correct behaviour in the important situations, namely for the significant blobs. Therefore, we may expect that this solution should not affect the result too seriously. It should also be mentioned, that in order to adapt to the current amplitude of the signal, V_m and V_σ are rescaled linearly from a least-squares fit between the actual behaviour and expected behaviour of these quantities.

5.5.3.3 Other Possibilities

If the mean value and the standard deviation would show the same qualitative behaviour over all levels of scale it would suffice to divide the current grey-level blob volume with the expected value of grey-level blob volume. Then the variation around this rescaled quantity would show the same behaviour at all levels of scale, and the information about standard deviation would not be necessary. Moreover, the rescaling to the current amplitude of the signal could be ignored, since it would affect all scale levels similarly. However, from Figure 5.13, where we have plotted the standard deviation of the blob volume as “function” of the mean value of the blob volumes, we see that the situation is not that simple. In

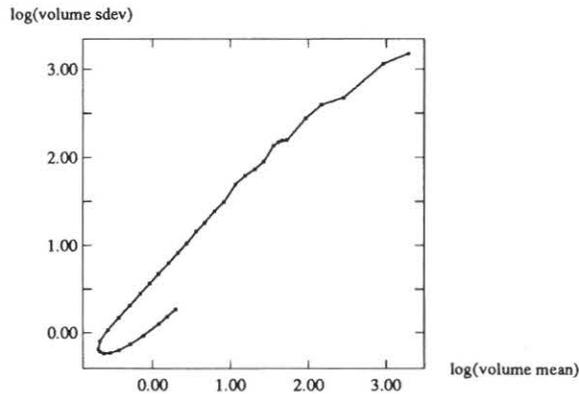


Figure 5.13: Standard deviation V_σ of grey-level blob volume as “function” of the mean value V_m of grey-level blob volumes in log-log scale.

later work we will instead investigate if it is possible to normalize the grey-level volumes by division with the V_m only, and then define a significance measure as in (5.24) based on the scale-space volumes instead. Since these volumes need not be accumulated, negative values will not be any problem. However, then also the underlying statistics must be based on scale-space volumes instead of grey-level volumes. As scale value for normalization we may choose the representative scale value defined in Chapter 8.

Another approach to determine a significance level for the grey-level blob contrast and the grey-level blob volume could be by estimation of characteristic variation amplitude in the image, similar to the method used by Voorhees and Poggio [Vor87]. The basic idea is to accumulate a histogram over the grey-level differences in the image over some characteristic length corresponding to the current level of scale, and extract the peak(s) from the histogram. If the contrast of a grey-level blob is lower than the estimated variation level then the blob can be regarded as non-significant. Similarly, if its contrast is much larger than the variation level in the image it may be regarded as significant. However, this approach assumes that there is a global grey-level variation level valid for the entire image, an assumption that is often violated in realistic imagery.

5.5.4 Resulting Representation — The Scale-Space Primal Sketch

To summarize, the data structure we propose is a tree-like multi-scale representation of blobs at all levels of scale in scale-space *including* the relations between blobs at different

levels of scales. Grey-level blobs should be extracted at all scales, the bifurcations occurring in scale-space be explicitly registered and grey-level blobs stable over scales be linked across scales into the higher-order objects called scale-space blobs.

Since the representation tries to capture the significant features and events occurring in scale-space with a small set of primitives we call it a *scale-space primal sketch*. In the resulting data structure constructed according to this description, every scale-space blob contains explicit information about which grey-level blobs it consists of. The grey-level blobs are given at (sampled) scale levels obtained from an adaptive scale linking and refinement procedure to be outlined in Chapter 7. Further, the (normalized) scale-space blob volume, the appearance scale, the disappearance scale and the scale-space lifetime have been computed (using straightforward numerical techniques). The scale-space blobs “know” about the type of bifurcations (annihilation, split, merge, creation) that have taken place at the appearance and disappearance scales. They also have links to the other scale-space blobs involved in the bifurcation processes. Hence, the representation¹⁰ we have computed explicitly describes the hierarchical relations between blobs at different levels of scale.

The intention with this representation is to capture *inherent geometric* properties of the underlying grey-level image and we suggest that the representation as such is useful in itself. Worth emphasizing is that the involved quantities (grey-level blobs and scale-space blobs) are defined solely in terms of *singularities*, namely local extrema, saddle points and bifurcations in scale-space and completely free from tuning parameters.

In Chapter 8 we will show how some directly available information from this scale-space primal sketch can be used for extraction of significant image structure. Before that we will in the next chapter investigate some of the theoretical properties of the representation and then in Chapter 7 describe an algorithm for actually computing it.

¹⁰More detailed information about what type of information can be contained in a data structure representing the scale-space primal sketch is given in Appendix A.6.3.

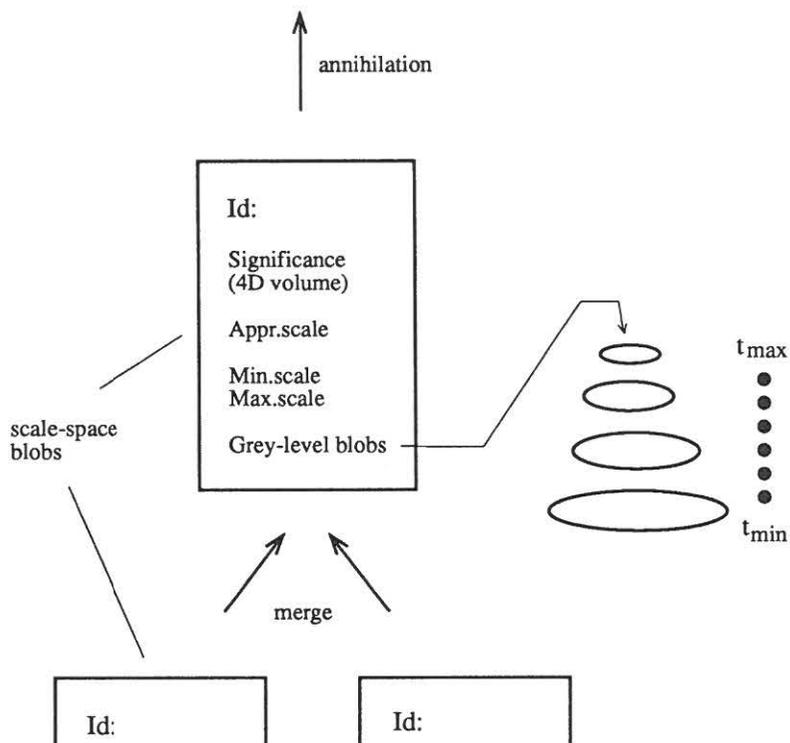


Figure 5.14: *The scale-space primal sketch is a tree-like multi-scale representation of blobs with the scale-space blobs as basic primitives (nodes) and the relations (bifurcations) between scale-space blobs at different levels of scale as branches.*

Chapter 6

Evolution Properties in Scale-Space: Drift Velocities and Bifurcation Events

It is well-known that scale-space smoothing leads to shape distortions. For example, features like local extrema, edges, blobs etc can be expected to drift when the underlying grey-level image is subject to blurring.

Aspects of this phenomenon have been studied by other authors from different viewpoints. Canny [Can86] discussed the general trade-off problem between detection and localization occurring in edge detection. Bergholm [Ber87] estimated the drift velocity of edges for a set of plausible configurations with the aim of estimating a step size for scale changes in the edge focusing algorithm. Berzins [Ber84] has analyzed the localization error for zero-crossings of the Laplacian of the Gaussian.

Other kinds of phenomena affecting the topology may also occur. Blobs can disappear, merge and split as developed by Koenderink and van Doorn [Koe86]. Similar transitions apply to edges, zero-crossings of the Laplacian, corners etc. Such events are usually called bifurcations.

In this work we will perform a study of critical points, that is local extrema and saddle points, and investigate in detail what happens to those features when the underlying image undergoes scale-space smoothing. We will essentially

- develop how these feature points can be expected to behave generically when the scale parameter in scale-space changes
- derive an expression for their drift velocity
- classify their behaviour at bifurcation situations into a discrete set of generic situations
- give a coarse estimate to the global problem of how the number of local extrema can be expected to vary with scale.

The results we will arrive at are not based on any specific models for the intensity variations in the image but are generally valid under rather weak a priori assumptions. Although the results are expressed in a general form the primary intention with the study is to provide a further theoretical basis of the scale-space primal sketch concept. In this context the results to be presented will find their main application to

- the formal construction and definition of the primitives (scale-space blobs) in the scale-space primal sketch. The scale-space blobs are defined as families of grey-level blobs,

which in turn are directly determined by pairs of critical points. This treatment allows for precise mathematical definitions of those concepts.

- providing a theoretical basis for the linking algorithm necessary when computing the representation.
- giving further motivations for the normalization process with respect to “expected scale-space behaviour”, which is necessary when defining the significance measures of the scale-space blobs.

In other words, we will try to explain what happens when scale changes in scale-space, especially with application to the scale-space primal sketch. Therefore, special attention will be given to the objects called grey-level blobs and scale-space blobs.

Before starting, let us point out that some of the results to be presented are (at least partly) known or touched upon before, see e.g. Koenderink and van Doorn [Koe84, Koe86] and Koenderink [Koe90a]. Bifurcations in scale-space have also been studied by Johansen et.al. [Joh86], who have shown that a band-limited one-dimensional signal up to a multiplicative constant is determined by its “toppoints”, that is the points in scale-space where bifurcations take place.

The purpose of this treatment is to develop systematically and comprehensively what can be said about the behaviour in scale-space of critical points using elementary mathematical techniques and to convey an intuitive feeling for the qualitative behaviour in the different generic cases. Detailed calculations will also be given showing the behaviour of blobs in a set of “characteristic examples”.

The scale-space concept we will deal with is the traditional diffusion based scale-space for continuous signals developed by Witkin [Wit83a], Koenderink, van Doorn [Koe84, Koe86] and Babaud et al. [Bab86], which is given by the solution to the diffusion equation, in one and two dimensions respectively,

$$\frac{\partial L}{\partial t} = \frac{1}{2} \frac{\partial^2 L}{\partial x^2} \quad (6.1)$$

$$\frac{\partial L}{\partial t} = \frac{1}{2} \left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \right) \quad (6.2)$$

with initial condition $L(\cdot; 0) = f(\cdot)$ where f indicates the original signal.

The chapter is organized as follows: In Section 6.1 we start by analysing the evolution of non-degenerate critical points as scale changes. This results in drift velocity estimates useful both for extremum points and straight edges as well as precise definitions of the notions of extremum paths, saddle paths and scale-space blobs. Then Section 6.2 gives a classification of the generic behaviour around degenerate critical points, which also leads to a classification of which blob events are possible in scale-space. Further illustrations to these results are given in Section 6.3 and Section 6.4 where detailed calculations are carried out for a set of characteristic examples. In Section 6.5 we study another problem, arising for instance when defining the concept of effective scale, concerning how the density of local extrema in a signal in scale-space can be expected to vary with scale. The analysis is carried out both for continuous and discrete signals and the results from the two approaches are compared. Finally, Section 6.6 gives a brief summary of the main results.

6.1 Trajectories of Critical Points in Scale-Space

In many situations it is of interest to estimate the drift velocity of critical points when the scale parameter varies. Such information is useful for instance when estimating the localization error of feature points due to blurring or when tracking local extrema or related entities between scales as done for instance by Lifshitz and Pizer [Lif87] or in the scale-space primal sketch. In non-degenerate situations, that is when the second differential is a non-degenerate quadratic form, we can base such an analysis on the implicit function theorem.

Definition 6.1 (Critical point (2D))

A point (x_0, y_0) is a critical point of a mapping $f : R^2 \rightarrow R$ if the gradient in this point

$$(\nabla f)(x_0, y_0) = \left(\begin{array}{c} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{array} \right)_{(x_0, y_0)} \quad (6.3)$$

is zero. The critical point is said to be non-degenerate if the Hessian matrix in this point

$$(\mathcal{H}f)(x_0, y_0) = \left(\begin{array}{cc} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial y \partial x} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{array} \right)_{(x_0, y_0)} \quad (6.4)$$

is non-singular. Otherwise it is called degenerate.

Lemma 6.1 (Behaviour of critical points in continuous scale-space (2D))

Let $L : R^2 \times R_+ \rightarrow R$ be the scale-space representation of a two-dimensional continuous signal given by the diffusion equation (6.2). Assume that at some scale level $t_0 > 0$ the point (x_0, y_0) is a non-degenerate critical point for the mapping $(x, y) \mapsto L(x, y; t_0)$.

Then there exist an open set $S_{(x_0, y_0; t_0)} \subset R^2 \times R_+$ and an open interval $I_{t_0} \subset R$ with $(x_0, y_0; t_0) \in S_{(x_0, y_0; t_0)}$ and $t_0 \in I_{t_0}$ having the following property: To every $t_1 \in I_{t_0}$ there corresponds a unique (x_1, y_1) such that $(x_1, y_1; t_1) \in S_{(x_0, y_0; t_0)}$ and (x_1, y_1) is a non-degenerate critical point for the mapping $(x, y) \mapsto L(x, y; t_1)$.

If this (x_1, y_1) is defined to be $r(t_1)$ then r is a continuously differentiable mapping $I_{t_0} \rightarrow R^2$ such that

- $r(t_0) = (x_0, y_0)$
- $r(t_1)$ is for every $t_1 \in I_{t_0}$ a non-degenerate critical point for the mapping $(x, y) \mapsto L(x, y; t_1)$.
- the derivative of r with respect to t in the point (x_0, y_0) is given by

$$\frac{dr}{dt}(t_0) = -\frac{1}{2} \left(\begin{array}{cc} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial y \partial x} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{array} \right)_{(x_0, y_0)}^{-1} \left[\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left(\begin{array}{c} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \end{array} \right) \right]_{(x_0, y_0)} \quad (6.5)$$

Proof: The result can be proved directly by a straight-forward application of the implicit function theorem to the current situation. For the sake of clarity we review its formulation as expressed and proved by Rudin, [Rud76]. We will adapt the following notation: Assume

that $\xi \in R^n$ and $\eta \in R^m$. The derivative of a mapping $f : R^{n+m} \rightarrow R^n$ in a point $(\xi_0; \eta_0)$ is given by the $n \times (n + m)$ matrix

$$A = [f'(\xi_0; \eta_0)] = \begin{pmatrix} \frac{\partial f_1}{\partial \xi_1} & \cdots & \frac{\partial f_1}{\partial \xi_n} & \frac{\partial f_1}{\partial \eta_1} & \cdots & \frac{\partial f_1}{\partial \eta_m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \xi_1} & \cdots & \frac{\partial f_n}{\partial \xi_n} & \frac{\partial f_n}{\partial \eta_1} & \cdots & \frac{\partial f_n}{\partial \eta_m} \end{pmatrix}_{(\xi_0; \eta_0)} \quad (6.6)$$

which can be decomposed into one $n \times n$ matrix B and one $n \times m$ matrix C , where

$$B = \begin{pmatrix} \frac{\partial f_1}{\partial \xi_1} & \cdots & \frac{\partial f_1}{\partial \xi_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \xi_1} & \cdots & \frac{\partial f_n}{\partial \xi_n} \end{pmatrix}_{(\xi_0; \eta_0)} \quad \text{and} \quad C = \begin{pmatrix} \frac{\partial f_1}{\partial \eta_1} & \cdots & \frac{\partial f_1}{\partial \eta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \eta_1} & \cdots & \frac{\partial f_n}{\partial \eta_m} \end{pmatrix}_{(\xi_0; \eta_0)} \quad (6.7)$$

Implicit function theorem:

Let f be a continuously differentiable mapping of an open set $E \subset R^{n+m}$ into R^n , such that $f(a; b) = 0$ for some point $(a; b) \in E$. Put $A = \begin{pmatrix} B & C \end{pmatrix} = [f'(a; b)]$ and assume that B is invertible. Then there exist open sets $U \subset R^{n+m}$ and $W \subset R^m$, with $(a; b) \in U$ and $b \in W$, having the following property: To every $\eta \in W$ corresponds a unique ξ such that

$$(\xi; \eta) \in U \quad \text{and} \quad f(\xi; \eta) = 0 \quad (6.8)$$

If this ξ is defined to be $g(\eta)$, then g is a continuously differentiable mapping of W into R^n satisfying

$$g(b) = a \quad \text{and} \quad f(g(\eta); \eta) = 0 \quad (\eta \in W). \quad (6.9)$$

The derivative of g with respect to η in b is given by

$$g'(b) = -B^{-1}C \quad (6.10)$$

Moreover, we can directly observe that, since here, L is a solution of the diffusion equation (6.2) for strictly positive t it follows that L and hence also the mapping $(x, y) \mapsto L(x, y; t)$ will be continuously differentiable (in fact infinitely continuously differentiable) for all (x, y) and t . Hence, the existence of derivatives of low order will be no problem in our treatment.

Given the scale-space representation $L : R^2 \times R_+ \rightarrow R$, we define an auxiliary function $h : R^2 \times R_+ \rightarrow R^2$ by

$$h(x, y; t) = \begin{pmatrix} h_1(x, y; t) \\ h_2(x, y; t) \end{pmatrix} = \begin{pmatrix} \frac{\partial L}{\partial x}(x, y; t) \\ \frac{\partial L}{\partial y}(x, y; t) \end{pmatrix} \quad (6.11)$$

Then a point (x_1, y_1) is a critical point of the mapping $(x, y) \mapsto L(x, y; t_1)$ if and only if $h(x_1, y_1; t_1) = 0$. The derivative of h in $(x_0, y_0; t_0)$ is given by

$$A = [h'(x_0, y_0; t_0)] = \begin{pmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial t} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial t} \end{pmatrix}_{(x_0, y_0; t_0)} = \begin{pmatrix} B & C \end{pmatrix} \quad (6.12)$$

where

$$B = \begin{pmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} \end{pmatrix}_{(x_0, y_0; t_0)} = \begin{pmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial y \partial x} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{pmatrix}_{(x_0, y_0; t_0)} \quad (6.13)$$

$$C = \begin{pmatrix} \frac{\partial h_1}{\partial t} \\ \frac{\partial h_2}{\partial t} \end{pmatrix}_{(x_0, y_0; t_0)} = \begin{pmatrix} \frac{\partial^2 L}{\partial t \partial x} \\ \frac{\partial^2 L}{\partial t \partial y} \end{pmatrix}_{(x_0, y_0; t_0)} \quad (6.14)$$

In the last equations we have also replaced h_1 with $\frac{\partial L}{\partial x}$ and h_2 with $\frac{\partial L}{\partial y}$. Since (x_0, y_0) is a non-degenerate critical point of the mapping $(x, y) \mapsto L(x, y; t_0)$ we have that $h(x_0, y_0; t_0) = 0$ and that B is non-singular. Hence, we can apply the implicit function theorem and accordingly there exist open sets $U \subset \mathbb{R}^2 \times \mathbb{R}$ and $W \subset \mathbb{R}$ with $(x_0, y_0; t_0) \in U$ and $t_0 \in W$ such that there to every $t \in W$ corresponds a unique (x, y) satisfying $(x, y; t) \in U$ and $h(x, y; t) = 0$. In other words, there exists an open neighbourhood U around $(x_0, y_0; t_0)$ such that for every $t \in W$ there exists a unique critical point (x, y) that we can define as $r(t)$. The derivative of this mapping $r : W \rightarrow \mathbb{R}^2$ is

$$\frac{dr}{dt}(t_0) = -B^{-1}C = -\frac{1}{2} \begin{pmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial y \partial x} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{pmatrix}_{(x_0, y_0; t_0)}^{-1} \begin{pmatrix} \frac{\partial^2 L}{\partial t \partial x} \\ \frac{\partial^2 L}{\partial t \partial y} \end{pmatrix}_{(x_0, y_0; t_0)} \quad (6.15)$$

Moreover, since L satisfies the diffusion equation (6.2) we can replace the derivatives with respect to t by derivatives with respect to x and y via

$$\frac{\partial}{\partial t} = \frac{1}{2} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \quad (6.16)$$

to obtain the result in Equation (6.5).

With the formulation so far, nothing ensures the critical point to be non-degenerate. Since, however, the Hessian $(\mathcal{H}L) = \frac{\partial^2 L}{\partial x^2} \frac{\partial^2 L}{\partial y^2} - \frac{\partial^2 L}{\partial x \partial y} \frac{\partial^2 L}{\partial y \partial x}$ is a continuous function of $(x, y; t)$ and is non-zero in $(x_0, y_0; t_0)$ it follows that there exists some open neighbourhood $V \subset U$ with $(x_0, y_0; t_0) \in V$ where $(\mathcal{H}L)$ is non-zero. If we let $S(x_0, y_0; t_0) = V$ and $I_{t_0} = r^{-1}(V) \cap W$ we are guaranteed that the critical points given by the mapping $r : I_{t_0} \rightarrow \mathbb{R}^2$ are non-degenerate. The uniqueness property will be trivially preserved. \square

A corresponding result does of course also hold in one dimension. For the sake of clarity we state the necessary definitions and the result. The proof is obvious from the two-dimensional case.

Definition 6.2 (Critical point (1D))

A point x_0 is a critical point of a mapping $f : \mathbb{R} \rightarrow \mathbb{R}$ if the first derivative in this point $\frac{df}{dx}(x_0)$ is zero. The critical point is said to be non-degenerate if the second derivative in this point $\frac{d^2 f}{dx^2}(x_0)$ is non-zero, otherwise degenerate.

Lemma 6.2 (Behaviour of critical points in continuous scale-space (1D))

Let $L : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$ be the scale-space representation of a one-dimensional continuous signal given by the diffusion equation (6.1). Assume that at some scale level $t_0 > 0$ the point x_0 is a non-degenerate critical point for the mapping $x \mapsto L(x; t_0)$.

Then there exist an open set $S_{(x_0; t_0)} \subset \mathbb{R} \times \mathbb{R}_+$ and an open interval $I_{t_0} \subset \mathbb{R}_+$ with $(x_0; t_0) \in S_{(x_0; t_0)}$ and $t_0 \in I_{t_0}$ having the following property: To every $t_1 \in I_{t_0}$ there

corresponds a unique x_1 such that $(x_1; t_1) \in S_{(x_0; t_0)}$ and x_1 is a non-degenerate critical point for the mapping $x \mapsto L(x; t_1)$.

If this x_1 is defined to be $r(t_1)$ then r is a continuously differentiable mapping $I_{t_0} \rightarrow R$ such that

- $r(t_0) = x_0$
- $r(t_1)$ is for every $t \in I_{t_0}$ a non-degenerate critical point for the mapping $x \mapsto L(x; t_1)$.
- the derivative of r with respect to t in the point x_0 is given by

$$\frac{dr}{dt}(t_0) = -\frac{1}{2} \frac{\frac{\partial^3 L}{\partial x^3}(x_0; t_0)}{\frac{\partial^2 L}{\partial x^2}(x_0; t_0)} \quad (6.17)$$

6.1.1 Interpretation: Drift Velocity Estimates

These lemmas express how critical points in general can be expected to behave in scale-space. As indicated above, one of the most immediate interpretations is that they give straightforward estimates of the drift velocity of critical points in under scale-space smoothing.

Proposition 6.3 (Drift velocity of critical points in scale-space (2D))

Given the scale-space representation $L : R^2 \times R_+ \rightarrow R$, assume that for some scale level $t_0 > 0$ the point (x_0, y_0) is a non-degenerate critical point for the mapping $(x, y) \mapsto L(x, y; t_0)$. Then the drift velocity of that critical point when the scale parameter changes is given by

$$\begin{aligned} \frac{dr}{dt}(t_0) &= -\frac{1}{2} \left(\begin{array}{cc} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial y \partial x} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{array} \right)_{(x_0, y_0)}^{-1} \left[\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left(\begin{array}{c} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \end{array} \right) \right]_{(x_0, y_0)} \\ &= -\frac{1}{2 \left(\frac{\partial^2 L}{\partial x^2} \frac{\partial^2 L}{\partial y^2} - \left(\frac{\partial^2 L}{\partial x \partial y} \right)^2 \right)} \left(\begin{array}{c} \frac{\partial^2 L}{\partial y^2} \frac{\partial^3 L}{\partial x^3} + \frac{\partial^2 L}{\partial y^2} \frac{\partial^3 L}{\partial x \partial y^2} - \frac{\partial^2 L}{\partial y \partial x} \frac{\partial^3 L}{\partial x^2 \partial y} - \frac{\partial^2 L}{\partial y \partial x} \frac{\partial^3 L}{\partial y^3} \\ -\frac{\partial^2 L}{\partial x \partial y} \frac{\partial^3 L}{\partial x^3} - \frac{\partial^2 L}{\partial x \partial y} \frac{\partial^3 L}{\partial x^2} + \frac{\partial^2 L}{\partial x^2} \frac{\partial^3 L}{\partial x^2 \partial y} + \frac{\partial^2 L}{\partial x^2} \frac{\partial^3 L}{\partial y^3} \end{array} \right)_{(x_0, y_0)} \end{aligned} \quad (6.18)$$

Proof: Follows directly from Lemma 6.1 above. In the last line we have used the well-known inversion formula for 2×2 matrices

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right)^{-1} = \frac{1}{ad - bc} \left(\begin{array}{cc} d & -b \\ -c & a \end{array} \right) \quad (6.19)$$

for writing down an explicit expression for the drift velocity in terms of spatial derivatives of the smoothed grey-level data. \square

Note that the drift velocity actually can become infinite when the the Hessian becomes singular. At such points bifurcations can occur, as will be developed in Sections 6.2-6.4.

Corollary 6.4 (Unbounded drift velocity of critical points)

The drift velocity of critical points may tend to infinity near bifurcations.

These conclusions are of course valid also in one dimension. The expression for the drift velocity is, however, much simpler in this case:

Proposition 6.5 (Drift velocity of critical points in scale-space (1D))

Given the scale-space representation $L : R \times R_+ \rightarrow R$, assume that for some scale level $t_0 > 0$ the point x_0 is a non-degenerate critical point for the mapping $x \mapsto L(x; t_0)$. Then the drift velocity of that critical point as the scale parameter changes is given by

$$\frac{dr}{dt}(t_0) = -\frac{1}{2} \frac{\frac{\partial^3 L}{\partial x^3}(x_0; t_0)}{\frac{\partial^2 L}{\partial x^2}(x_0; t_0)} \quad (6.20)$$

This estimate can easily be extended to comprise edges as well. For simplicity, assume that the edge under study is sufficiently long and sufficiently close to a straight line such that a one-dimensional analysis is a valid approximation. Further, without loss of generality assume that the coordinate system is oriented such that the edge is perpendicular to the x -axis. Then, we can use for instance non-maximum suppression to define the location of the edge as those points where the first derivative along the gradient direction (that is here the x -direction) has a local maximum. In other words, the edge is defined by those points where the second derivative along the gradient direction is zero. Now, since under these conditions, critical points are given by zeros in the first derivative and edge points by zeros in the second derivative, we can apply Proposition 6.5 to this situation just by replacing L by $\frac{\partial L}{\partial x}$. Hence,

Proposition 6.6 (On the drift velocity of straight edges in scale-space (2D))

Given the scale-space representation $L : R^2 \times R_+ \rightarrow R$, assume that for some scale level $t_0 > 0$ the point (x_0, y_0) is an edge point along a long straight line. Moreover, assume that the coordinate system is aligned to the edge such that the x -direction is perpendicular to the edge and further that the third derivative in this direction is non-zero. Then the drift velocity of the edge point as the scale parameter changes is given by

$$\frac{dr}{dt}(t_0) = -\frac{1}{2} \frac{\frac{\partial^4 L}{\partial x^4}(x_0, y_0; t_0)}{\frac{\partial^3 L}{\partial x^3}(x_0, y_0; t_0)} \quad (6.21)$$

A similar idea, although with just an approximate derivation, has been expressed by Zhuang and Huang [Zhu86].

This analysis is applicable also to edges given by zero-crossings, provided that the second derivative along the edge direction (here the y -direction) is sufficiently small to be neglected. Trivially, an identical result holds for the edges of a one-dimensional signals. Note also, that we have not made any specific assumptions about the shape of the intensity profile perpendicular to the edge. Hence, the result is valid for any configuration that can be described by a one-dimensional analysis.

Corollary 6.7 (Unbounded drift velocity of straight edges)

The drift velocity of edges may tend to infinity when two adjacent parallel edges are just about to merge into one.

This result can, for instance, be used for explaining a recent observation by Zhang and Bergholm [Zha91], where they noted that configurations consisting of two adjacent edges, a so-called “staircase edge” — see Figure 6.1, can lead to a rapid edge drift when the scale changes, which in turn violates the assumptions behind the step size estimate used in the edge focusing algorithm [Ber87]. In this situation the third derivative is in fact very close to zero.

Finally, regarding the drift velocity estimates for local extrema and edges, let us point out that although the drift velocity *momentarily* may tend to infinity, the total drift (integrated over a scale interval of finite length) will always be finite. What the results mean, is that it is not possible to derive any *uniform* upper bound for the drift velocity of these features. Given any scale interval of length Δt and any distance Δx it is always possible to find a signal such that the total drift of a feature during the time Δt exceeds Δx . This property emphasizes the need for algorithms based on *adaptive* sampling along the scale direction.

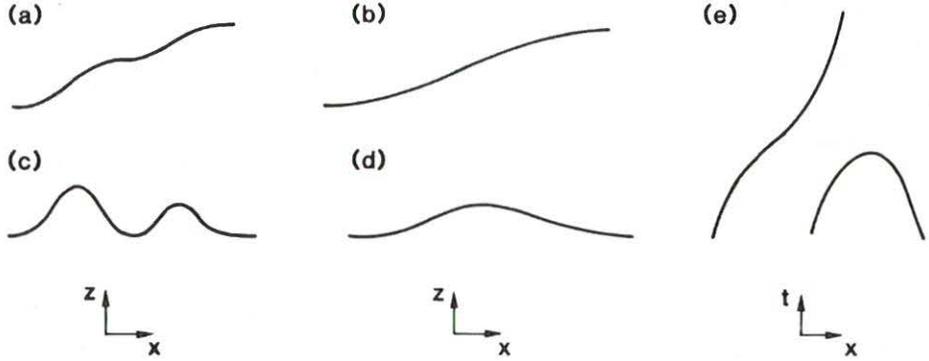


Figure 6.1: (a) A “staircase edge” can lead to a rapid edge drift. This behaviour can be explained by noting that (b) after sufficient amount of blurring the configuration will tend to a “diffuse step edge” and by studying the derivatives of (c) the original signal (d) the signal after strong smoothing. (e) By considering the paths the zero-crossings of the Laplacian will describe as scale changes it is easy to realize that when the edge points tend to each other the drift velocity will tend to infinity. See also Sections 3-5 for a more detailed description of the behaviour at bifurcation situations, in particular Section 5.2 concerning this configuration.

6.1.2 Interpretation: Extremal Paths

Another consequence of Lemma 6.1 and Lemma 6.2 is that a non-degenerate critical point existing at a certain level of scale in general can be traced to a similar critical point both at a slightly coarser and a slightly finer scale. By continuation, such local paths obtained from the implicit function theorem can be extended to curves as long as the Hessian determinant remains non-zero. Hence, we get trajectories of critical points that in general will be regular curves, delimited from above and below by some scale values, t_{min} and t_{max} , at which the Hessian is zero and the critical point hence degenerate. One easily shows, that the type of critical point will not change along such a path, see below. Therefore, we have:

Proposition 6.8 (Extremal paths and saddle paths (2D))

Given the scale-space representation $L : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$, assume that for some scale level $t_0 > 0$ the point (x_0, y_0) is a non-degenerate maximum (minimum/saddle) point for the mapping $(x, y) \mapsto L(x, y; t_0)$. Then there exists a unique trajectory of maximum (minimum/saddle) points $r : I_{t_0} \rightarrow \mathbb{R}^2$ with $t_0 \in I_{t_0}$ such that $r(t_0) = (x_0, y_0)$ and that $r(t)$ for every t is a local

maximum (minimum/saddle) point for the mapping $(x, y) \mapsto L(x, y; t)$. This trajectory is called the extremal (extremal/saddle) path through $(x_0, y_0; t_0)$ and is denoted by $M_{(x_0, y_0; t_0)}^+$ ($M_{(x_0, y_0; t_0)}^-/S_{(x_0, y_0; t_0)}$). The associated scale interval, where the path is defined, is delimited by a minimum scale t_{\min} and a maximum scale t_{\max} . At those scales the critical paths end up in degenerate critical points unless the minimum scale is zero or the maximum scale infinite. At all interior points the associated critical points are non-degenerate,

Proof: The existence of trajectories of critical points is evident from Lemma 6.1 and the previous discussion about continuation. What remains to verify is that the nature of the critical point does not change under scale-space smoothing. It is obvious that a local maximum (minimum) cannot be transformed into a saddle point or vice versa, because if the Hessian would change sign then it would first become zero, since it is a continuous function of the scale parameter. However, then, by definition, the trajectory would be cut off by a degenerate critical point into two separate segments.

Moreover, a maximum point cannot be transformed into a minimum point or opposite, since then (at least) the partial derivative $\frac{\partial^2 L}{\partial x^2}$ would need to change sign. However, such a sign change implies that this derivative would first become zero (because of continuity), which in turn means that the quadratic form would become indefinite, i.e. the point would get transformed into a saddle point. Above we have shown that such a transition has to go through a degenerate critical point which means that the trajectory would be cut off into at least two parts. \square

The one-dimensional situation is similar, although simpler, since there are no stable saddle points in this case.

Proposition 6.9 (Extremal paths (1D))

Given the scale-space representation $L : R \times R_+ \rightarrow R$, assume that for some scale level $t_0 > 0$ the point x_0 is a non-degenerate maximum (minimum) point for the mapping $x \mapsto L(x; t_0)$. Then there exists a unique trajectory of maximum (minimum) points $r : I_{t_0} \rightarrow R$ with $t_0 \in I_{t_0}$ such that $r(t_0) = x_0$ and $r(t)$ is for every t local maximum (minimum) point for the mapping $x \mapsto L(x; t)$. This trajectory is called the extremal path through $(x_0; t_0)$ and is denoted by $M^+(x_0; t_0)$ ($M^-(x_0; t_0)$). The associated scale interval, where the path is defined, is delimited by a minimum scale t_{\min} and a maximum scale t_{\max} . At those scales the critical paths end up in degenerate critical points unless the minimum scale is zero or the maximum scale infinite. At all interior points the associated extremum points are non-degenerate,

6.1.3 Formal Definition of Scale-Space Blob

The treatment of extremal paths above allows for a more formal definition of scale-space blobs — the basic primitives in the scale-space primal sketch. In Chapter 5 grey-level blobs were defined as local extrema with extent and scale-space blobs in turn as families of those. More precisely, a grey-level blob of a two dimensional signal was given by a pair consisting of a local extremum and a saddle point and in one dimension by a maximum and minimum point, implying a one-to-one correspondence between local extrema and grey-level blobs. The previous definition of scale-space blob was, however, intuitive: “similar blobs at adjacent levels of scale were linked into scale-space blobs”. The linking process proceeded until no such linking could be performed, i.e., until a bifurcation was encountered. The idea behind this construction was to identify and group similar features at different scales into higher order and unified objects.

The notion of extremal paths makes it possible to express this linking criterion in a more formal way. Consider the two-dimensional case and study a non-degenerate local extremum point at some level of scale. Then, by Proposition 6.8 there exists a unique trajectory of local extrema associated with this point such that all points along this path are local extrema of the same kind and this path is delimited by two scales, t_{min} and t_{max} . At all interior scales of this interval the associated extremum points will be non-degenerate, while at the end points they will, by definition, be degenerate. The scale-space blob associated with the local extremum point we originally started with, will be defined as a subset of the union of all grey-level blobs corresponding to the local extrema along the extremum path through the original extremum point. To be more precise we first define a natural concept:

Definition 6.3 (Delimiting saddle point (2D))

Let E be an extremum point and S a saddle point together defining the extent of a grey-level blob. Then S is said to be the delimiting saddle point of E , denoted $S_{delimit}(E)$.

The delimiting saddle points associated with the extremum points of an extremum path need of course not all be on the same saddle path, but may jump between different saddle paths. Generically this occurs at a discrete set of scales at which the extremum point and the (two) involved saddle points are non-degenerate.

If the delimiting saddle point (or the extremum point) is involved in a bifurcation then we say that a blob event has occurred for the scale-space blob associated with (the segment of) the extremum path. It is therefore natural to proceed with the linking as long as the extremum points and their delimiting saddle points are non-degenerate critical points and to stop it when either of the critical points degenerates. Hence, a scale-space blob will be given as the union of the grey-level blobs along a subset $[t'_{min}, t'_{max}]$ of the previous scale interval $[t_{min}, t_{max}]$. In order to obtain a closed object it might however be convenient to define the *scale-space blob* as the closure of the previously suggested set. To summarize:

Definition 6.4 (Scale-space blob (2D))

Given the scale-space representation $L : R^2 \times R_+ \rightarrow R$ of a two-dimensional signal $f : R^2 \rightarrow R$, let $r : [t_{min}, t_{max}] \rightarrow R^2$ be an extremal path as formulated in Proposition 6.8 such that $r(t_{min})$ and $r(t_{max})$ are degenerate critical points. Further let $[t'_{min}, t'_{max}] \subset [t_{min}, t_{max}]$ be a scale interval where for all interior scales the delimiting saddle points of the extremum points along the extremum path are non-degenerate and at the end points, either of $r(t'_{min})$ and $S_{delimit}(r(t'_{min}))$ and also either of $r(t'_{max})$ and $S_{delimit}(r(t'_{max}))$ are degenerate critical points. Then the scale-space blob associated with the segment $\bar{r} : [t'_{min}, t'_{max}] \rightarrow R^2$ of the extremal path is defined as the set

$$S_{blob}(r) = \text{Closure}(\{ (x, y, z; t) \in R^2 \times R_+ \times R : (t'_{min} < t < t'_{max}) \wedge ((x, y, z) \in G_{blob}(r(t))) \}) \quad (6.22)$$

where the symbol $G_{blob}(r(t))$ denotes the grey-level blob associated with the extremum point $r(t)$.

It is natural to define the *support region* of the scale-space blob as

$$S_{support}(r) = \text{Closure}(\{ (x, y; t) \in R^2 \times R_+ : (t'_{min} < t < t'_{max}) \wedge ((x, y) \in G_{support}(r(t))) \}) \quad (6.23)$$

One easily verifies that this construction implies that

$$S_{support}(r) = \{(x, y; t) \in R^2 \times R_+ : (x, y, z; t) \in S_{blob}(r) \text{ for some } z\} \quad (6.24)$$

In most figures presented in earlier papers it is this projection of the four-dimensional scale-space blob that has been illustrated.

Strictly, in this original coordinate system the *scale-space blob volume* is given by

$$S_{volume}(r) = \int \int \int \int_{S_{blob}(r)} dx dy dz dt = \int_{t=t'_{min}}^{t'_{max}} G_{volume}(r(t)) dt \quad (6.25)$$

where $G_{volume}(r(t))$ is the grey-level blob volume of the grey-level blob associated with the extremum point $r(t)$. However, when the scale-space blob volume is to be used as a significance measure in the scale-space primal sketch it turns out that some transformations need to be done in order to enable a uniform treatment of structures over scale. The aim with that normalization is to achieve a significance measure that neither favours fine scales to coarse scales nor the opposite. Therefore, we define a *normalized scale-space blob volume* as

$$S_{volume,norm}(r) = \int_{t=t'_{min}}^{t'_{max}} V_{trans}(G_{volume}(r(t)); t) d(\tau_{eff}(t)) \quad (6.26)$$

where $t_{eff} : R \rightarrow R$ is a transformation function mapping the ordinary scale parameter into a transformed scale parameter called effective scale and $V_{trans} : R \times R_+ \rightarrow R$ is a corresponding transformation function normalizing the grey-level blob volumes into a more uniform behaviour over scale, see Chapter 5 for details.

For one-dimensional signals the treatment is similar and a *scale-space blob* associated with a segment of an extremum path is defined as follows: We express the definition for bright blobs only. The case with dark blobs is similar.

Definition 6.5 (Delimiting minimum point (bright blobs 1D))

Let M^+ be a maximum point and M^- a minimum point together defining the extent of a bright grey-level blob. Then M^- is said to be the delimiting minimum point of M^+ , denoted $M_{delimit}^-(M^+)$.

Definition 6.6 (Scale-space blob (bright blobs in 1D))

Given the scale-space representation $L : R \times R_+ \rightarrow R$ of a one-dimensional signal $f : R \rightarrow R$, let $r : [t_{min}, t_{max}] \rightarrow R$ be a maximum path as formulated in Proposition 6.9 such that $r(t_{min})$ and $r(t_{max})$ are degenerate critical points. Further let $[t'_{min}, t'_{max}] \subset [t_{min}, t_{max}]$ be a scale interval where for all interior scales the delimiting minima of the maximum points along the maximum path are non-degenerate and at the end points, either of $r(t'_{min})$ and $M_{delimit}^-(r(t'_{min}))$ and also either of $r(t'_{max})$ and $M_{delimit}^-(r(t'_{max}))$ are degenerate critical points. Then the scale-space blob associated with the segment $r' : [t'_{min}, t'_{max}] \rightarrow R$ of the extremal path is defined as the set

$$S_{blob}(r) = \text{Closure}(\{(x, z; t) \in R \times R \times R_+ : (t'_{min} < t < t'_{max}) \wedge ((x, z) \in G_{blob}(r(t)))\}) \quad (6.27)$$

where the symbol $G_{blob}(r(t))$ denotes the grey-level blob associated with the extremum point $r(t)$.

It should be obvious how the related entities, support region and scale-space blob volume, should be defined in an analog manner.

6.2 Behaviour Near Singularities: Classification

The results so far describe the evolution properties in scale-space of non-degenerate critical points. When we want to investigate the behaviour of degenerate critical points, the approach with the implicit function theorem is no longer applicable, since at those points the Hessian matrix is singular. One can show that the critical points of a solution to the diffusion equation will, in general, be non-degenerate and that, generically, critical points will be degenerate at isolated points only.

Useful methods for analysing the behaviour around these points, where bifurcations can occur, can be obtained from a branch of mathematics known as singularity or catastrophe theory. In this treatment we will make use of some existing results from this field to express what kind of behaviour can be expected at the singularities in the scale-space representation of a one-dimensional or two-dimensional signal. We will not make any attempt to summarize the full theory behind the important theorems, but instead just briefly review some of the definitions and results of highest relevance to this application. We refer the reader to e.g. Poston and Stewart [Pos78], Gibson [Gib79] or Bruce and Giblin [Bru84] for an application-oriented introduction and to e.g. Arnold et al. [Arn81, Arn85, Arn88], Golubitsky and Schaeffer [Gol85] or Lu [Lu76] for a more rigorous treatment of the subject.

6.2.1 Background

The main purpose with the analysis in singularity or catastrophe theory is to deal with the qualitative behaviour of functions in the neighbourhoods of points where singularities occur. An important concept in this context is the notion of equivalence. Two functions and or two families of functions are said to be equivalent if they show the same kind of qualitative behaviour. More precisely, the notion of equivalence of means that functions (or families of functions) are similar up to a diffeomorphic change of variables:

Equivalence of functions:

Two functions $f_1, f_2 : R^n \rightarrow R$ are said to be (right) equivalent around 0 if there is a local diffeomorphism $y : R^n \rightarrow R^n$ around 0 and a constant γ such that

$$f_2(x) = f_1(y(x)) + \gamma \quad (6.28)$$

in some neighbourhood around the point 0.

Equivalence of families of functions:

Two r -parameter families of functions $L_1, L_2 : R^n \times R^r \rightarrow R$ are said to be equivalent if there exist

- a diffeomorphism $e : R^r \rightarrow R^r$
- a smooth map $y : R^n \times R^r \rightarrow R^n$ such that for each $s \in R^r$ the map $y_s : R^n \rightarrow R^n$ defined by $y_s(x) = y(x; s)$ is a diffeomorphism
- a smooth map $\gamma : R^r \rightarrow R$

defined in a neighbourhood around the point 0 such that

$$g(x, s) = f(y_s(x), e(s)) + \gamma(s) \quad (6.29)$$

for all $(x; s) \in R^n \times R^r$ in that neighbourhood.

In other words, two r -parameter families of functions are treated as equivalent if there exists a set of diffeomorphisms such that one of the families can be smoothly transformed into the other one. This notion of equivalence implies that the singularity sets of the families, which are the sets of points where the first differential with respect to the state variable $x \in R^n$ is zero,

$$S_{L_1} = \{(x; u) \in R^n \times R^r : (D_x L_1)(x; u) = 0\} \quad (6.30)$$

$$S_{L_2} = \{(x; u) \in R^n \times R^r : (D_x L_2)(x; u) = 0\} \quad (6.31)$$

will also be equivalent sets up to a diffeomorphic change of variables. In this sense the concept of equivalence can be seen as capturing the property of qualitative similarity.

The equivalence concept is closely linked to the concept of structural stability. Intuitively a function or a family of functions is structurally stable if a sufficiently small perturbation does change the qualitative behaviour of the function or the family. This property is more formally expressed in terms of transversality:

Transverse intersection:

Let X and Y be affine subspaces of R^n of dimensions s and t respectively. They are said to meet transversely if either

- *their intersection $X \cap Y$ is empty, or*
- *$s + t \geq n$ and $\dim(X \cap Y) = s + t - n$*

Two submanifolds of R^n meet transversely at a given point provided either they do not meet or their tangent affine hyperplanes meet transversely.

One of the fundamental results in singularity theory is that the typical qualitative behaviour of families given by a small number of parameters can be expressed completely by the qualitative behaviour of a finite set of families. A famous theorem by Thom classifies the generic behaviour of families of functions with the number of parameters $r \leq 4$ into seven elementary catastrophes. We cite a summarizing result as expressed by Poston and Stewart [Pos78]:

Thom's classification theorem:

Typically an r -parameter family $R^n \times R^r \rightarrow R$ of smooth functions $R^n \times R^r \rightarrow R$, for any n and $r \leq 4$, is structurally stable and is in every point (locally) equivalent to one of the following forms:

- *non-critical: x_1*
- *non-degenerate critical, or Morse: $x_1^2 + \dots + x_i^2 - x_{i+1}^2 - \dots - x_n^2$ ($0 \leq i \leq n$)*
- *degenerate critical — catastrophe*
 - *fold (A_2): $x_1^3 + u_1 x_1 + (M)$*
 - *cusp (A_3): $\pm(x_1^4 + u_2 x_1^2 + u_1 x_1) + (M)$*
 - *swallowtail (A_4): $x_1^5 + u_3 x_1^3 + u_2 x_1^2 + u_1 x_1 + (M)$*
 - *butterfly (A_5): $\pm(x_1^6 + u_4 x_1^4 + u_3 x_1^3 + u_2 x_1^2 + u_1 x_1) + (M)$*
 - *elliptic umbilic (D_4^-): $x_1^2 x_2 - x_2^3 + u_3 x_1^2 + u_2 x_2 + u_1 x_1 + (N)$*
 - *hyperbolic umbilic (D_4^+): $x_1^2 x_2 + x_2^3 + u_3 x_1^2 + u_2 x_2 + u_1 x_1 + (N)$*
 - *parabolic umbilic (D_5): $\pm(x_1^2 x_2 + x_2^4 + u_4 x_2^2 + u_3 x_1^2 + u_2 x_2 + u_1 x_2) + (N)$*

The symbols A_2, A_3 , etc. denote the types of singularities of which the families constitute generic unfoldings while (M) and (N) indicate Morse functions on the forms

$$(M) = x_2^2 + \dots + x_i^2 - x_{i+1}^2 - \dots - x_n^2 \quad (2 \leq i \leq n) \quad (6.32)$$

$$(N) = x_3^2 + \dots + x_i^2 - x_{i+1}^2 - \dots - x_n^2 \quad (2 < i \leq n) \quad (6.33)$$

which must be added on to the previously mentioned expressions to match up the dimensions. This is in strong analogy with the Morse splitting lemma, which states a singularity at a degenerate critical point can be separated into two components:

Splitting lemma for families:

Let $L : R^n \times R^r \rightarrow R$ be smooth. Denote a point in $R^n \times R^r$ by $(x; u) = (x_1, \dots, x_n, u_1, \dots, u_r)$. Suppose that the Hessian

$$HL = \left\{ \frac{\partial^2 L}{\partial x_i \partial x_j} \right\}_{1 \leq i, j \leq n} \quad (6.34)$$

has corank m at $(x; u) = 0$. Then L is equivalent to a family of the form

$$\tilde{L}(y_1(x; u), \dots, y_m(x; u), u) \pm y_{m+1}^2 \pm \dots \pm y_n^2 \quad (6.35)$$

If the Hessian is non-degenerate, i.e. has corank zero, this result reduces to the Morse Lemma for families, containing the ordinary Morse lemma as a special case:

Morse lemma for families:

Let $L : R^n \times R^r \rightarrow R$ be smooth and assume that the Hessian

$$HL = \left\{ \frac{\partial^2 L}{\partial x_i \partial x_j} \right\}_{1 \leq i, j \leq n} \quad (6.36)$$

is non-degenerate at $(x; c) = 0$. Then L is equivalent to a family of the form

$$\pm y_1^2 \pm y_2^2 \pm \dots \pm y_n^2 \quad (6.37)$$

If the number of parameters in the family is increased to five, then a few more catastrophes, not mentioned in Thom's original treatment, will also be possible, see e.g. Poston and Stewart [Pos78]:

- wigwam (A_6) $x_1^7 + u_5 x_1^5 + u_4 x_1^4 + u_3 x_1^3 + u_2 x_1^2 + u_1 x_1 + (M)$
- second elliptic umbilic (D_6^-): $x_1^2 x_2 - x_2^5 + u_5 x_2^3 + u_4 x_2^2 + u_3 x_1^2 + u_2 x_2 + u_1 x_1 + (N)$
- second hyperbolic umbilic (D_6^+): $x_1^2 x_2 + x_2^5 + u_5 x_2^3 + u_4 x_2^2 + u_3 x_1^2 + u_2 x_2 + u_1 x_1 + (N)$
- symbolic umbilic (E_6): $\pm(x_1^3 + x_2^4 + u_5 x_1 x_2^2 + u_4 x_2^2 + u_3 x_1 x_2 + u_2 x_2 + u_1 x_1) + (N)$

According to Poston and Stewart [Pos78] the latter four cases contain geometry not significantly altered from the previous cases, however comprising interchanges between maxima and minima that may be important in certain situations. Material on the classification of singularities of higher order can be found in for instance Arnold et.al. [Arn81, Arn85, Arn88].

6.2.2 Application to Scale-Space Representation: One-Parameter Families

Treating the scale-space embedding of a two-dimensional signal, we can restrict the treatment to one-parameter families. Then the only possible catastrophe will be the one of fold type. To summarize,

Thom's classification theorem applied to one-parameter families:

Typically a one-parameter family $R^n \times R \rightarrow R$ of smooth functions $R^n \rightarrow R$, for any n , is structurally stable and is in every point (locally) equivalent to one of the following forms:

- *non-critical: x_1*
- *non-degenerate critical, or Morse: $x_1^2 + \dots + x_i^2 - x_{i+1}^2 - \dots - x_n^2$ ($0 \leq i \leq n$)*
- *degenerate critical — fold catastrophe (A_2): $x_1^3 + u_1 x_1 \pm x_2^2 \pm \dots \pm x_n^2$*

The A_2 type of singularity in the fold singularity means that the first and second derivatives in one direction are zero while the third derivative in that direction is non-zero. At the singularity point the function is locally equivalent to the function $x^3 \pm y^2$ and from the concept of equivalence between families of functions it follows that any transversal unfolding through a singularity of this type will be equivalent to the family $G_1(x, y; u) = x^3 + ux \pm y^2$, where x and y should be interpreted as state variables and u serve as the parameter.

Therefore, if one is interested in the behaviour of the critical points of a signal during the evolution of the diffusion equation, it should in principle be sufficient to study this situation. For two-dimensional signals the singularity set is given by the solutions of

$$\frac{\partial G_1}{\partial x}(x, y; u) = 3x^2 + u = 0 \tag{6.38}$$

$$\frac{\partial G_1}{\partial y}(x, y; u) = \pm 2y = 0 \tag{6.39}$$

and the bifurcation set by the solution of

$$\frac{\partial G_1}{\partial x}(x, y; u) = 3x^2 + u = 0 \tag{6.40}$$

$$\frac{\partial G_1}{\partial y}(x, y; u) = \pm 2y = 0 \tag{6.41}$$

$$\frac{\partial^2 G_1}{\partial x^2}(x, y; u) = 6x = 0 \tag{6.42}$$

We easily observe that the singularity set is given by $(x_1(u), y_1(u)) = (-\sqrt{-\frac{u}{3}}, 0)$ and $(x_2(u), y_2(u)) = (+\sqrt{-\frac{u}{3}}, 0)$ ($u \leq 0$) and that the bifurcation is an isolated point $(x, y; u) = (0, 0; 0)$. From the sign of the Hessian determinant $|(\mathcal{H}G_1)|(x, y; u) = \pm 12x$ it follows that $(x_1(u), y_1(u))$ are saddle/maximum points and $(x_2(u), y_2(u))$ are minimum/saddle points for every $u < 0$. At $u = 0$ the points merge and then disappear, see also Figure 6.2.

By similar arguments the scale-space representation of a one-dimensional signal will at a bifurcation point be locally equivalent to the unfolding $G_1(x; u) = x^3 + ux$. The same type of calculations as above show that in this case the fold catastrophe instead describes the merging of a maximum point and a minimum point with increasing u , see also Figure 6.2. To summarize,

Corollary 6.10 (Generic behaviour at singularities in scale-space (2D))

The typical behaviour to be expected at singularities in a one-parameter family of two-dimensional continuous signals are annihilations or creations of pairs of local extrema and saddle points.

Corollary 6.11 (Generic behaviour at singularities in scale-space (1D))

The typical behaviour to be expected at singularities in a one-parameter family of one-dimensional continuous signals are annihilations or creations of pairs of local maxima and local minima.

Observe in this context that in the scale-space representation of a one-dimensional signal the number of local extrema cannot increase when the scale parameter increases. This means that creations of pairs of local maxima and minima with increasing scale are impossible if the special structure of the diffusion equation is taken into account. However, as will be demonstrated below, creations of saddle-extremum pairs with increasing scale are possible in the scale-space representation of a two-dimensional signal.

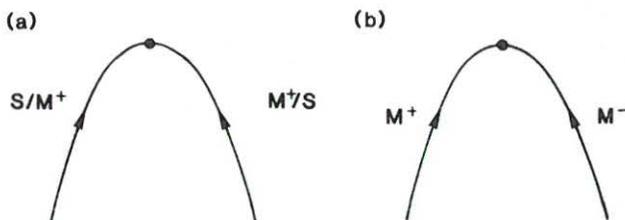


Figure 6.2: (a) The generic behaviour at a singularity of a one-parameter family of two-dimensional functions is described by the unfolding $G_1(x, y; u) = x^3 + ux \pm y^2$. The singularity set of this family, that is the set of critical points to the mapping given by $x \mapsto G_1(x, y; u)$, describes an extremum point and a saddle point that merge along a parabola and then disappearing. (b) For a one-parameter family of one-dimensional functions the behaviour is instead given by $G_1(x; u) = x^3 + ux$. The singularity set corresponds to a similar merge of a maximum point and a minimum point.

6.2.3 Interpretations

By comparisons with earlier theoretical and experimental results we know that these corollaries describe the qualitative behaviour of critical points in scale-space. However, when to give a more detailed interpretation of those results there is one apparent complication. From the equivalence concepts we know that there exist diffeomorphisms such that the singularity set of a solution to the one-dimensional diffusion equation around a bifurcation point $(x_0; t_0)$ in scale-space can be transformed into the singularity set of G_1 around $(0; 0)$. However, there is obviously some directional information lost in the equivalence concept: In which direction should we interpret the u parameter? If we treat u and t as increasing simultaneously, then the situation describes a local minimum and a local maximum merging with increasing t . On the other hand, if u runs in a direction opposite to t then the interpretation would be that a pair with a local maximum and a local minimum would be created when t increases. However, as indicated above, we know from the scale-space theory for continuous signals that the latter phenomenon is impossible, since the number of local

extrema in a solution to the one-dimensional diffusion equation cannot increase when the scale parameter increases.

The diffusion equation apparently introduces a directional preference to its solutions (due to the causality requirements), which makes such creations impossible. How should this information be incorporated into the analysis of the singularities in scale-space? One way of avoiding the previous blindness of the equivalence concept to the structural property of the diffusion equation is by requiring the previous polynomial unfoldings in Thom's transversality lemma to satisfy the diffusion equation. Then we are ensured that artifacts cannot be introduced. In the one-dimensional case this can be accomplished by interpreting the parameter u as a rescaled scale parameter, i.e. by replacing u by $3t$. Then the unfolding of a one-dimensional signal would be¹,

$$\tilde{G}_1(x; t) = x^3 + 3xt \tag{6.43}$$

which satisfies the one-dimensional diffusion equation. Obviously, with this interpretation, creations of pairs of local maxima with increasing t are no longer possible. Moreover, since the family is still on the generic form it seems as if we could treat it as a general representative of the solutions to the diffusion equation and the only possible bifurcation events would be pairs of local maxima and minima merging with increasing scale.

On the other hand, if similar heuristic arguments are applied to the two-dimensional case, the corresponding unfolding would instead be

$$\tilde{G}_1(x, y; t) = x^3 + 3xt \pm (y^2 + t) \tag{6.44}$$

We have added a t term to the previous expression for G_1 in order to have \tilde{G}_1 satisfying the diffusion equation. Adding such a term does not affect the equivalence concept, since the t term can be treated as a constant with respect to the state variables (x, y) and, hence, be included in the γ term in the definition of equivalence of families of functions.

However, there is more complexity in the two-dimensional situation, since in this case, pairs of extremum and saddle points actually can be created with increasing scale, see e.g. the example in Figure 6.4. Obviously this kind of phenomenon is not captured by the unfolding in (6.44). Therefore, the directional constraint on the parameter u in terms of t implies that we can no longer treat the catastrophe of fold type as exhausting all possible types of behaviours at a singularity in a solution to the diffusion equation.

One way of addressing this problem could possibly be by trying to develop results similar to Thom's classification theorem, which instead of being expressed in terms of the ordinary standard basis of polynomials could be expressed in terms of polynomials satisfying the diffusion equation. A possible set of candidates for such a treatment in the two-dimensional case is listed in Appendix A.5.1.

Another approach is to use the previous classifications in Corollary 6.10 and Corollary 6.11 to state which configurations are possible in general one-parameter families of functions. Then, after this classification has been performed the special structure of the diffusion equation can be taken into account for judging which of the resulting possibilities apply to the scale-space representation when the directional constraint of the diffusion equation has been added. Such a treatment will be the subject of the next section.

¹Analyses of this type have been carried out by e.g. Koenderink and van Doorn [Koe84, Koe86].

6.2.4 Classification of Blob Events at Bifurcations in Scale-Space

A natural question that arises in connection with the scale-space primal sketch concerns which types of blob events are possible in bifurcation situations. Since scale-space blobs are defined in terms of paths of critical points, the behaviour of a scale-space blob at a singularity will be solely determined by the behaviour of those paths during a short scale interval around the bifurcation moment.

Compared to the previous treatment where we were analysing the behaviour of critical points only there is, however, an additional factor that must be taken into account when dealing with scale-space blobs, namely the fact that saddle points, delimiting the extent of grey-level blobs involved a bifurcation, can be associated with other grey-level blobs as well. This leads to natural coupling between scale-space blobs sharing the same saddle path (of delimiting saddle points) in a neighbourhood of a bifurcation. We define:

Definition 6.7 (Non-shared saddle path (2D))

Given a saddle path involved in a structurally stable bifurcation of a two-dimensional signal we say that the saddle path is non-shared before (after) the bifurcation if there exists some scale interval before (after) the bifurcation during which every saddle point of the saddle path is not contained in more than one grey-level blob. Otherwise, the saddle path is said to be shared.

More formally, a saddle path is called non-shared before (after) a bifurcation at t_{bifurc} if there exists some $\epsilon > 0$ such that for all scales in the interval $t \in]t_{bifurc} - \epsilon, t_{bifurc}[$ ($t \in]t_{bifurc}, t_{bifurc} + \epsilon[$) the saddle point of the saddle path at that scale does not belong to more than one grey-level blob, see also Figure 6.3. Another way to express this property is that a shared saddle point is the delimiting saddle point of two (or more) grey-level blobs of the same polarity, while a non-shared saddle point either is the delimiting saddle point of one or no grey-level blobs.

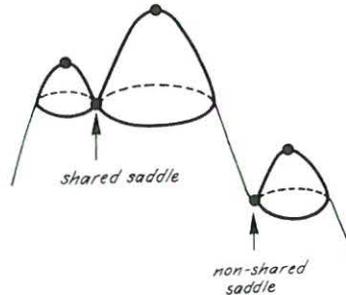


Figure 6.3: Illustration of the definition of grey-level blob for a two-dimensional signal. Every local extremum gives rise to a blob and the extent of the blob is given by a saddle point. A saddle point is said to be shared if it is contained in more than one grey-level blob, i.e. if it is a delimiting saddle point of two (or more) grey-level blobs of the same polarity.

This definition implies that a *non-shared* saddle path participating in, say, an extremum-saddle pair disappearing with increasing scale describes an isolated blob that disappears. We call this event a blob *annihilation*. On the other hand, a *shared* saddle path involved in a similar event describes a blob disappearing under the influence of a neighbour blob, a blob *merge*. Similarly, a *shared* saddle point taking part in an extremum-saddle pair that is created with increasing scale describes a blob *split*, while a *non-shared* saddle path participating in a similar event describes a blob *creation*. From the classification of the canonical

behaviour of the critical points of one-parameter families of functions in Corollary 6.10 we therefore have: (Below, the term annihilation (creation) of an extremum-saddle pair will mean that a pair consisting of an extremum path and a saddle path disappears (appears) when the scale parameter increases.)

Theorem 6.12 (Classification of scale-space blob events (2D))

In the scale-space representation of two-dimensional continuous signal, the following blob events are possible at a structurally stable bifurcation:

- *blob annihilation* — annihilation of an extremum-saddle pair where the saddle path is non-shared before the bifurcation.
- *blob merge* — annihilation of an extremum-saddle pair where the saddle path is shared with another scale-space blob before the bifurcation.
- *blob split* — creation of an extremum-saddle pair where the saddle path is shared with another scale-space blob after the bifurcation.
- *blob creation* — creation of an extremum-saddle pair where the saddle path is non-shared after the bifurcation.

These four cases constitute the definitions of the terms annihilation, merge, split and creation with respect to grey-level blobs and scale-space blobs in the two-dimensional case.

Proof: From Corollary 6.10 we have that the typical behaviour at singularities are pairwise annihilations and creations of extremum-saddle pairs. Combined with the definition of shared saddle path this means that the class of possible blob events is restricted to the given four types, provided that we deal with structurally stable bifurcations.

What remains to verify is that all these four types can be instantiated and that they also are structurally stable. It is well-known that blob annihilations and blob merges can take place in scale-space, see also Section 6.4 for illustrative examples. The fact that splits can occur is known as well, see e.g. the example given by Lifshitz and Pizer [Lif87] illustrated in Figure 6.4(a) and also Figure 3.1. The latter configuration can also be modified to describe a blob creation as well, if the higher one of the two peaks is replaced by a double peak, see Figure 6.4(b). Then the extent of the two smaller blobs at the higher peak will be delimited by the grey-level in the valley between them, which means that when the narrow ridge has eroded and given rise to the creation of a saddle-extremum pair the saddle path in the created saddle-extremum pair will not be shared by any other blob. \square

The assumption of structural stability is important in this context, since otherwise, there is an infinite variety of possible events. For instance, three or more blobs could merge into one blob at the same moment. Such events will however be unstable, since a slight perturbations of the input signal would perturb such a simultaneous merge of three blobs into a sequence of two successive pairwise merges. Note in this context that for Morse functions, see e.g. Arnold [Arn81], no pair of critical points will have the same values. In other words, for generic functions all critical points will be distinct. Although, by definition, the grey-level function will not be Morse at a bifurcation, we can, in general, assume this latter property to hold at bifurcations, which means that situations with three or more blobs simultaneously merging into one can be expected not to occur.

Algorithmically, this means that an encountered actual situation with, say, three blobs at a fine scale seeming to belong all to the same blob at a coarser scale, can in general be

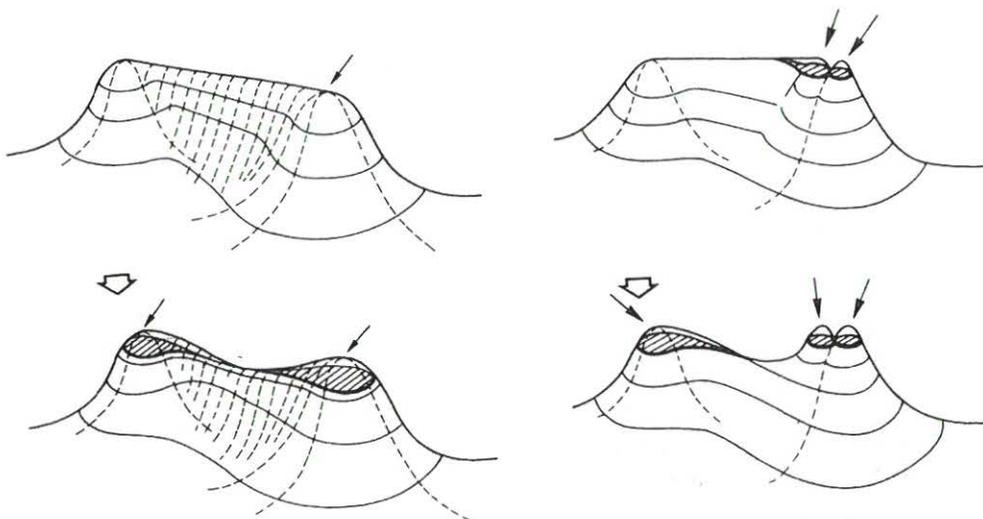


Figure 6.4: (a) Example illustrating the property that new local extrema can be created with increasing scale in the scale-space representation of a two-dimensional signal. Interpreted in terms of blobs the configuration describes a blob split. (b) By modifying the example slightly (by replacing the higher one of the two peaks with a double peak) one realizes that blob creations can occur as well. The base levels of the different grey-level blobs have been indicated.

decomposed into transitions of the four primitive types. This principle forms the idea behind the automatic scale refinement algorithm to be described in Chapter 7, which essentially refines the scale sampling until all relations between scale-space blobs in scale-space can be decomposed into events of the previously listed types.

For one-dimensional signals the analogies of Definition 6.7 and Proposition 6.12 will be as follows: We express the formulations for bright blobs only. The case with dark blobs is similar.

Definition 6.8 (Non-shared extremum path (bright blobs in 1D))

Consider the case with bright blobs in the scale-space representation of a one-dimensional signal. Given an extremum path of minimum points involved in a structurally stable bifur-

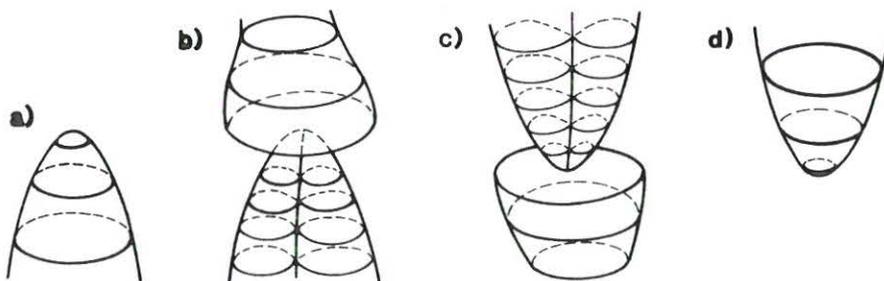


Figure 6.5: Illustration of the various events possible for the scale-space blobs of a two-dimensional signal: (a) blob annihilation (b) blob merge (c) blob split (d) blob creation.

cation we say that the extremum path is non-shared before (after) the bifurcation if there exists some scale interval before (after) the bifurcation during which every minimum point of the extremum path is contained only one bright grey-level blob. Otherwise the saddle path is said to be shared.

Theorem 6.13 (Scale-space blob events (bright blobs in 1D))

Typically, in the scale-space representation of one-dimensional continuous signal, the following blob events are possible at a structurally stable bifurcation:

- *blob annihilation* — annihilation of a minimum-maximum pair where the extremum path of minimum points is non-shared before the bifurcation.
- *blob merge* — annihilation of an extremum-saddle pair where the extremum path of minimum points is shared with another bright scale-space blob before the bifurcation.

Proof: From Corollary 6.11 we have that the typical behaviour at singularities in a one-parameter family of functions are pairwise annihilations and creations of minimum-maximum pairs. However, as discussed above the number of local extrema cannot increase with scale in the scale-space representation of a one-dimensional signal if follows that new minimum-maximum pairs cannot arise with increasing scale. This means that blob splits and blob creations are impossible in the one-dimensional case because of the causality requirements. □

These bifurcations between scale-space blobs define the hierarchical relations across scales between scale-space blobs at different scales in the tree-like representation scale-space primal sketch. The generated data structure will, however, not constitute a strict tree because of the blob annihilations and the blob splits.

6.3 Behaviour Near Singularities: Examples in 1D

In the previous section we classified the qualitative behaviour at bifurcation points. In this section we will illustrate the quantitative behaviour at singularities and give examples demonstrating how the blob descriptors vary with scale for a set of characteristic examples.

We will start by exploring the one-dimensional situation in more detail and show how one with very simple techniques can arrive at an expression similar to the generic representative of the fold unfolding (6.43) just by studying a third order Taylor expansion of the scale-space embedding. Then we will investigate the consequences of some of the other unfoldings in Thom’s classification theorem with application to grey-level blobs and scale-space blobs.

Since the main intention with this section is to mediate an intuitive feeling for what will happen at bifurcations in scale-space we will in most cases, for the sake of clarity, display the full calculations and sometimes also redo calculations carried out in previous sections. The technical details can be skipped by the hasty reader without loss of continuity.

6.3.1 Third Order Taylor Expansion of the Scale-Space Embedding

Given a scale-space embedding $L : R \times R_+ \rightarrow R$ of a one-dimensional signal $f : R \rightarrow R$ consider a third order Taylor expansion of the mapping $x \mapsto L(x; t_0)$ around a given point x_0 at some scale level t_0 :

$$f_{t_0}(x) = \alpha + \beta(x - x_0) + \gamma(x - x_0)^2 + \epsilon(x - x_0)^3 \tag{6.45}$$

where

$$\alpha = L(x_0, y_0; t_0); \quad \beta = \frac{\partial L}{\partial x}(x_0, y_0; t_0); \quad \gamma = \frac{1}{2} \frac{\partial^2 L}{\partial x^2}(x_0, y_0; t_0); \quad \epsilon = \frac{1}{6} \frac{\partial^3 L}{\partial x^3}(x_0, y_0; t_0) \quad (6.46)$$

Requiring this function to satisfy the diffusion equation

$$\frac{\partial L}{\partial t} = \frac{1}{2} \frac{\partial^2 L}{\partial x^2} \quad (6.47)$$

with initial condition $L(x; t_0) = f_{t_0}$ ($x \in Z$) we obtain

$$L(x; t) = \alpha + \beta(x - x_0) + \gamma(x - x_0)^2 + \epsilon(x - x_0)^3 + \delta_1(t - t_0) + \delta_2(x - x_0)(t - t_0) \quad (6.48)$$

where $\delta_1 = \gamma$ and $\delta_2 = 3\epsilon$. For simplicity, introduce new (offset) variables by

$$u = x - x_0; \quad v = t - t_0 \quad (6.49)$$

Then,

$$\tilde{L}(u; v) = L(u + x_0; v + t_0) = \alpha + \beta u + \gamma(u^2 + v) + \epsilon(u^3 + 3uv) \quad (6.50)$$

The critical points of the function $u \mapsto \tilde{L}(u; t)$ are given by

$$\frac{\partial \tilde{L}}{\partial u}(u; v) = \beta + 2\gamma u + 3\epsilon(u^2 + v) = 0 \quad (6.51)$$

If $\epsilon = 0$ we get one single root $x = -\frac{\beta}{2\gamma}$, whose location is independent of t . Obviously, this case is not interesting, since it implies a totally stationary solution. Therefore, from now on, we will only consider the solutions when $\epsilon \neq 0$. Then we get two trajectories of critical points

$$u_1(v) = -\frac{\gamma}{3\epsilon} + \sqrt{\frac{\gamma^2}{9\epsilon^2} - (v + \frac{\beta}{3\epsilon})} \quad (6.52)$$

$$u_2(v) = -\frac{\gamma}{3\epsilon} - \sqrt{\frac{\gamma^2}{9\epsilon^2} - (v + \frac{\beta}{3\epsilon})} \quad (6.53)$$

These paths only exist when the argument of the root function $\frac{\gamma^2}{9\epsilon^2} - (v + \frac{\beta}{3\epsilon})$ is non-negative, i.e. if and only if $v \leq \frac{\gamma^2}{9\epsilon^2} - \frac{\beta}{3\epsilon}$. The critical paths meet and a bifurcation takes place at

$$(u_{bifurc}; v_{bifurc}) = \left(-\frac{\gamma}{3\epsilon}; \frac{\gamma^2}{9\epsilon^2} - \frac{\beta}{3\epsilon} \right) \quad (6.54)$$

From the second derivative $\frac{\partial^2 \tilde{L}}{\partial u^2}(u; v) = 2\gamma + 6\epsilon u = 0$ we obtain:

$$\frac{\partial^2 \tilde{L}}{\partial u^2}(u_1(v); v) = +6\epsilon \sqrt{\frac{\gamma^2}{9\epsilon^2} - (v + \frac{\beta}{3\epsilon})} \quad (6.55)$$

$$\frac{\partial^2 \tilde{L}}{\partial u^2}(u_2(v); v) = -6\epsilon \sqrt{\frac{\gamma^2}{9\epsilon^2} - (v + \frac{\beta}{3\epsilon})} \quad (6.56)$$

i.e. the second derivative has different sign in the two critical points (provided that $\epsilon \neq 0$). Thus, the bifurcation consists of one maximum point and minimum point that meet and

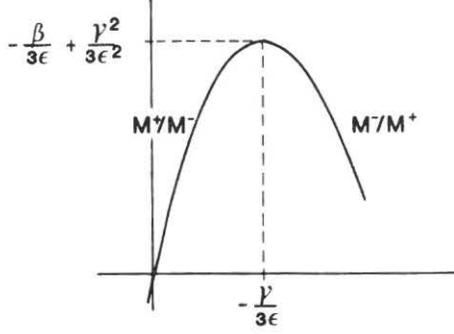


Figure 6.6: Third order Taylor expansion of the scale-space embedding: Schematic view over the loci of the critical points as scale changes. The bifurcation consists of a maximum point and a minimum point that meet and annihilate.

annihilate, see also Figure 6.6. At the bifurcation $\frac{\partial^2 \bar{L}}{\partial u^2}$ is of course zero. Note that, as expressed in Corollary 6.4, the drift velocity actually tends to infinity as the critical points approach the singularity. At $v = 0$ the drift velocity is

$$\frac{\partial u_1}{\partial v}(0) = -\frac{3\epsilon}{\gamma} = -\frac{1}{2} \frac{\frac{\partial^3 \bar{L}}{\partial x^3}(0; 0)}{\frac{\partial^2 \bar{L}}{\partial u^2}(0; 0)} = -\frac{1}{2} \frac{\frac{\partial^3 L}{\partial x^3}(x_0; t_0)}{\frac{\partial^2 L}{\partial x^2}(x_0; t_0)} \quad (6.57)$$

which agrees with the result (6.20) in Proposition 6.5.

Now, assume that for the scale level t_0 the point $x = x_0$ is a critical point for the mapping $x \mapsto L(x; t_0)$, i.e. that for $v = 0$ the point $u = 0$ is a critical point for the mapping $u \mapsto \bar{L}(u; 0)$. Then $\beta = 0$ and we can estimate the time Δv_{bifurc} as well as the distance Δu_{bifurc} until bifurcation by

$$\Delta u_{bifurc} = |u_1(v_{bifurc}) - u_1(0)| = \left| \frac{\gamma}{3\epsilon} \right| = \left| \frac{\frac{\partial^2 \bar{L}}{\partial u^2}(0; 0)}{\frac{\partial^3 \bar{L}}{\partial u^3}(0; 0)} \right| = \left| \frac{\frac{\partial^2 L}{\partial x^2}(x_0; t_0)}{\frac{\partial^3 L}{\partial x^3}(x_0; t_0)} \right| \quad (6.58)$$

$$\Delta v_{bifurc} = \left(\frac{\gamma}{3\epsilon} \right)^2 = \left(\frac{\frac{\partial^2 \bar{L}}{\partial u^2}(0; 0)}{\frac{\partial^3 \bar{L}}{\partial u^3}(0; 0)} \right)^2 = \left(\frac{\frac{\partial^2 L}{\partial x^2}(x_0; t_0)}{\frac{\partial^3 L}{\partial x^3}(x_0; t_0)} \right)^2 \quad (6.59)$$

where we have also inserted the actual expressions for γ and ϵ . To summarize,

Observation 6.14 (Coarse estimate of the scale-step when linking grey-level blobs into scale-space blobs (Distance to a bifurcation) (1D))

A coarse estimate of the scale-step when linking grey-level blobs to into scale-space blobs is given by (6.58) and (6.59).

So far we have not made any numerical experiments testing the feasibility of using this estimate as the basis for the step size selection in the actual blob linking. Note, however, that despite the pessimistic upper bound on the drift velocity induced by Proposition 6.5, the local extremum will hardly escape far outside the support region of its associated grey-level blob. This property has proved to be very useful in the blob linking algorithm to be described in Section 7.2

Observation 6.15 (Coarse bound on the drift of local extrema (1D))

Although the drift velocity of a local extremum point may momentarily be very large (tend to infinity near a bifurcation), when scale changes, the grey-level blob support region defines a natural spatial region to search for blobs in at the next level of scale.

To simplify our further considerations we introduce new coordinates again by

$$\xi = u + \frac{\gamma}{3\epsilon}; \quad \tau = v + \frac{\beta}{3\gamma} - \frac{\gamma^2}{9\epsilon^2} \quad (6.60)$$

Then, the expressions for the scale-space representation reduces to

$$\tilde{\tilde{L}}(\xi; \tau) = \tilde{L}\left(\xi - \frac{\gamma}{3\epsilon}; \tau - \frac{\beta}{3\gamma} - \frac{\gamma^2}{9\epsilon^2}\right) = \dots = \epsilon\xi^3 + 3\epsilon\xi\tau + \left(\alpha - \frac{\gamma\beta}{3\epsilon} + \frac{2\gamma^3}{27\epsilon^2}\right) \quad (6.61)$$

Finally, we let

$$\lambda(\xi; \tau) = \tilde{\tilde{L}}(\xi; \tau) - \left(\alpha - \frac{\gamma\beta}{3\epsilon} + \frac{2\gamma^3}{27\epsilon^2}\right) = \epsilon(\xi^3 + 3\xi\tau) \quad (6.62)$$

These coordinate shifts from $(x; t)$ to $(u; v)$ and at last $(\xi; \eta)$ only mean that we have *translated* the coordinate axes such that the bifurcation occurs for $(\xi; \tau) = (0; 0)$ and *subtracted* a constant to achieve $\lambda(0; 0) = 0$. Therefore, no derivatives are affected, which in turn means that $\lambda : R \times R_+ \rightarrow R$ satisfies

$$\frac{\partial \lambda}{\partial \tau} = \frac{1}{2} \frac{\partial^2 \lambda}{\partial \xi^2} \quad (6.63)$$

and can be regarded as the general third order approximation to the solution of the original diffusion equation. Moreover, the calculations show that for one-dimensional functions any third order polynomial satisfying the diffusion equation can be reduced to the canonical form (6.43) just by a simple translation and rescaling of the coordinate axes. This property cannot, however, be expected to hold in higher dimensions.

In the next section, we will develop how one from this analytically simple expression can derive closed form results for the evolution properties of grey-level blobs and scale-space blobs over scale.

6.3.2 Evolution Properties of Local Extrema, Grey-Level Blobs and Scale-Space Blobs in 1D Continuous Scale-Space

Consider again the generic unfolding of the scale-space embedding in the neighbourhood of a bifurcation.

$$\lambda(x; t) = \tilde{G}_1(x; t) = x^3 + 3xt \quad (6.64)$$

where x and t can be interpreted local coordinates in a coordinate system centered at the bifurcation point. As mentioned above, the critical points of this function are given by

$$\frac{\partial \lambda}{\partial x}(x; t) = 3(x^2 + t) = 0 \quad (6.65)$$

that is by

$$\xi_1(t) = -\sqrt{-t}; \quad \xi_2(t) = +\sqrt{-t} \quad (6.66)$$

Moreover, from this analytically simple expression one can easily analyze what happens to

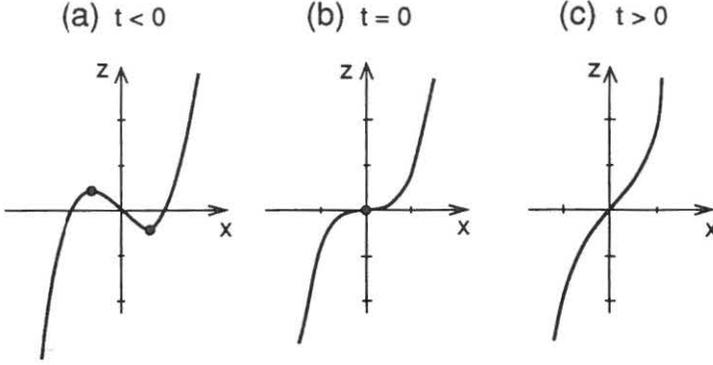


Figure 6.7: Fold unfolding in one dimension: Schematic view of the smoothed signal (a) before the bifurcation (b) at the bifurcation (c) after the bifurcation.

the grey-level blobs of a one-dimensional continuous signal as scale changes. The critical values, λ_1 and λ_2 , in the two extremum points are given by

$$\lambda_1(t) = \lambda(\xi_1(t); t) = +2(-t)^{\frac{3}{2}}; \quad \lambda_2(t) = \lambda(\xi_2(t); t) = -2(-t)^{\frac{3}{2}} \quad (6.67)$$

Hence, the *contrasts*, C_1 and C_2 , of the two blobs have equal magnitude given by

$$C_1(t) = C_2(t) = |\lambda_2(t) - \lambda_1(t)| = 4(-t)^{\frac{3}{2}} \quad (6.68)$$

The level crossings, ρ_1 and ρ_2 , for the grey-level blob associated with the extremum points ξ_1 and ξ_2 respectively are given by the roots of single multiplicity to the equations

$$\lambda(\rho_1, t) = \lambda_2(t); \quad \lambda(\rho_2, t) = \lambda_1(t) \quad (6.69)$$

One easily shows that these equations have the set of roots $\{-\sqrt{-t}, -\sqrt{-t}, 2\sqrt{-t}\}$ and $\{-2\sqrt{-t}, \sqrt{-t}, \sqrt{-t}\}$ respectively, which leads to

$$\rho_1(t) = +2\sqrt{-t}; \quad \rho_2(t) = -2\sqrt{-t} \quad (6.70)$$

The support regions of the two blobs are the intervals

$$G_{\text{support}}(r_1(t)) = \{x : \rho_2(t) \leq x \leq \xi_2(t)\} \quad (6.71)$$

$$G_{\text{support}}(r_2(t)) = \{x : \xi_1(t) \leq x \leq \rho_1(t)\} \quad (6.72)$$

Hence, the magnitudes of the blob support regions are

$$A_1(t) = \|G_{\text{support}}(r_1(t))\| = |\xi_2(t) - \rho_2(t)| = 3\sqrt{-t} \quad (6.73)$$

$$A_2(t) = \|G_{\text{support}}(r_2(t))\| = |\rho_1(t) - \xi_1(t)| = 3\sqrt{-t} \quad (6.74)$$

The grey-level blobs are the two sets

$$G_{\text{blob}}(r_1(t)) = \{(x, \zeta) : (\rho_2(t) \leq x \leq \xi_2(t)) \wedge (\lambda_2(t) \leq \zeta \leq \lambda(x; t))\} \quad (6.75)$$

$$G_{\text{blob}}(r_2(t)) = \{(x, \zeta) : (\xi_1(t) \leq x \leq \rho_1(t)) \wedge (\lambda(x; t) \leq \zeta \leq \lambda_1(t))\} \quad (6.76)$$

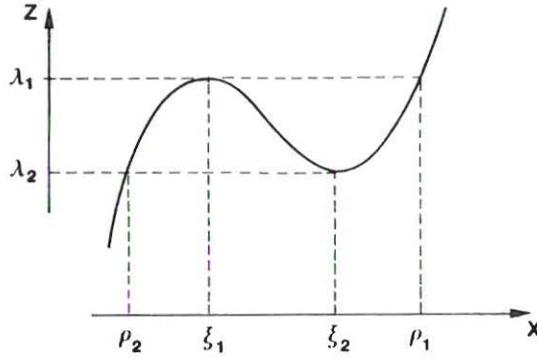


Figure 6.8: The situation before the bifurcation occurs. Illustration of the definitions of ξ_1 , ξ_2 , λ_1 , λ_2 , ρ_1 and ρ_2 .

Finally, the grey-level blob volumes, V_1 and V_2 , of the two blobs are given by

$$V_1(t) = \|G_{blob}(r_1(t))\| = \int_{x=\rho_2(t)}^{\xi_2(t)} |\lambda(x; t) - \lambda_2(t)| dx = \dots = \frac{27(-t)^2}{4} \quad (6.77)$$

$$V_2(t) = \|G_{blob}(r_2(t))\| = \int_{x=\xi_1(t)}^{\rho_1(t)} |\lambda_1(t) - \lambda(x; t)| dx = \dots = \frac{27(-t)^2}{4} \quad (6.78)$$

There is a natural dimensionless quantity associated with these blob measures:

$$\frac{V_1}{A_1 C_1} = \frac{V_2}{A_2 C_2} = \frac{9}{16} \quad (6.79)$$

Assuming that the scale-space blob has some minimum scale t_{min} (of course less than zero) we can compute its scale-space blob volume by

$$S_1 = \int_{t_{min}}^0 V_1(t) dt = \frac{9(-t_{min})^3}{4} \quad (6.80)$$

From the previous discussion it follows that we can treat this situation as a general representative of the generic behaviour around a structurally stable singularity in a solution to the one-dimensional diffusion equation.

Proposition 6.16 (Generic behaviour at singularities in scale-space (1D))

The generic behaviour at a singularity in the scale-space representation of a continuous signal can be represented by the fold unfolding $\tilde{G}_1(x; t) = x^3 + 3xt$. This singularity describes a maximum point and a minimum point that meet with increasing scale along the two branches of a parabola and then disappear. Interpreted in terms of blobs the situation describes a blob annihilation (or, possibly, a blob merge if the delimiting saddle point is shared before bifurcation). Above we have illustrated how the blob descriptors contrast, support region and blob volume evolve with scale near the bifurcation in this case.

6.4 Behaviour Near Singularities: Examples in Two Dimensions

The analysis in the previous section can in a sense be said to be complete, since the restricted unfolding $\tilde{G}_1(x; t) = x^3 + 3xt$ exhausts the possible events between critical points in the

scale-space representation of a one-dimensional signal. In this section we will examine the more difficult two-dimensional case and investigate the consequences of some of the other unfoldings in Thom's classification theorem with application to the diffusion equation.

6.4.1 Fold

As mentioned several times above the general unfolding in the fold case is on the form

$$G_1(x; u) = x^3 + ux \quad (6.81)$$

Earlier we have said that in order for this function to satisfy the diffusion equation it is necessary that $u = 3t$. Another way of reaching to the same conclusion is by replacing every monomial in G_1 by a corresponding polynomial satisfying the diffusion equation. Hence, we replace the x^3 term by $x^3 + 3xt$ (and the x term by x), see also Appendix A.5.1. Moreover, since we are here interested in the two-dimensional case we have to add a $\pm y^2$ term, which then because of the diffusion equation leads to a $\pm t$ term. We get the unfolding

$$L(x, y; t) = x^3 + (u + 3t)x \pm (y^2 + t) \quad (6.82)$$

which is still of the same type as G_1 . Here, one can observe that the u parameter will not affect the qualitative behaviour² of the singularity set, since a change of variable $3t' = u + 3t$ would move the u -dependence to the constant term. Hence, without loss of generality, we may set u to zero and study the polynomial:

$$L(x, y; t) = x^3 + 3tx \pm (y^2 + t) \quad (6.83)$$

which, as earlier indicated, is the same one as we would have got just by setting the original parameter u to $3t$. For simplicity, first assume that the sign of the $\pm(y^2 + t)$ is positive. Then the scale-space family to be studied is

$$L(x, y; t) = x^3 + 3xt + y^2 + t \quad (6.84)$$

where x , y and t should again be interpreted as offset coordinates. The critical points of this mapping are given by

$$\begin{cases} \frac{\partial L}{\partial x} = 3(x^2 + t) = 0 \\ \frac{\partial L}{\partial y} = 2y = 0 \end{cases} \quad (6.85)$$

If $t < 0$ we obtain two solutions:

$$r_1(t) = (x_1(t), y_1(t)) = (-\sqrt{-t}, 0); \quad r_2(t) = (x_2(t), y_2(t)) = (+\sqrt{-t}, 0) \quad (6.86)$$

At $t = 0$ they degenerate into a double root (the bifurcation moment), and for $t > 0$ they cease to exist. Hence, the trajectories of the critical points will be similar to the one-dimensional case described in Section 6.3.2. The Hessian of L is

$$(\mathcal{H}L) = \frac{\partial^2 L}{\partial x^2} \frac{\partial^2 L}{\partial y^2} - \left(\frac{\partial^2 L}{\partial x \partial y} \right)^2 = 12x \quad (6.87)$$

Hence, $(\mathcal{H}L)(r_1(t)) = -12\sqrt{-t} < 0$ and $(\mathcal{H}L)(r_2(t)) = +12\sqrt{-t} > 0$. Further, $\frac{\partial^2 L}{\partial x^2}(r_2(t)) = 6\sqrt{-t} > 0$. Therefore,

- r_1 describes the trajectory of a saddle point and
- r_2 describes the trajectory of a minimum point.

²The interpretation of the v parameter is that it translates the singularity along the t -axis.

Hence, for every $t < 0$ the point $r_2(t)$ gives rise to a dark grey-level blob. The values in the critical points are

$$L_1(t) = L(r_1(t); t) = t - 2t\sqrt{-t}; \quad L_2(t) = L(r_2(t); t) = t + 2t\sqrt{-t} \quad (6.88)$$

The dark grey-level blob associated with the minimum point $r_2(t)$ is delimited by the base-level $L_1(t)$. At a fixed value of t we get the intersection curve between the base-level and the grey-level surface by solving the equation:

$$L(x, y; t) = L_1(t) \quad (6.89)$$

which can be reduced to

$$x^3 + 3tx + y^2 - 2(-t)^{\frac{3}{2}} = 0 \quad (6.90)$$

Hence, the curve is symmetric with respect to the y -axis. Solving for y as a function of x and t we obtain two solutions:

$$y^-(x; t) = -\sqrt{2(-t)^{\frac{3}{2}} - x^3 - 3tx}; \quad y^+(x; t) = +\sqrt{2(-t)^{\frac{3}{2}} - x^3 - 3tx} \quad (6.91)$$

Setting $y = 0$ and solving for x we obtain one single root at $x = 2\sqrt{-t}$ and one double root at $x = -\sqrt{-t}$. See also Figure 6.9. Equation (6.90) gives the equation for the boundary of

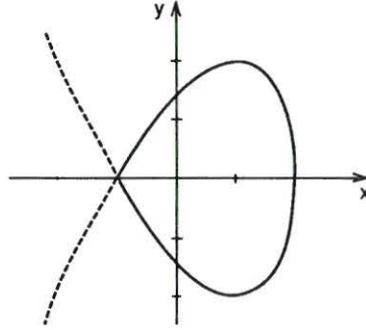


Figure 6.9: The blob support region of the grey-level blob at a specific level of scale. Outside the blob the level curve corresponding to the clipping level of the grey-level blob has been indicated with a dashed line. This figure can be regarded as describing the general appearance of the support region of a grey-level blob delimited by a non-shared saddle point.

the support region of the grey-level blob provided that $x \in [-\sqrt{-t}, 2\sqrt{-t}]$ and $t < 0$. Now we can easily compute closed-form expressions for the blob descriptors.

$$C_{blob}(r_2(t)) = L_1(t) - L_2(t) = 4(-t)^{\frac{3}{2}} \quad (6.92)$$

$$\begin{aligned} \|G_{support}(r_2(t))\| &= \iint_{A_{blob}(r_2(t))} dx dy = \int_{x=-\sqrt{-t}}^{2\sqrt{-t}} \int_{y=y^-(x;t)}^{y=y^+(x;t)} dy dx = \dots = \\ &= 2 \int_{x=-\sqrt{-t}}^{2\sqrt{-t}} (x + \sqrt{-t}) \sqrt{2\sqrt{-t} - x} dx = \dots = \frac{24\sqrt{3}(-t)^{\frac{5}{4}}}{5} \end{aligned} \quad (6.93)$$

$$\|G_{blob}(r_2(t))\| = \iiint_{G_{blob}(r_2(t))} dx dy dz = \iint_{G_{support}(r_2(t))} (L_1(t) - L(x, y; t)) dx dy = (6.94)$$

$$\int_{x=-\sqrt{-t}}^{2\sqrt{-t}} \int_{y=y^-(x;t)}^{y^+(x;t)} (2(-t)^{\frac{3}{2}} - 3tx - x^3 - y^2) dy dx =$$

$$= \int_{x=-\sqrt{-t}}^{2\sqrt{-t}} \frac{4}{3} (\sqrt{-t} + x)^3 (2\sqrt{-t} - x)^{\frac{3}{2}} dx = \dots = \frac{3456\sqrt{3}u^{\frac{11}{4}}}{385}$$

These quantities give rise to a natural dimensionless ratio:

$$\frac{\|G_{blob}(r_2(t))\|}{C_{blob}(r_2(t))\|G_{support}(r_2(t))\|} = \frac{36}{77} \quad (6.95)$$

If the sign of the $(y^2 + t)$ term in (6.83) instead would have been selected negative, the trajectories of critical points would have been similar. The only difference would have been that the minimum point would have been replaced by a saddle point and the saddle point by a maximum point. Hence, the situation would have described the annihilation of a bright blob instead of a dark one.

Observation 6.17 (Evolution properties, Fold case (2D))

The unfolding in the two-dimensional fold case $L(x, y; t) = x^3 + 3xt \pm (y^2 + t)$ describes a minimum (maximum) point and a saddle point that merge along the two branches of a parabola and disappear at $t = 0$. In other words, it describes the annihilation of a dark (bright) grey-level blob.

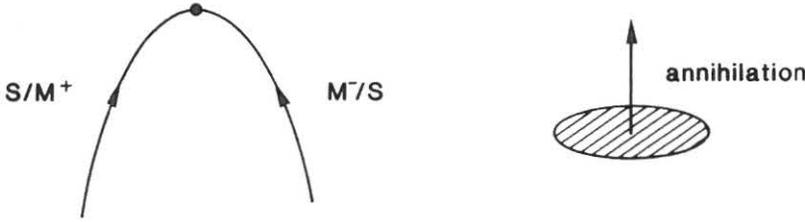


Figure 6.10: The fold unfolding in two dimensions $L(x, y; t) = x^3 + 3xt \pm (y^2 + t)$ describes (a) a saddle point and a minimum (maximum) point that merge or equivalently (b) the annihilation of a dark (bright) grey-level blob.

Finally, if we assume that the scale-space blob is from below by a minimum scale t_{min} we can compute its scale-space blob volume. Then,

$$|S_{blob}(r_2)| = \int_{t=t_{min}}^0 |G_{blob}(r_2(t))| dt = \dots = \frac{4608\sqrt{3}(-t_{min})^{\frac{15}{4}}}{1925} \quad (6.96)$$

6.4.1.1 Comparisons with Zero-Crossings of the Laplacian

In this context it is interesting to compare the results with the locations of the zero-crossings of the Laplacian. Since the sign of the $\pm(y^2 + t)$ term affects the qualitative behaviour, we introduce a parameter α such that

$$L_\alpha(x, y; t) = x^3 + 3xt + \alpha(y^2 + t) \quad (6.97)$$

Then, the zero-crossings are given by

$$\frac{\partial^2 L_\alpha}{\partial x^2} + \frac{\partial^2 L_\alpha}{\partial y^2} = 6x + 2\alpha = 0 \quad (6.98)$$

Hence, under variations in scale, the zero-crossings of the Laplacian will always be on a vertical straight line at $x = -\frac{\alpha}{3}$. Moreover, the zero-crossings will always be on the same side of the local extremum, see also Figure 6.11. This property will, however, not hold in the cusp unfolding to be considered in our next example.

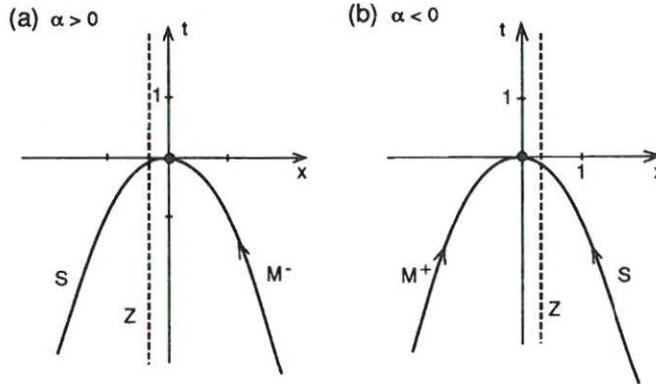


Figure 6.11: *Fold case unfolding: Locations of the zero-crossings over scale (marked with Z) compared to the trajectories of the local extrema (marked with M^- and M^+) and the saddle points (marked with S). (a) Positive α corresponding to $+(y^2+t)$. (b) Negative α corresponding to $-(y^2+t)$.*

6.4.2 Cusp

The general unfolding in the cusp case is given by

$$G_2(x; u, v) = x^4 + ux^2 + vx \quad (6.99)$$

In order to have this function satisfying the diffusion equation we replace x^4 by $x^4 + 6tx^2 + 3t^3$ and x^2 by $x^2 + t$. Moreover, we add a $\pm y^2$ term which also leads to a $\pm t$ term to get:

$$L(x, y; t) = x^4 + (6t + u)x^2 + vx + ut + 3t^2 + \pm(y^2 + t) \quad (6.100)$$

Notable is that also this unfolding is of the same type as the previous one. The ut , $3t^2$ and t terms can all be treated as constants with respect to x and y . Further, the u parameter will not affect the singularity set since a translation of the t axis would eliminate the u -dependence from the terms depending on x and y . Hence, we may set u to zero without loss of generality. Thus,

$$L(x, y; t) = x^4 + 6tx^2 + vx + 3t^2 \pm (y^2 + t) \quad (6.101)$$

First, we assume that the sign of the y^2 term is positive. Then, the polynomial to be studied is:

$$L(x, y; t) = x^4 + 6x^2t + vx + 3t^2 + y^2 + t \quad (6.102)$$

where x , y and t should again be interpreted as offset coordinates, while v is a free parameter. The critical points of this mapping are given by

$$\begin{cases} \frac{\partial L}{\partial x} = 4x^3 + 12tx + v = 0 \\ \frac{\partial L}{\partial y} = 2y = 0 \end{cases} \quad (6.103)$$

and their type by the sign of the Hessian

$$(\mathcal{H}L)(x, y; t) = 24(x^2 + t) \quad (6.104)$$

The existence of roots to this system of equations can obviously be reduced to the existence of roots to $h(x) = 4x^3 + 12tx + v = 0$. After some calculations, see Appendix A.5.2, one can easily show the following:

- If $t > -(\frac{v}{8})^{\frac{2}{3}}$ then $h(x) = 0$ has only one real root and there exists a unique stationary point. For $t > 0$ this point obviously has a strictly positive Hessian and is accordingly a local minimum.
- If $t < -(\frac{v}{8})^{\frac{2}{3}}$ then $h(x) = 0$ has three distinctly different roots. These roots satisfy $x_1(t) < -\sqrt{-t} < x_2(t) < +\sqrt{-t} < x_3(t)$. Hence by (6.104) we have that $x_1(t)$ and $x_3(t)$ are local minima and that $x_2(t)$ is a saddle point.
- If $t = -(\frac{v}{8})^{\frac{2}{3}}$ then $h(x)$ has either one root of multiplicity three or one root of multiplicity two and another root of multiplicity one. The root of multiplicity greater than one is at $x = (\frac{v}{8})^{\frac{1}{3}}$, and at this point a bifurcation occurs. The behaviour around this point depends on the value of v , see also Figure 6.12:
 - If $v > 0$ then x_2 and x_3 meet and disappear while x_1 remains unaffected.
 - If $v < 0$ then x_1 and x_2 meet and disappear while x_3 remains unaffected.
 - If $v = 0$ then all three roots meet in $x = 0$. This is obviously a degenerate case.

Hence, this situation describes a minimum point and a saddle point that meet and annihilate under the influence of another maximum. In other words, it describes two dark grey-level blobs merging into one. To summarize,

Observation 6.18 (Evolution properties, Cusp case (dark blobs, 2D))

The unfolding in the two-dimensional cusp case $L(x, y; t) = x^4 + 6x^2t + vx + 3t^2 + (y^2 + t)$ describes a minimum point and a saddle point that merge under the influence of another minimum point. In other words, it describes two dark grey-level blobs merging into one.

Interpreted in terms of blobs all the bifurcation situations above correspond to two grey-level blobs merging into one. In this sense the bifurcation relations between grey-level blobs at different levels of scale will be more stable to small perturbations than bifurcation relations between critical points only, since the blob bifurcation remains unaffected by a change in the v parameter while expressed in terms of critical points only the topology of the situation is completely changed with the sign of v .

If the sign of the $\pm(y^2 + t)$ term in (6.101) instead would have been selected negative, then $x_1(t)$ and $x_3(t)$ would have been saddle points and $x_2(t)$ a local maximum. In that case the unfolding would have described a maximum point and a saddle point merging under the influence of another saddle point.

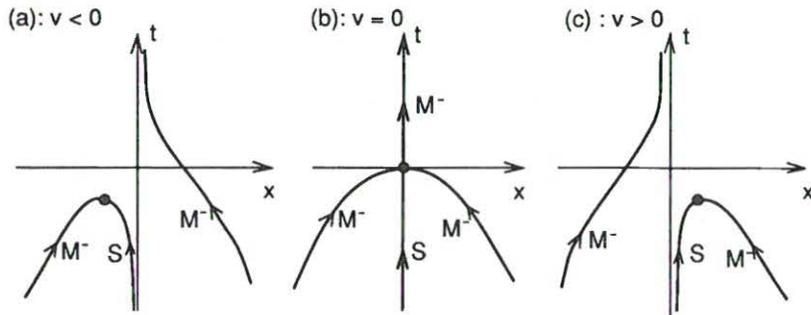


Figure 6.12: The cusp unfolding in two dimensions $L(x, y; t) = x^4 + 6x^2t + vx + 3t^2 + (y^2 + t)$ describes a minimum point and a saddle point that merge under the influence of another minimum point provided that $\alpha > 0$. (a)-(c) Depending on the value of v different events may occur. Equivalently the situation describes the two dark grey-level blobs merging into one.

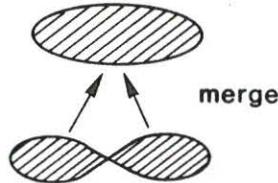


Figure 6.13: Independent of the value of v all the three situations above describe two dark grey-level blobs merging into one. In this sense the bifurcation relations between grey-level blobs are more stable than the corresponding relations between critical points, for which the topology of the situation is in fact changed by a variation in the sign of v .

Observation 6.19 (Evolution properties, Cusp case (bright blobs, 2D))

The unfolding in the two-dimensional cusp case $L(x, y; t) = x^4 + 6x^2t + vx + 3t^2 - (y^2 + t)$ describes a maximum point and a saddle point that merge under the influence of another saddle point. In other words, it describes the annihilation of a bright grey-level blob.

If instead the sign of the entire unfolding would have been changed, then, depending on the sign of the $\pm(y^2 + t)$ term, the situation would have described either a maximum point and a saddle point merging under the influence of another maximum point or a local minimum and a saddle merging under the influence of another saddle. Interpreted in terms of blobs this corresponds to either two bright blobs merging into one or the annihilation of a dark blob.

6.4.2.1 Comparisons with Zero-Crossings of the Laplacian

By introducing a parameter α such that

$$L_\alpha(x, y; t) = x^4 + 6x^2t + vx + 3t^2\alpha(y^2 + t) \tag{6.105}$$

the zero-crossings of the Laplacian will in this case be given by

$$\frac{\partial^2 L_\alpha}{\partial x^2} + \frac{\partial^2 L_\alpha}{\partial y^2} = 12x^2 + 12t + 2\alpha = 0 \tag{6.106}$$

which leads to two solutions $x = -\sqrt{-t - \frac{\alpha}{6}}$ and $x = +\sqrt{-t - \frac{\alpha}{6}}$ if $t \leq -\frac{\alpha}{6}$. One can observe that these curves do not give a correct subdivision around the local extrema for all t , see also Figure 6.14.

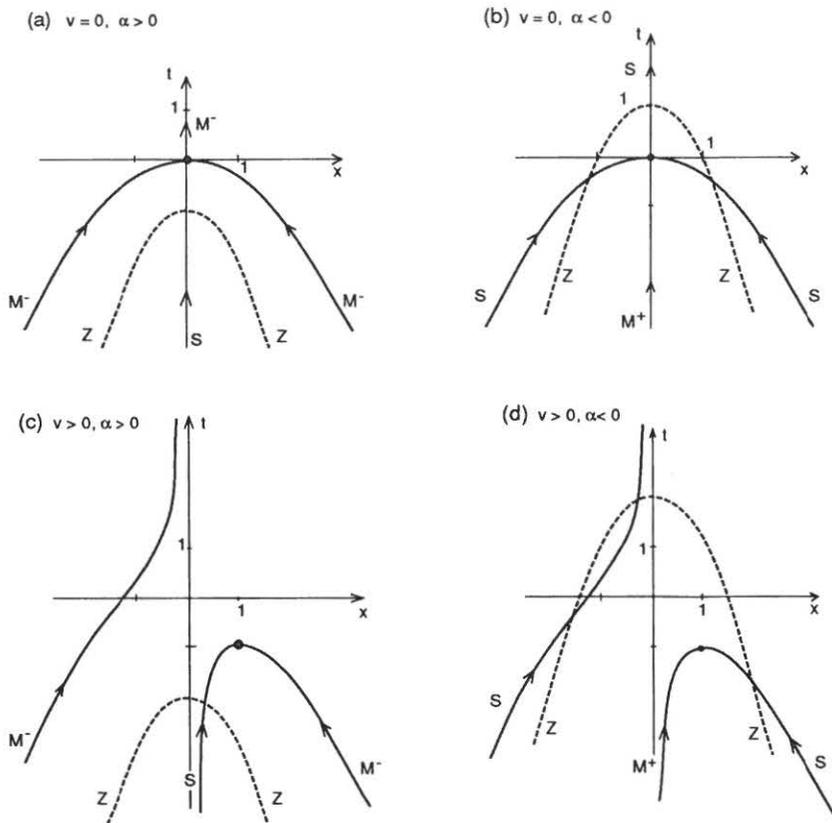


Figure 6.14: Cusp unfolding in the two-dimensional case: Locations of the zero-crossings over scale compared to the trajectories of the local extrema. (a-b) The degenerate case when $v = 0$. (a) $\alpha > 0$. Note that during a certain scale interval the zero-crossings of the Laplacian fail to enclose isolated local extrema — a property pointed out also by Koenderink and van Doorn (1984). (b) $\alpha < 0$. Note that during a certain scale interval there is no local extremum between the two zero-crossing curves. (c-d) Similar examples for the non-degenerate case when $v > 0$. (c) $\alpha > 0$. (d) $\alpha < 0$.

This example shows that, strictly speaking, in two (and higher) dimensions there is no absolute relation between the locations of the Laplacian zero-crossing curves and the local extrema of a signal. We have seen that it may happen that a zero-crossing curve encloses either no extremum, one extremum or more than one local extremum. In the one-dimensional case, though, the zero-crossings of the Laplacian will always divide local extrema correctly. In other words, between any two consecutive Laplacian zero-crossing points of a one-dimensional signal there will always be exactly one local extremum point.

One could say that by summing up the second order derivatives as done in the Laplacian operator we mix the behaviours from the different coordinate directions into a single scalar quantity that cannot fully describe the geometry of the two-dimensional grey-level landscape.

6.4.2.2 Cusp Unfolding in One Dimension

If we restrict this treatment to the one-dimensional case the cusp unfolding will be

$$L(x, y; t) = x^4 + 6x^2t + vx + 3t^2 \quad (6.107)$$

and the singularity set be given by

$$\frac{\partial L}{\partial x} = 4x^3 + 12tx + v = 0 \quad (6.108)$$

From similar calculations as in the previous two-dimensional case it follows that this unfolding describes a maximum point and a minimum point merging under the influence of another minimum point.

Observation 6.20 (Evolution properties, Cusp case (1D))

The unfolding in the one-dimensional cusp case $L(x, y; t) = x^4 + 6x^2t + vx + 3t^2$ describes a minimum point and maximum merging under the influence of another minimum point. In other words, it describes two dark grey-level blobs merging into one.

By changing the sign of the entire unfolding we will instead describe a maximum point and a minimum point merging under the influence of another maximum point, or equivalently two bright blobs merging into one.

6.4.2.3 Drift Velocity Analysis

In order to analyse the drift velocity of the local extremum point *not* involved in the bifurcation we differentiate (6.108) with respect to t :

$$\frac{\partial x}{\partial t} = -\frac{x}{x^2 + t} \quad (6.109)$$

To find the scale where the drift velocity assumes its maximum value we differentiate again and set the derivative to zero:

$$\frac{\partial^2 x}{\partial t^2} = \frac{2xt}{(x^2 + t)^3} = 0 \quad (6.110)$$

Here, we are not interested in the case $x = 0$ since the behaviour at the bifurcation has already been analysed. Thus, as expected, the maximum drift velocity occurs for $t = 0$. At this scale we have $x = (-\frac{v}{4})^{1/3}$. Hence,

$$\left| \frac{\partial x}{\partial t} \right|_{max} = -\frac{1}{x} = \left(\frac{4}{v} \right)^{\frac{1}{3}} \quad (6.111)$$

which shows that the maximum drift velocity of the extremum point increases towards infinity as v decreases towards zero and the configuration tends to the non-generic case. This exemplifies a further consequence of Proposition 6.3, namely that *even for critical points not directly involved in bifurcations there is no absolute upper bound on their drift velocity*, a conclusion valid both in one and two dimensions.

This analysis gives further explanation to some of the problems occurring when edge focusing is applied to “staircase edges”, see Figure 6.1 and the brief discussion in Section 6.1.1. From experiments [Ber90] it is known that, in general, only one of the two edges in such a configuration will be found by the focusing algorithm and that sometimes even that edge

might get lost when scale decreases. The fact that only one of the edges will be found is obvious from the bifurcation diagram in Figure 6.12 provided that the focusing procedure is initiated from a sufficiently coarse scale and the bifurcation takes places sufficiently far away from the edge subject to tracking. The bifurcation diagram and the previous analysis for local extrema also indicate that the drift velocity of an edge point may increase rapidly even though the edge is not directly involved in any bifurcation, and hence exceed the finite drift velocity estimate used by the edge focusing algorithm.

6.4.3 Elliptic Umbilic

If we are to find a polynomial that both satisfies the diffusion equation and captures the creation of a saddle-extremum pair with increasing scale (which constitutes the basic ingredient in a blob split or a blob creation) then neither the swallowtail nor the butterfly unfolding from Thom's classification theorem will be applicable, since they describe singularities with extent only in one (essential) coordinate direction. The elliptic umbilic does, however, contain a singularity appropriate for such an analysis. The general unfolding in this case is given by

$$G_5(x, y; u, v, w) = x^2y - y^3 + wx^2 + vy + ux \quad (6.112)$$

In order to obtain a polynomial satisfying the diffusion equation, we again replace every monomial $x^m y^n$ with a corresponding polynomial $p_{m,n}(x, y)$ satisfying the diffusion equation. In other words, we replace the x^2y term with $y(x^2 + t)$, the y^3 term with $y^3 + 3yt$ and the wx^2 term with $w(x^2 + t)$, see also Appendix A.5.1, and get

$$L(x, y; t) = x^2y - y^3 + wx^2 + (v - 2t)y + ux + wt \quad (6.113)$$

which is an unfolding still of the same type as G_5 . One observes that the v parameter will not affect the qualitative nature of the singularity set. It corresponds just to a translation along the t axis and can therefore be set to zero without loss of generality. To summarize, the polynomial to be studied is:

$$L(x, y; t) = x^2y - y^3 + wx^2 - 2ty + ux + wt \quad (6.114)$$

Its singularity set is given by

$$\begin{cases} \frac{\partial L}{\partial x} = 2x(y + w) + u = 0 \\ \frac{\partial L}{\partial y} = x^2 - 3y^2 - 2t = 0 \end{cases} \quad (6.115)$$

and the types of the critical points determined by

$$\begin{cases} (AL) = \frac{\partial^2 L}{\partial x^2} = 2(y + w) \\ \frac{\partial^2 L}{\partial x \partial y} = 2x \\ \frac{\partial^2 L}{\partial y^2} = -6y \\ (\mathcal{H}L) = \frac{\partial^2 L}{\partial x^2} \frac{\partial^2 L}{\partial y^2} - \frac{\partial^2 L}{\partial x \partial y} \frac{\partial^2 L}{\partial y \partial x} = -4(3y(y + w) + x^2) \end{cases} \quad (6.116)$$

The solution to this system of equations is analysed in Appendix A.5.3. There it is shown that bifurcations can occur provided that

$$|w| > w_0 = \frac{2}{3} \sqrt{\frac{2|u|}{3}} \quad (6.117)$$

If $w > w_0$ the unfolding describes the creation of a pair with a saddle point a minimum point with increasing t . If t is increased further the minimum point joins with another saddle point and minimum-saddle pair will be annihilated with increasing t , see also Figure 6.15 and Figure 6.16. On the other hand if $w < -w_0$ we will have a creation of a maximum-saddle pair followed by the annihilation of another maximum-saddle pair. In both cases the same extremum point is involved in the two bifurcations.

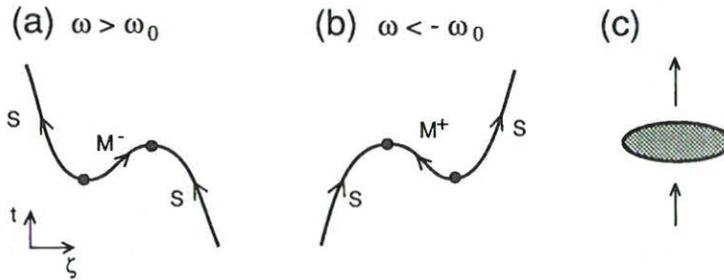


Figure 6.15: (a) When $w > w_0$ the elliptic umbilic unfolding $L(x, y, t) = x^2y - y^3 + wx^2 - 2ty + ux + wt$ describes the creation of a minimum-saddle pair with increasing scale. If the scale parameter is increased further then the created minimum point will merge with another saddle point and minimum-saddle pair will be annihilated. (The ξ coordinate is a coordinate along one of the two branches of the hyperbola in the next figure.) (b) If $w < -w_0$ then the elliptic umbilic instead describes the creation of a maximum-saddle pair followed by the annihilation of a maximum-saddle pair. (c) Interpreted in terms of blobs these events correspond to the creation of a dark (bright) blob followed by the annihilation of a dark (bright) blob (provided that all the saddle points involved in the process are non-shared).

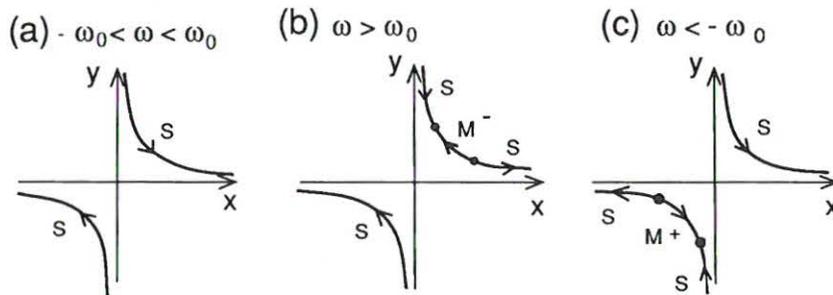


Figure 6.16: The trajectories and the qualitative behaviour of the critical points described by the elliptic umbilic unfolding in the case when $u < 0$. (a) If $-w_0 < w < w_0$ then L describes two saddle points. (b) If $w > w_0$ then L describes the creation of a pair with a saddle point and a minimum point under the influence of another saddle point. Later the minimum point annihilates with the other saddle point and there is only one saddle point left. In this figure the arrows indicate increasing values of the scale parameter while the marked dots show the bifurcation point. (c) If $w < -w_0$ then L instead describes the creations of a maximum-saddle pair followed by the annihilation of a maximum-saddle pair.

Interpreted in terms of blobs these events correspond to the creation of a dark (bright)

blob followed by the annihilation of the same dark (bright) blob provided that the delimiting saddle points involved in the processes are non-shared. If instead the saddle point in the created minimum-saddle (maximum-saddle) pair would have been shared, then the corresponding blob event would have been one dark (bright) blob splitting into two dark (bright) blobs. To summarize,

Observation 6.21 (Evolution properties, Elliptic umbilic case (2D))

The unfolding in the two-dimensional elliptic umbilic case $L(x, y; t) = x^2y - y^3 + wx^2 - 2ty + ux + wt$ describes the creation of a minimum (maximum) point and a saddle point followed by the annihilation of a minimum (maximum) and a saddle point. In other words, it describes the creation of a dark (bright) blob followed by the annihilation of a dark blob (or, possibly, a dark (bright) blob splitting into two followed by a merge of two dark (bright) blobs if the saddle points involved in the process are shared).

The singularity in the elliptic umbilic unfolding contains essential variations taking place in two (or more) (essential) coordinate directions. Therefore, there do not exist any one-dimensional analogies to these blob events.

6.4.4 Summary

Thom’s classification theorem provides a catalogue of elementary catastrophes. In this section we have investigated a few of those, the fold, the cusp and the elliptic umbilic and observed that when restricted to satisfy the diffusion equation they describe a blob annihilation, a blob merge and either of a blob creation or a blob split respectively. When considering the zero-crossings of the Laplacian of the Gaussian we have noticed that in two dimensions there is no simple relation between the locations of these curves and the locations of the local extrema.

6.5 Density of Local Extrema as Function of Scale

In some applications it is of interest to know how the density of local extrema can be expected to vary with scale. One example is the derivation of the effective scale, a transformed scale parameter intended to capture the concept of “scale-space lifetime” in a proper manner, see Section 5.5. Of course, this question seems to be very difficult or even impossible to answer to generally, since such a quantity can be expected to vary substantially from one image to another. How should one then be able to talk about “expected behaviour”? Should one consider all possible (realistic) images, study how this measure evolves with scale and then form some kind of average?

In this section we will perform a simple study. We will consider random noise data with normal distribution. Under these assumptions it turns out to be possible to derive a compact closed form expression for this quantity. We will base the analysis on a treatment by Rice [Ric45] about the expected density of zero-crossings and local maxima of stationary normal processes, see also Papoulis [Pap72] or Cramer and Leadbetter [Cra67].

6.5.1 Continuous Analysis

The density of local maxima μ for a stationary normal process is given by the second and fourth derivatives of the autocorrelation function R ([Ric45] Section 3.6 or [Pap72]):

$$\mu = \frac{1}{2\pi} \sqrt{-\frac{R^{(4)}(0)}{R''(0)}} \quad (6.118)$$

This expression can also be written as [Ric45, Pap72]

$$\mu = \frac{1}{2\pi} \sqrt{\frac{\int_{-\infty}^{\infty} \omega^4 S(\omega) d\omega}{\int_{-\infty}^{\infty} \omega^2 S(\omega) d\omega}} \quad (6.119)$$

where S is the spectral density

$$S(\omega) = \int_{-\infty}^{\infty} e^{-i\omega\tau} R(\tau) d\tau \quad (6.120)$$

Since the scale-space representation L is generated from the input signal f by a linear transformation, the spectral density of L , denoted S_L , is given by

$$S_L(\omega) = |H(i\omega)|^2 S_f(\omega) \quad (6.121)$$

where S_f is the spectral density of f and $H(i\omega)$ the system function

$$H(\omega) = \int_{-\infty}^{\infty} h(t) e^{-i\omega t} dt \quad (6.122)$$

that is the Fourier transform of the the impulse response h . In the scale-space case, h is of course the Gaussian kernel

$$g(\xi; t) = \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} \quad (6.123)$$

which has the Fourier transform

$$G(\omega; t) = \frac{1}{2} e^{-\omega^2 t/2} \quad (6.124)$$

Assuming that f is generated by white noise with $S_f(\omega) = 1$ this gives

$$S_L(\omega) = \frac{1}{4} e^{-\omega^2 t} \quad (6.125)$$

Using the formula

$$\int_0^{\infty} x^m e^{-ax^2} dx = \frac{\Gamma(\frac{m+1}{2})}{2a^{\frac{m+1}{2}}} \quad (6.126)$$

(see e.g. Spiegel [Spi68] 15.77) we can calculate a closed form expression for the density of local maxima of a continuous signal, $p_c(t)$:

$$p_c(t) = \frac{1}{2\pi} \sqrt{\frac{\int_{-\infty}^{\infty} \omega^4 \frac{1}{4} e^{-\omega^2 t} d\omega}{\int_{-\infty}^{\infty} \omega^2 \frac{1}{4} e^{-\omega^2 t} d\omega}} = \frac{1}{2\pi} \sqrt{\frac{2 \frac{\Gamma(\frac{5}{2})}{2t^{\frac{5}{2}}}}{2 \frac{\Gamma(\frac{3}{2})}{2t^{\frac{3}{2}}}}} = \frac{1}{2\pi} \sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} \quad (6.127)$$

Of course an identical result applies to local minima. To summarize,

Proposition 6.22 (Density of local extrema in scale-space (white noise, 1D))

In the scale-space representation of a one-dimensional continuous signal generated by a white noise stationary normal process, the expected density of local maxima (minima) in a smoothed signal at a certain scale decreases with scale as $t^{-\frac{1}{2}}$. Interpreted in terms of $\sigma = \sqrt{t}$ the expected density of local extrema is inversely proportional to σ .

This scale dependence implies that a graph showing the density of local maxima (minima) as function of scale can be expected to be a straight line in a log-log diagram.

$$\log(p_c(t)) = \frac{1}{2} \log\left(\frac{3}{2}\right) - \log(2\pi) - \frac{1}{2} \log(t) = \text{constant} - \frac{1}{2} \log(t) \quad (6.128)$$

Of course one cannot expect that a graph showing this curve for a particular signal to be a straight line. This would require some type of ergodicity assumption that in general not will be satisfied. However, the average behaviour over many different types of imagery could be expected to be close to this situation. In Section 5.5.2 we showed that a natural way to convert the ordinary scale parameter t into a transformed scale parameter called effective scale τ is by $\tau(t) = A + B \log(p(t))$ where $p(t)$ again denotes the expected density of local extrema at a certain scale t and A and B are arbitrary constants. This result shows that

Corollary 6.23 (Effective scale as function of the ordinary scale parameter (1D))

For continuous one-dimensional signals the effective scale parameter τ_c as function of the ordinary scale parameter t is (up to an arbitrary affine transformation) given by a logarithmic transformation

$$\tau_c(t) = A' + B' \log(t) \quad (6.129)$$

where A' and B' are arbitrary constants.

An interesting question concerns what will happen if the uncorrelated white noise model for the input signal is changed. A spectral density applicable to e.g. fractals, see e.g. Barnsley et.al. [Bar88] or Gårding [Går88], is given by $S_f(w) = w^{-\beta}$. For one-dimensional signals, reasonable values of β are obtained between 1 and 3 [Bar88]. Of course, such a distribution is somewhat non-physical since $S_f(w)$ will tend to infinity as t tends to zero and neither one of the spectral moments will be convergent. However, when multiplied by a Gaussian function the second and fourth order moments used in (6.119) will converge provided that $\beta < 3$. We obtain,

$$p_{c,\beta}(t) = \frac{1}{2\pi} \sqrt{\frac{\int_{-\infty}^{\infty} \omega^4 \frac{1}{4} e^{-\omega^2 t} \omega^{-\beta} d\omega}{\int_{-\infty}^{\infty} \omega^2 \frac{1}{4} e^{-\omega^2 t} \omega^{-\beta} d\omega}} = \frac{1}{2\pi} \sqrt{\frac{\int_{-\infty}^{\infty} \omega^{(4-\beta)} e^{-\omega^2 t} d\omega}{\int_{-\infty}^{\infty} \omega^{(2-\beta)} e^{-\omega^2 t} d\omega}} = \quad (6.130)$$

$$= \frac{1}{2\pi} \sqrt{\frac{2 \frac{\Gamma(\frac{5-\beta}{2})}{2t^{\frac{5-\beta}{2}}}}{2 \frac{\Gamma(\frac{3-\beta}{2})}{2t^{\frac{3-\beta}{2}}}}} = \frac{1}{2\pi} \sqrt{\frac{3-\beta}{2}} \frac{1}{\sqrt{t}} \quad (\beta < 3) \quad (6.131)$$

Proposition 6.24 (Density of local extrema in scale-space (fractal noise, 1D))

In the scale-space representation of a one-dimensional continuous signal generated by a stationary normal process with spectral density $\omega^{-\beta}$ the expected density of local maxima (minima) in a smoothed signal at a certain scale decreases with scale as $t^{-\frac{1}{2}}$.

Note that also this graph will be a straight line in a log-log diagram.

$$\log(p_{c,\beta}(t)) = \frac{1}{2} \log\left(\frac{3-\beta}{2}\right) - \log(2\pi) - \frac{1}{2} \log(t) \quad (6.132)$$

The slope will be the same as in the case with uncorrelated white noise, but the dependence on β means that the graph has been translated by $\frac{1}{2}(\log(3) - \log(3 - \beta))$ in the negative vertical direction or equivalently by $\log(3) - \log(3 - \beta)$ in the negative horizontal direction. This corresponds to a multiplication of the t -value by $(1 - \frac{\beta}{3})^{-1}$.

6.5.2 Discrete Analysis

From the previous continuous analysis we have that the density of local extrema may tend to infinity as the scale parameter tends to zero. It is obvious that this result is not applicable to discrete signals, since in this case the density of local extrema will have an upper bound because of the finite sampling. Hence, in order to capture what will happen in this case, a genuinely discrete treatment is necessary. We will base the analysis on the discrete scale-space concept from Chapter 2:

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t) f(x - n) \quad (6.133)$$

where $T(n; t) = e^{-t} I_n(t)$ is the discrete analogue of the Gaussian kernel and I_n are the modified Bessel functions of integer order.

The probability that a point at a certain scale is say a local maximum point is equal to the probability that its value is greater than (or possibly equal to)³ the values of its nearest neighbours. In one-dimension we have:

$$p(x_i \text{ is a local maximum at scale } t) = p((L(x_i; t) \geq L(x_{i-1}; t)) \wedge (L(x_i; t) \geq L(x_{i+1}; t))) \quad (6.134)$$

If we assume that the input signal f is generated by a stationary normal process then also L will be a stationary normal process and the distribution of any triple $(L_{i-1}, L_i, L_{i+1})^T$, from now on denoted by $\xi = (\xi_1, \xi_2, \xi_3)^T$, will be jointly normal, which means that its statistics will be completely determined by the mean vector and the autocovariance matrix. Trivially, we have that the mean of ξ is zero provided that the mean of f is zero. Since the transformation from f to L is linear, the autocovariance C_L for the smoothed signal L will be given by

$$C_L(\cdot; t) = T(\cdot; t) * T(\cdot; t) * C_f(\cdot) = T(\cdot; 2t) * C_f(\cdot) \quad (6.135)$$

where C_f denotes the autocovariance of f . In the last equality we have made use of the semigroup property $T(\cdot; s) * T(\cdot; t) = T(\cdot; s + t)$ for the family of convolution kernels. If the input signal consists of white noise then C_f will be the discrete delta function and $C_L(\cdot; t) = T(\cdot; 2t)$. Taking the symmetry property $T(-n; t) = T(n; t)$ into account as well,

³There are actually several ways to define a local extremum of a discrete signal using different combinations of "strictly greater than" and "greater than or equal to". However, those distinctions are not important in the expression below since they will differ only for non-generic signals and in addition the numerical value of the integral below will anyway be the same if some of the \geq signs are replaced by $>$.

the distribution of ξ will be jointly normal with mean vector m_{3D} and covariance matrix C_{3D} given by:

$$m_{3D} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad C_{3D} = \begin{pmatrix} T(0; 2t) & T(1; 2t) & T(2; 2t) \\ T(1; 2t) & T(0; 2t) & T(1; 2t) \\ T(2; 2t) & T(1; 2t) & T(0; 2t) \end{pmatrix} \quad (6.136)$$

Using the probability density for multivariate normal random variables, see e.g. Papoulis [Pap72], we can express

$$p_d(t) = \{\text{probability that at scale } t \text{ a certain point is a local maximum}\} \quad (6.137)$$

as

$$p_d(t) = \int \int \int_{\{\xi=(\xi_1, \xi_2, \xi_3): (\xi_2 \geq \xi_1) \wedge (\xi_2 \geq \xi_3)\}} \frac{1}{\sqrt{(2\pi)^3 |C_{3D}|}} e^{-\frac{1}{2} \xi^T C_{3D}^{-1} \xi} d\xi_1 d\xi_2 d\xi_3 \quad (6.138)$$

where $|C_{3D}|$ denotes the determinant of C_{3D} and C_{3D}^{-1} its inverse.

To reduce the dimensionality of the integral we introduce new variables $\eta_1 = \xi_2 - \xi_1$ and $\eta_2 = \xi_2 - \xi_3$. Then also $\eta = (\eta_1, \eta_2)^T$ will be jointly normal and its statistics completely determined by

$$m_{2D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \quad C_{2D} = \begin{pmatrix} a_0(t) & a_1(t) \\ a_1(t) & a_0(t) \end{pmatrix} \quad (6.139)$$

From well-known rules for the covariance $C(\cdot, \cdot)$ of a linear combination of stochastic variables it follows that

$$a_0(t) = C(\eta_1, \eta_1) = C(\eta_2, \eta_2) = 2(T(0; 2t) - T(1; 2t)) \quad (6.140)$$

$$a_1(t) = C(\eta_1, \eta_2) = C(\eta_2, \eta_1) = T(0; 2t) - 2T(1; 2t) + T(2; 2t) \quad (6.141)$$

From $a_0(t) - a_1(t) = T(0; t) - T(2; t)$ and the unimodality property of T ($T(i; t) > T(j; t)$ if $|i| > |j|$) it follows that $a_0(t) > a_1(t)$ and trivially $a_0(t) > 0$ for all t . Now $p_d(t)$ can be expressed in terms of a two-dimensional integral

$$p_d(t) = \int \int_{\{\eta=(\eta_1, \eta_2): (\eta_1 \geq 0) \wedge (\eta_2 \geq 0)\}} \frac{1}{\sqrt{(2\pi)^2 |C_{2D}|}} e^{-\frac{1}{2} \eta^T C_{2D}^{-1} \eta} d\eta_1 d\eta_2 \quad (6.142)$$

Here $|C_{2D}|$ does not depend on η and can thus be moved out of the integral. After some calculations, see Appendix A.5.4, it follows that

$$p_d(t) = \frac{1}{4} + \frac{1}{2\pi} \arctan \left(\frac{a_1(t)}{\sqrt{a_0^2(t) - a_1^2(t)}} \right) \quad (6.143)$$

Observe that for any $a_0(t)$ and $a_1(t)$ this value is guaranteed to never be outside the interval $[0, \frac{1}{2}]$. With our expressions for $a_0(t)$ and $a_1(t)$, given by smoothing with the discrete analogue of the Gaussian kernel, the maximum value over variations in t is obtained for $t = 0$:

$$p_d(0) = \frac{1}{3} \quad (6.144)$$

Proposition 6.25 (Density of local extrema in discrete scale-space (1D))

In the scale-space representation (6.133) of a one-dimensional discrete signal generated by a white noise stationary normal process the expected density of local maxima (minima) in a smoothed signal at a certain scale t is given by

$$p_d(t) = \frac{1}{4} + \frac{1}{2\pi} \arctan \left(\frac{a_1(t)}{\sqrt{a_0^2(t) - a_1^2(t)}} \right) \quad (6.145)$$

where $a_0(t) = 2(T(0; 2t) - T(1; 2t))$, $a_1(t) = T(0; 2t) - 2T(1; 2t) + T(2; 2t)$ and T in turn denotes the discrete analogue of the Gaussian kernel.

It is interesting to compare this discrete expression with the earlier continuous results. The scale value where the continuous estimate gives a density equal to the discrete density at $t = 0$ is given by the equation $p_c(t) = p_d(0)$, that is by

$$\frac{1}{2\pi} \sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} = \frac{1}{3} \quad (6.146)$$

which has the solution

$$t_{c-d} = \frac{27}{8\pi^2} \approx 0.3420 \quad (6.147)$$

This corresponds to a σ -value of about 0.5848. Below this scale the continuous analysis is, from that point of view, definitely not a valid approximation of what will happen to discrete signals.

6.5.2.1 Asymptotic Behaviour at Fine Scales

A second order Taylor expansion of $p_d(t)$ around $t = 0$, see Appendix A.5.5, yields

$$p_d(t) = \frac{1}{3} - \frac{1}{2\sqrt{3}\pi} t + \frac{1}{6\sqrt{3}\pi} t^2 + O(t^3) \quad (6.148)$$

This means that the effective scale $\tau_d(t)$ can be Taylor expanded around $t = 0$ and, see Appendix A.5.5,

$$\tau_d(t) = \log \left(\frac{p_d(0)}{p_d(t)} \right) = \frac{\sqrt{3}}{2\pi} t + \left(\frac{1}{2\sqrt{3}\pi} + \frac{3}{8\pi^2} \right) t^2 + O(t^3) \quad (6.149)$$

In other words, at fine scales the effective scale increases approximately linearly with the ordinary scale parameter t .

Corollary 6.26 (Asymptotic behaviour of the effective scale at fine scales (1D))

For one-dimensional discrete signals the effective scale is approximately a linear function of the ordinary scale parameter t for small t .

6.5.2.2 Asymptotic Behaviour at Coarse Scales

A Taylor expansion of $p_d(t)$ at coarse scales, see Appendix A.5.6, gives

$$p_d(t) = \frac{1}{2\pi} \sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} \left(1 + \frac{1}{8t} + O\left(\frac{1}{t^2}\right) \right) \quad (6.150)$$

which asymptotically agrees with the continuous result in (6.127). By inserting this expression into the expression for effective scale and using $p_d(0) = \frac{1}{3}$ we get

$$\tau_d(t) = \log \left(\frac{p_d(0)}{p_d(t)} \right) = \log \left(\frac{2\pi}{3} \sqrt{\frac{2}{3}} \right) + \frac{1}{2} \log(t) + \log \left(1 - \frac{1}{8t} + O\left(\frac{1}{t^2}\right) \right) \quad (6.151)$$

which shows that at coarse scales the effective scale increases approximately logarithmically with the ordinary scale parameter t .

Corollary 6.27 (The effective scale at coarse scales (1D))

For one-dimensional discrete signals the effective scale is approximately (and up to an arbitrary affine transformation) a logarithmic function of the ordinary scale parameter t for large t .

The term $\log(1 - \frac{1}{8t} + O(\frac{1}{t^2}))$ expresses how much the effective scale derived for discrete signals differs from the effective scale derived for continuous signals, provided that the same values of the (arbitrary) constants A and B are selected in both cases.

6.5.3 Comparisons Between the Continuous and Discrete Results

In order to illustrate the difference between the density of local maxima in the scale-space representation of a continuous and a discrete signal we show the graphs of p_c and p_d in Figure 6.17 (linear scale) and Figure 6.18 (log-log scale). As expected, the curves differ significantly for small t and approach each other as t increases. Numerical values quantifying

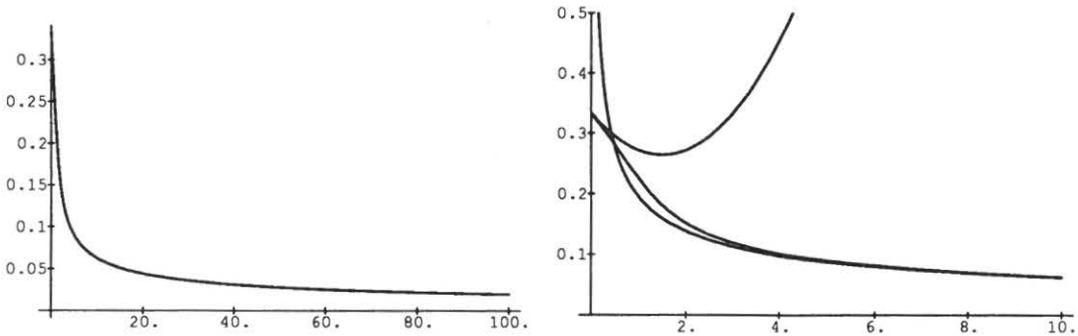


Figure 6.17: The density of local maxima of a discrete signal $p_d(t)$ as function of the ordinary scale parameter t in linear scale. (a) Graph for $t \in [0, 100]$. (b) Enlargement of the interval $t \in [0, 10]$. For comparison the graphs showing the density of local extrema for a continuous signal $p_c(t)$ and the second order Taylor expansion of $p_d(t)$ around $t = 0$ have also been drawn. As expected, the continuous and discrete results differ significantly for small values of t but approach each other as t increases. The MacLaurin expansion is a valid approximation only in a very short interval around $t = 0$.

this difference for a few values of t are given in Table 6.1. We have tabulated the ratio

$$\tau_{diff}(t) = \frac{\tau_d(t) - \tau_c(t)}{\tau_c(2t) - \tau_c(t)} = \frac{\tau_d(t) - \tau_c(t)}{\frac{\log(2)}{2}} \quad (6.152)$$

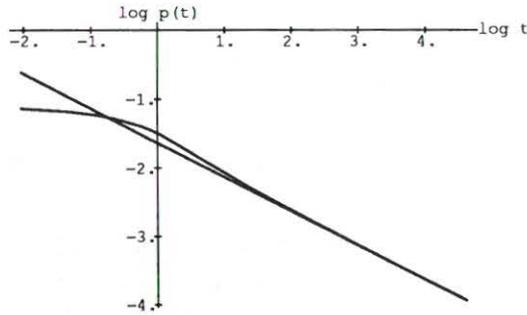


Figure 6.18: The density of local maxima of a continuous p_c and a discrete p_d signal as function of the ordinary scale parameter t in log-log scale ($t \in [0, 100]$). The straight line shows $p_c(t)$ and the other curve $p_d(t)$. One observes p_c and p_d approach each other as the scale parameter increases. When t tends to zero $p_c(t)$ tends to infinity while $p_d(t)$ tends to a constant ($\frac{1}{3}$).

which is a natural measure for how much the effective scale obtained from a continuous analysis differs from a discretely determined effective scale. The quantity is normalized such that one unit in τ_{diff} corresponds to the increase in τ_c induced by an increase in t with a factor of two.

6.5.4 Extension to Two Dimensions

The same type of analysis can, in principle, be carried out also for two-dimensional signals. The probability that a specific point at a certain scale is a local maximum point is again equal to the probability that its value is greater than the values of its neighbours. Depending on the connectivity concept (four-connectivity or eight-connectivity for a square grid) we then obtain either a four-dimensional or an eight-dimensional integral to solve. However, because of the dimensionality of the integrals we have not made any attempts to calculate explicit expressions for the variation of the density as function of scale. Instead, for implementational purpose, the behaviour over scale has been simulated for various uncorrelated random noise signals, see Section 5.5.2.2 for more details. From those experiments it has been empirically demonstrated that the $t^{-\gamma}$ dependence of the density of local extrema as function of scale constitutes a reasonable approximation at coarse levels of scale.

6.6 Summary

We have analysed the behaviour of critical points in scale-space and shown that non-degenerate critical points will in general form regular curves across scales. Along those we have provided generally valid estimates of the drift velocity. At degenerate critical points the behaviour is more complicated and bifurcations may take place. For one-dimensional signals, the only bifurcation events possible when the scale parameter increases, are annihilations of pairs of local maxima and minima, while for two-dimensional signals both annihilations and creations of pairs of local extrema and saddle points can occur. Applied to grey-level and scale-space blobs only annihilations and merges will take place in the one-dimensional case,

t	$\tau_{diff}(t)$
0	∞
0.0625	250.30 %
0.25	67.46 %
1.0	-41.82 %
4.0	-10.47 %
16.0	-2.32 %
64.0	-0.56 %
256.0	-0.14 %
∞	0

Table 6.1: *Indications about how the effective scale obtained from a discrete analysis differs from the effective scale given by the continuous scale-space theory. The quantity $\tau_{diff}(t)$ expresses the difference between $\tau_d(t)$ and $\tau_c(t)$ normalized such that one unit (100 %) in $\tau_{diff}(t)$ corresponds to the increase in τ_c induced by an increase in t with a factor of two.*

while the list of possibilities in two-dimensions comprises four types: annihilations, merges, splits and creations.

Let us finally point out that this analysis has been mainly concerned with the scale-space concept for continuous signals. When one is to implement this theory computationally it is obvious that one has to consider sampled, that is, discrete data. At coarse scales, when a characteristic length of features in the image can be regarded as large compared to the distance between adjacent grid points, it seems plausible that the continuous results should constitute a reasonable approximation to what will happen in the scale-space representation of a discrete signal and vice versa. However, as indicated above in Section 6.5 this similarity will not necessarily hold⁴ at fine scales. In those cases a genuinely discrete theory might be needed. We believe that a thorough understanding of what happens to continuous signals under scale-space smoothing constitutes a first step towards this goal.

⁴There are also some conceptual complications arising in this context, for instance, concerning what should be meant by drift velocity for discrete signals. It seems very difficult to estimate such a quantity accurately, especially at fine scales, since in the discrete case local extrema will not move continuously — but rather in steps; from one pixel to the next. Thus, one cannot talk about velocity, but rather about how long time it takes until an extremum point moves, say one pixel. An alternative approach to this problem would be by analysing the feature points with sub-pixel accuracy although this idea has not been carried out. Other conceptual problems concern what should be meant by singularities or degenerate and nondegenerate critical points in the discrete case? One possibility is to define those in terms of transitions, say e.g. blob bifurcations. But, will the classification of possible blob events still be valid in the discrete case?

Chapter 7

Computing the Representation

When building a representation of the proposed type, there are several computational aspects that must be considered in addition to those already dealt with. We need algorithms for

- detecting grey-level blobs in smoothed grey-level images
- registering bifurcations
- linking grey-level blobs across scales into scale-space blobs
- computing the scale-space blob volumes.

In this chapter we will briefly describe how this can be done. Some algorithmic descriptions will by necessity be somewhat technical, and as a general guideline those details can be skipped by the hasty reader.

7.1 Grey-Level Blob Detection

We start by outlining an algorithm for detecting blobs in a grey-level image. We will describe the case with bright blobs on a dark background only. The case with dark blobs on a bright background can be solved by application of the bright-blob detection algorithm on the inverted grey-level image.

7.1.1 The One-Dimensional Case

Detecting grey-level blobs in a one-dimensional discrete signal is trivial. In this case it suffices to start from each local maximum point and initiate search procedures in each one of the two possible directions, see Figure 7.1. Every search procedure continues until it finds a local minimum point, i.e., as long as the grey-level values are decreasing. As soon as a minimum point has been found the search procedure is stopped and the grey-level value is registered. The base-level of the blob is then given by the maximum value of the these two registered grey-levels. From this information the grey-level blob is given by those pixels that can be reached from the local maximum point without descending below the base-level.

The two-dimensional case is more elaborate, since the search then may be performed in a variety of directions. In Section 7.1.4 we will describe a methodology that avoids the search problem and instead performs a global blob detection based on a pre-sorting of the grey-levels. However, we will first state some basic properties that turn out to be useful for the algorithm.

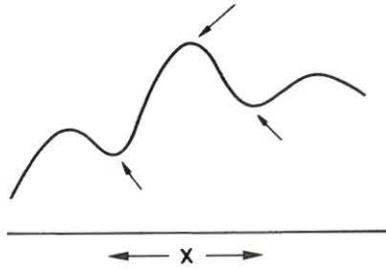


Figure 7.1: The blob detection algorithm for a one-dimensional discrete signal is trivial. The base-level of a bright blob is equal to the maximum value of the grey-levels in the two local minimum points surrounding the local maximum point of the blob.

7.1.2 Grey-Level Blob Invariants

From the definition of a grey-level blob one easily realizes that the following basic properties hold in the classification of the bright blobs of a discrete signal, see also Figure 7.2. To simplify the presentation, let the notation “higher-neighbour” stand for “neighbour pixel having a higher grey-level value”. Further, the concept “background” will mean a pixel that has been classified as not belonging to a blob. (Remember that the saddle point will not be included in the grey-level blob in the discrete case.)

1. If a pixel has no higher-neighbour then it is a local maximum and will be the seed of a blob.
2. Else, if it has at least one higher-neighbour which is background then it cannot be part of any blob and must be background.
3. Else, if it has more than one higher-neighbour and if those higher-neighbours are parts of different blobs then it cannot be a part of any blob, but must be background.
4. Else, it has one or more higher-neighbours, which are all parts of the same blob. Then it must also be a part of that blob.

Starting from these properties sequential or parallel blob detection algorithms can be easily constructed.

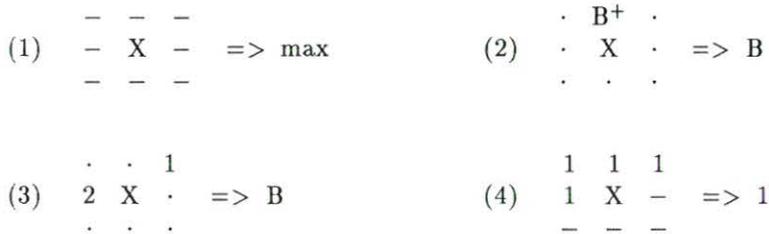


Figure 7.2: Illustration of the grey-level blob invariants numbered from 1 to 4 above. In these figures the symbol 'X' denotes the central point that is to be classified, the symbol '-' a pixel having a lower grey-level than the central point, 'B' a pixel classified as background, 'B+' a background pixel with a higher grey-level than the central point, '1' and '2' pixels classified as belonging to regions labeled 1 and 2 respectively and '.' an arbitrary pixel.

7.1.3 Generic v.s. Non-Generic Signals

One aspect to be considered in this context is generic v.s. non-generic signals. Actually, the properties stated in the previous section are valid only for generic signals, that is signals for which all pixel values are different. Non-generic signals, where connected pixels may have equal values, can lead to complications. Consider for instance the detection of a local extremum. There are several possible ways to define this concept in a discrete situation:

Definition 7.1 (Weak local maximum)

A point (x, y) is said to be a weak local maximum point for a discrete function $g : Z^2 \rightarrow R$ if for $g(\xi, \eta) \leq g(x, y)$ holds for all neighbours (ξ, η) of (x, y) .

Definition 7.2 (Strict local maximum)

A point (x, y) is said to be a strict local maximum point for a discrete function $g : Z^2 \rightarrow R$ if $g(\xi, \eta) < g(x, y)$ holds for all neighbours (ξ, η) of (x, y) .

Definition 7.3 (Semi-weak local maximum)

A point (x, y) is said to be a semi-weak local maximum point for a discrete function $g : Z^2 \rightarrow R$ if $g(\xi, \eta) \leq g(x, y)$ holds for all neighbours (ξ, η) of (x, y) and in addition $g(\xi, \eta) < g(x, y)$ for at least some neighbour (ξ, η) of (x, y) .

Definition 7.4 (Region-based local maximum)

A point (x, y) is said to be a region-based local maximum point for a discrete function $g : Z^2 \rightarrow R$ if $g(\xi, \eta) \leq g(x, y)$ holds for all neighbours (ξ, η) of (x, y) and in addition, by following connected points having the same grey-level value as (x, y) , it is impossible to reach a neighbour having a higher grey-level value.

For generic signals, all these formulations will be equivalent. On the other hand, in situations where adjacent pixels have equal values, they can give quite different results, see Figure 7.3 for an illustration.

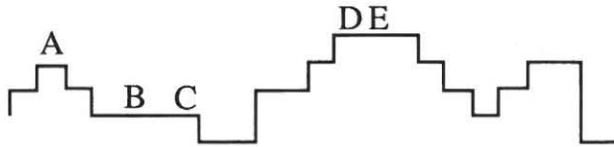


Figure 7.3: For non-generic signals, special care must be taken when defining the concept of a local maximum point. In this figure, point A represents a maximum point as it would appear in a generic signal and accordingly it satisfies all the definitions of a local maximum; it is a weak local maximum, a strict local maximum, a semi-weak local maximum as well as a region-based local maximum. B is a weak local maximum while C is both a weak local maximum and a semi-weak local maximum. D is both a weak local maximum, a semi-weak local maximum and a region-based local maximum, while E is both a weak local maximum and a region-based local maximum. The only one of the definitions that gives reasonable results in all these cases is the region-based local maximum.

Therefore when dealing with degenerate data, as will be the case when the grey-level values have been quantized, the *region-based local maximum* is apparently the appropriate

definition to work with. When implementing this concept computationally, it is convenient to pre-process the input data with the connected-component-labelling-algorithm, which assigns a unique identifier to each set of connected pixels having the same value. Then, the comparisons between neighbouring grey-level values, which were earlier based on neighbouring *pixels*, can instead be made based on neighbouring *regions*. In other words, the use of “pixel” in the grey-level blob invariants in Section 7.1.2 can be replaced by “region”. For instance, from this viewpoint a point is a region-based maximum if and only if it belongs to a region that only borders upon regions having a lower grey-level value. Of course, it is also possible to implement an algorithm for region-based maximum detection by first detecting weak local maxima and then carrying out a neighbourhood search in each individual degenerate case. This method may in fact be faster if it is known in advance that the number of degenerate situations are few. However, here we have made use of the approach with connected regions because of its algorithmic simplicity.

7.1.4 Sequential Implementation

The idea with the algorithm is to initiate a blob seed in every local maximum point and then let each maximum region grow until it meets with some other maximum region. If the growth procedure is performed in descending grey-level order, we are guaranteed that no maximum region will grow too much. The case where adjacent points have equal values might lead to some practical problems and we avoid those by pre-processing the image with the connected-component-labelling-algorithm. Hence, given a finite discrete real-valued or integer-valued image perform the following steps:

1. Run the connected-component-labelling-algorithm on the grey-level image in order to group connected points with equal values into regions¹. After this step connected pixels having same grey-level will be given the same unique region label.
2. Sort the regions with respect to their grey-levels. For integer-valued images this may be done efficiently by indexing.
3. For each region, create a list of its neighbour regions having a higher grey-level.
4. Group the regions into blobs, i.e., for each region in descending grey-level order: count how many references it has to neighbour regions with a higher grey-level.
 - (a) If the region has no such neighbours, then it is a local maximum point and will be the seed of a blob. Set a flag allowing the blob to grow, and store the grey-level of the region as the maximum grey-level of the blob.
 - (b) Else, if the region has a neighbour region with higher grey-level, which has been classified as background, then the current region cannot be a part of a bright blob and must also be classified as background.

¹This step can be omitted if it is known in advance that no two connected pixels have equal values. Then every pixel can be regarded as a region in the description below. For data given by scale-space smoothing it can in general be assumed that adjacent pixels in fact have different values, provided that the calculations are carried out in floating point precision and that the output image is stored on that format. Therefore this step will in some situations be superfluous. Another possible way to ensure that the input data is non-degenerate is by modifying the least significant bits in the floating point numbers such that no pair of neighbouring pixels have equal values. The effects of such a modification should be negligible if the error introduced by this operation is kept below the numerical error in the implementation of the scale-space smoothing.

- (c) Else, if the region has more than one higher neighbour region, and if those neighbour regions are not parts of the same blob, then the region cannot be a part of a blob and must be set to background. For blob, containing any of the neighbour regions, carry out the following:
 - If the blob is still allowed to grow then clear the flag, which allows it to grow, and store the current grey-level as the base-level of the blob. Store this region as a² saddle region associated with the blob.
 - (d) Else, if none of the previous conditions are true then the neighbour regions having a higher grey-level than the current region are all parts of the same blob. If that blob is still allowed to grow then the current region should be included as a part of that blob. Otherwise the region should be set to background.
5. Create a blob image where all pixels in a region classified as blob are given the same (unique) label of the blob.
 6. Traverse the grey-level image and the blob image simultaneously and compute the contrast, area and volume for each blob. Store these values in a data structure together with the extremum regions and the saddle regions of the blobs.

7.1.4.1 Alternatives

There are several simplifications that could be made if it is known for sure that the algorithm only needs to handle generic signals. Also the approach with a pre-sorting of the pixels with respect to their grey-level values can be changed. One can initiate a seed in each local maximum and let the classification propagate in a grass-fire-like way. Then, it will be sufficient to process only the frontier. The grey-level blob invariants can also form the basis of a parallel implementation, see [Lin91a] for a brief description.

7.2 Linking Grey-Level Blobs into Scale-Space Blobs

Linking blobs across scales could be a potential source to difficult matching problems, since blobs can move, disappear, merge, split or be created when the scale parameter changes. However, the notion of a scale-space with a *continuous* scale parameter gives us a simple way to circumvent these problems in many cases, since the scale step may be varied at will. If one is confronted with a problematic matching situation, then the matching difficulties can often be avoided by a refinement of the scale sampling. If the scale step is *adaptively* made just fine enough it should be trivial to judge which grey-level blobs belong to the same scale-space blob.

According to the classification of blob events in the previous chapter, there are four possible types of blob events for generic signals: annihilation, merge, split and creation. Assuming that this continuous property is valid for discrete signals, all we have to look for are those four possibilities. This implies that if a situation is encountered with, say, three blobs at a fine scale, seeming all to belong to the same coarse scale blob, then the situation can (under the assumption of generic signals) be resolved into bifurcations of these types by a sufficient number of refinements of the sampling along the scale direction. This

²Observe that for degenerate signals a grey-level blob can be delimited by more than one saddle point (where all such saddle points have the same grey-level), and that these “saddle points” in turn can be regions.

constitutes the basic principle behind the *adaptive scale linking algorithm*, which essentially refines the scale sampling until all relations between blobs at adjacent levels of scales can be decomposed into those primitive transitions.

7.2.1 Blob-Blob Matching

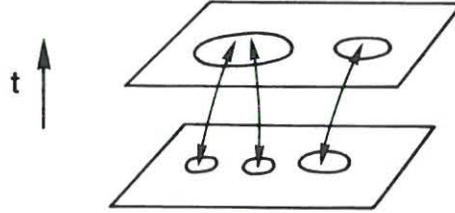


Figure 7.4: The blob linking between scale levels is based on spatial coincidence, i.e., if two grey-level blobs at adjacent levels of scale have a spatial point in common they are registered as matching candidates of each other. In this example the left situation will be registered as a possible blob merge, while the right situation will be a possible plain link within the same scale-space blob.

Based on this idea, the blob linking between two levels of scale can be performed based on spatial coincidence. A straightforward strategy is to start with a relatively fine initial sampling in scale and then for each pair of scale levels traverse all pixels and for each point investigate if it is included in a blob both at the lower scale and at the higher scale. If so, the lower blob is registered as a match candidate of the higher blob, and the higher blob is registered as a match candidate of the lower blob. By inclusion in a blob, we here mean that a pixel belongs to the *support region* of the blob. It is convenient to introduce a notation formalizing this statement:

Definition 7.5 (Blob-blob matching candidate)

Let S_F and S_C be the support regions of two grey-level blobs G_F and G_C existing at two adjacent scale levels t_F and t_C respectively where $t_F < t_C$. G_F is said to be a blob-blob matching candidate from above of G_C denoted

$$G_F \swarrow_{b-b} (G_C) \tag{7.1}$$

if there exists some pixel in S_C that is contained in S_F . Similarly G_C is said to be a blob-blob matching candidate of G_F from below denoted

$$G_C \nwarrow_{b-b} (G_F) \tag{7.2}$$

if there exists some pixel in S_F that is contained in S_C .

Obviously, the definition implies that matching candidates of this types are bidirectional

$$(G_F \swarrow_{b-b} (G_C)) \iff (G_C \nwarrow_{b-b} (G_F)) \tag{7.3}$$

Given such relations between the grey-level blobs from the scale-space representations at two adjacent levels of scale, we can discern the following primitive types of elementary matching situations; possible link within a scale-space blob, possible annihilation, possible merge, possible split and possible creation; see also Figure 7.5.

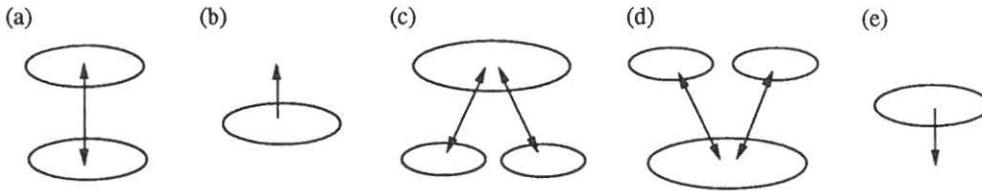


Figure 7.5: Elementary matching situations given by matching relations between blobs at different scales. (a) plain link (b) annihilation (c) merge (d) split and (e) creation.

As indicated above, the idea behind the adaptive scale linking algorithm is basically that it should be possible to decompose all relations between blobs at adjacent levels of scale into primitive relations of the types listed above by successive refinements of the scale sampling. For instance, if any blob has more than two matching candidates then a refinement should be made.

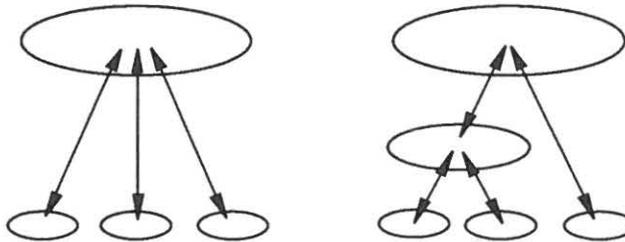


Figure 7.6: An encountered situation with, say, three blobs at a fine scale that all seem to belong to the same coarse scale blob can generically be resolved in to a sequence of two successive blob merges by a refinement of the scale sampling. Observe that this figure shows only the support regions of the blobs.

7.2.2 Extremum-Blob Matching

There are, however, some situations where this methodology might lead to an unnecessarily large number of refinements. Consider for instance a pair of neighbouring blobs, that is two blobs sharing the same delimiting saddle point, which slowly drift with the scale-space smoothing, see Figure 7.7(a). Then a very large number of refinements might actually be needed in order to resolve the situation into two plain links.

The efficiency in such situations can be substantially improved by allowing for extremum-blob matching. The idea is to perform an additional gathering of matching candidates based on the inclusion of the extremum points at one level scale in the grey-level blobs at the other level of scale. In other words, if the maximum point of a blob B at one scale is included in the support region of grey-level blob A at the other level of scale then blob A is registered as match candidate of blob B , see Figure 7.7(b).

Definition 7.6 (Extremum-blob matching candidate)

Let $t_F < t_C$ be two scale levels and let G_F be a grey-level blob at scale t_F and G_C a grey-level blob at scale t_C with blob support regions S_F and S_C and extremum points E_F and E_C

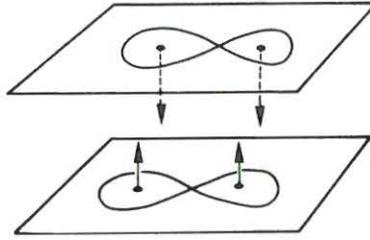


Figure 7.7: (a) Basing the matching just on blob-blob matching candidates might lead to an unnecessarily large number of refinements for configurations with two neighbouring blobs that slowly drift due to the scale-space smoothing. (b) In such situations extremum-blob matching can be used for improving the matching, especially at coarser levels of scale where pairs of double candidates appear relatively frequently. The idea is to gather additional matching candidates based on inclusion of the local maximum points at one level of scale in the grey-level blobs at the other level of scale. If the matching candidates are unique and mutual then a match will be accepted without refinement.

respectively. G_F is said to be an extremum-blob matching candidate from above of G_C denoted

$$G_F \swarrow_{e-b} (G_C) \quad (7.4)$$

if E_C is contained in S_F . Similarly G_C is said to be an extremum-blob matching candidate of G_F from below denoted

$$G_C \nwarrow_{e-b} (G_F) \quad (7.5)$$

if E_F is contained in S_C .

It is clear that these matching relations will not necessarily be bidirectional. The idea behind this construction is that if the extremum-blob matching candidates obtained in this way are mutual and if they resolve a situation with a pair of double candidates then the situation can be registered as a pair of possible plain links. It turns out that these types of situations are rather common at coarser levels of scale, compare with Figure 5.8 and Figure 5.9, where two blobs “hang together” but anyway drift slowly due to the scale-space blurring. As we shall see later, these relations can also be used for stating stronger matching conditions than the blob-blob coincidence requirements.

7.2.3 Registering Bifurcations in Scale-Space

What remains to decide is when a blob match should be accepted. In our current implementation we, in principle, perform a scale refinement each time an unclear matching situation occurs, and accept matches in principle only when all blob events between the two scale levels can be classified as belonging to either one of the primitive cases: plain link within a scale-space blob, blob annihilation, blob merge, blob split or blob creation.

7.2.3.1 Notation

There are several possible ways to define situations which are candidates of being bifurcation situations. In order to enable a clear statement of what we mean by that, we will first introduce some notation describing the number of matching relations associated with a certain grey-level blob G :

- $\# \swarrow_{b-b} (G)$ denotes the total number of blob-blob matching relations *from above*, *starting* at G , that G is involved in.
- $\# \searrow_{b-b} (G)$ denotes the total number of blob-blob matching relations *from above*, *ending* at G , that G is involved in.
- $\# \swarrow_{e-b} (G)$ denotes the total number of extremum-blob matching relations *from above*, *starting* at G , that G is involved in.
- $\# \searrow_{e-b} (G)$ denotes the total number extremum-blob matching relations *from above*, *ending* at G , that G is involved in.

Similarly, the symbols $\# \nwarrow_{b-b} (G)$, $\# \nearrow_{b-b} (G)$, $\# \nwarrow_{e-b} (G)$ and $\# \nearrow_{e-b} (G)$ describe the number of matching candidates *from below* that are associated with a certain grey-level blob G . One observes that

$$\# \swarrow_{b-b} (G) = \# \nearrow_{b-b} (G) \quad (7.6)$$

$$\# \searrow_{e-b} (G) \leq \# \searrow_{b-b} (G) \quad (7.7)$$

$$\# \swarrow_{e-b} (G) \leq \# \swarrow_{b-b} (G) \quad (7.8)$$

$$\# \swarrow_{e-b} (G) = \text{either 0 or 1 for generic signals} \quad (7.9)$$

(7.6) is a direct consequence of the property that blob-blob matching relations are bidirectional. (7.7) and (7.8) simply mean that the number of blob-extremum matching relations cannot exceed the number of blob-blob matching relations, since both these types of relations are obtained from spatial coincidence and the number of pixels that satisfy the definition of a region-based local extremum cannot exceed the number of pixels in the blob support region. (7.9) is guaranteed to hold only for generic signals and simply means that a point that is an extremum point at one level of scale cannot be contained in more than one grey-level blob at an other level of scale.

7.2.3.2 Weak Conditions for Bifurcation Situations

Given these relations we can state when a set of relations between blobs at adjacent scales should be interpreted as a candidate for being either a link within the same scale-space blob, a bifurcation situation or a complex situation to be subject to simplification by further refinements. In the following definitions, grey-level blobs existing at the finer of the two scales will be throughout denoted G_F , G_{F1} and G_{F2} , while grey-level blobs at the coarser scale level will be written G_C , G_{C1} and G_{C2} :

Definition 7.7 (Weak link candidate)

$\{G_F, G_C\}$ are said to form a weak link situation between t_F and t_C if

$$(\# \searrow_{b-b} (G_F) = 1) \wedge (\# \swarrow_{b-b} (G_C) = 1) \wedge (G_C \nwarrow_{b-b} (G_F)) \quad (7.10)$$

Definition 7.8 (Weak annihilation candidate)

$\{G_F\}$ is said to form a weak annihilation situation between t_F and t_C if

$$(\# \nwarrow_{b-b} (G_F) = 0) \quad (7.11)$$

Definition 7.9 (Weak merge candidate)

$\{G_{F1}, G_{F2}, G_C\}$ are said to form a weak merge situation between t_F and t_C if

$$(\# \searrow_{b-b} (G_{F1}) = 1) \wedge (\# \searrow_{b-b} (G_{F2}) = 1) \wedge (\# \swarrow_{b-b} (G_C) = 2) \wedge (G_C \searrow_{b-b} (G_{F1})) \wedge (G_C \searrow_{b-b} (G_{F2})) \quad (7.12)$$

Definition 7.10 (Weak split candidate)

$\{G_F, G_{C1}, G_{C2}\}$ are said to form a weak split situation between t_F and t_C if

$$(\# \searrow_{b-b} (G_F) = 2) \wedge (\# \swarrow_{b-b} (G_{C1}) = 1) \wedge (\# \swarrow_{b-b} (G_{C2}) = 1) \wedge (G_F \swarrow_{b-b} (G_{C1})) \wedge (G_F \swarrow_{b-b} (G_{C2})) \quad (7.13)$$

Definition 7.11 (Weak creation candidate)

$\{G_C\}$ is said to form a weak creation situation between t_F and t_C if

$$(\# \swarrow_{b-b} (G_C) = 0) \quad (7.14)$$

7.2.3.3 Strong Conditions for Bifurcation Situations

In these statements there are, however, a lot of available information that we have not made use of. We have not taken the locations of the extremum points into account and not the relations between delimiting saddle points that hold in split and merge situations. Therefore it is natural to define the following (below $S_{delimit}(G)$ denotes the delimiting saddle point of a grey-level blob G , see Definition 6.3, and $nonshared(S)$ means that the saddle point S is non-shared, see Definition 6.7):

Definition 7.12 (Strong link candidate)

$\{G_F, G_C\}$ are said to form a strong link situation between t_F and t_C if they form a weak link situation between t_F and t_C and in addition

$$(G_C \searrow_{e-b} (G_F)) \wedge (G_C \swarrow_{e-b} (G_F)) \wedge (nonshared(S_{delimit}(G_F))) \wedge (nonshared(S_{delimit}(G_C))) \quad (7.15)$$

Definition 7.13 (Strong merge candidate)

$\{G_{F1}, G_{F2}, G_C\}$ are said to form a strong merge situation if between t_F and t_C they form a weak merge situation between t_F and t_C and in addition

$$(G_C \searrow_{e-b} (G_{F1})) \wedge (G_C \searrow_{e-b} (G_{F2})) \wedge (S_{delimit}(G_{F1})S_{delimit}(G_{F2})) \quad (7.16)$$

Definition 7.14 (Strong split candidate)

$\{G_F, G_{C1}, G_{C2}\}$ are said to form a strong split situation between t_F and t_C if they form a weak split situation between t_F and t_C and in addition

$$(G_F \swarrow_{e-b} (G_{C1})) \wedge (G_F \swarrow_{e-b} (G_{C2})) \wedge (S_{delimit}(G_{C1})S_{delimit}(G_{C2})) \quad (7.17)$$

Note that one cannot in general require that the extremum point of the coarser scale blob involved in a blob merge should necessarily belong to some of the blobs at the finer scale or that any corresponding relation should hold in a blob split. When formalizing the matching criterion for pairs of double candidates described in Figure 7.7 we get:

Definition 7.15 (Strong double link candidate)

$\{G_{F1}, G_{F2}, G_{C1}, G_{C2}\}$ are said to form a strong double link situation between t_F and t_C if

$$\begin{aligned}
& (\# \nwarrow_{b-b} (G_{F1}) = 2) \wedge (\# \nwarrow_{b-b} (G_{F2}) = 2) \wedge \\
& (\# \swarrow_{b-b} (G_{C1}) = 2) \wedge (\# \swarrow_{b-b} (G_{C2}) = 2) \wedge \\
& (G_{C1} \nwarrow_{b-b} (G_{F1})) \wedge (G_{C1} \nwarrow_{b-b} (G_{F2})) \wedge \\
& (G_{C2} \nwarrow_{b-b} (G_{F1})) \wedge (G_{C2} \nwarrow_{b-b} (G_{F2})) \wedge \\
& (G_{C1} \nearrow_{e-b} (G_{F1})) \wedge (G_{C2} \nearrow_{e-b} (G_{F2})) \wedge \\
& (G_{F1} \swarrow_{e-b} (G_{C1})) \wedge (G_{F2} \swarrow_{e-b} (G_{C2})) \wedge \\
& (S_{delimit}(G_{F1}) = S_{delimit}(G_{F2})) \wedge (S_{delimit}(G_{C1}) = S_{delimit}(G_{C2})) \quad (7.18)
\end{aligned}$$

When this condition is satisfied G_{C1} will be regarded as belonging to the same scale-space blob as G_{F1} and G_{C2} is regarded as belonging to the same scale-space blob as G_{F2} .

To express similar stronger conditions for blob annihilations and blob creations is not as easy, since in this case we have to ensure that we have not failed to find any relevant matching candidate that should have been registered. Of course, one could require that the delimiting saddle point (see Section 6.1.3) in such a situation should be non-shared. But such a condition will be far from sufficient.

7.2.3.4 Extended Neighbourhood Search

The conditions mentioned so far will however not be sufficient when tracking blobs covering just a small number of pixels. For example, the drift of a blob with an area of say one pixel will be impossible to capture with the previously outlined criteria, unless some additional gathering of matching candidates is carried out. This means that a situation that should have been registered as a plain link can give rise to one annihilation and one creation unless some additional precautions are taken. Therefore, in our current implementation, we perform an extended neighbourhood search in a region (of width one pixel) around every point involved in a weak creation situation. The purpose is to investigate if there are other blobs near it, which are involved weak annihilation situations. A blob creation is accepted only if no such blobs can be found and if, in addition, the same conclusion holds through a small number of refinements.

Another possible way of improving the performance could be by analysing the variation of the volume and contrast of the grey-level blobs and compare with analytical results as those derived in Section 6.4.1. One could also use drift velocity estimates as those derived in Section 6.4.2.3 or build up a model of the motion of the extremum point as function of scale. No such methods have, however, been implemented.

7.2.3.5 Bidirectional Matching

Let us point that in contrast to many matching algorithms in e.g. motion analysis, where the matching is performed only in one direction, that is with increasing time, this matching procedure, in its current form, is purely geometric and bidirectional. The matching candidates are always registered from both directions. Therefore, the scheme can *equivalently* be started either at a fine scale or at a coarse scale. The first approach can be advantageous if the scale-smoothing is implemented as cascade smoothing. The second approach could on the other hand have advantages when focusing the attention, that is when zooming in to a

particular object, since significant image features are more often found at the coarser scales than at the very finest levels of scale.

7.2.4 Delimiting the Refinement Depth

The decomposition property, meaning that relations between grey-level blobs at different levels of scale can be resolved into relations of the five primitive types (shown in Figure 7.5), is guaranteed to hold only for generic signals. Therefore in order to avoid a possible infinite number of refinements in situations when the algorithm is presented with a degenerate signal, it is necessary to introduce an upper bound on the number of refinements allowed to take place. If this number is reached, then a *complex* bifurcation will be registered by the algorithm. Although we have not yet found this situation to occur in any realistic images we have sometimes seen it happening for highly regular and noise-free *synthetic* data.

7.2.5 Scale Levels and Computation of the Refinement Scale

The algorithm is initiated with a relatively fine sampling along the scale direction corresponding to about $\frac{1}{3} - \frac{1}{2}$ octave in t at coarse levels of scale, distributed such that the scale step measured in effective scale is approximately constant. The maximum scale is determined from the size of the image (the outer scale) and the minimum scale is set to a low value³ (the inner scale). When refinements are needed, the refinement scale is computed from the existing scale levels t_1 and t_2 based on the notion of effective scale

$$t_{refine}(t_1, t_2) = \tau^{-1}\left(\frac{\tau(t_1) + \tau(t_2)}{2}\right) \quad (7.19)$$

where τ denotes the transformation function from the ordinary scale parameter to the effective scale parameter and τ^{-1} its inverse. The function values are computed from interpolation in a table with simulation data accumulated from point noise images, compare with Section 5.5.2.2.

7.2.6 Basic Blob Linking Algorithm

To summarize, an algorithm for linking grey-level blobs across scales into scale-space blobs can be based on the following steps. In the treatment below we will base the matching on the weak matching relations only in order to illustrate the idea. It should be obvious how the strong criteria can be incorporated in an analog manner.

1. Determine an initial set of scale levels, from some minimum scale value t_{min} , given by the inner scale of the image, to some maximum value t_{max} , which is given by the outer scale of the image. Distribute the intermediate scale levels such that the scale step, measured in effective scale is approximately constant. At coarse scales this means that the ratio between successive scale values will be about constant. At fine scales instead the differences between successive scale values will be approximately equal. Push these scales onto a stack of scale levels to be processed later.
2. Extract the grey-level blobs from the image at the finest scale using the grey-level blob detection algorithm.

³This scale value may be zero, but because of computational aspects it might be practical to use a higher value. During our experiments we have consequently let it be either 0, 1 or 2.

3. Get the next scale-level from the stack of scale levels.

- (a) Extract the blobs at the current level of scale.
- (b) For each grey-level blob at the current scale level, determine how many matching candidates it finds at the previous scale level. Similarly, for each grey-level blob at the previous scale level determine how many matching candidates it finds at the current scale level.
- (c) If some grey-level blob has more than two match candidates then the matching is non-trivial. Similarly, if there is a pair of double candidates⁴, i.e., if there is a blob having two matching candidates and one of its matching candidates in its turn also has two possible match candidates, then the matching is also difficult. In these cases perform a refinement, i.e.,
 - i. push the current scale level into the set of scale levels to be calculated.
 - ii. compute a refinement scale between the current scale level and the previous one.
 - iii. Continue with Step 3.
- (d) Else, if some grey-level blob at the coarser level does not find a match candidate at the finer level then the situation is more complicated. According to the scale-space theory this situation may in fact occur (but not very often). There could also be some other natural explanations why we may fail to find match candidates:
 - i. The blob may have moved outside the spatial region it covered at the previous level of scale. This phenomenon applies mostly to blobs with small areas — particularly blobs consisting of one pixel only. Such blobs will always be lost when moving if the matching is based on common pixels only.
 - ii. Numerical errors may have violated the scale-space properties.

In this implementation we perform an extended search in a small neighbourhood (of distance 1) around the coarse-scale blob in order to gather more matching candidates. If exactly one such candidate has been found and if that blob has no other match candidates, then a blob match will be accepted and the two grey-level blobs will be linked into the same scale-space blob. Otherwise, a refinement will be performed. However, if the refinement depth is too deep then a blob creation will instead be registered.

- (e) Else, each blob has either one or two matching candidates, and the matching candidates will be accepted.
 - i. If a blob at the fine scale has exactly one match candidate at the coarse scale, and if that candidate in turn has exactly one match candidate to the fine scale then link the grey-level blobs into the same scale-space blob.
 - ii. If a grey-level blob at the finer level does not have a match candidate at the coarser level then register a blob annihilation.
 - iii. If a blob at the coarser level finds two match candidates at the finer level and if these blobs in turn have exactly one match candidate each at the coarser level then register a blob merge.

⁴Many situations of this type can, as indicated above, be resolved with an extended extremum-blob matching — especially at coarser levels of scale. Then the refinement step will not be necessary.

- iv. If a blob at the finer level finds two match candidates at the coarser level and if these blobs in turn have one match candidate each, then register a blob split.
4. Store the registered relations between grey-level blobs at different scales. Then traverse all the scale levels and compute the scale-space blob volume and the scale-space lifetime for each scale-space blob (see Section 7.3 for the details).

7.2.7 Continuous Scale Parameter v.s. Fixed Scale Sampling

Note that this refinement principle cannot be applied as easily in pyramid representations where the scale levels have been set in advance. Then there is a fixed scale step beyond which refinements are no longer possible. Often, the pyramid representations also imply quite a coarse sampling in scale (a factor of 2 or $\sqrt{2}$ in σ that is a factor of 4 or 2 in t), that makes the matching problem more difficult.

7.2.8 Blob Linking v.s. Extremum Linking

It should be stressed that the grey-level blobs are much easier to trace across scales than are local extrema. This is mainly because the blob concept associates a region with every local extremum point. If one instead would have based the scale-level analysis on local extrema only, then the matching problem would often be more difficult, since local extrema may move much faster than the blobs. Ambiguous situations could easily occur. Especially bifurcation situations would be harder to identify. If at some level of scale one has lost the track of a local extremum point, then it is hard to say if it is because the extremum point has moved much faster than expected, been annihilated or because it has merged with another extremum point. It is in this context the blob regions are important, since they give natural spatial regions in which there are no other local extrema. They also define natural regions, to search for blobs in, at the next level of scale, compare also with Observation 6.15.

7.3 Computing the Scale-Space Blob Volumes

Once the scale linking has been performed and the bifurcations have been registered, it is straightforward to compute the scale-space blob volumes. At first every grey-level blob volume, as computed by the grey-level blob detection algorithm, is transformed according to the relation

$$V_{eff}(t) = \begin{cases} 1 + V_{eff,prel} & \text{if } V_{eff,prel} \geq 0 \\ e^{V_{eff,prel}} & \text{otherwise} \end{cases} \quad (7.20)$$

where

$$V_{eff,prel}(t) = \frac{V(t) - V_m(t)}{V_\sigma(t)} \quad (7.21)$$

and $V_m(t)$ and $V_\sigma(t)$ denote interpolated values from the tables of the mean values and standard deviations of the grey-level blob volumes for point noise data. In order to reduce the sensitivity of these values for the actual scaling of the grey-level values in the image, the tabulated values of V_m are rescaled with a uniform scaling factor determined from a least squares fit between the experimental values and the tabulated values at the finest levels of scale. Given these normalized grey-level blob volumes, the scale-space blob volumes are computed from the trapezoid rule of integration using the effective scale as integration

variable. The scale of a bifurcation is localized to the mean value (computed in effective scale) of the nearest coarser and finer scale around the bifurcation. The grey-level blob volume at a bifurcation is set to zero for annihilations and creations as well as for the two smaller blobs involved in merges and splits. As grey-level blob volume for the larger blob involved in a blob merge or blob split is taken the value of the grey-level blob volume in the nearest scale level included in that scale-space blob.

7.4 Data Structure

In order to give a rough idea of what information can be available in a data structure representing the scale-space primal sketch, we have in Appendix A.6.3 briefly described what kinds of objects could be defined in an actual implementation of this concept and also what types of data can be stored in those.

7.5 Possible Improvements of the Algorithm

The main concern behind this implementation has been to compute the representation as accurately as possible. We have not focused very much upon the computational efficiency of the algorithm⁵, since the main objective with this work has been to investigate what type of information can be obtained once a representation of the proposed type has been computed. In this section we will briefly describe some obvious improvements that could be made in order to speed up the performance.

7.5.1 Local Refinements

Currently, the refinements are made globally. In other words when a difficult situation has been encountered, necessitating refinements, then all grey-level blobs at the involved scales are subject to this process. One of the main improvements that could be made to the algorithm would be by restricting those refinements to comprise only the grey-level blobs involved in the ambiguous situation. Then only a window instead of the entire image needs to be processed.

7.5.2 Drift Velocity Estimates

The success of the linking algorithm depends very much on the fact that it is initiated with a fine initial sampling in scale. In later work we hope to incorporate an estimate of the drift of the blobs in order to get further verification of the bifurcation situations and the blob matches, compare also with Section 7.2.3.4. Such an estimate could also give more precise information about how dense the scale sampling really needs to be, possibly implying that a fewer number of scale levels needs to be treated. Observe that some discrete aspects may have to be introduced if such an approach is taken, see also Section 6.6.

7.5.3 Approximate Description

As was said earlier, we have throughout this work tried to introduce as few computational and numerical errors as possible when computing the representation. However, it often

⁵Coarse estimates about the performance of the algorithm are given in Appendix A.6.1.

turns out that many of the situations leading to refinements, correspond to structures later on rejected as being non-significant. Therefore, it seems plausible that the performance could be improved if those refinements could be avoided. In other words, if just an approximate description could be computed. However, such approximations require extensive experimentations and have not carried out.

7.5.4 Subsampling at Coarser Scales

The representations at coarser levels of scale are highly redundant. Another approximation to make would be by subsampling the images at coarser levels of scales, as is done in pyramids, in order to reduce the number of pixel values that need to be computed. An important issue to consider if such an approach is taken is to ensure that “no severe discontinuities” are introduced in the scale direction, compare also with Section 3.6.5.

7.5.5 Other Normalization Methods

As indicated above, the subtraction by the mean value carried out when computing the normalized grey-level blob volumes is sensitive to the actual scaling of the data. The transformation performed by scaling the tabulated data with a constant determined from a least squares fit to the experimental data is intended to compensate for this phenomenon. A possible way to avoid this rescaling would be by computing the normalized grey-level blob volumes from

$$V_{eff}(t) = \frac{V(t)}{V_m(t)} \quad (7.22)$$

and then integrate those values into scale-space blob volumes. As significance values for comparisons across scales one could conceive taking the ratio

$$S_{eff}(t) = \frac{S(t) - S_m(t)}{S_\sigma(t)} \quad (7.23)$$

where S_m and S_σ denote mean values and standard deviations for *scale-space blobs* computed from point noise data (based on the same normalization (7.22)). As scale values for the scale-space blobs we could take their appropriate scale, that is the scale where they assume their maximum grey-level blob volumes, see Chapter 8 for further explanation. This method has not yet been implemented, mainly because the amount of simulation work required when building the tables is much larger.

Part III

The Scale-Space Primal Sketch: Applications

Chapter 8

Detecting Image Structures

8.1 Detecting Significant Image Structures and Their Scales

One motivation for this research was to investigate if the scale-space model really allows for determination and detection of global and stable phenomena. In this section we will demonstrate that this is indeed possible and that the proposed representation can be used for extraction of important regions from an image, in a solely data-driven way. The treatment is based on the *assumption* that:

- *Features, which are significant in scale-space, correspond to relevant details in the image.*

More precisely, since the primitives we intend to use are scale-space blobs we formulate the assumption as follows:

Assumption 8.1 (Significant image structure)

A scale-space blob having a large scale-space volume in scale-space corresponds to a relevant region in the image.

A scale-space blob will in general exist over some scale interval in scale-space. When there is a need to reduce the amount of data represented and to select an appropriate scale and a spatial region for a scale-space blob, we make use of the following postulates:

Assumption 8.2 (Scale selection)

The scale-level, at which a scale-space blob assumes its maximum grey-level blob volume, is a relevant scale for treating that individual blob.

Assumption 8.3 (Spatial representative)

The spatial extent of a scale-space blob can be represented by the blob support region corresponding to its grey-level blob at the relevant scale.

The ranking of events in order of significance depends on the actual scaling of the four coordinate axes in the scale-space representation. Therefore, the extraction method implicitly relies upon the assumption that it is sufficient to transform the coordinate axes once and for all as was done in Section 5.5, and that this normalization can be carried out based on the behaviour in scale-space of point noise signals.

Assumption 8.4 (Normalization)

The coordinate axes in the scale-space representation can be normalized based on the behaviour in scale-space of point noise data.

Below, we will give experimental results showing that these assumptions, combined with a careful computational treatment of the scale-space, really segment out perceptually relevant regions in the image.

8.2 Motivation for the Assumptions

A central issue in low-level vision concerns what should be meant with “image structure”. In other words, which features in an image can be regarded as important, and which ones should be rejected as noise. Notably, Lowe [Low85] defines structure based on non-accidentalness. However, such an approach requires a probabilistic model of the situation. It is well-known that it is difficult to find a statistical model generally valid for the image formation process.

8.2.1 Transformational Invariance: Structure

In this work we take an alternative viewpoint and suggest *a definition of structure based on features, which are stable with respect to (appropriately selected) transformations and/or parameter variations*. For this specific treatment the transformation family of interest is the semi-group of convolution transformations associated with the scale-space smoothing. The parameter we vary is the scale parameter. We think that features stable or invariant with respect to variations in scale can be regarded as significant. In more general situations one could also imagine the probing transformation as given by variations in viewing distance (focusing), spatial resolution, regularization parameters etc.

One can motivate such a standpoint by a pragmatic argument. If a feature is to be useful for recognition, it must necessarily be stable with respect to small disturbances. Otherwise it can hardly be practically useful, since then, it inherently cannot be computed accurately. This definition of structure in terms of *transformational invariance* also induces a straightforward and general method for detecting significant image features, namely by *subjecting the image to systematic parameter variations*. In line with that idea we believe that those features, that are the most stable ones during such a parameter variation process, can be regarded as strong candidates for being useful for later processing and possibly recognition.

The reverse statement does of course not hold. There are many other sources of information, i.e., lines in line-drawings, that are not captured by a blob concept and scale-space smoothing. In this work we focus mainly on one aspect of image structure, namely regions that are brighter or darker than the background and stand out from the surrounding.

Note, that this use of transformational invariance is different than to what is usually meant by invariance in the algebraic or geometric sense. Here, we consider invariance as stability over a finite interval, that is a limited range of parameter values, and define features based on this property.

8.2.2 Stability in Scale-Space: Perceptual Salience

The approach is closely related to Witkin’s [Wit83a] observation about correspondence between stability in scale-space and perceptual salience. However, here we base the stability measure on the scale-space blob volumes instead of the scale-space lifetime. The intention is that this choice also should reflect the size of the blobs and how strongly they manifest themselves with respect to the background. As mentioned in Section 5.4 we have observed that small blobs due to noise can survive over a large range of scales if they are located

in regions with slowly varying grey-level, which shows that scale-space lifetime alone is not sufficient as a significance measure.

8.2.3 Reduction of the Representation: Abstraction

Because of complexity arguments, the entire parameter variation information from the low-level modules cannot not be transferred to modules intended to perform higher-level processing tasks. Instead we think that low-level modules working after this paradigm should be able to extract stable intervals, and that it should suffice to determine a representative descriptor for each important stability region.

The second and third assumptions express such a desire to represent a scale-space blob with a grey-level blob at a single level of scale in order to give a more compressed representation, an abstraction, for further processing. We believe that a relevant scale of a scale-space blob should be a scale where the grey-level blob manifests itself “as its best”, i.e., it should be the scale level where the blob response “is maximally strong”. This selection method is similar to Marr’s, [Mar76], idea about the choice of an appropriate mask size for edge detection. Empirically we have found that this suggested scale value will give a good description of the situation. It turns out that it often will be close to the appearance scale of the scale-space blob, except at blob splits and blob creations, for which the grey-level blob volume at the appearance scale will be zero.

Worth noting is that Assumption 8.2 implies a projection from a four-dimensional scale-space blob to a three-dimensional scale-space blob and that Assumption 8.3 implies a projection from that three-dimensional grey-level blob to its two-dimensional blob support region.

8.3 Basic Extraction Method for Image Structures

The basic methodology, in our suggested algorithm for extraction of important image structure, should be obvious from the previous presentation.

- Generate the suggested multi-scale representation, where blobs are extracted at all levels of scale and linked across scales into scale-space blobs.
- Compute the scale-space volume for each scale-space blob based on the notion of effective scale and transformed grey-level blob volumes.
- For each scale-space blob determine the scale where it assumes its maximum grey-level blob volume, and extract the blob support region of the grey-level blob at that level of scale.
- Sort the scale-space blob in descending significance order, i.e., with respect to their scale-space blob volumes.

8.4 Experimental Results

In Figure 8.1, Figure 8.2 and Figure 8.3 we show the results of applying this procedure to three different images with toy blocks, a telephone and a calculator and a dot pattern. The reader is encouraged to study these images carefully.

For display purpose we have extracted the N dark scale-space blobs having the largest blob volumes. Each blob is represented at its representative scale, that is the previously

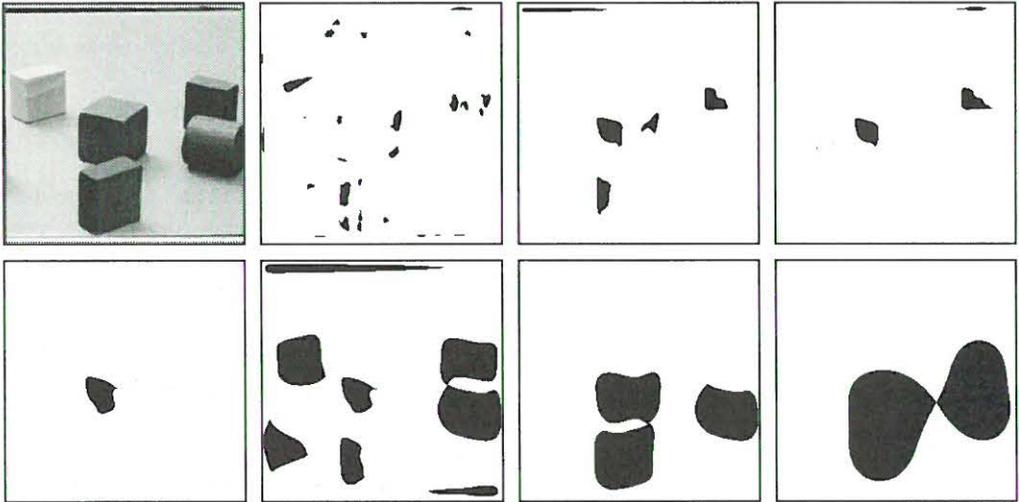


Figure 8.1: *The 50 most significant dark blobs from a toy block image. (Note how these images have been produced — they are not just blob images at a few levels of scale. Instead every blob has been marked at its representative scale. Finally, the blobs have been drawn in different images as to avoid overlap.)*

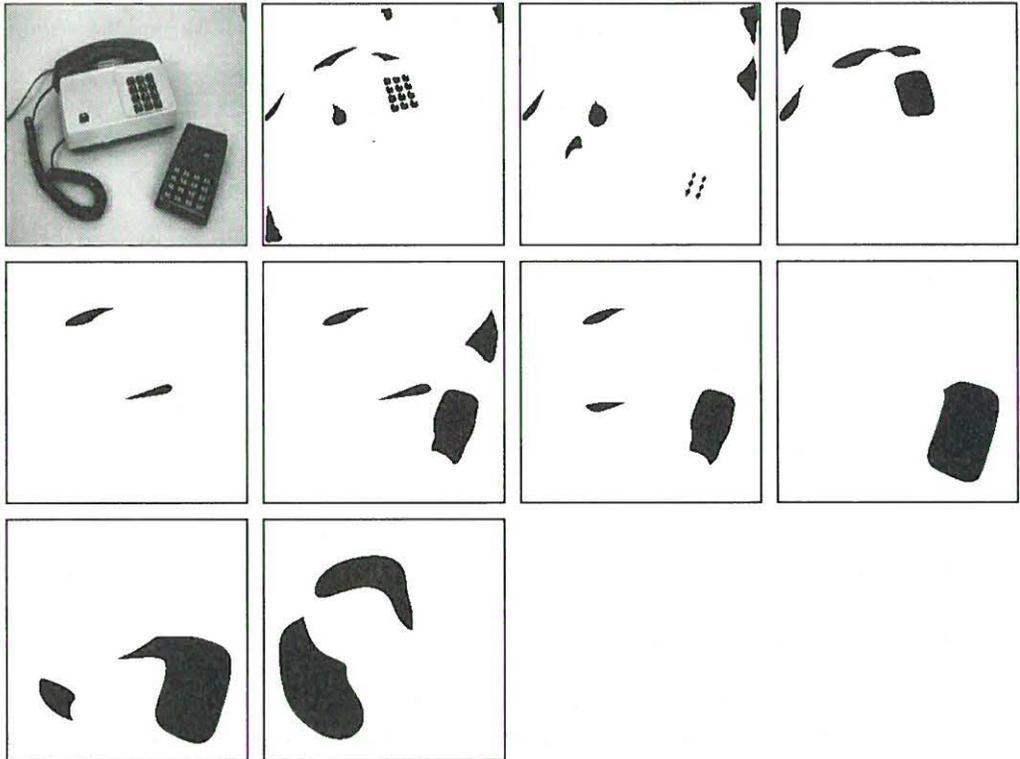


Figure 8.2: *The 50 most significant dark blobs from a telephone and calculator image.*

Blob label	Scale	Significance
1760	32.00	1450.55
1767	64.00	1266.43
1764	50.80	1030.53
1768	80.60	591.16
1770	812.90	297.60
1769	645.10	284.72
1761	45.25	150.64
1758	28.51	131.99
1763	45.25	73.69
1065	35.91	63.51
1759	28.51	35.92
1753	22.65	35.42
1703	8.00	20.45
1702	8.99	17.43
1723	11.99	12.84
1757	28.51	9.94
1256	4.00	6.84
1708	9.53	6.20
1725	14.25	5.33
1440	2.00	5.10
1471	1.40	4.85
1610	2.87	4.03
1731	16.00	3.84
1679	4.73	2.41
1078	1.00	2.30
1265	1.10	2.27
1713	10.10	2.21
1706	8.99	2.21
251	1.22	2.07
1072	1.00	2.02
1070	1.00	1.98
1187	1.00	1.96
1243	2.65	1.95
1686	5.00	1.95
1371	1.05	1.93
1611	6.40	1.87
1286	1.00	1.83
1183	1.00	1.83
1083	1.00	1.79
1336	1.00	1.75
1393	1.10	1.72
212	1.00	1.71

Table 8.1: Table over the relevant scales and significance values of the 40 most significant blobs obtained from the scale-space primal sketch representation of the toy block image. Note that a few blobs have significance values clearly standing out from the other ones.

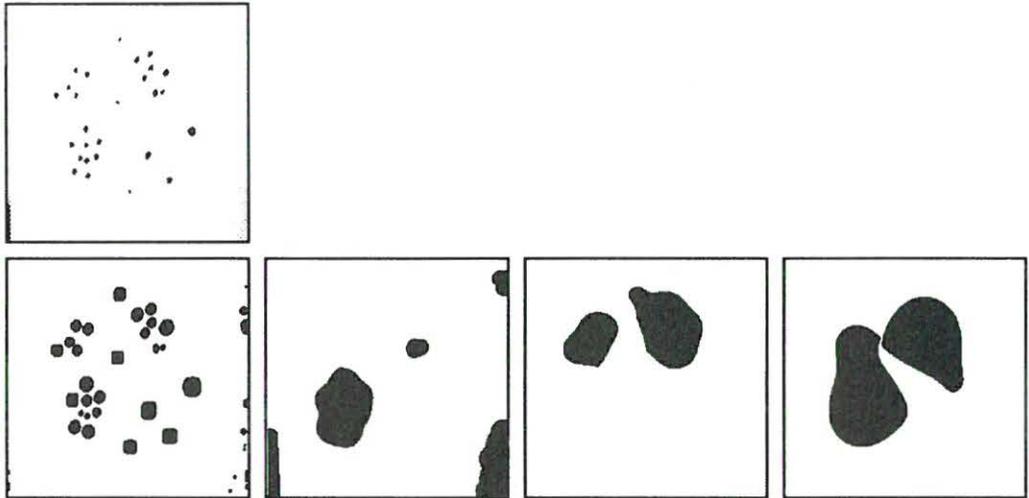


Figure 8.3: *The 50 most significant dark blobs from a dot pattern image.*

mentioned scale where the scale-space blob has its maximum grey-level blob volume. Finally, the spatial region of each blob (which is the blob support region at its representative scale) has been marked in a binary image, where black indicates the existence of a significant blob and white represents background. In order to avoid overlap in the display, we have shifted to a new fresh image each time the addition of a new blob would have implied overlap between two different blobs.

We can see that the blocks are extracted from the toy block scene. Also, at coarser scales, adjacent blocks become grouped into coarser scale units and the imperfections of the image acquisition near the boundaries are pointed out. In the telephone scene, the buttons, the keyboard, the calculator, the cord and the receiver are detected as single units. Finally, in the dot pattern image the algorithm finds at first all the dots and then also performs those groupings we find perceptually reasonable.

In order to show the spatial relations between the blobs at the various levels of scale we have also drawn the blob boundaries for the images from the previous examples in Figures 8.4-8.5.

Let us conclude by stressing that we *extract the intrinsic shape of the grey-level landscape in a completely bottom-up data-driven way without any assumptions about the shape of the primitives* (except for the fact that the scale-space smoothing favours blob-like objects, since it is equivalent to correlation with a Gaussian-shaped kernel).

We get a segmentation that is coarse in the sense that the localization of object boundaries may be poor, due to the natural distortions of shape which occur in scale-space. However, the segmentation is safe in the sense that those regions, which are given by the scale-space blobs with large scale-space volume, really serve as landmarks of significant structure in the image, with information about

- the *approximate location and extent* of relevant regions in the image.
- an *appropriate scale* for treating those regions.

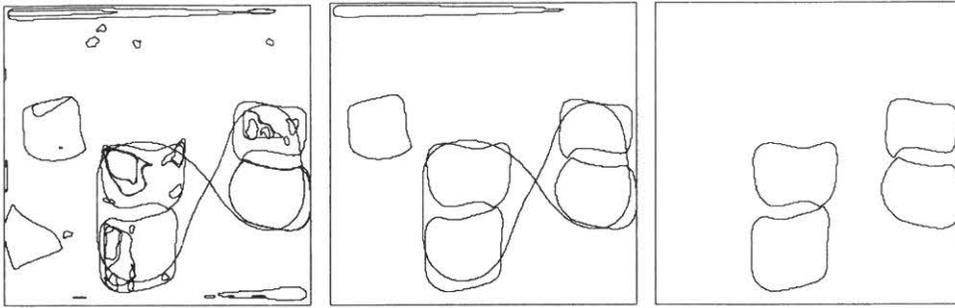


Figure 8.4: Boundaries of the dark blobs extracted from the toy block image. (a) The 50 most significant dark blobs. (b) Low threshold on the significance measure set in one of the “gaps” (between 74 and 131) in the sequence of significance values. (c) High threshold on the significance measure set in another “gap” (between 298 and 591). (The significance values are shown in Table 8.1).

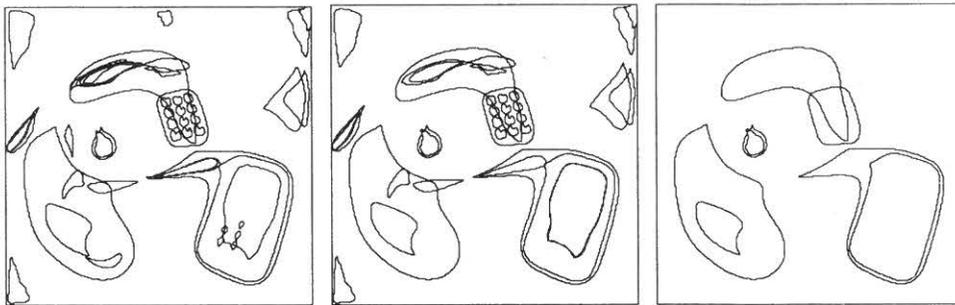


Figure 8.5: Boundaries of the dark blobs extracted from the telephone and calculator image. (a) The 50 most significant dark blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.

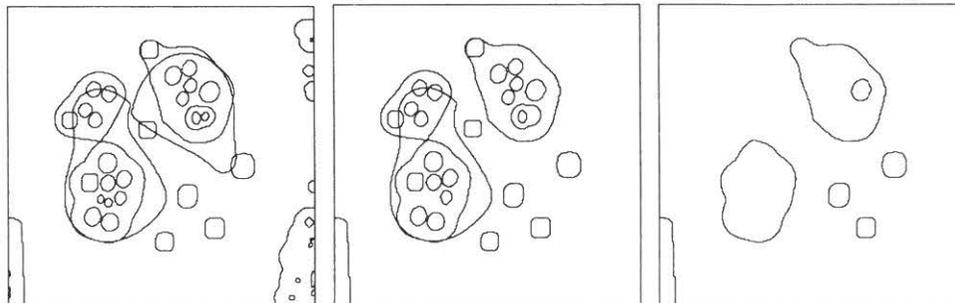


Figure 8.6: Boundaries of the dark blobs extracted from the dot pattern image. (a) The 50 most significant dark blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.

This is exactly the kind of coarse information¹ that is necessary for many higher-level processes, see e.g. the application to edge detection in Section 9.1.

8.5 Further Treatment of the Generated Blob Hypotheses

The number of scale-space blobs selected for display above is, of course, rather arbitrary. However, note that there is a well-defined ranking between the blobs. If one studies the significance values of the blobs, (see Table 8.1 regarding the toy block image), one can observe that those blobs we regard as the most significant ones have significance values standing out from the significance values of the other ones.

Hence, it seems plausible that a few image regions could be extracted just from the criterion that their significance values should stand out from the significance values of the other ones. In more general situations there is a need for feed-back or reasoning.

The output information from this algorithm should not be over-estimated. Since it is a low-level processing module, the output results should be interpreted as such, namely *as indicators signalling that "there might be something there of about that size — now some other module should take a closer look"*. From this viewpoint it can be noted how well the extracted blobs describe the images in the previous examples, considering that the blobs have been extracted almost without *any* a priori information.

In principle we think that a reasoning process, working on the output from the scale-space primal sketch, could operate in either of two possible modes:

1. Use a threshold on the significance measure. In a real system such a threshold could in some applications be set from given context information and expectations.
2. Evaluate the generated hypothesis in decreasing order of significance, i.e., first try to interpret the first hypothesis in a feed-back loop. Then consider the second one etc. Continue as long as the hypotheses deliver meaningful interpretations for the higher-level modules.

Note also that the output from the scale-space primal sketch can work both in a static and a dynamic mode. Consequently, we believe that it can really serve as a guide to the focus-of-attention. In Chapter 9 will show how such integration of this kind of information with later stage processing modules can be done.

Another inherent property with this representation is that it does not have any limiting requirement that there is just one possible interpretation of a situation. Instead it generates a variety of hypotheses. Given some region in space, several hypotheses may be active for it (or parts of it) concerning structures at different levels of scale.

8.6 About the Selection of Appropriate Scale

In this section we will now describe some consequences of the suggested definition of appropriate scale of a scale-space blob. The presentation to be given is not intended to be theoretically rigorous in any way, but rather to convey an intuitive understanding for what qualitative properties the stated assumption leads to.

¹The scale-space primal sketch contains much more information than is presented in this rudimentary output. For instance, we have not illustrated the registered blob bifurcations in scale-space. Nor have we shown or made use of the hierarchical relations between blobs at different levels of scale induced by the blob events. This information is however explicit in the computed representation.

8.6.1 Relations between Appropriate Scale and Object Size

The scale value given by Assumption 8.2 does not necessarily reflect the size of the blob region in the image. Although large values of the scale parameter in general will lead to images with large size features, there is no *direct* relation between the size of an object and its associated appropriate scale. In certain situations large size objects may in fact be assigned relatively small values of t as their appropriate scale (although the opposite situation can be expected not to occur). The scale value given by Assumption 8.2 should therefore rather be interpreted as an *abstract scale parameter* or as giving *the smallest amount of smoothing for which a region in the image manifests itself as a single blob entity*.

Consider for example an image with, say, a few squares of fixed size. The scale value, where for the first time one of the squares appears as a blob, can vary substantially depending on the noise level in the image and on where the squares are located relative to each other. In the ideal noise-free and texture-free case, i.e. when there are no interfering fine-scale structures present, the appropriate scale for each one of the squares will be zero. Only for coarse scale structures, which only exist as groupings of other primitive fine scale structures, will the appropriate scale be non-zero in the ideal noise-free case. (For example, a letter formed by arranging the blocks in a certain pattern with some spacing between them).

8.6.2 Partial Ordering

Hence, these scale values do not induce any total ordering of regions with respect to their relative size, but rather a *partial ordering*. By and large the following property holds: If two structures overlap, i.e., if a fine scale structure is superimposed onto a coarser scale structure, then the coarser scale structure will be given a greater scale value than its superimposed fine scale structure. On the other hand, if similar structures are located sufficiently far apart from each other in an image then the reverse relation may actually hold.

However, the situation is even further complicated. At blob splits, the blob existing after the bifurcation will be larger than the blobs existing before the bifurcation. Therefore, the scale values² given by Assumption 8.2 give useful information about the relative size³ of two objects only when the objects overlap and in addition they can be related to each other through a series of bifurcations free from blob splits and blob creations.

8.6.3 Several Instances of a Region

As can be seen e.g. with the calculator in Figure 8.2 it may happen that given some region in the image, *several instances of blobs* can be detected corresponding to that region. This is a common phenomenon in the scale-space primal sketch, arising because a large (significant) blob merges with a small (insignificant) blob and thus forms a new scale-space blob. From the definition of a scale-space blob, we have that it is delimited by two

²However, the scale interval between the appearance scale and the disappearance scale, during which an object exists, should be applicable for such determination and it seems plausible that there should exist some scale level within the scale interval that could be mapped to the size of the object, at least for regions of relatively round shape. The appearance scale of a blob is mainly determined by the interaction between the blob and interfering finer scale structures. When no fine-scale structures are present, the appearance scale will be zero. Similarly, the disappearance scale of a blob is determined by the interference between the blob and the coarser scale structures in the surrounding. When no coarser scale structures are present, the disappearance scale will be infinite.

³On the other hand it is not even clear that it is desirable to use the scale values for size comparisons, since the size of a region can be easily estimated from the size of its blob support region.

scales where bifurcations occur. This means that every time a bifurcation takes place, the (involved) grey-level blobs existing before the bifurcation will be treated as belonging to different scale-space blobs than the (involved) grey-level blobs existing after the bifurcation.

8.6.4 Number of Layers: Complexity

Observe that the number of layers in the output gives a coarse measure on the complexity of an image. A relatively simple image will in general give rise to fewer layers than a complex one. In the case when all significant blobs of an image can be drawn in the same layer without overlap, it is natural to say that to every point in the image there corresponds just one stable scale, or shorter, that there is only one scale level in the image, (even though the actual value of the scale parameter may vary substantially between different blobs). Similarly, the keyboard of the telephone can be said to have two scales — one for the set of buttons and one for the keyboard as a whole.

8.7 Additional Experiments

In order to further demonstrate the properties of the scale-space primal sketch and the suggested way to extract image structures from this representation, we give some more experimental results⁴ in Figures 8.8-8.27. See also Section 9.4 for an application to the analysis of aerosol images and Section 9.5 for examples with textures and medical data.

Figure 8.8 and Figure 8.9 show an indoor table scene and the 50 most significant bright and dark blobs extracted from the grey-level image. In Figures 8.10-8.13 we display the boundaries of these blobs and also the results of superimposing the blob boundaries onto the original image. In order to give a rough idea of the significance values, we have manually set different thresholds in “gaps” in the sequences of significance values. One can observe that in this scene, most of the meaningful objects are brighter than the background and that those objects are found. In addition, the bright blobs respond also to illumination phenomena on the table, in the background as well as to specularities. The detected dark blobs correspond to the two background regions and various shadows due to the objects on the table.

For one object, the curved pipe in the right part of the image, only the specularity on its surface is detected. The object fails to stand out as a single blob unit. This illustrates a characteristic property of the representation, namely that a region, which borders upon both a darker region and a brighter region, cannot be expected to be detected as a single blob region by this method. According to the blob definition (compare with Figure 5.1 and Figure 5.6), only regions that are either brighter or darker than their background will be treated as “blobs”, see Figure 8.7. In order to be able to detect regions also of this latter type it seems necessary to include more information into the analysis. This issue will be considered in further work.

Figures 8.14-8.17 show similar results from a scattered office scene, where most of the important objects are darker than their background. One can see that the handle of the

⁴All these experiments have been performed with images of size 256×256 pixels. The scale-space convolutions were carried out with floating point calculations and the image boundaries were treated in the following way: When detecting *bright* blobs on dark background the image was extended using its *minimum* image value. Conversely, when looking for *dark* blobs on bright background the image was extended with its *maximum* grey-level value. The infinite support convolution kernel (the discrete analogue of the Gaussian kernel) was truncated at the tails such that the truncation error ϵ was guaranteed not to exceed 0.0005.

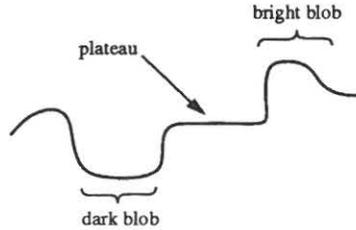


Figure 8.7: According to the definition of grey-level blob, only regions that are either brighter or darker than their background will be classified as blobs. A region, which borders upon both a darker and a brighter region, does not satisfy the blob definition, which means that the plateau in the figure will not be detected as one unit by the algorithm. In order to extract such regions it seems necessary to include some kind of gradient information into the analysis.

hammer, the heap of screws, the black tape reel, the label of the hammer and some other dark regions are all detected as dark blobs. The grey tape reel is not detected as a blob, since it bounds upon both a region that is darker and a region that is brighter. One can observe that the blob corresponding to the handle of the hammer spreads relatively far from the boundary of the actual object. This phenomenon occurs for isolated objects far away from other competing blobs of the same polarity. However, this effect does not imply any severe problems and can be easily compensated for, for instance when matching blobs to edges, see Section 9.1.2.5 for a description. As bright blobs we find the holes in the two tape reels as well as various regions on the table.

Figure 8.18 and Figure 8.20 display the extracted dark and bright blobs from an outdoor image of a house, a scene where there are both dark and bright objects with meaningful interpretation. Figure 8.19 and Figure 8.21 show the boundaries of these blobs. One can observe that the windows of the house are detected as well as various parts of the wall, the sky and parts of the tree.

Finally, as a test of the stability of the representation, Figures 8.22-8.25 display the results from another image of the same telephone and calculator as in Figure 8.2, where we have changed the background to a textured piece of cloth and also moved the camera and the objects in the scene. One can observe that the important regions (receiver, cord, keyboard, buttons, calculator) are still being found. The bright blobs respond to the telephone, the hole in the cord, other regions delimited by dark objects, various illumination phenomena in the background, the bright buttons of the calculator and, actually, the regions between the buttons of the telephone.

For comparison, corresponding results for the original telephone and calculator image are shown in Figures 8.26-8.27. One observes that similar types of regions are extracted in the two cases, both for the dark and the bright blobs, although the interference effects for the coarse scale blobs are different, mainly because the distance between the telephone and the calculator has been changed.

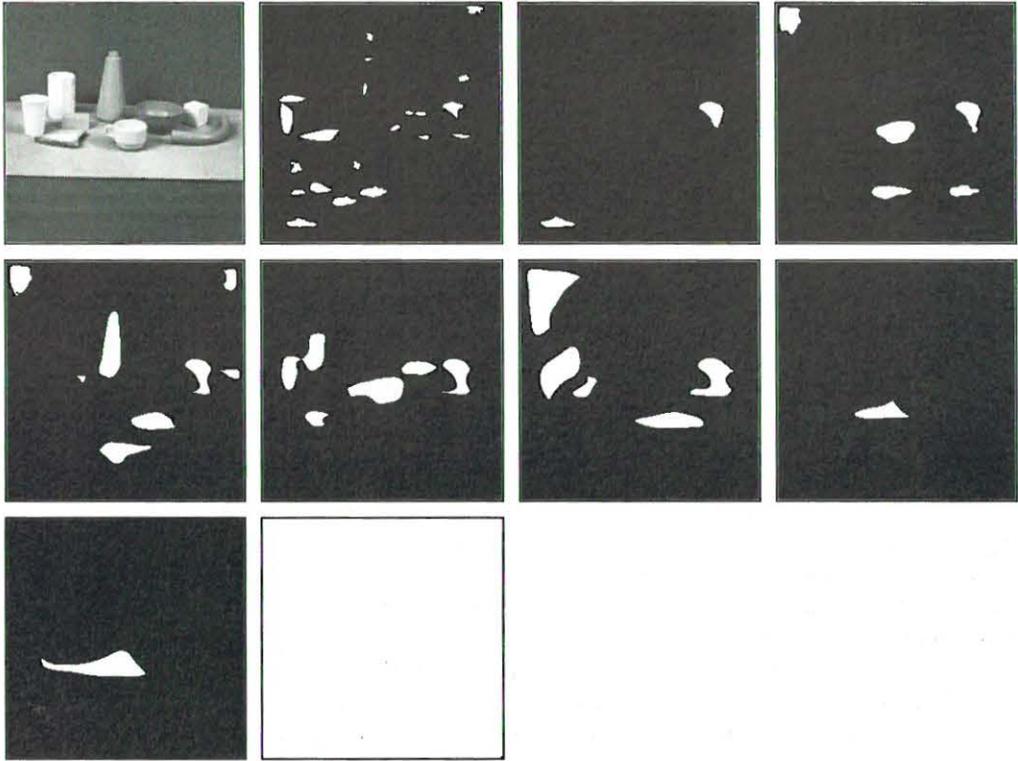


Figure 8.8: *The 50 most significant bright blobs from a table scene. (In the last case the whole image has been classified as one blob).*

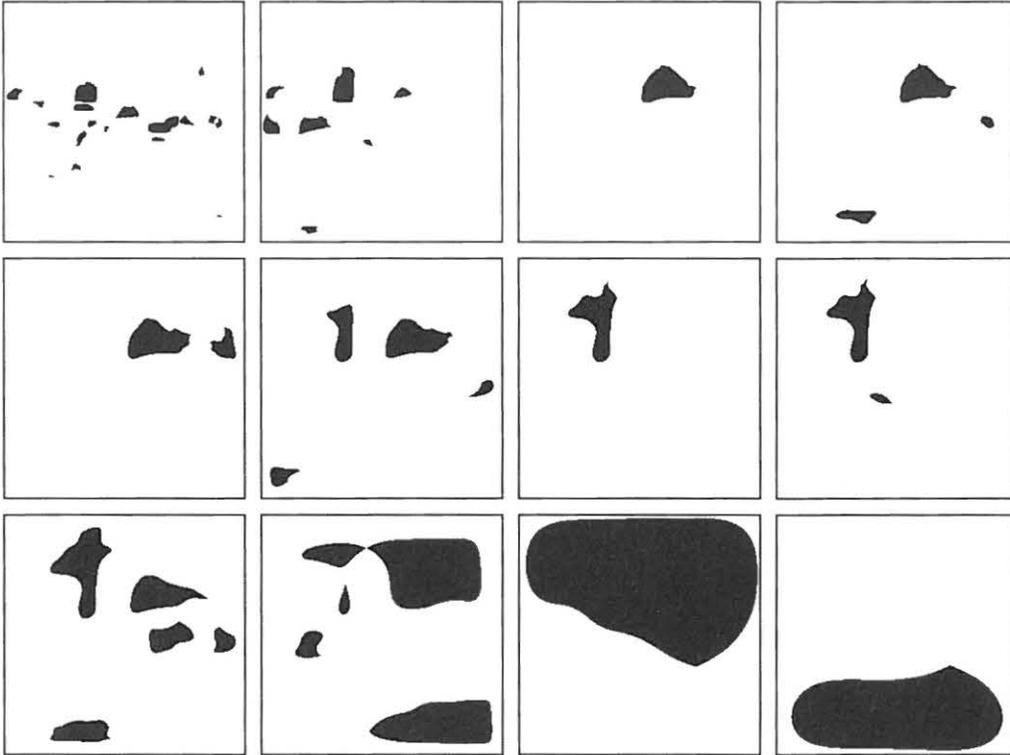


Figure 8.9: *The 50 most significant dark blobs from a table scene.*

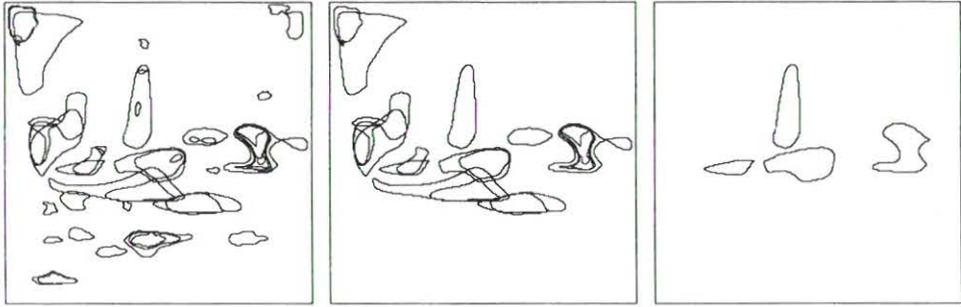


Figure 8.10: Boundaries of the bright blobs extracted from the table scene. (a) The 50 most significant bright blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.

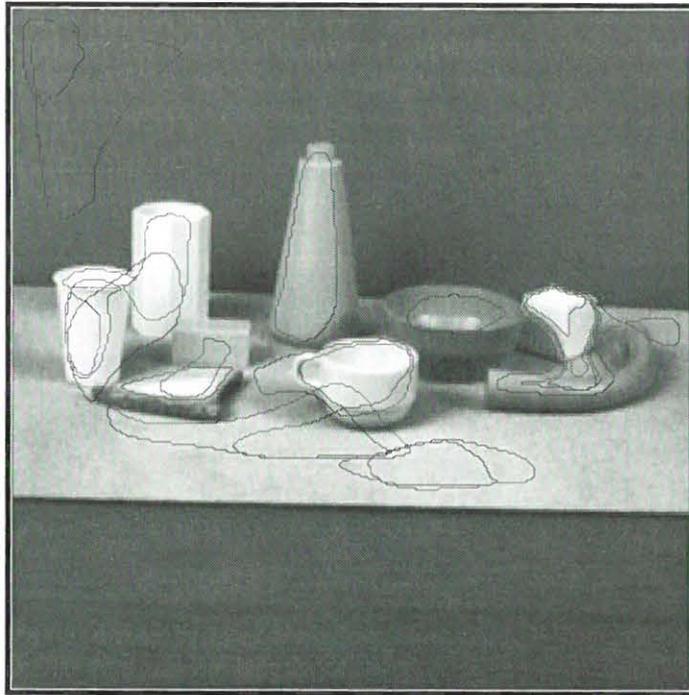


Figure 8.11: Boundaries of the extracted bright blobs superimposed onto the original grey-level image. The lower threshold has been used on the significance values.



Figure 8.12: Boundaries of the dark blobs extracted from the table scene. (a) The 50 most significant dark blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.

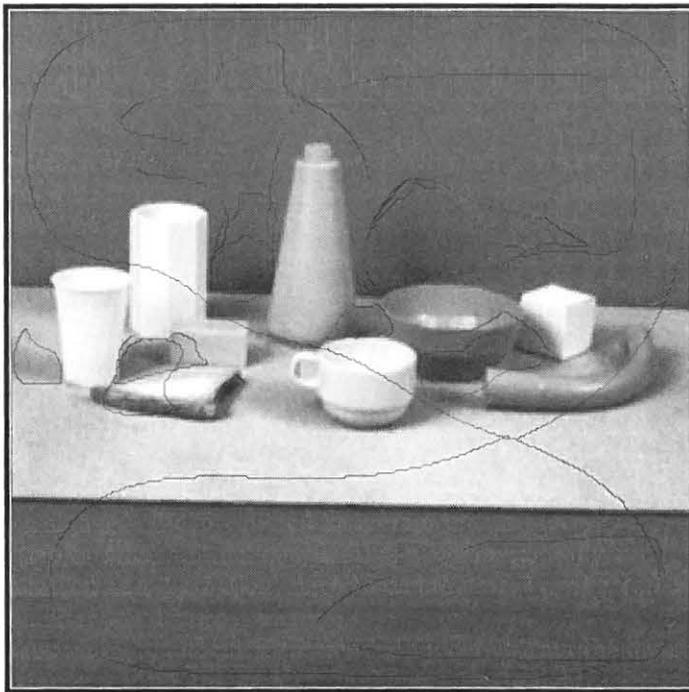


Figure 8.13: Boundaries of the extracted dark blobs superimposed onto the original grey-level image. The lower threshold has been used on the significance values.

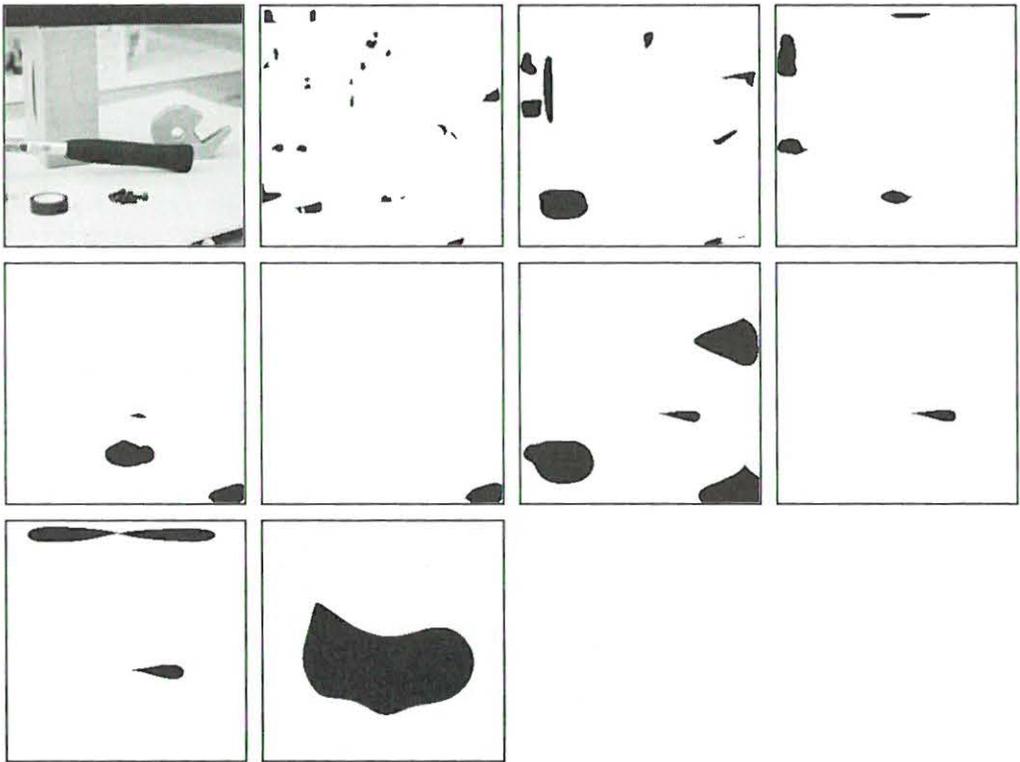


Figure 8.14: *The 50 most significant dark blobs from a scattered office scene.*



Figure 8.15: *Boundaries of the dark blobs extracted from the scattered office scene. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the "gaps" in the sequence of significance values. (c) High threshold on the significance measure set in another "gap".*

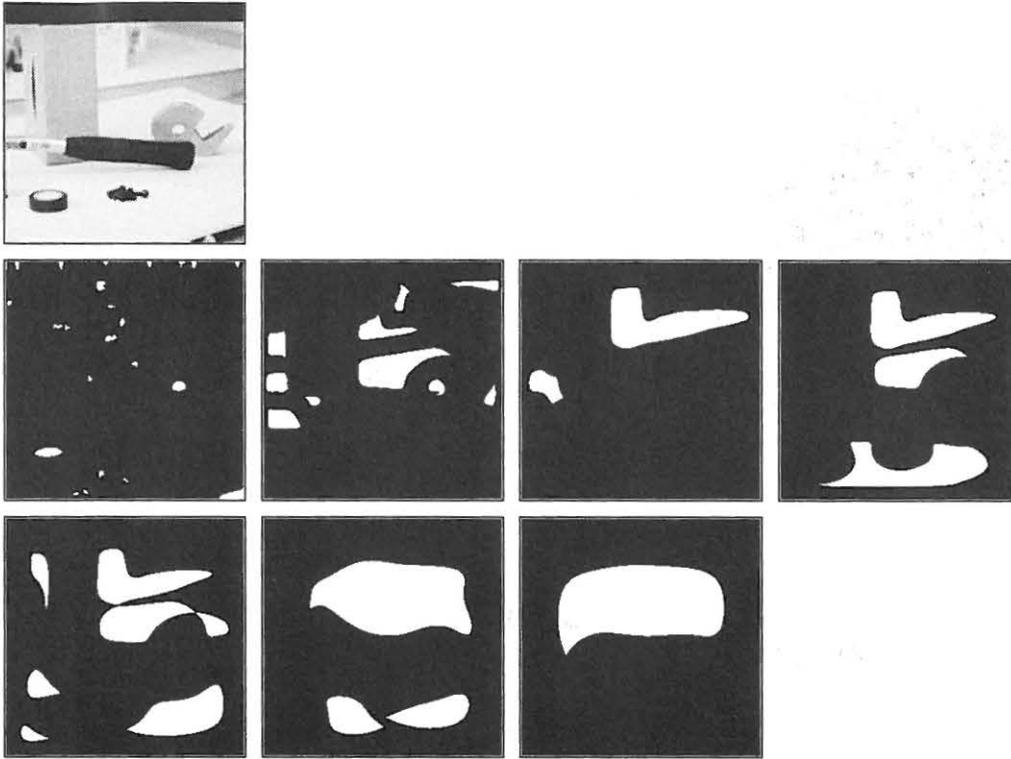


Figure 8.16: *The 50 most significant bright blobs from a scattered office scene.*

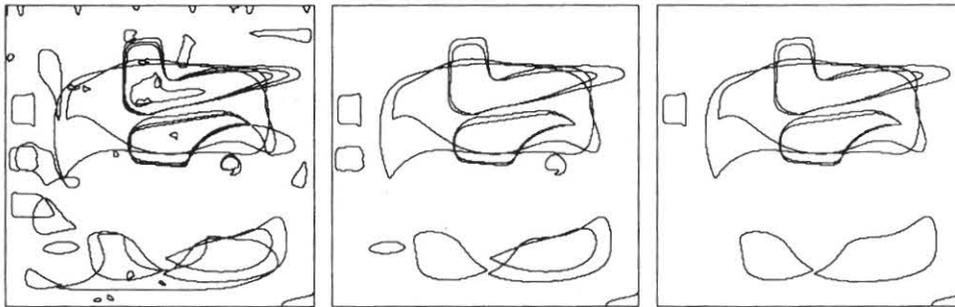


Figure 8.17: *Boundaries of the bright blobs extracted from the scattered office scene. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.*

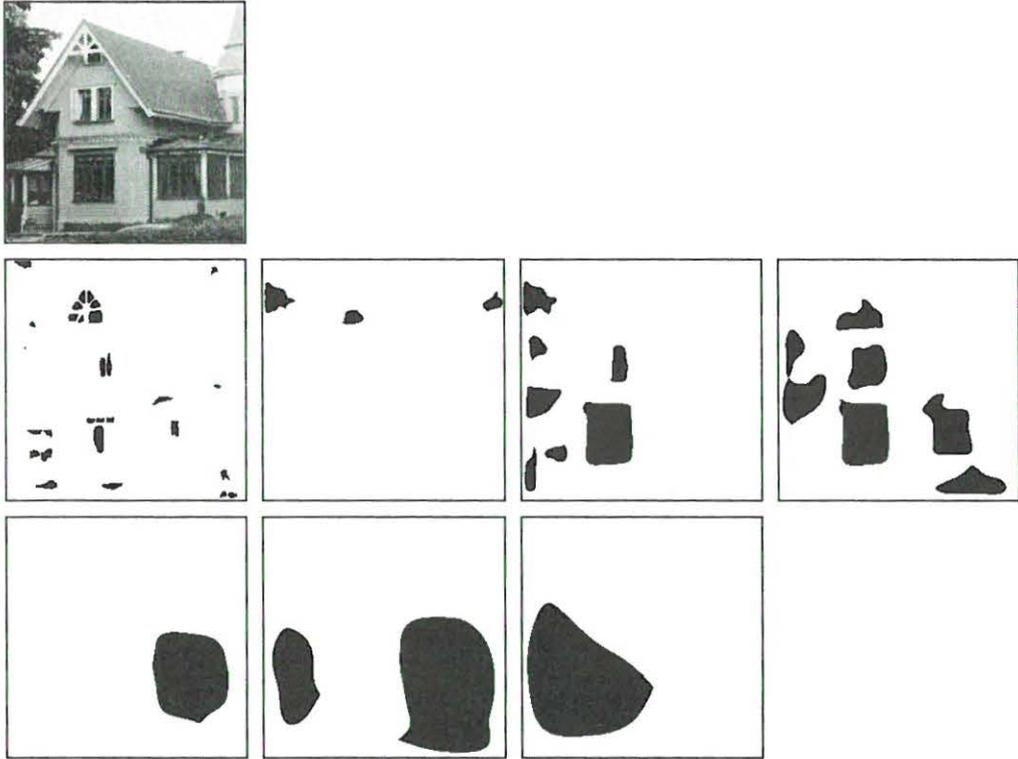


Figure 8.18: *The 50 most significant dark blobs from an image of the Godthem Inn at Djurgården, Stockholm.*

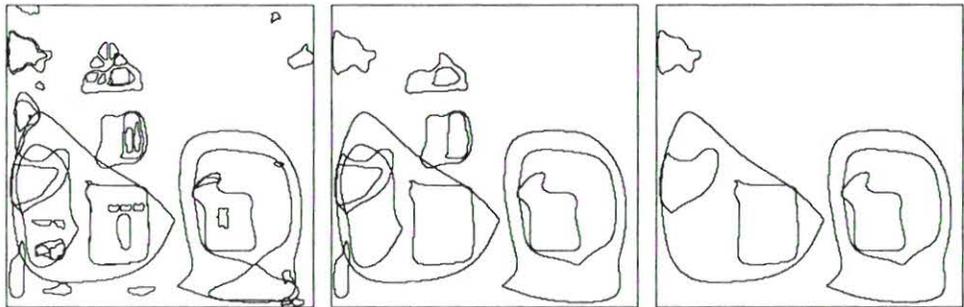


Figure 8.19: *Boundaries of the dark blobs extracted from the Godthem Inn image. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.*

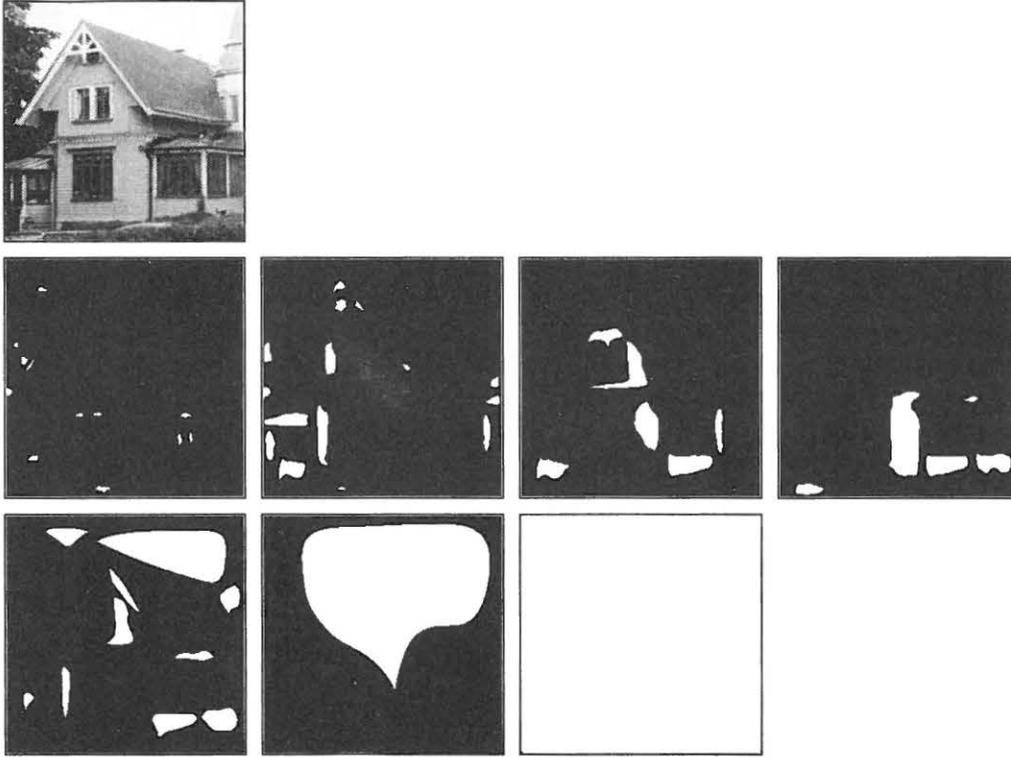


Figure 8.20: *The 50 most significant bright blobs from an image of the Godthem Inn at Djurgården, Stockholm.*

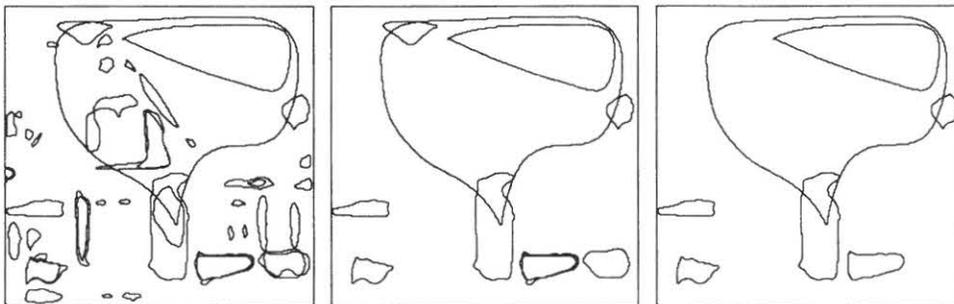


Figure 8.21: *Boundaries of the bright blobs extracted from the Godthem Inn image. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.*

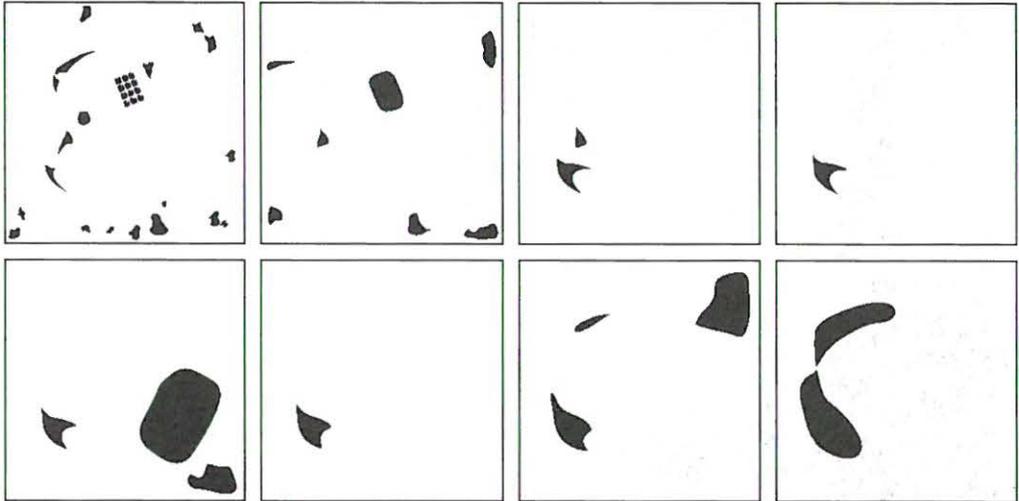


Figure 8.22: The 50 most significant dark blobs from a telephone and calculator image. The background is textured.

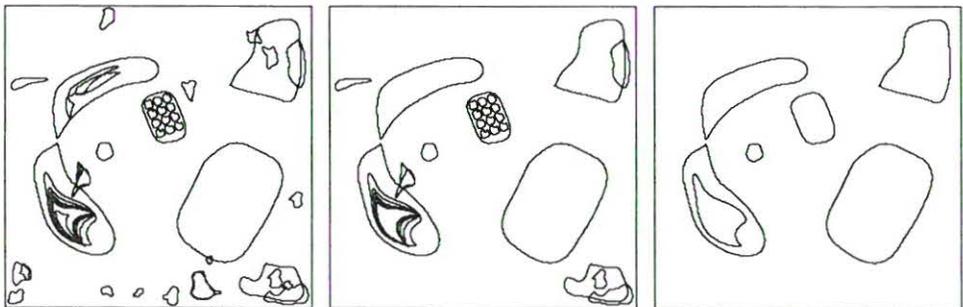


Figure 8.23: Boundaries of the dark blobs extracted from the telephone and calculator image with textured background. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.

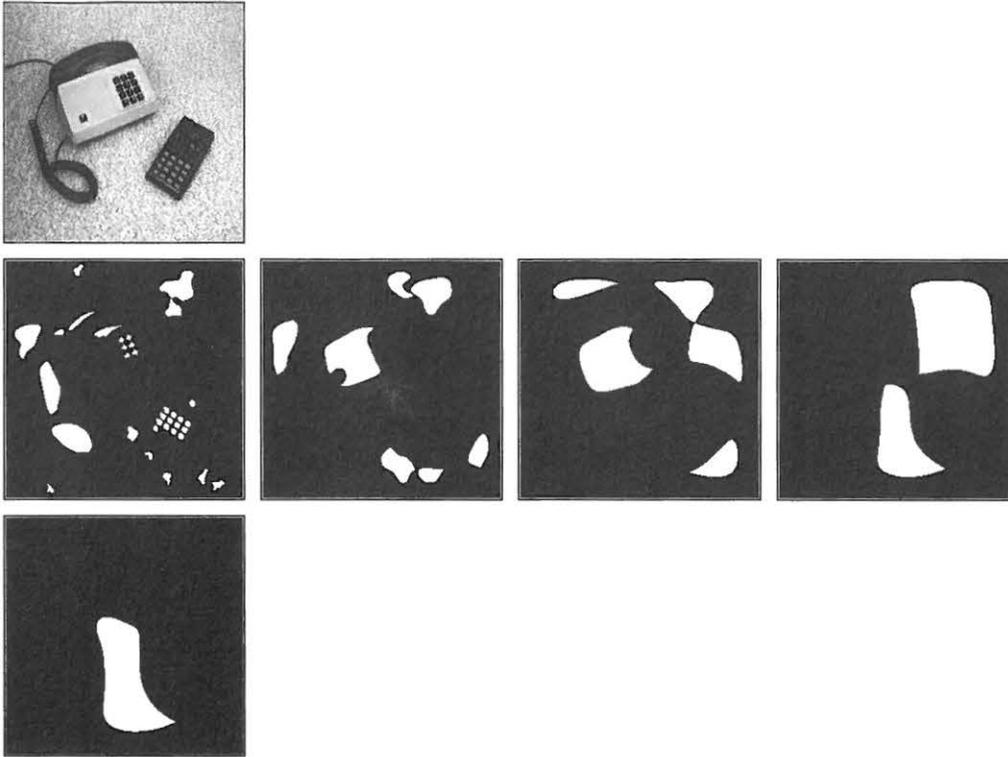


Figure 8.24: *The 50 most significant bright blobs from a telephone and calculator image. The background is textured.*

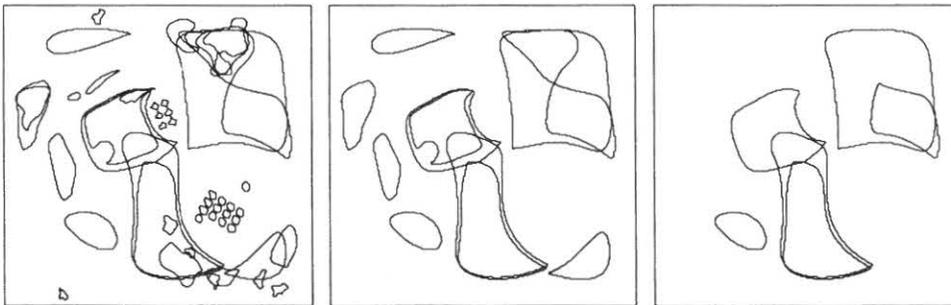


Figure 8.25: *Boundaries of the bright blobs extracted from the telephone and calculator image with textured background. (a) The 50 most significant blobs. (b) Low threshold on the significance measure set in one of the “gaps” in the sequence of significance values. (c) High threshold on the significance measure set in another “gap”.*

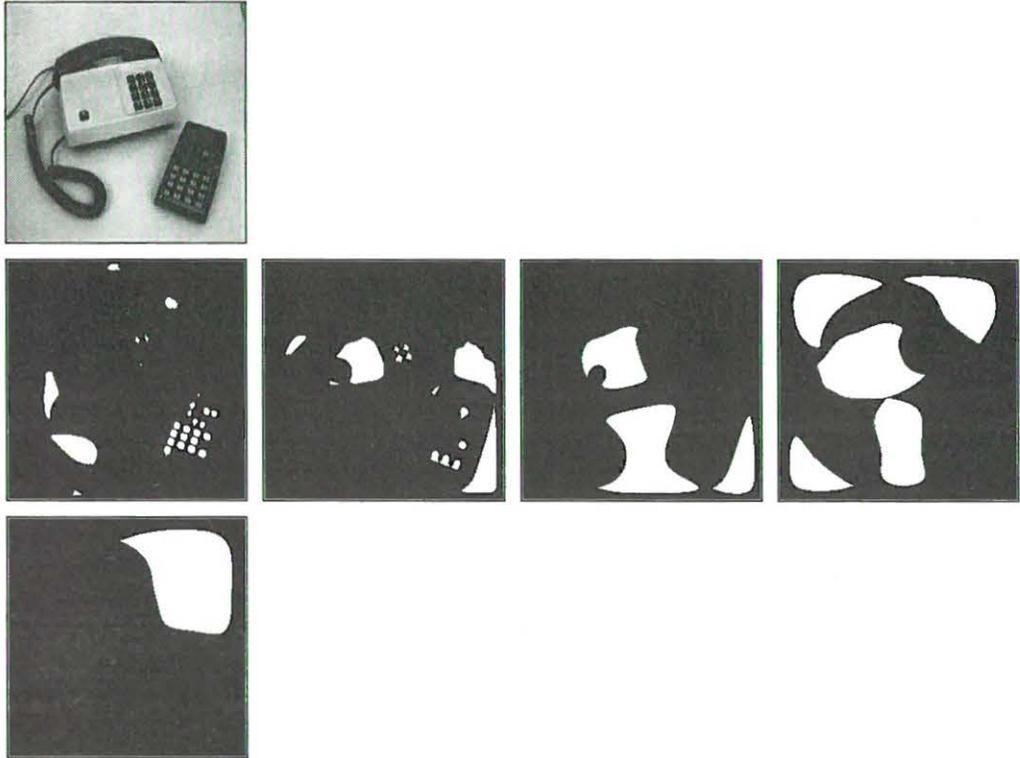


Figure 8.26: *The 50 most significant bright blobs from the telephone and calculator image with smooth background. (The dark blobs were shown in Figure 8.2.)*

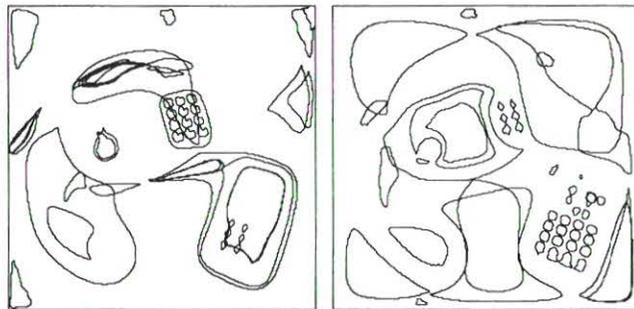


Figure 8.27: *Boundaries of the dark and bright blobs extracted from the telephone and calculator image with smooth background. (a) The 50 most significant dark blobs. (b) The 50 most significant bright blobs.*

Chapter 9

Guiding Early Visual Processes

Many methods in computer vision and image analysis implicitly assume that the problems of scale detection and initial segmentation have already been solved. One example is in edge detection, where the selection of step size for the gradient computations leads to a trade-off problem. A small step size gives a small truncation error, but the noise sensitivity might be severe. Conversely, a large step size will in general reduce the noise sensitivity, but at the cost of an increased truncation error. In the worst case one may even miss the slope of interest and get meaningless results if the difference quotient approximating the gradient is formed over a wider distance than the size of the object in the image. The problem originates from the basic scale problem, namely that the issue of inherent scale must be considered when selecting a mask size for computing spatial derivatives. Other examples can be obtained from most “shape from X” methods, which in general assume that they are applied to a domain in the image where the underlying assumptions are valid, corresponding to e.g. a region in the image corresponding to one facet of a surface etc.

The methodology we will develop in this section states that *the qualitative scale and region information extracted from the scale-space primal sketch can be useful for guiding other visual processes and will simplify their tasks*. More specifically, we propose that when spatial derivatives are needed, they can be computed from the scale-space representation at the scale given by a scale-space blob. Furthermore, the blob support regions can provide coarse size information to other algorithms. We suggest that this type of information can be used for delimiting the search space for further processing, for example such that matching could be carried out regionally in a neighbourhood of a blob instead of globally over the entire image.

Of course, the amplitude of spatial derivatives can in general be expected to decrease by the scale-space smoothing. Therefore, one cannot expect the actual numerical values of derivatives computed from the coarse scale representations to be *quantitatively* accurate. However, for finding *qualitative* features, not depending on the actual scaling of the intensity, like e.g. edges, local extrema, singularities in general etc, the *detection* step can be carried out at a coarse scale. Then, once the existence of a feature has been established, if precise numerical values are required, it should be possible to compute those in a second step e.g. by fitting an appropriate model, to the original data.

9.1 Application to Edge Detection

As a first example of the suggested way to use the scale-space primal sketch, for guiding other processes in early vision, we present an integration of the output from this representation

with an edge detection method known as edge focusing, developed by Bergholm [Ber87, Ber89].

The leading idea is to use the output scale information to guide an *edge detection scheme working at an adaptively determined level of scale*. We demonstrate that this task can be relatively easy and that there is no need for thresholding on gradient magnitude, since the image has been subjected to an appropriately selected amount of blurring. Hence, the detection step will be safe. The localization could on the other hand be poor due to the natural shape distortions that occur at coarser levels of scale in scale-space. However, the localization can be improved using the edge focusing method, which traces the safely detected edges at coarse scales to corresponding and better localized edges at finer scales. Hence, the resulting method will achieve a good compromise between the two conflicting goals in edge detection, namely eliminating the noise without distorting the localization of the edges. Another way to phrase this property is that we circumvent the problems connected with *simultaneous* detection and localization, that have been discussed by e.g. Canny [Can86]. Thus, we do not only perform edge detection without any need for thresholding. We are also more likely to get edge elements with meaning, since they correspond to boundaries of objects, which have given rise to significant blobs in the scale-space primal sketch.

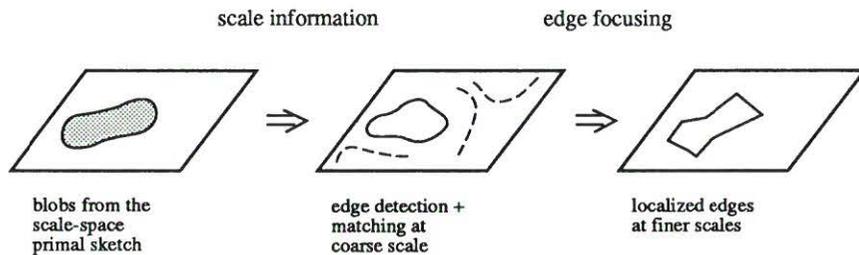


Figure 9.1: Schematic view over the proposed integration of the scale-space primal sketch module with edge detection. Given a significant scale-space blob, edge detection is performed at the appropriate scale of the blob scale-space blob. Then a matching step between the support region of the blob and the edges is carried out. Finally, the matched edges are localized to finer scales using edge focusing.

We do not maintain that this part of the presentation describes any “optimal way” to solve every occurring subproblem. Instead, the intention is to illustrate how a connection between the scale-space primal sketch with other modules can be done. The application supports the claim we make, that *if the image contains significant structures, which stand out from the surrounding, then they are extracted in such a way that the output information from the scale-space primal sketch is useful for further processing*. We will now describe the actions of the different submodules in more detail.

9.1.1 Edge Detection at a Proper Scale

The edge detection method used here is by intention simple, since we want to illustrate that edge detection becomes easier once the earlier mentioned scale and region information is available. The image is smoothed to the scale level given by a significant blob from the scale-space primal sketch. Then x - and y -gradients are computed with the Sobel operator and a non-maximum-suppression step is performed to get thin edges. In order to suppress

spurious noise points at the finest levels of scale we accept only edge segments having a length exceeding, say, 2 pixels.

9.1.2 Matching Blobs to Edges

Associating blobs with edges leads to a matching situation. However, we argue that also this task becomes simpler when performed at a proper level of scale. The matching criterion we have made use of in this work is based on spatial coincidence, and is a combination of three different conditions:

9.1.2.1 Geometric Coincidence

The edge segment should “encircle” or be “included” in the blob. A convenient way to formulate such a criterion is as follows. Let B be the set of pixels contained in the support region of a blob and let E be the set of pixels covered by an edge segment. Further, given any region R define the quantities x_{min} , x_{max} by

$$x_{min}(R) = \min_{(x,y) \in R} x; \quad x_{max}(R) = \max_{(x,y) \in R} x; \quad (9.1)$$

and the quantities y_{min} , y_{max} analogously. Now, an edge segment E will be regarded as a matching candidate of a blob B if¹

$$\begin{aligned} x_{min}(E) &\leq x_{max}(B); & x_{max}(E) &\geq x_{min}(B) \\ y_{min}(E) &\leq y_{max}(B); & y_{max}(E) &\geq y_{min}(B) \end{aligned} \quad (9.2)$$

In order to reduce the directional sensitivity of this criterion it is suitable to require that similar conditions hold also in a coordinate system rotated by 45 degrees. This criterion constitutes an approximation to the property that it should be impossible to draw a straight line separating the edge from the blob, see Figure 9.2 for an illustration. The latter property is satisfied if (9.2) holds in an arbitrarily rotated coordinate system. We define

Definition 9.1 (Extreme coordinate blob-edge matching candidate)

An edge E is said to be a (four-directional) extreme-coordinate matching candidate of a blob B if the conditions in (9.2) hold in the standard xy -coordinate system as well as in a similar coordinate system rotated by 45 degrees.

9.1.2.2 Proximity

The edge segment should *not be too far away from the blob boundary*. In other words the edge segment should comprise at least some pixel located near the boundary of the blob. We state the necessary condition

$$\min_{(x_E, y_E) \in E; (x_B, y_B) \in B} \sqrt{(x_E - x_B)^2 + (y_E - y_B)^2} \leq \frac{d(t)}{2} \quad (9.3)$$

where $d(t)$ is a typical spatial length at the current level scale². To summarize,

¹This condition is not the same requirement as $E \cap B \neq \emptyset$.

²Here, we have set this distance to the square root of an experimentally determined typical blob area, $A_m(t)$, at the current level of scale, (see Section 5.5.3.1 for further details).

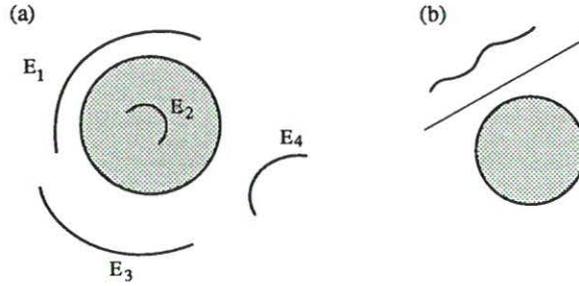


Figure 9.2: (a) The geometric extreme-coordinate condition means that the edge should either surround the blob or be included in it. In this example edges E_1 and E_2 are treated as matching candidates of the blob while E_3 and E_4 are not. (b) This criterion is an approximation to the requirement that it should be impossible to draw a straight line separating the line from the blob.

Definition 9.2 (Proximity blob-edge matching candidate)

An edge E at scale t is said to be a (weak) proximity blob-edge matching candidate of a blob B at the same scale if the minimum distance between the blob and the edge is less than $d(t)/2$, where $d(t)$ is a characteristic length at scale t .

The main purpose of the stripe around the boundary of the blob is to avoid edges corresponding to the interior of a blob from being interpreted as belonging to the blob boundary. It will prevent interior edges corresponding to e.g. surface markings from being matched to the blob boundary and also rule out edges far outside the blob. The width of this stripe is not critical, since at coarse scales the edges will usually have a substantial width and there will be an interval around the edge where there are no other edges.

9.1.2.3 Voronoi Diagram of the Grey-Level Blobs

The edge segment should not be too strongly associated with other blobs. We compute a *Voronoi diagram* of the grey-level blob image at the given level of scale, using a distance transformation. An edge segment is regarded as a matching candidate of a blob if it has at least have one pixel in common with the Voronoi region associated with the grey-level blob, see Figure 9.3(b). We define

Definition 9.3 (Voronoi blob-edge matching candidate)

Given a blob B at a certain scale, let V be the Voronoi region corresponding to B in the Voronoi diagram of the grey-level blob image at that scale. Then, an edge E at the same scale is said to be a (weak) voronoi blob-edge matching candidate of B if the edge has at least one pixel in V .

This condition prevents edges, which are closely related to one particular blob, from being associated with other blobs. For instance, if two grey-level blobs share the same delimiting saddle point, then the stripe around one of the blobs will cover a part of the other blob.

9.1.2.4 Resulting Matching Procedure

For an edge segment to be accepted as a matching candidate of a blob, it must be a matching candidate with respect to all these three criteria. Hence, the matching is relatively

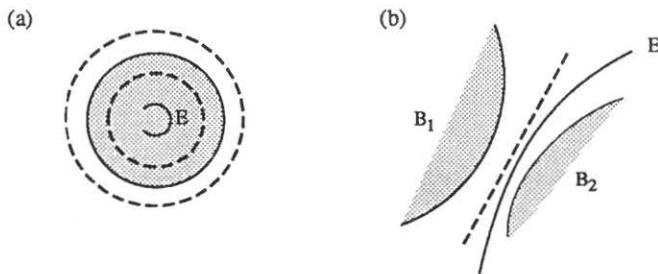


Figure 9.3: (a) The main purpose with the stripe around the blob is to prevent edges corresponding to the interior of the blob and edges far away from the blob from being associated to the blob boundary. (b) The purpose with the Voronoi region is to prevent edges strongly related to one blob from being associated with other nearby blobs.

restrictive. But again, the situation is improved by the fact that it is performed at a coarser scale. Once we know that a spatial region has given rise to a large blob at some level of scale, it seems very improbable that conflicting edges could appear at the same level of scale, since most interfering fine-scale structures ought to be suppressed by the scale-space smoothing. Figure 9.4 and Figure 9.5 illustrate two such matching situations from the toy block image and the telephone and calculator image respectively. We display the blob to be matched, the extracted edges at the scale level given by the blob, the grey-level blobs at the same level of scale, the Voronoi diagram of the grey-level blob image and the matched edges.

The main problem with this matching procedure is that it does not include any mechanism for breaking up long edge segments into shorter ones. The edge segment grouping is based just on connectivity between adjacent edge pixels. This means that the edge segments at coarser levels of scale may be very long, and spread far away from the boundary of the actual blob, see Figure 9.5 and also the examples in Figure 9.8 and Figure 9.10. It seems probable that further clues for distinguishing which edges should be associated with a certain blob could be obtained by studying the behaviour and the connectivity of the edges during the focusing procedure, see also Figure 9.10.

Another situation where the matching could fail is for severely fragmented edges. Then the matching may be rejected by Definition 9.1 if the edge segments encircle but are located outside the blob support region. However, we have not found any such problems to occur in any experiments.

More generally, we find the regions defined by the Voronoi diagram of the grey-level blob image and the region around the blob boundary as useful spatial regions (see Figure 9.4(e-f) and Figure 9.5(e-f)) to be associated with the blobs also for other types of matching purposes, see e.g. the work with junction classification in Section 9.3 and the more extensive discussion in Section 9.1.5.

9.1.2.5 Improving the Localization of the Blob Boundary

For single isolated blobs, the proximity matching criterion in its original formulation can lead to problems. In such cases the boundary of the grey-level blob support region can spread far away from the boundary of the actual “object” in the image, since there will be no competing blobs in its neighbourhood delimiting its growth. Hence, the blob might extend far beyond the “actual boundary”, but with a relatively flat intensity slope (compare with

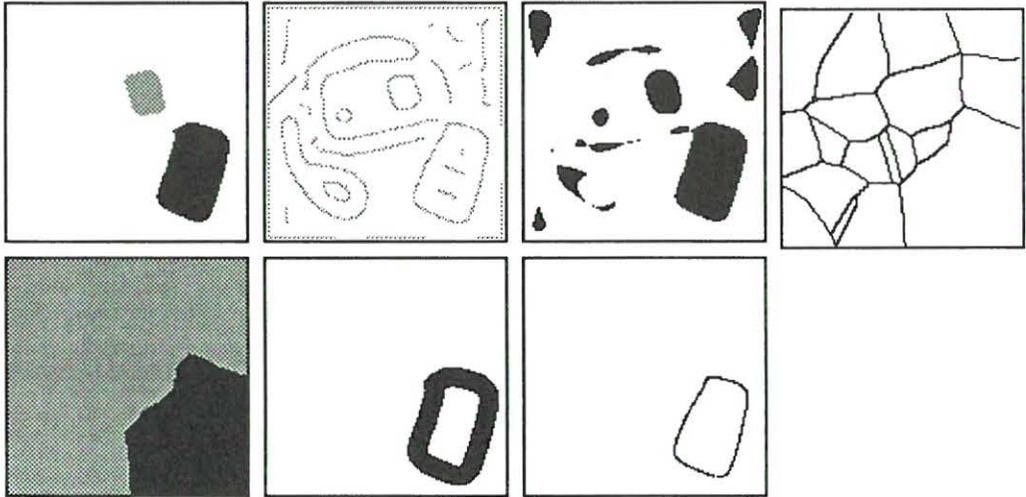


Figure 9.4: *Illustration of the matching procedure between blobs and edges for a blob from the telephone and calculator image. (a) A dark blob from the scale-space primal sketch which is to be matched (marked with black). (b) Extracted edges (non-maximum suppression without thresholding) at the scale level given by the blob. (c) The grey-level blobs at the same level of scale, i.e. all grey-level blobs at that scale level. (d) Voronoi diagram of the previous grey-level blob image. (e) The region in the Voronoi diagram corresponding to the treated blob. (Used in Criterion 3). (f) The stripe around the blob edge. Its width has been set to a characteristic length at the current level of scale. (Used in Criterion 2). (g) The resulting matched edges, that is the edges that have at least one pixel in both of the regions marked in Figures (e) and (f) and in addition satisfy the min-max coordinate criterion (Criterion 1) above about geometric coincidence with the blob.*

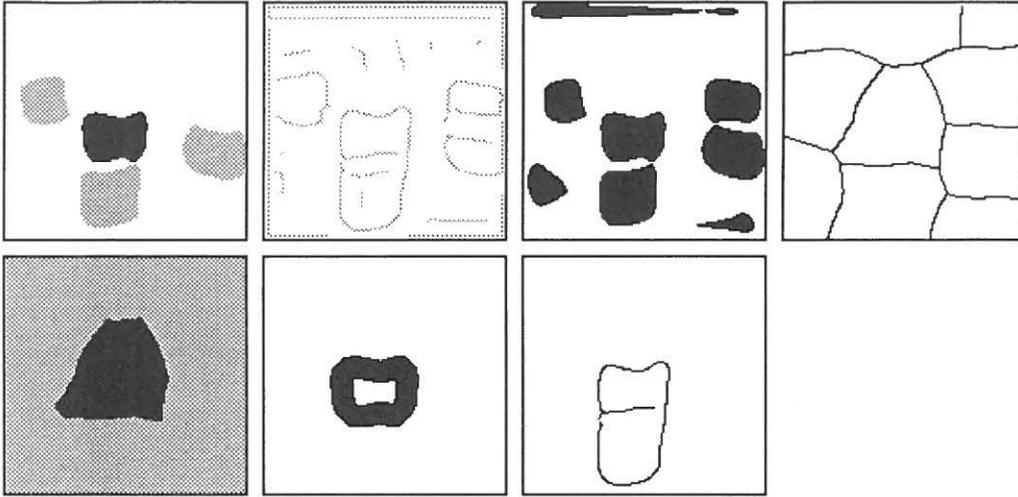


Figure 9.5: Similar illustration of the matching procedure as above but for a blob from the toy block image. Note that one of the edge segments spreads far away from the blob since the matching algorithm does not include any mechanism for breaking up long edge segments into shorter ones. However, we will demonstrate below that the focusing procedure itself provides a cue for such determination — at finer levels of scale the elongated edge will break up into two well separated sets of edges (see Figure 9.13).

the shape of the Gaussian at the tails). This means that the edge matching by Definition 9.2 might fail. In the extreme case, when there is just one local extremum in an image, the corresponding blob will actually get an infinite support region. In order to compensate for this effect we clip³ the grey-level blob at a higher grey-level for bright blobs (and a lower level for dark blobs) than the previously defined base level, as to obtain a better localized blob boundary. This modified blob is then used for determining the stripe around the blob for matching according to (9.3). Empirically we have found that a clipping level⁴ of about 35% ($\approx \frac{1}{3}$) of the range between the minimum and maximum grey-levels within the blob gives a reasonable improvement in localization without seriously affecting blobs actually having nearby competitors. In this region the slope of the grey-level intensity function of the blob is normally relatively steep and we will obtain a smaller blob with a better localized boundary. For blobs that are not isolated the effect of this clipping will usually be minor.

The actual value of the clipping level is not critical for the matching, since the stripe around the blob boundary is anyway intended just as a coarse descriptor of a region around the blob boundary for guiding the blob-edge matching. During all our experiments this parameter has been kept unchanged. Other possible ways of overcoming this problem could be using a more advanced reasoning process in the matching step or by applying e.g. a snake [Kas87] attracted by high values of gradient magnitude in order to obtain the better localized “blob boundary” for matching. The initial position of the snake can be determined from

³This clipping is performed only for the purpose of computing the blob boundary for matching. In all other situations we stick to the original definition of a grey-level blob.

⁴In these units a clipping level of 0% corresponds to the original definition of grey-level blob and a clipping level of 100% to clipping at the extremum point of the blob.

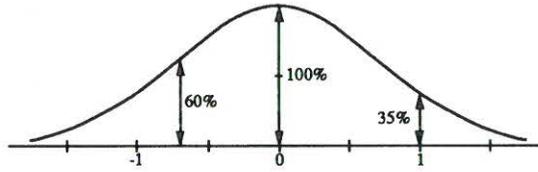


Figure 9.6: In the extreme case, a single isolated blob will have an infinite support region. In order to compute a spatial representative more useful for matching purpose, the blob boundary must be modified. A straightforward way is to clip the blob at a grey-level corresponding to a position closer to the edge. For a Gaussian intensity profile the position of an edge defined by non-maximum suppression corresponds to a clipping level of about 61% ($\approx e^{-1/2}$). Here we use a clipping level of about 35% ($\approx e^{-1}$).

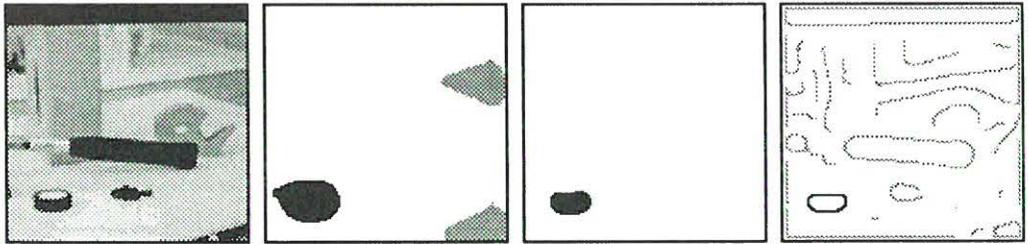


Figure 9.7: Illustration of the effect of clipping. (a) Original grey-level image (b) The (unclipped) blob corresponding to the tape reel in the lower left corner. (c) The effect of clipping that blob. (d) The edges matched to the clipped blob.

the position and the extent of the (possibly clipped) blob. However, in this implementation we have used the clipping method because of its algorithmic simplicity.

9.1.3 Edge Focusing

Edge focusing, developed by Bergholm [Ber87, Ber89], is a method for tracing edges through scale-space. The basic principle is to detect edges at a coarse scale in scale-space and then trace them to finer scales. Hence, the method achieves a good compromise between the two conflicting goals in edge detection, namely; eliminating noise without distorting the localization of the edges.

It has been shown [Ber87, Ber89] that if the focusing procedure is performed such that the scale step $\Delta\sigma$, expressed in $\sigma = \sqrt{t}$, is less than $\frac{1}{2}$ then for most common edge configurations the edges are guaranteed to move not more than one pixel from one level to the next. In that case the matching will be trivial — to find the corresponding edges at the finer level of scale, it suffices⁵ to perform edge detection in a one-pixel neighbourhood around the edges at the coarser scale.

⁵Obviously, there are situations where such a fixed scale sampling can lead to problems, see Section 6.1.1 for a description.

In this application we initiate the focusing procedure from several scale levels, since the significant blobs from the scale-space primal sketch manifest themselves at different levels of scale. Hence, we resort to the significant blobs in decreasing scale order. We start with the coarsest scale blob, detect edges at that level of scale and match the obtained edges to the blob. This gives the input for the focusing procedure, which then follows these edges to the scale given by the second blob. The edge detection and matching steps are repeated at this new level of scale and the resulting edges are added to the output from the previous focusing step. This new edge image serves as input for another focusing procedure, tracing the edges to the next finer level of scale etc.

9.1.4 Experimental Results

In Figure 9.8 and Figure 9.9 we illustrate some steps from the composed edge detection, blob-edge matching and edge focusing procedure for the telephone and calculator image. The left column shows the blob support region of the blob. The blob considered at the current level of scale is black. The other blobs from the scale-space primal sketch are displayed in grey. The middle column shows the edge image at the same level of scale. The matched edges have been marked with black, while the other ones are grey. Finally, the right column shows the result after focusing, just before a new blob is considered. In order to reduce the number of blob hypotheses treated, we have used a threshold on the significance value. The final result of the focusing procedure is shown in the lower right corner of Figure 9.9. Figure 9.10 and Figure 9.11 show corresponding results for the toy block image.

Let us again mention that this method, which we call *blob-initiated edge focusing*, is not just another edge detector, but that the edge elements obtained are more meaningful entities, since they are associated with blobs and explicit scale information.

9.1.5 Alternative: Individual Treatment of the Blob Hypotheses

Instead of treating all hypotheses generated by the scale-space primal sketch simultaneously during the edge focusing phase they can of course also be treated separately. I.e. one can let each blob start up its own focusing scheme, which processes the edge segments matching that blob independently of the edges corresponding to other blobs. Then the relations between blobs and edges will be obvious and the interference effects between edge segments from different blobs during the edge focusing will be eliminated.

Note that such an approach need not require much more computations than the previously described method, provided that the edge images used for the focusing procedure are pre-computed⁶. Then, the part of the processing that is repeated for different blobs will be just the matching between the scale levels, which is computationally inexpensive, since it only comprises a search in the eight-neighbourhoods around the processed pixels.

Figure 9.12 and Figure 9.13 show the results after such individual focusing schemes applied to the blobs used for the matching illustrations in Figure 9.4 and Figure 9.5. Observe that in the second case the focusing algorithm resolves the problem with the elongated edge segment that previously spread far from the blob. At a fine level of scale it has split into two distinctly separated (groups of) edge segments.

With this *individual blob-initiated edge focusing* we achieve a way of avoiding the commonly occurring step of tracking and grouping related edge elements into edge segments. The matching between a blob and the edges at coarse levels of scale has already induced

⁶The edge focusing algorithm is normally implemented using a fixed set of scale levels [Ber90].

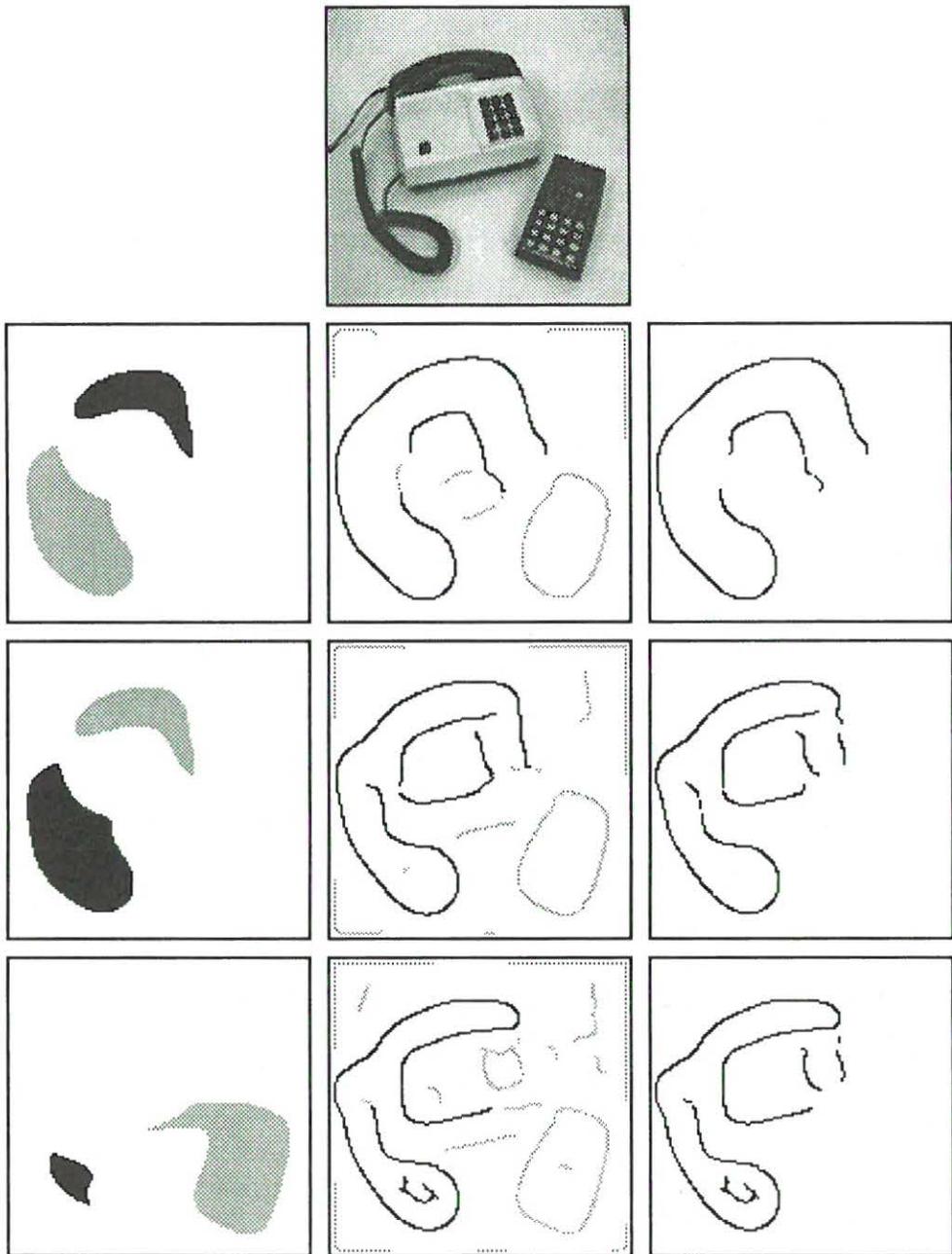


Figure 9.8: Illustration of the composed blob-edge focusing procedure for the telephone and calculator image. The left column shows the active blob hypothesis. Its blob support region has been marked with black. The middle column shows the edge image at the level of scale given by the previous blob. The matching edge segments have been drawn black while the other edge pixels are grey. The right column shows the focused edge. The scale and significance values for the different blobs are from top to bottom (101.6, 14.1), (50.8, 252.8), (32.0, 11.4) respectively.

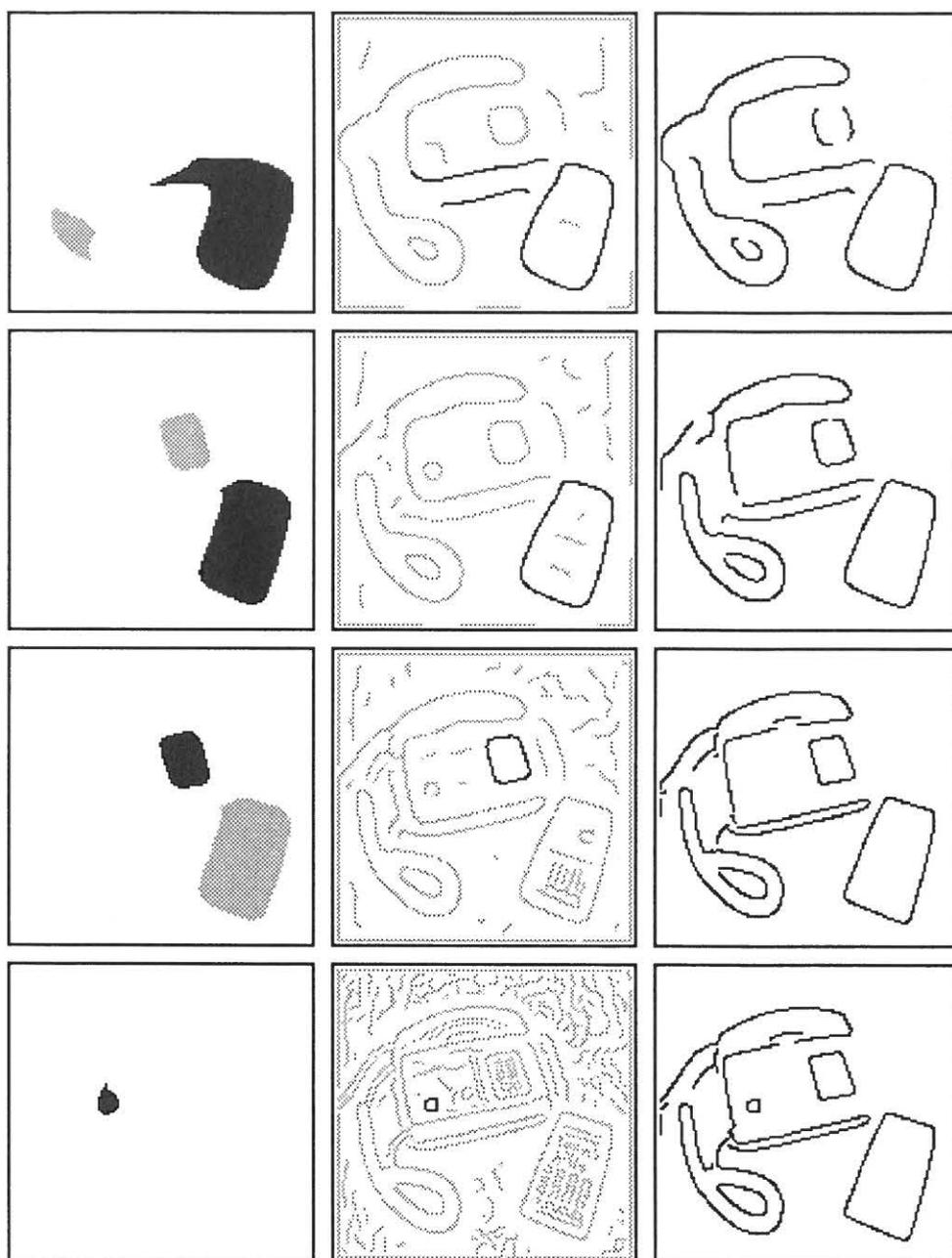


Figure 9.9: Illustration of the composed blob-edge focusing procedure for the telephone and calculator image continued. The scale and significance values for the different blobs are from top to bottom (25.4, 660.9), (14.3, 40.8), (6.4, 63.6) and (1.3, 13.2) respectively.

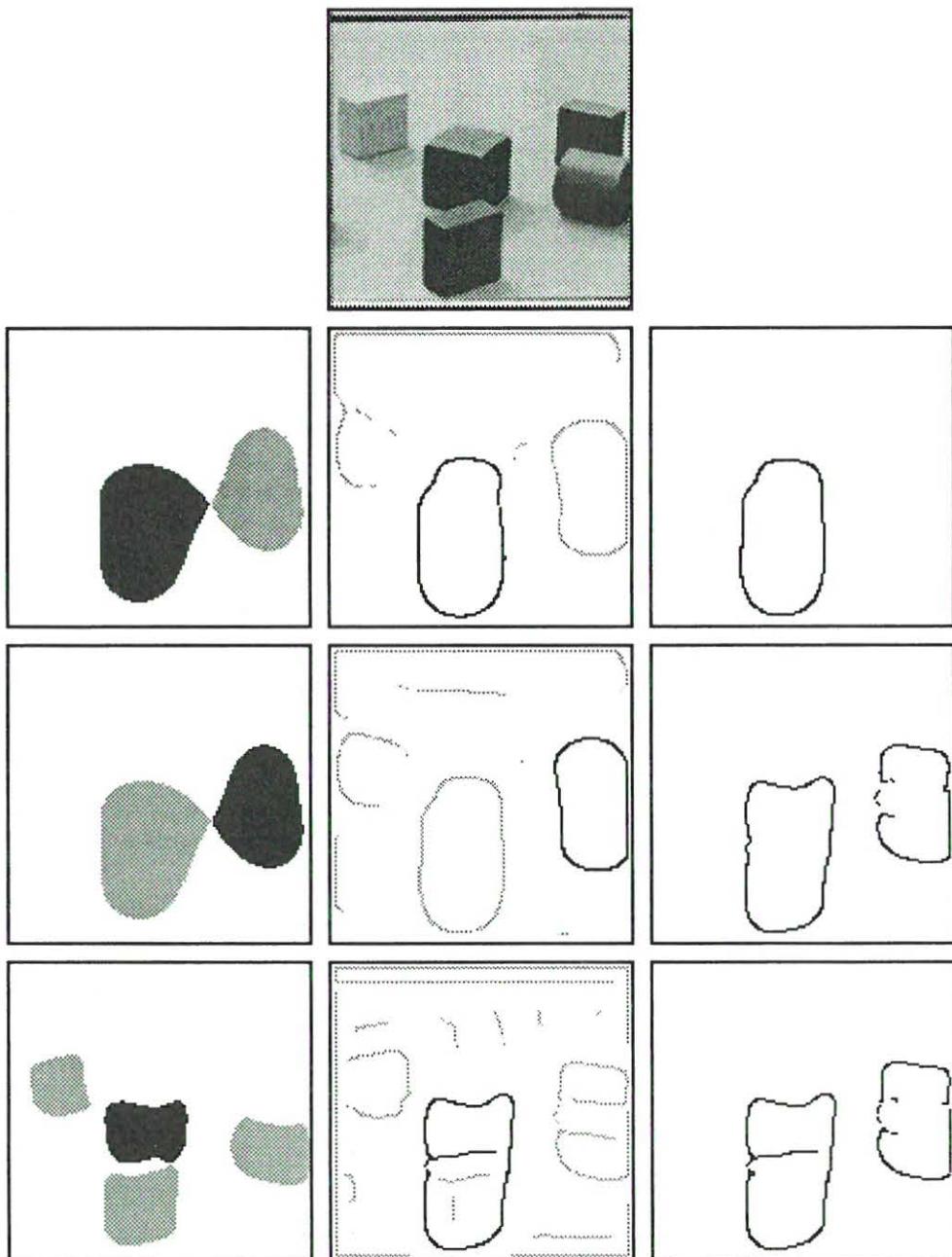


Figure 9.10: Illustration of the composed blob-edge focusing procedure for the toy block image. The left column shows the active blob hypothesis. Its blob support region has been marked with black. The middle column shows the edge image at the level of scale given by the previous blob. The matching edge segments have been drawn black while the other edge pixels are grey. The right column shows the focused edge. The scale and significance values for the different blobs are from top to bottom (203.2, 15.5), (161.3, 10.7), (20.2, 20.3) respectively.

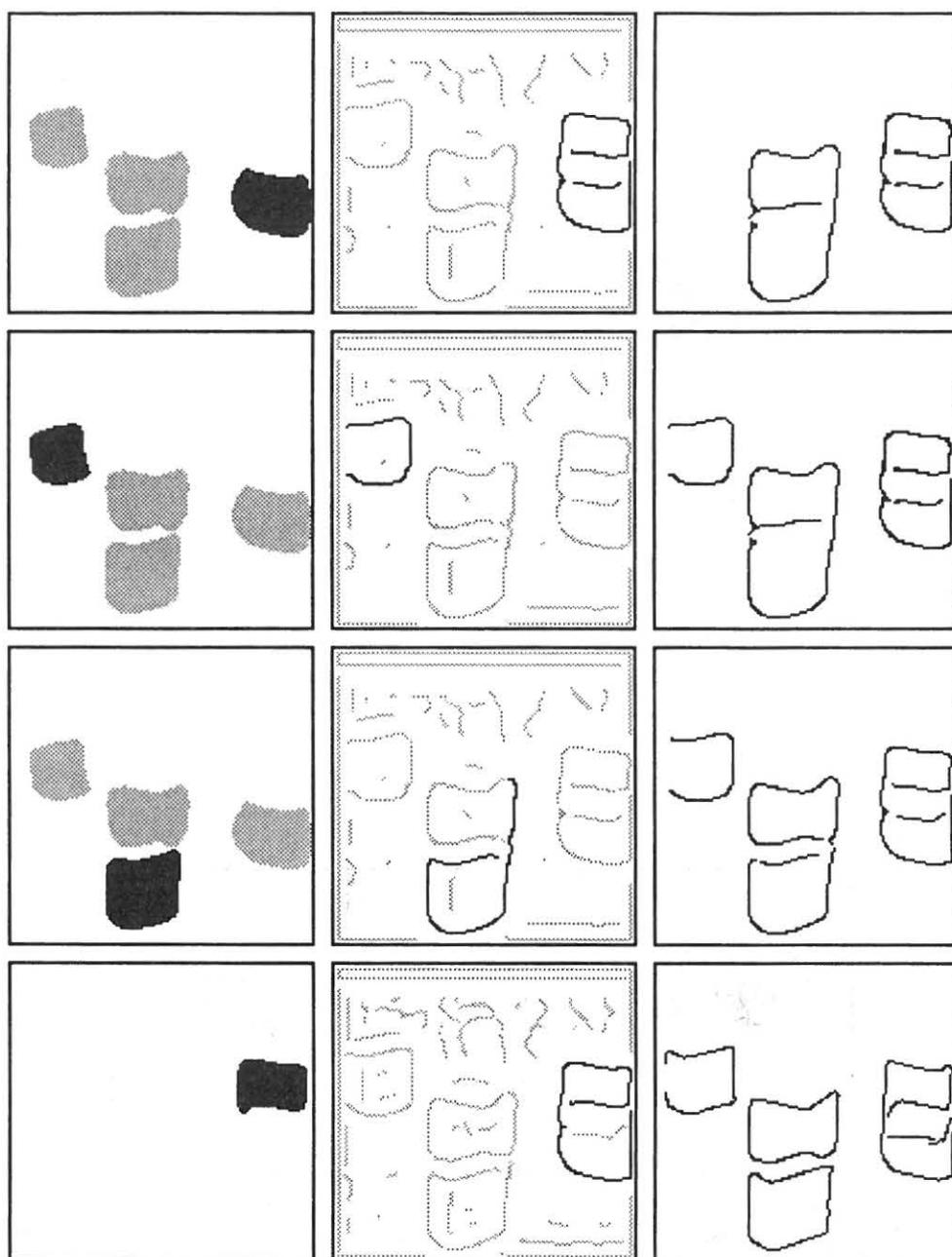


Figure 9.11: Illustration of the composed blob-edge focusing procedure for the toy block image continued. The scale and significance values for the different blobs are from top to bottom (12.7, 52.2), (12.7, 4.6), (12.7, 35.7) and (8.0, 48.9) respectively.

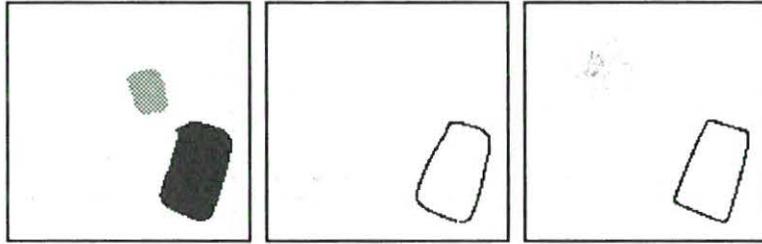


Figure 9.12: *Illustration of the individual blob hypothesis treatment for a blob from the telephone and calculator image. (a) A blob from the scale-space primal sketch. (b) Matched edges according to the procedure described above. (c) The result after focusing the edge down to finer scales ($t = 1.0$).*

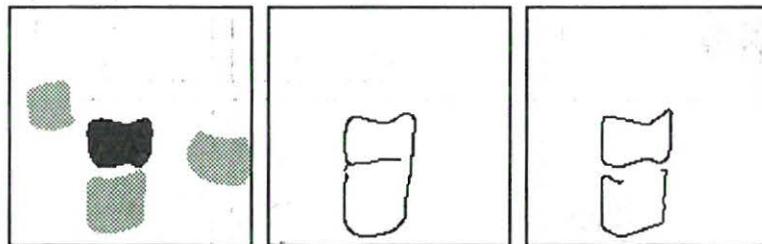


Figure 9.13: *Illustration of the individual blob hypothesis treatment for the a blob from the toy block image. (a) A blob from the scale-space primal sketch. (b) Matched edges according to the procedure described above. (c) The result after focusing the edge down to finer scales ($t = 1.0$). Note that the long edge segment that spread far from the actual blob has split into two separate edge segments.*

a *coarse grouping of edge pixels* into higher order units. In addition, we *explicitly* have the relation to the blob, which gives scale information and coarse spatial information. Similarly, this approach induces a way of actually verifying or rejecting various kinds of blob hypotheses generated by the blob detection module. For instance, blobs due to noise (or illumination variations) can probably be rejected by studying the behaviour of their corresponding edges under defocusing in a manner similar to the classification of diffuse edges by Sjöberg, Bergholm [Sjö88, Ber89] and Zhang, Bergholm [Zha91].

We also find it possible that this kind of coarse edge grouping, combined with the blob information (and possibly also the Voronoi diagram of the grey-level blob image), can be used for delimiting the search space for higher order interpretations. We see potential applications in various grouping and matching problems like model matching, Hough transforms, tests for parallel lines, abstractions of edge descriptors etc. In other words, we believe that this kind of coarse information lends itself naturally to *qualitative reasoning*. However, there is still more work to be done in order to explore these suggestions.

9.1.6 Conclusions

The result from these experiments can be interpreted in many ways. We have used the output from the scale-space primal sketch to control an edge focusing procedure. Hence, we have eliminated two of the tuning parameters used in the edge focusing algorithm, namely the initial scale for edge detection and the threshold on gradient magnitude. What remains undetermined is the stop scale, i.e., the scale down to which the edge focusing should be performed. In this work it has throughout been set to $t = 1$, a scale where the sampling effects due to the discrete grid start to become important, (see e.g. the comparisons between different methods for implementing scale-space smoothing in Chapter 4). It seems plausible that some further guidance for this selection could be obtained by studying the behaviour of the focused edges in scale-space, compare with the classification of diffuse edges in Sjöberg, Bergholm [Sjö88, Ber89] and Zhang, Bergholm [Zha91].

This integration of the two algorithms exemplifies the previously mentioned guidance of the focus-of-attention. Note that the processing initiated by the scale-space primal sketch is performed only for a small subset of the image data. Hence, the resulting method relates to the idea of a “focused beam”, derived by Tsotsos [Tso90] from complexity arguments.

A more immediate interpretation of the results is that we have selected a subset of the edges in the edge image at the finer level of scale. In contrast to the result from a raw edge detection scheme, we know that these edge elements are more meaningful entities. They are associated with significant blobs, and the scale information is explicit. Note that label information for the edge segments can be easily inherited during the edge focusing process. Hence, even if the different blob hypotheses are treated simultaneously in the same focusing process, we can keep track of which blobs have given rise to a specific edge at any level of scale during the edge focusing.

The experiment also illustrates that the dual concepts “region” and “edge” can be matched without severe difficulties, provided that the matching is performed at a proper level of scale.

The most important conclusion one can draw from this experiment is that it clearly demonstrates that the qualitative output information from the scale-space primal sketch is useful in guiding and simplifying later stage processing. We shall now exemplify this in multi-spectral classification.

9.2 Application to Histogram Analysis

The scale-space primal sketch is well suited for automated cluster detection, since it is designed for detection of bright blobs on dark background and vice versa. Hence, it lends itself as a natural module for peak detection in algorithms based on histogramming techniques. Although it is well-known that histogram-based segmentation hardly can be expected to work globally on entire images (due to illumination variations, interference because of many regions etc), such methods can often give useful results *locally* in small windows where only a few regions of distinctly different characteristics (e.g. colour or grey-level) are present.

9.2.1 Experimental Results: Histogram-Based Colour Segmentation

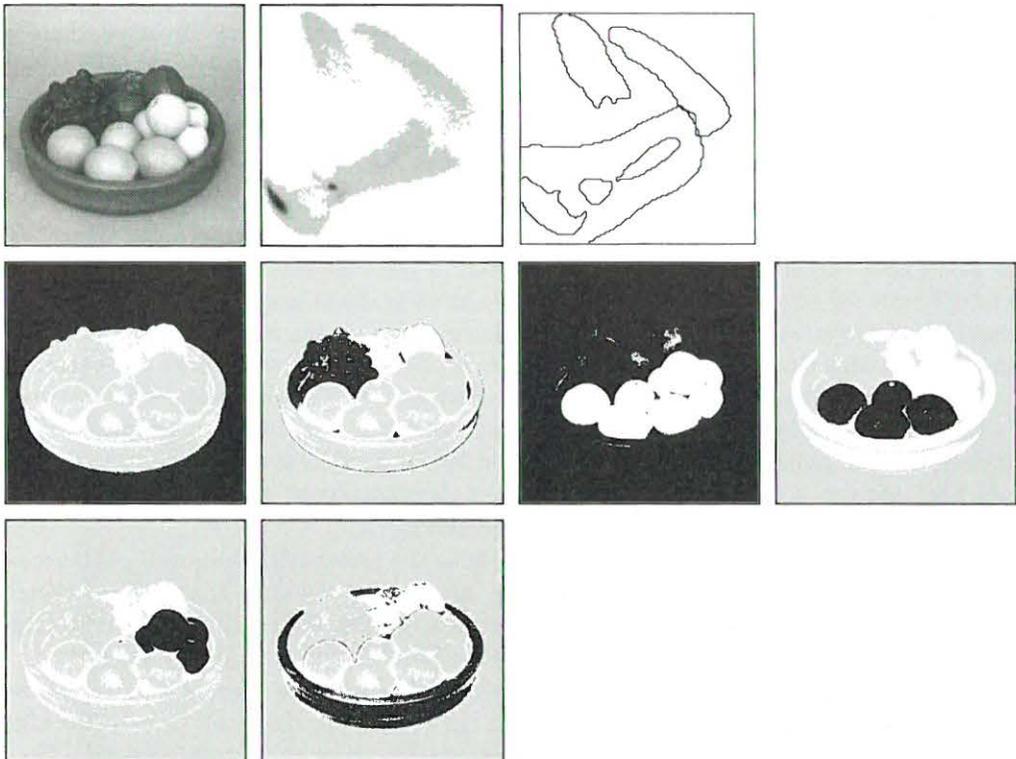


Figure 9.14: Histogram-based colour segmentation of a fruit bowl image: (a) Grey-level image. (b) Histogram over the chroma information. (c) Boundaries of the 6 most significant blobs detected by the scale-space primal sketch. (d)-(i) Backprojections of the different histogram blobs to the original image (in decreasing order of significance). The pixels corresponding to the various blobs have been marked in black. (The region in Figure (f) is the union of the regions in Figures (d), (e) and (i)).

In Figure 9.14 and Figure 9.15 we illustrate how the scale-space primal sketch can constitute a helpful tool in such histogram modality analysis of multi-spectral data. We have accumulated histograms⁷ over the chroma information and used the scale-space primal sketch

⁷The colour images have been converted from the usual RGB format to the CIE u^*v^* 1976 format, see e.g. [Bil82], which separates the intensity and the chroma information. The histogram is formed only over the chroma information, ignoring the intensity information.

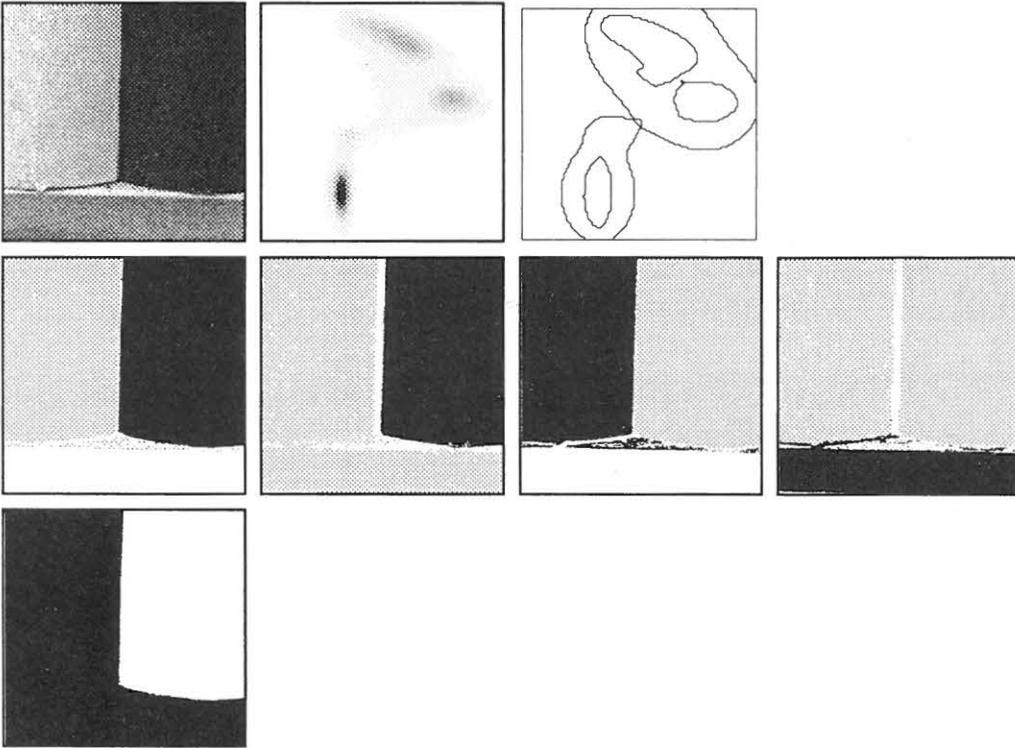


Figure 9.15: Similar histogram-based colour segmentation of a detail from an office scene. The image shows a small window from a bookcase with two binders (yellow and blue) on a shelf made of (yellowish) wood. (a) Grey-level image. (b) Histogram over the chroma information. (c) Boundaries of the 5 most significant blobs detected by the scale-space primal sketch. (d)-(h) Backprojections of the different histogram blobs to the original image (in decreasing order of significance).

for detecting peaks and clusters in the histograms. We see that the extracted blobs induce a meaningful partitioning of the histogram corresponding to regions in the image with distinctly different colours.

Of course, there is a decision finally to be made about which peaks in the histogram should be counted as being significant. However, we hypothesize that the significance values given by the scale-space blob volumes reflect the situation in a manner useful for such reasoning, especially since the regions around the peaks are extracted automatically. In these examples (single) thresholds have been set manually in “gaps” in the sequences of significance values. For the fruit bowl scene the accepted blobs had significance 42.6 (background), 8.3 (grapes), 3.6 (oranges), 3.1 (apples), 3.0 (bowl) while the significance values of the rejected blobs were 2.0 and less (in decreasing order 2.0, 1.9, 1.8, 1.4, 1.3, 1.1, 1.1, 1.0, ...).

The significance values of the displayed blobs from the office scene were 187.9 (blue binder, large blob), 173.7 (blue binder, small blob), 170.1 (yellow binder), 80.6 (shelf) and 66.7 (yellow binder and shelf). As we see, two blobs corresponding to the blue binder have been detected. This is a common phenomenon in the scale-space primal sketch, that arises because a large blob merges with a small (insignificant) blob and forms a new scale-space blob. Two such duplicate blobs corresponding to the yellow binder (significance 18.0) and the shelf (significance 17.9) have been suppressed. The remaining blobs had significance values 2.5, 2.0, 2.0, 2.0, 1.2, 1.2, 1.2, 1.1 and less.

9.2.2 Sensitivity to Quantization Effects

It can also be noted that this peak detection concept will be less sensitive to quantization effects in the histogram acquisition than many traditional peak detection methods. The problems due to too fine a quantization in the accumulator space will be substantially reduced, since the scale-space blurring will lead to a propagation of information between different accumulator cells. Thus, even though the original histogram might have been acquired using “too many and too small” accumulator cells, large scale peaks will be detected anyway, since the contents of their accumulator cells will merge to large scale blobs in scale-space after sufficient amounts of blurring.

Finding peaks in histograms is a problem that arises in many contexts. Let us point out that the case with colour-based histogram segmentation has been considered just as one possible application of the scale-space primal to histogram analysis. Because of the general purpose nature of this tool we think that it could be applicable also to other types of similar techniques such as Hough transforms, texture classification etc.

9.3 Application to Junction Classification

Brunnström et al. [Bru89, Bru90a] have shown that a reliable classification of junctions can be performed by analysing the modalities of local intensity and directional histograms during an active focusing process.

In this section we will first briefly review the main ideas behind the approach and then outline how the scale-space primal sketch can be useful in providing context information necessary for this procedure. It should be emphasized that the treatment here describes on-going work. Anyway, we find the presentation useful in illustrating some basic ideas about how the scale-space primal sketch can interact with other processing modules in an active vision situation.

9.3.1 Background: Classifying Junctions by Active Focusing

The basic principle of the junction classification method is to accumulate local histograms over the grey-level values and the directional information around candidate junction points (given by some interest point operator). Then, the numbers of peaks in the histograms can be related to the type of junction according to the following table:

Intensity	Edge direction	Classification hypothesis
unimodal	any	noise spike
bimodal	unimodal	edge
bimodal	bimodal	L-junction
trimodal	bimodal	T-junction
trimodal	trimodal	3-junction

Table 9.1: Basic classification scheme for local intensity and directional distributions around a candidate junction point (adapted from Brunnström et al (1990b)).

The motivation for this scheme is that for example, in the neighbourhood of a point where three edges join, there will (generically) be three dominant intensity peaks corresponding to the three surfaces. If that point is a 3-junction (an arrow-junction or a Y-junction) then the edge direction histogram will (generically) contain three main peaks, while for a T-junction the number of directional peaks will be two. Similarly, at an L-junction there will be two intensity and two directional peaks. Noise spikes and edges must be considered, since interest point operators like those proposed by Moravec [Mor77] or Kitchen and Rosenfeld [Kit82] tend to give false alarms near such points. Situations with more than three peaks in either the intensity or the directional histogram are treated as non-generic or as corresponding to surface markings.

Of course, the result from this type of histogram analysis cannot be regarded as a final classification, since the spatial information will be lost in the histogram accumulation. One obtains a hypothesis that must be verified in some way, e.g. by backprojection into the original data. Therefore, this algorithm is embedded in a classification cycle, see Figure 9.16 for an overview. More detailed information about the different submodules and how they communicate is given in [Bru90a, Bru90b].

9.3.2 Setting Window Size from Blob Information

However, taking such local histogram properties as the basis for a classification scheme leads two obvious questions: Where should the window be located and how large should it be? We believe that the scale-space primal sketch can provide valuable clues for both these tasks.

In order to estimate the number of peaks in the histogram, some minimum number of samples will be required. With a precise model for the imaging process as well as the noise characteristics, one could conceive deriving bounds on the resolution, at least in some simple cases. However, as will be developed further below, direct setting of a single window size immediately valid for correct histogram classification seems to be a very difficult or even impossible task.

Therefore, what is made use of instead is the process of *focusing*. Focusing means that

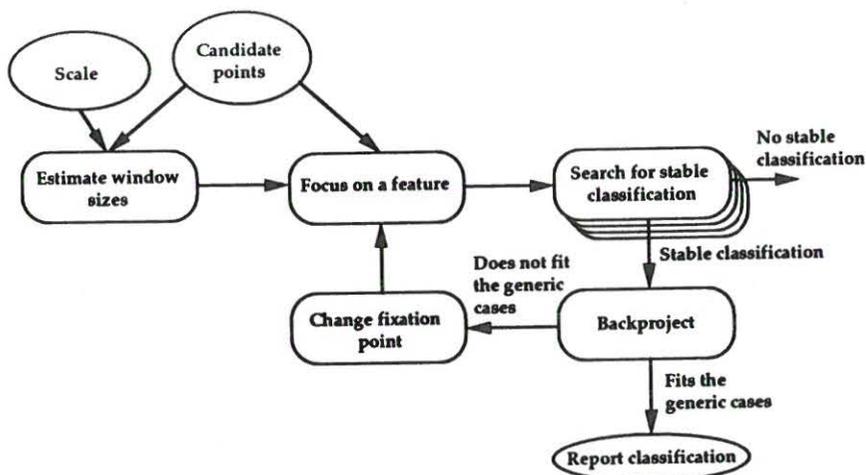


Figure 9.16: Schematic view over the junction classification cycle. A pair consisting of an interest point matched to a blob from the scale-space primal sketch gives rise to a series of windows of different size and resolution over which histograms are accumulated. The modalities of these histograms are compared to the generic cases and their stability measured. This gives a hypothesis about the possible nature of the junction candidate, which is tested by backprojection into the original image. If it cannot be verified then additional data are acquired, invoking a new analysis cycle in a closed-loop fashion. (From Brunnström et al. (1990b)).

the resolution is increased⁸ locally in a *continuous* manner (even though we still have to sample at discrete resolutions). The method is based on the assumption that stable responses will occur for the models that best fit the data, which relates to the systematic parameter variation principle described in Section 8.2.1.

Assuming that we have found a point of interest, we are to invoke the focusing procedure analysing local histograms. This calls for some mechanism for actually setting an initial range of window sizes, since the size of a suitable neighbourhood region around a junction candidate will in general vary both within and between images.

9.3.2.1 The Scale Problem in Junction Classification

If the window is too large, then other structures than the actual corner region around the point of interest might be included in the window, and the histogram modality would be affected. Conversely, if it is too small then the directional histogram could be severely biased and deviate far from the ideal appearance in case the physical corner is slightly rounded — a scale phenomenon that seems to be commonly occurring in realistic scenes⁹. A too small window might also fall outside the actual corner if the interest point is associated with a localization error. An example illustrating these effects for a rounded corner of a plastic detail is shown in Figure 9.17.

⁸Currently, for experimentation purpose, this process has been *simulated* by changing the window size in an (already taken) image having a sufficiently high resolution to *clearly resolve* the structures we are interested in. However, the long term goal of the work is to integrate the analysis with a camera system allowing the algorithm to *acquire new images* of higher resolution in situations where the current sampling density is not sufficient for resolving the structures under study.

⁹This effect does not occur for an ideal (sharp) corner, for which the inner scale is zero.

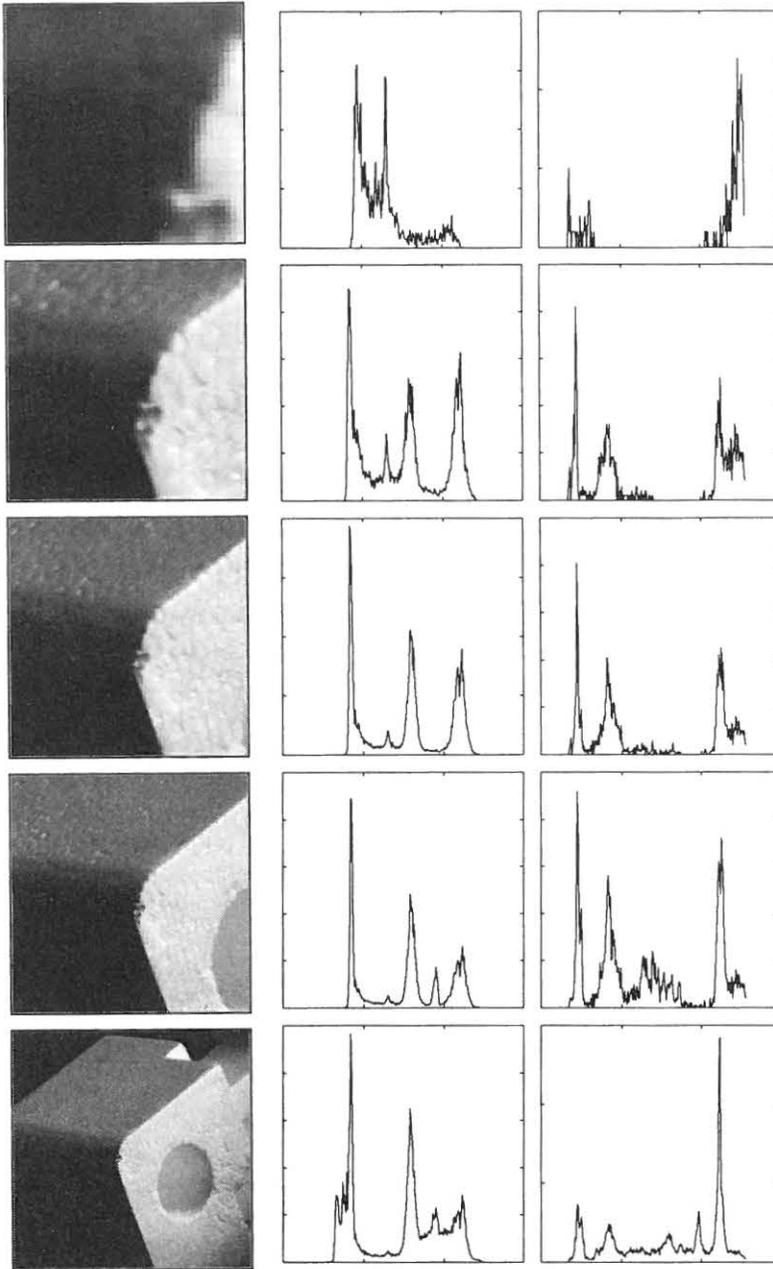


Figure 9.17: *Illustration of the scale problem in junction classification. We show the effects of varying the window size around a point near a rounded corner. The left column displays the treated subwindow, the middle column the intensity histogram and the right column the directional histogram. One observes that a correct classification based on histogram modalities can be made only within a certain range of window sizes. (The directional histograms have been accumulated only for the edge pixels in the window.)*

Therefore, the methodology we have adopted is to use the context information from the blobs for setting just coarse approximate values giving (generous) upper and lower bounds on a focusing interval. Then, the intention is that the systematic variation of window size combined with the consistency check over parameter variation should allow for a more robust modality determination.

9.3.2.2 Matching Interest Points to Blobs

Associating blobs to interest points leads to a matching situation. The method we use here is based on similar criteria as the matching between blobs and edges described in Section 9.1.2. Currently, an interest point is related to a blob if the following two conditions about spatial coincidence are satisfied:

1. The interest point should be at least coarsely associated with the blob, without being too strongly associated with other blobs. We compute a Voronoi diagram of the grey-level blob image at the given level of scale, using a distance transformation, and require the interest point to be included in the Voronoi region associated with the support region of the grey-level blob. This condition prevents points, which are closely related to one particular blob, from being associated to other blobs.
2. The interest point should not be located too far away from the blob boundary. We require the minimum distance between the interest point and the blob not to exceed a typical spatial length at the current level of scale¹⁰.

However, compared to the previous blob-edge matching, this implementation suffers from a few shortcomings. One situation where this procedure will face problems is at sharp corners, see e.g. Figure 9.18 for an example. Because of the rapid edge drift in such cases, the blob

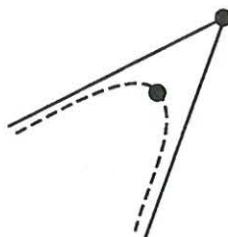


Figure 9.18: *At sharp corners, the matching procedure between blobs and edges will face serious problems unless some additional precautions are taken. Because of the rapid edge drift in such situations, the boundary of the blob may spread far away from the actual corner, which means that the matching may be rejected by the distance criterion.*

boundary may be ill localized at coarser scales. Hence, an interest point located at a sharp corner could be missed, because the distance between the interest point and the boundary of the blob is too large. Of course, one could imagine increasing the width of the stripe, but then the number of false matches would increase.

A possible way of explaining this weakness of the blob-point matching compared to the earlier mentioned blob-edge matching is that the scale information from the blobs is not

¹⁰Here, we have (similarly to the blob-edge matching) set this parameter to the square root of an experimentally determined typical blob area, $A_m(t)$, at that scale level, (see Section 5.5.3.1).

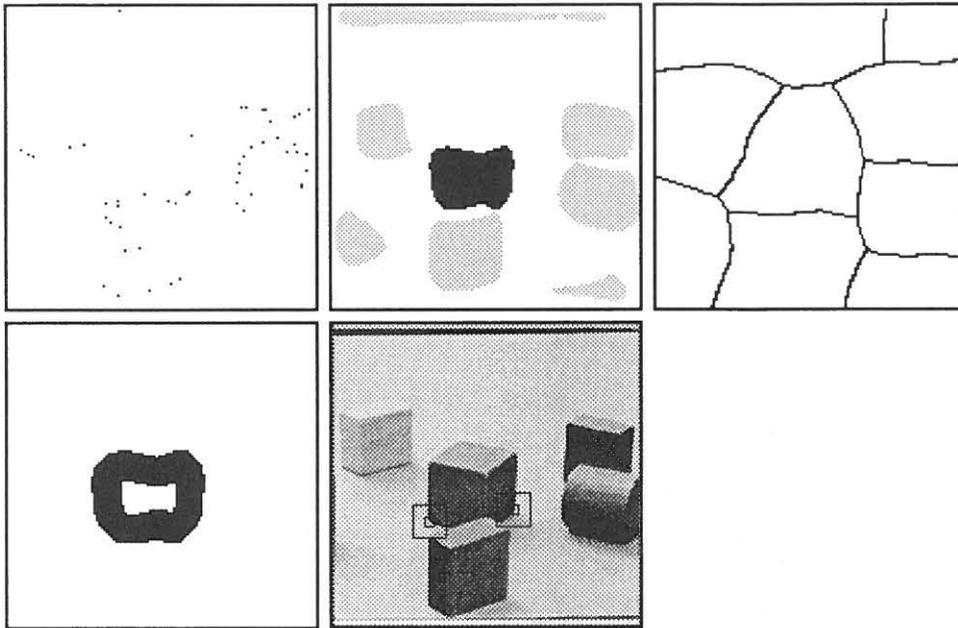


Figure 9.19: *Illustration of the matching procedure between blobs and interest points for the toy block image. (a) The 50 most significant interest points obtained with the Moravec interest point operator. (b) A significant blob extracted by the scale-space primal sketch (black) and the other grey-level blobs (grey) at the scale level given by the scale-space blob. (c) Voronoi diagram of the previous grey-level blob image (Criterion 1). The boundaries between different regions are marked with black. (d) Stripe around the boundary of the blob. The width of the stripe has been set to a characteristic length at the current level of scale (Criterion 2). (e) The interest points matched to the blob. (f) Resulting minimum and maximum window size around two of the interest points as set from the data given by the blob information.*

used as extensively for the interest points as for the edges. In the case with edges, the detection step was performed at the same scale as the blob manifested itself, which meant that interfering structures at finer levels of scale had to a large extent been suppressed by the scale-space smoothing, something that simplified the matching problem considerably. Here, the interest points are detected directly from the raw grey-level image at the finest level of scale without any use of scale knowledge. However, we are currently investigating different promising approaches for including this information already in the phase of detecting the interest points, with the purpose of reducing the number of false blob-point matches and, possibly also, obtaining more reliable points of interest. By and large, we are considering three main strategies:

- Incorporate some gradient information in the matching as well. One possibility could be to first compute edges at the scale given by the scale-space blob, follow the edges to finer scales using edge focusing and then carry out a matching between the interest points and the edges, which in turn would give the matching relations between the interest points and the blobs.
- An extension of this method could be to start from the edges delivered by the edge focusing algorithm and then detect interest points in a neighbourhood of appropriate size around those. Then, also the operator size for the interest point operator could be set from the coarse scale information.
- Investigate if the interest points can be computed at the scale given by the scale-space blob. Some experiments with this approach will be described in Section 9.3.5.

9.3.3 Computing Window Size from Blob Size

Once we have a relation between a blob and an interest point, we fit an ellipse to the blob in order to get a characteristic length associated with the blob. The ellipse is given by a 2×2 correlation matrix around the center of gravity of the blob support region.

$$\begin{pmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{pmatrix} \quad (9.4)$$

The eigenvalues λ_1, λ_2 of this matrix are extracted from

$$\lambda^2 - (C_{xx} + C_{yy})\lambda + C_{xx}C_{yy} - C_{xy}^2 = 0 \quad (9.5)$$

Then the minimum and maximum window widths are set to some constants ($\alpha_{min} \approx \frac{1}{3} - \frac{1}{2}$ and $\alpha_{max} \approx 2 - 3$) times the lengths of the shorter and longer semi-axes respectively.

$$w_{min} = \alpha_{min} \min(\lambda_1, \lambda_2); \quad w_{max} = \alpha_{max} \max(\lambda_1, \lambda_2) \quad (9.6)$$

Figure 9.19 shows a set of windows computed for a few interest points from the toy block image. This information constitutes the input data for the focusing procedure.

9.3.4 Experimental Results

Figure 9.20 shows the result of applying the composed classification procedure to a junction candidate near a corner of one of the dark blocks from the toy block image. We display the minimum and maximum window sizes as set from the blob information, together with the

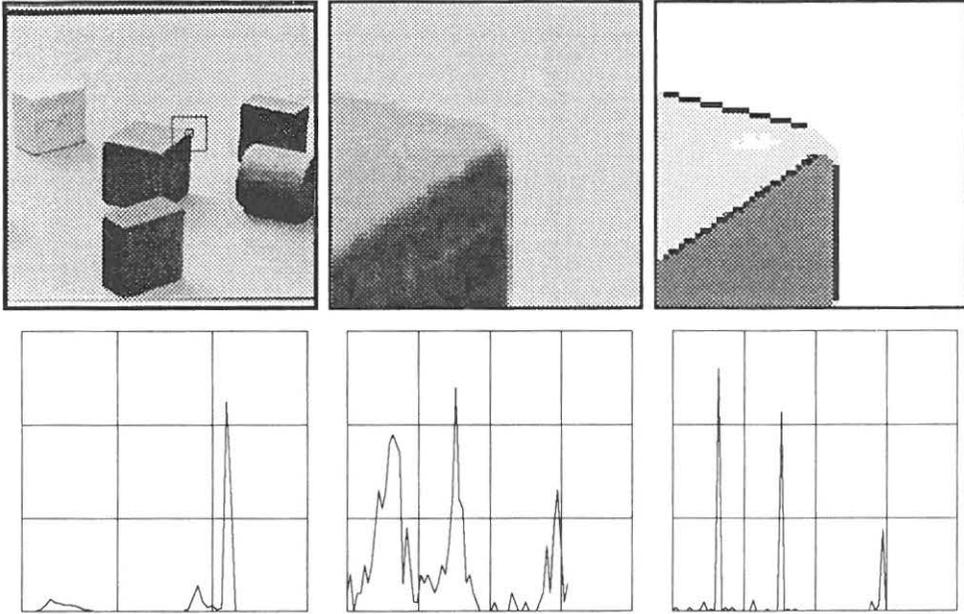


Figure 9.20: Illustration of the results of applying the classification procedure to a junction candidate near a corner of one of the dark blocks from the toy block image. (a) Maximum and minimum window sizes as set from the blob information. (b) Enlargement of the corner for a window size taken as representative of the classification. (c) Backprojections of the various regions in the final classification (together with the edges superimposed). (d) Grey-level histogram. (e) Directional histogram. (f) Peak-sharpened directional histogram. This junction was classified as a 3-junction since both the grey-level and the directional histograms contained three prominent peaks.

grey-level and directional histograms for a representative window size. For that window size, we also display an enlargement of the region around the corner as well as the backprojections of the different histogram peaks together with the superimposed edges. Observe that the noise level is much higher in the directional histogram than in the grey-level histogram, since the number of samples for the directional statistics is substantially smaller. This junction candidate was classified as a 3-junction, since three prominent peaks were found in both the grey-level and the directional histograms.

Figure 9.21 shows a more difficult situation with a detail from the hammer image. Here, the boundary between two of the regions in the corner is slightly curved, which implies that one of the directional peaks is relatively weak and widened. The point was classified as a T-junction, since three intensity peaks and two directional peaks were found in the histograms. More experiments with the method can be found in [Bru90b].

9.3.5 Detection of Candidate Junction Points Initiated by the Scale-Space Primal Sketch

What has not really been considered in the treatment above is the problem of actually detecting candidate junction points. We used Moravec's interest point operator, which leads to thresholding problems if applied uniformly all over an image. In addition, there

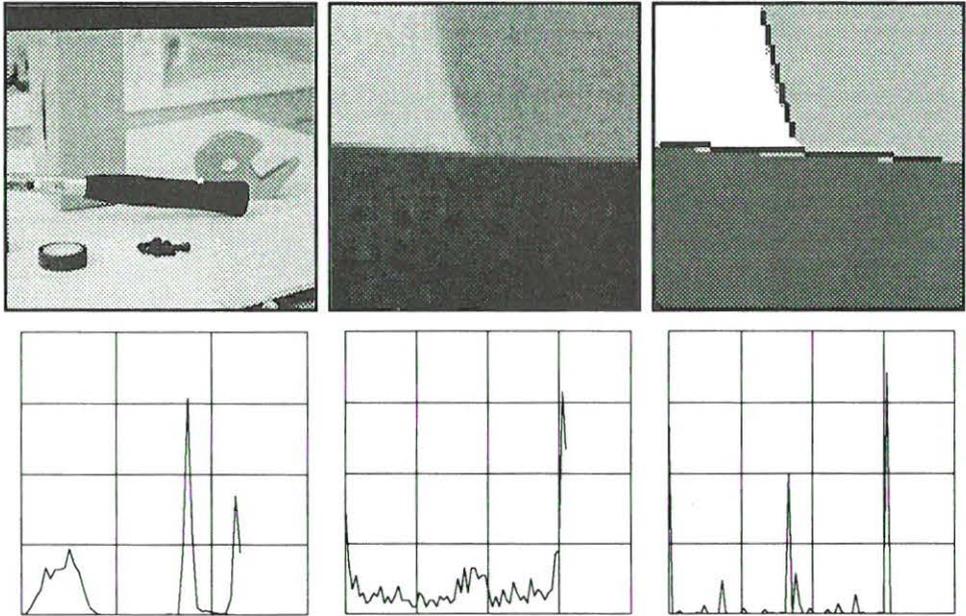


Figure 9.21: Illustration of the results of applying the classification procedure to a junction candidate in the scattered laboratory scene. (a) The point under consideration is marked with a white pad (near the junction between the upper edge of the handle of the hammer and the left edge of the tape reel). (b) Enlargement of the corner for a representative window size. (c) Backprojections of the various regions in the final classification (together with the edges superimposed). (d) Grey-level histogram. (e) Directional histogram. (f) Peak-sharpened directional histogram. This junction was classified as a T-junction, since three peaks were found in the grey-level distribution and two peaks in the directional histogram.

is one scale problem we have neglected, namely, that of the scale at which the interest points should be detected. As we discussed in Section 9.3.2.1, realistic corners are usually rounded, which means that small size operators will have problems in detecting those from the original image. Moreover, we faced problems when matching interest points to blobs, to a large extent because no scale information was included in the matching procedure.

Therefore, one would like to make use of the scale and region information already in the phase of detecting the interest points. In other words, we would like to detect the interest points at a coarser scale in order to simplify the detection and matching problems. Now, this poses another problem. Corners are usually treated as pointwise properties and are therefore regarded as very fine scale features. At first glance, smoothing the image before detecting such points seems like a contradiction, because of the risk that important interest points disappear by this operation. However, for detecting coarse scale corners, corresponding to the rough outline of say a polygon-like object, this approach can be applicable for finding the major corners, provided that the intensity contrast is sufficient. Therefore, it is desirable to have an interest point operator with a good behaviour in scale-space. A quantity with

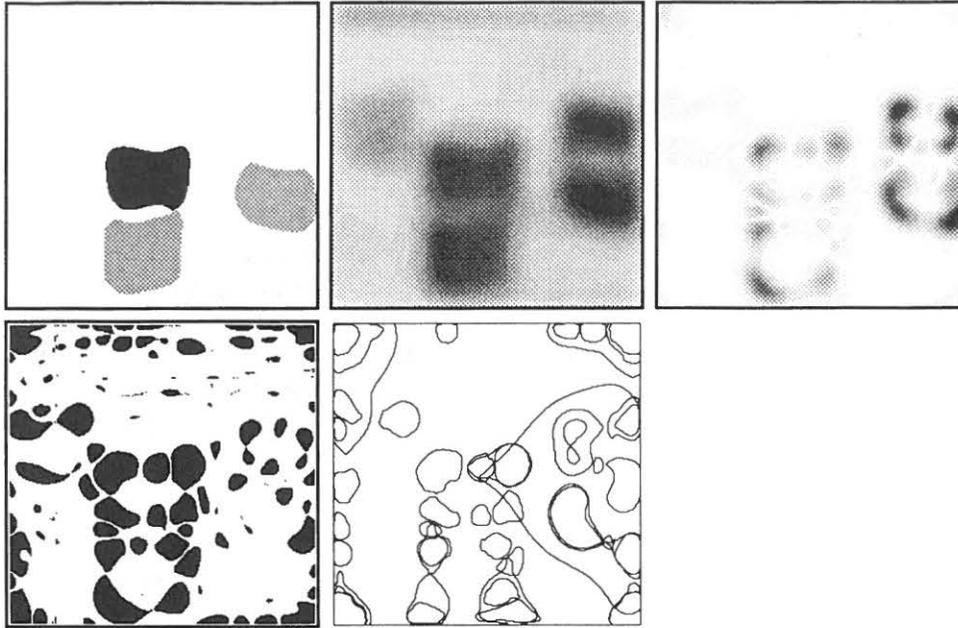


Figure 9.22: Illustration of the result of applying the level curve curvature operator at a coarse scale given by a significant blob from the scale-space primal sketch. (a) A significant scale-space blob from the scale-space primal sketch (marked with black). (b) The smoothed grey-level image at that level of scale. (c) The absolute value of the rescaled level curve curvature. (The image has been inverted so that dark regions correspond to high values). (d) Raw grey-level blobs detected from that image. (Note the high noise sensitivity in detecting blobs from a single level of scale). (e) Boundaries of the support regions of the 50 most significant scale-space blobs detected from the same image.

reasonable such properties is the *rescaled level curve curvature* given by

$$\tilde{\kappa} = |L_{xx}L_y^2 + L_{yy}L_x^2 - 2L_{xy}L_xL_y| \quad (9.7)$$

This expression is basically equal to the curvature of a level curve, which can be expressed as

$$\kappa = \frac{L_{xx}L_y^2 + L_{yy}L_x^2 - 2L_{xy}L_xL_y}{(L_x^2 + L_y^2)^{3/2}} \quad (9.8)$$

The level curve curvature has, however, been multiplied with the gradient magnitude¹¹ as to give a stronger response where the gradient is high. The motivation behind this approach is that corners basically can be characterized by two properties: (i) high curvature in the grey-level landscape and (ii) high intensity gradient. Using just the level curve curvature is not sufficient, since then a large number of false alarms would be obtained in regions with smoothly varying grey-level intensity. By taking the absolute value of the curvature, we treat positive and negative in the same way. Different versions of this operator, usually the level curve curvature multiplied by the gradient magnitude raised to the power of one, have

¹¹Raised to the power of 3 (to avoid the division operation).

been used by several authors, see e.g. Kitchen, Rosenfeld [Kit82], Koenderink, Richards [Koe88], Noble [Nob88], Blum [Blu90], and Deriche, Giraudon [Der90].

Figure 9.22 shows the result of applying this procedure to a blob extracted from the toy block image. In (a) we show the treated blob, in (b) the grey-level image at the scale given by the scale-space blob and in (c) the rescaled level curve curvature computed in this way¹². Figure (d) displays the result of applying raw grey-level blob detection to the curvature image and (e) the boundaries of the 50 most significant scale-space blobs extracted from this data. One observes that curvature operation gives a relatively strong response near the corners of the treated blob, and that the blob-like regions detected from that data give coarse indications of where one search for candidate junction points. A coarse estimate of the position of the candidate junction can be obtained from the curvature extremum in the blob region. In our further work we will investigate if it is possible to localize the interest point to finer scales in a way similar to edge focusing, or if the interest points can be computed within the blob support region directly from the data, given the coarse scale information.

9.3.6 Summary and Discussion

We have outlined how the scale-space primal sketch can be useful in dynamic situations like focus-of-attention. We believe that such mechanisms are necessary in computer vision systems, if they are to perform their tasks in a complex, dynamic world. More specifically, we discussed how the scale-space primal sketch together with foveation, which means examining selected regions of the visual world at high resolution, can be incorporated in an active vision system. The main reason to why foveation is carried out is because the resolution in normal overview images will not always be sufficient to clearly resolve the fine-scale structures under study. Further motivations for this methodology are given in [Bru90b].

In this integration of the scale-space primal sketch with junction classification, the scale-space blobs can be used for controlling the classification procedure as well as when detecting the junction candidates. The approach is similar to the blob-initiated edge focusing described in Section 9.1, in the sense that coarse hypotheses are generated about where to look, with associated coarse size information. However, there is still more work to be done in order to integrate these processing steps into a reasoning system.

9.4 Example: Analysis of Aerosol Images

As one example of how the scale-space primal sketch can be used for various image analysis tasks, we will in this section briefly describe a specific application that has arisen from a physical problem. We will be concerned with the analysis of a certain type of high-speed photographs of aerosols generated by nozzles for fluid atomization. A typical example of such an image is shown in Fig 9.23. What one perceives are some kind of clusters in the drop distribution, seemingly periodically spaced in the spread direction of the aerosol. If these events really exist, then the physical interpretation would be that there are periodical

¹²In the discrete implementation of this operation, the first order derivatives have been approximated by central differences, the second order derivatives with respect to x and y by the three-point operator ∇_3^2 and the mixed derivative by repeated application of the central differences. In other words, the discrete curvature is computed directly from the discrete N-jet representation by pointwise operations (without any need for nearest-neighbour communications).

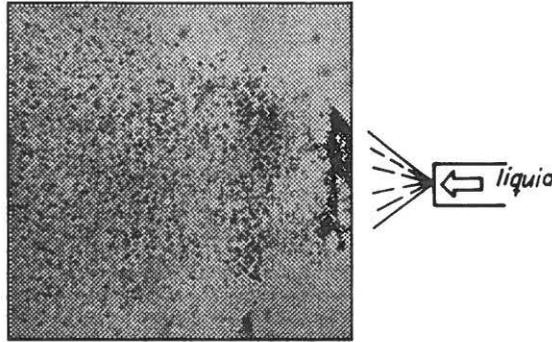


Figure 9.23: Photograph of an aerosol generated by a nozzle for fluid atomization (fuel injector). The time of exposure is approximately 20 nanoseconds.

(or oscillatory) phenomena taking place in the fuel atomization process. This is a theoretically interesting question, important for the deeper understanding of the combustion processes in combustors¹³. Usually it is assumed that fuel injectors produce aerosols with a relatively uniform droplet distribution, but the high-speed photograph in Figure 9.23 seems to indicate that this is not always the case. One may speculate that the occurrence of such non-uniformities could represent a possible driver for abnormal combustion events, which in turn could result in a deteriorated emission situation possibly affecting the exhaust production and/or the fuel consumption. However, it is not easy to say directly that these periodic structures really are there, since they correspond to coarse scale phenomena while the dominating kind of objects in the image is small dark blobs, i.e., fine scale phenomena. Therefore it is of interest to develop objective methods for analysing these structures.

Here we will demonstrate in a straightforward manner that: (1) these structures can be *enhanced* by a scale-space representation of the image and (2) they can be *extracted automatically* with the scale-space primal sketch.

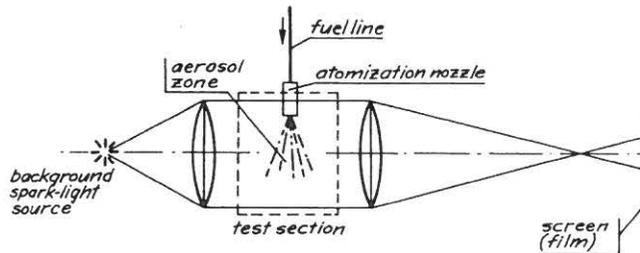


Figure 9.24: Schematic view of a shadowgraph optical system used in the physical experiments for acquiring the aerosol images. The fluid, subjected to a pressure of about 0.1 MPa, enters the nozzle and becomes atomized. The short exposure time is accomplished by performing the experiments in a dark room and illuminating the test section with a short flash.

9.4.1 Experimental Results

¹³ Further information about the physical background to the problem can be found in [Lin90f] and [Val89a, Val89b, Val89c].

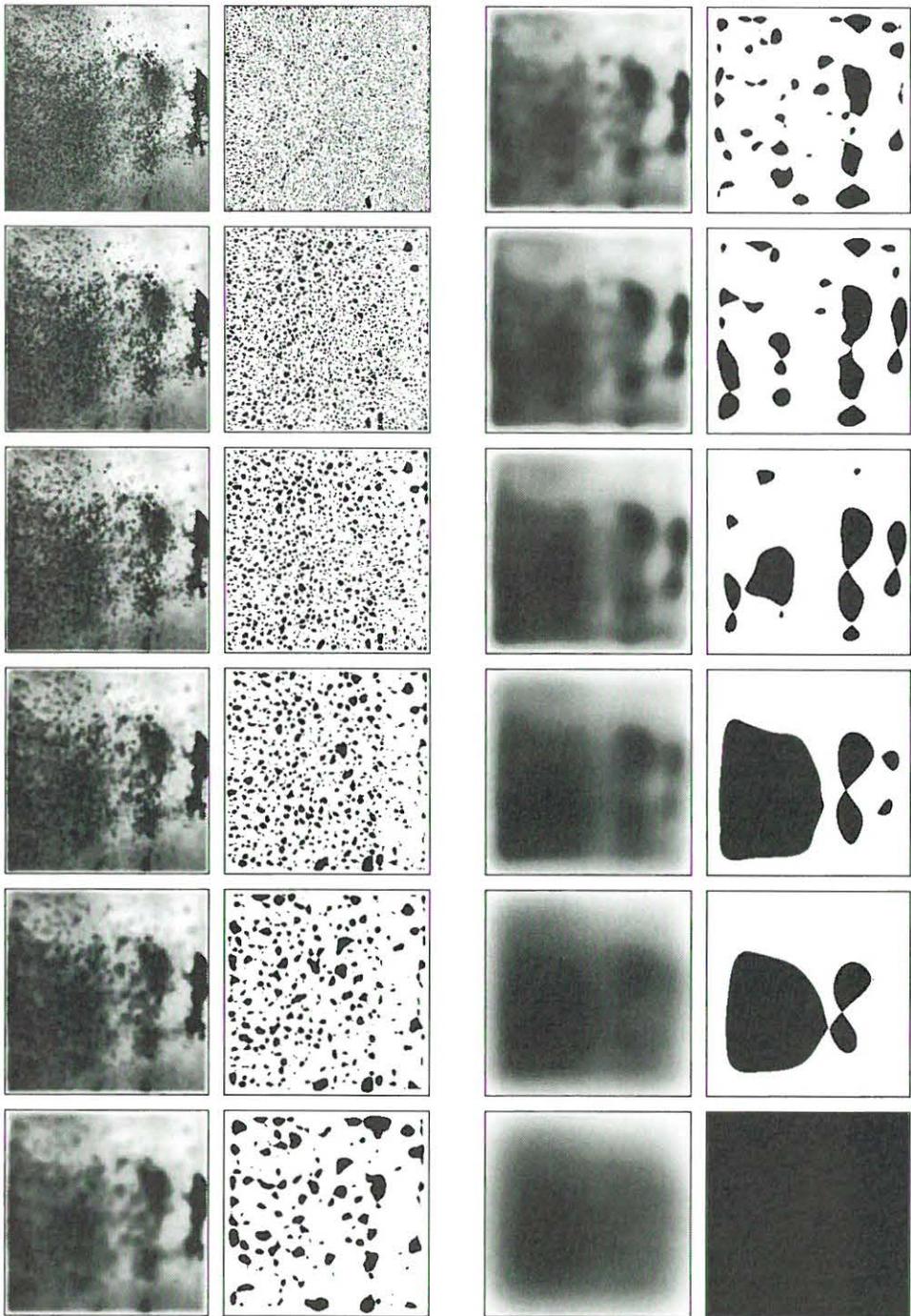


Figure 9.25: Grey-level and dark grey-level blob images of the aerosol image at scale levels $t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 (from top left to bottom right).

It should be stressed that these data are extremely irregular with a very high noise level.

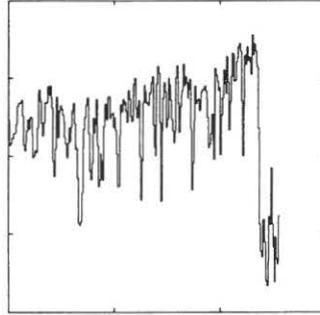


Figure 9.26: *Intensity variations in a central cross-section along the spread direction of the aerosol.*

Figure 9.26 displays the intensity variations in a cross-section of the image along the spread direction of the aerosol. Therefore, one could expect conventional segmentation techniques to have problems when applied to this type of data.

In Figure 9.25 we show the resulting scale-space representations together with the extracted blobs for a set of (logarithmically distributed) scale levels. As we can see, the drop clusters, that we earlier perceived as periodic structures in the original image, now appear as large dark blobs at the coarser levels of scale ($t = 128, 256, 512$). Although the scale-space representation enhances these clusters, we still rely on a visual and subjective observer in order to extract and verify the existence of these periodic phenomena. Some natural questions that were raised from the application point of view were:

- Can any one(s) of these smoothed images be regarded as a proper description(s) of the original image ?
- Which blobs can be regarded as significant structures in the image ?

In Figure 9.27 we display the result of extracting the 50 most significant dark blobs from this image together with the boundaries of the significant blob. One can observe that the periodically occurring drop clusters we perceived in the image are detected as significant structures in the scale-space primal sketch. Since, in contrast to many other methods used in image analysis, this method is essentially free from ad hoc “tuning parameters”, and arbitrarily selected error criterions or thresholds, we feel that the features detected by this algorithm can be regarded as reflecting inherent properties of the image — they are not just enforced effects of the analysis method.

9.4.2 Conclusions

We have seen that the scale-space primal sketch concept is a useful tool for automatic extraction of those periodic structures that were brought out by a scale-space representation of the aerosol image. Further experiments with this approach for analysing aerosol images are given in [Lin90f]. This presentation is mainly intended to demonstrate the potential of using the scale-space primal sketch as a primary tool in this kind of image analysis applications. Of course, more work needs to be done in order to arrive at a fully automated analysis method for this particular problem. Nevertheless, we believe that there is a potential in the approach also for other kinds of very noisy or irregular data, as e.g. medical imagery.

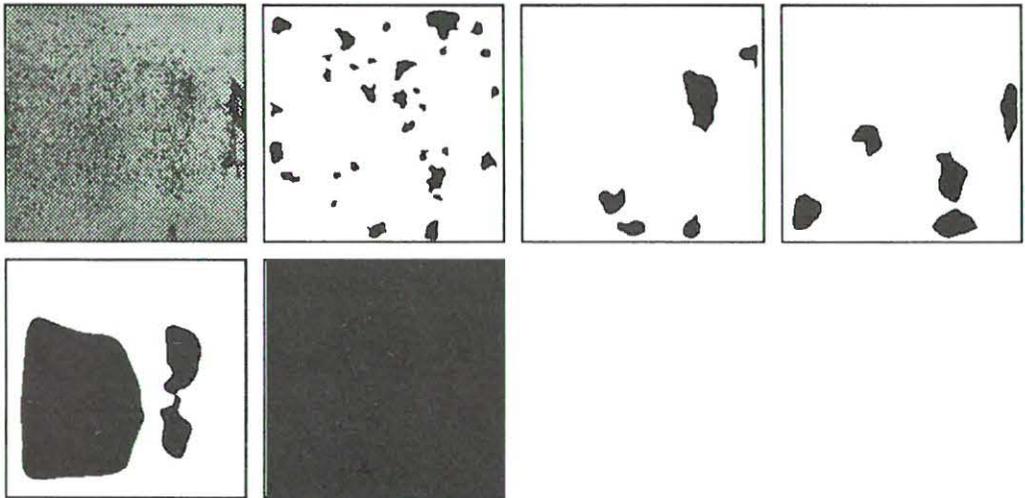


Figure 9.27: *Original aerosol image and the 50 most significant dark blobs determined from the scale-space primal sketch.*

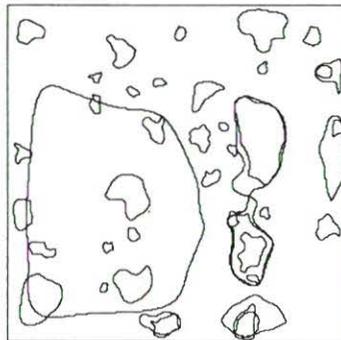


Figure 9.28: *The boundaries of the 50 most significant dark blobs in the aerosol image.*

9.5 Other Possible Applications

Let us finally mention a few other problem areas, where we believe that this approach can also be applicable:

9.5.1 Texture Analysis

A basic problem in many shape from texture algorithms concerns how to detect texture elements, *texels*. We believe that the blobs extracted from the scale-space primal sketch can be useful for such extraction in cases when the texture elements are blob-like. Compared

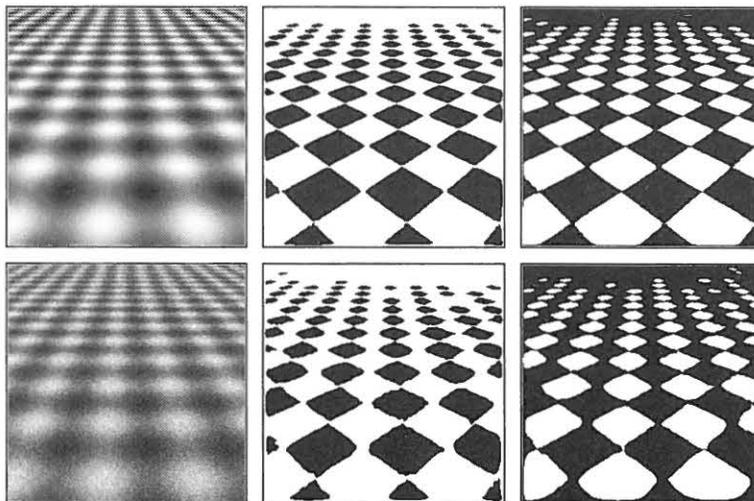


Figure 9.29: *The result of applying the scale-space primal sketch to a synthetic texture image generated from perspective projection of a planar surface painted with a sinusoidal grey-level (of the form $f(\xi, \eta) = \sin \xi \sin \eta$). (a) Grey-level image with 1 % added noise. (b) The 75 most significant dark blobs. (c) The 75 most significant bright blobs. (d) Grey-level image with 10 % added noise. (e-f) The 75 most significant dark blobs. (A few large blobs corresponding to coarse scale groupings have been suppressed as to simplify the presentation.) (The noise levels refer to uncorrelated point noise with normal distribution, where the percentage values relate the standard deviation of the noise to the maximum range of grey-level values in the original image).*

to e.g. the approach by Blostein and Ahuja [Blo87], where texture elements are detected based on the zero-crossings of the Laplacian of the Gaussian at a small set of pre-specified scale levels, this method does not require any such prior scale information. Further, the scale levels are automatically adapted to the size variations over the image. In Figure 9.29 we show the result of applying this blob detection scheme to a synthetic regular texture¹³. Figure 9.30 shows a less regular example, where the plane has been painted with a sum

¹³The image has been generated from an infinite planar surface painted with a grey-level intensity of the form $f(\xi, \eta) = \sin \xi \sin \eta$. After the perspective projection step, Gaussian noise of different amplitude has been added to the grey-level image. The extracted patterns consist of square-like regions, since in the ideal noise-free case the delimiting saddle points of the grey-level blobs will be located in the points $(\xi_n, \eta_m) = (n\pi, m\pi)$ on the plane, and the level curves through these points will be straight lines.

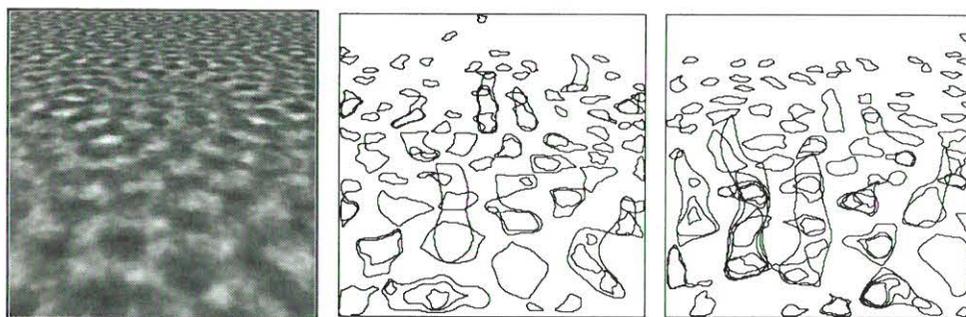


Figure 9.30: *The result of applying the scale-space primal sketch to a synthetic texture image generated from perspective projection of a planar surface painted with an intensity distribution generated from a sum of sinusoidals of random phase. (a) Grey-level image. (b) Boundaries of the 75 most significant dark blobs. (c) Boundaries of the 75 most significant bright blobs.*

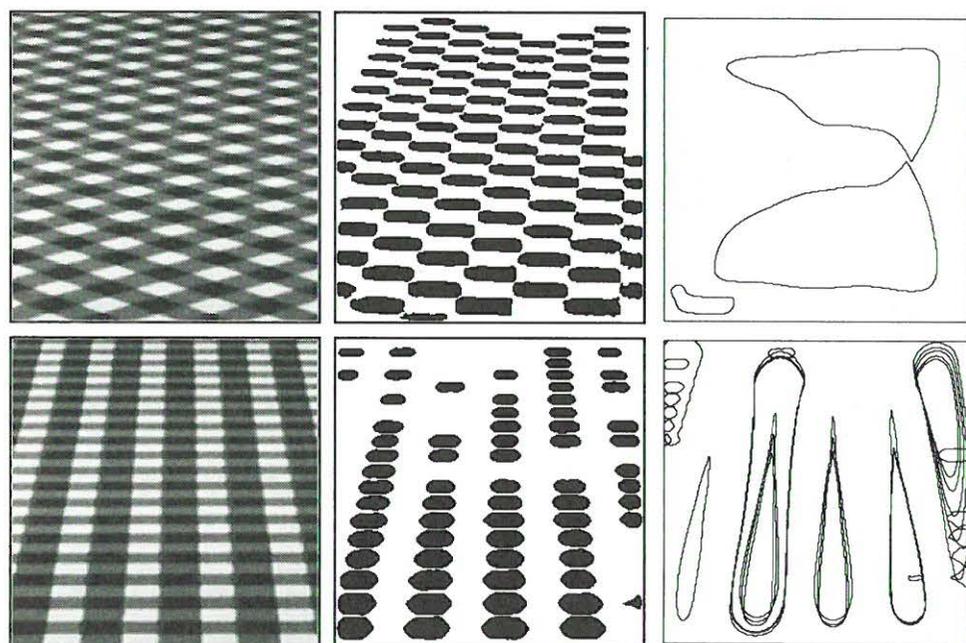


Figure 9.31: *The result of applying the scale-space primal sketch to two different views of a textured wallpaper with squares of three different grey-levels. (a) Grey-level image. (b-c) The 75 most significant dark blobs (marked either as blobs or as blob boundaries). (d) Grey-level image. (e-f) The 75 most significant dark blobs (marked either as blobs or as blob boundaries). Similar patterns are extracted when detecting bright blobs from these images.*

of several sinusoidals of different amplitude and phase. Note that in both these cases the algorithm extracts a set of blobs with a size gradient that could provide a cue to the three-dimensional structure.

Figure 9.31 show corresponding results for two different projections of the same wall-paper, with a texture consisting of squares of three different grey-levels. Observe that in the first case the algorithm mainly ranks blobs corresponding to squares on the wall-paper as important, while in the second case both the individual squares and a set of blob groupings are extracted¹⁴. Because of this phenomenon, some of the dark squares in the second image are no longer among the selected number of the most significant blobs.

When using these blobs as primitive cues to the three-dimensional structure, there are a few problems that must be considered. Given a set of blobs obtained from the scale-space primal sketch, with varying size and different significance values, one has to determine whether the size of some reasonable subset of the blobs varies in a way consistent with the projection of a three-dimensional surface. Some coarse scale groupings or fine scale blobs due to noise may have to be suppressed from the analysis. Another problem is that the scale-space smoothing can have introduced shape distortions of the blobs, which for instance will reduce the foreshortening effect. Therefore, some improvements of the localization of the blob boundaries (e.g. by combination with edge detection) may be needed in order to reduce the systematic underestimation of the slant that would otherwise occur. See e.g. Gårding [Går91] for an extensive treatment of the shape from texture problem.

9.5.2 Perceptual Grouping

We have seen that the blobs extracted from the scale-space primal sketch induce a perceptually reasonable grouping of various patterns. For example, in Figure 9.31(a) in principle only the individual squares were ranked as important, while in Figure 9.31(d-f) also the lines one perceives when looking at the image were found. See also the dot pattern in Figure 8.3. Note that the grouping operation is not given by any set of pre-specified logical rules, but by a process generated from a differential equation, which has been combined with a set of geometric constructions.

9.5.3 Matching

As we described in Section 9.1.5 above, we believe that the blobs delivered from the scale-space primal sketch can serve as coarse landmarks for different types of matching purposes. The relations given say by matches between a blob and a set of edges together with similar matches between the same blob and a set classified junctions provides a sparse feature set that could be used for simplifying e.g. object model matching. Another possible application is to use blobs for initiating deformable models like those proposed by e.g. Terzopoulos et al. [Ter86, Ter87, Ter88, Kas87, Wit87] or Pentland [Pen88, Pen90]. In addition, these blobs could possibly serve as to establish a coarse correspondense between regions from different images of the same scene, a problem arising in stereo and motion analysis. In fact, a type of similar approach has been recently used for motion matching by Koller and Nagel [Kol90]. If these blobs are to be used for stereo matching, then of course, if actual disparity measurements are needed, the computations must be based on better localized features like e.g. edges or corners.

¹⁴These higher order groupings take place mainly along the line of sight, since objects are closer to each other in the image measured in this direction than in the perpendicular direction.

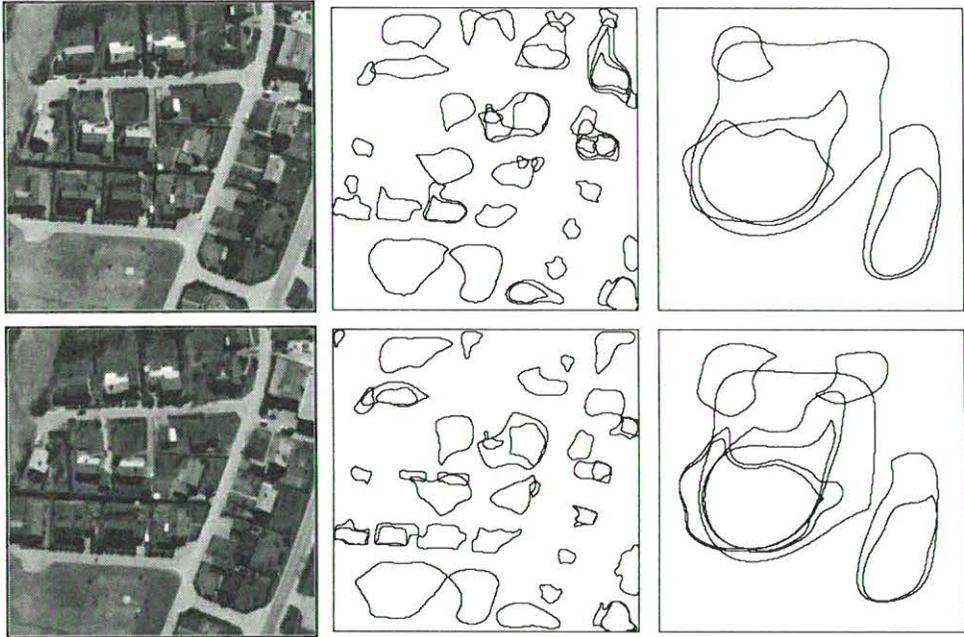


Figure 9.32: *Extracted dark scale-space blobs from a stereo pair (an aerial photograph of a suburb). (a) Left grey-level image. (b-c) Boundaries of the 50 most significant dark blobs extracted from the left image. (d) Right grey-level image. (e-f) Boundaries of the 50 most significant dark blobs extracted from the right image. (In order to simplify the presentation, the blob boundaries have been drawn in two different images instead of one.) (The upper row corresponds to the left image and the lower row to the right image.)*

Chapter 10

Summary and Discussion

10.1 The Scale-Space Primal Sketch

The representation that we build is similar to the primal sketch suggested by Marr [Mar76, Mar82], in the sense that it is a two-dimensional representation of the significant grey-level structures in the image. It is also computed under extremely weak assumptions. However, besides that it is a region-based and not an edge-based representation it is more *qualitative*, without strong assumptions about the shape of the primitives. Moreover, the proposed representation consists of coarse features like blobs represented at multiple scales and allows for

- Automatic detection of salient (stable) scales, if they exist.
- Ranking of events in order of significance.
- Generation of hypotheses for grouping and segmentation.

This implies that candidate regions for further processing are generated, as well as information about their scale. We see that the representation gives clues to subsequent analysis and can, hence, guide focus-of-attention mechanisms and simplify later stage processing. At the same time it is obtained with no a priori assumptions and, in principle, with no tuning parameters.

10.1.1 Qualitative Properties

We have also tried to demonstrate the effects of one as we believe very promising methodology, namely that *simple methods and qualitative reasoning can perform surprisingly well if the treatment is performed at a proper scale and over an appropriately selected region in space, provided that the resolution is sufficient to clearly resolve the phenomena we are studying*¹. For instance, the primitives (grey-level blobs, scale-space blobs and edges from non-maximum suppression) used for extracting image structure were defined solely in terms of *singularities and geometric properties* in scale-space. These entities can be very noise sensitive when considered at a single level of scale only. However, here we have shown that they can give useful results if combined with a careful treatment of the scale issue.

¹In a major part of the work we have assumed that the images have been acquired with a sufficient resolution. However, in Section 9.3 we indicated how this issue can be further coped with in an active vision context.

Of course the actual numerical values cannot be trusted, since the amplitude of the Gaussian derivatives will in general decrease by the scale-space smoothing. Therefore, in order to avoid a systematic bias if accurate values are required for computing *quantitative* properties related to the environment, as needed in many “shape from X” methods, we believe that a two-stage process could be applied: (1) First detect the qualitative type of actual situation. (2) Then fit a model, corresponding to the situation at hand, to the data (over a region in space determined from the first step).

10.1.2 Extraction of Structure — Transformational Invariance

The underlying principle we use when extracting image structures is that structure should be invariant under transformations in parameter space. Our method consists of three steps:

- Vary the parameters systematically.
- Detect (locally) stable states (intervals).
- Choose a representative descriptor as an abstraction of each stable interval and pass only this information on to the higher level modules.

In this specific case the parameter we vary is the scale parameter in the scale-space representation. However, we believe that a similar kind of methodology could be applicable also in other types of situations.

10.2 Scale-Space Experiences

Let us point to a few aspects of scale-space representations that have been given little or insufficient attention in the literature and that have to be dealt with in creating a representation of the sort we want.

10.2.1 Suppression of Local Extrema due to Noise

First, it is noteworthy, that the amount of noise in real images usually leads to a large number of local extrema. These extrema may disappear rather early, provided that they are subsumed by some more prominent extremum. However, if they occur in a region with smoothly varying grey-levels, then they will exist over a large range of scale. This effect is alleviated, but not remedied, by annihilation between nearby noise extrema. Even though their amplitudes decrease rapidly, it is not clear that one can set a threshold on objective grounds. This problem is related to the issue of estimating the noise level in an image, which hardly can be addressed without some constraining assumptions, like e.g. in Voorhees and Poggio [Vor87].

10.2.2 Stable Scale is a Local Property

Another property, indicated in Section 5.4, is that images of scenes of even moderate complexity rarely have a global scale, at which all structure above the noise level is present. This aspect is explicitly dealt with in our representation. Stable scales are local properties associated with objects not with entire images. Bischof and Caelli [Bis88] treat a similar question for zero-crossings. However, their measure of stability seems to be more arbitrary.

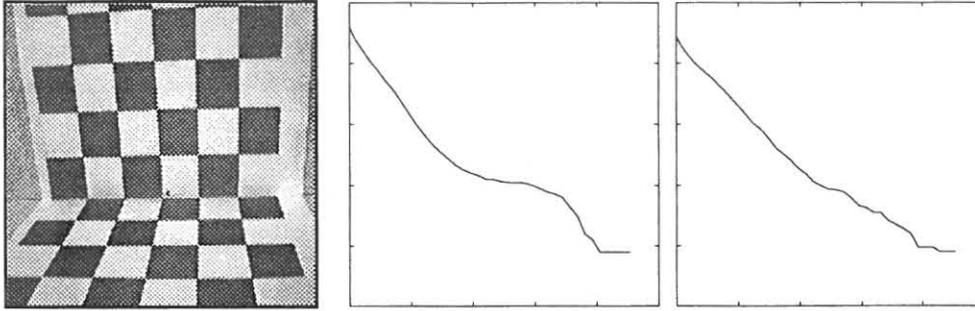


Figure 10.1: (a) An unusual situation, where one could possibly talk about a global scale for a whole image. (b) This property appears as a plateau in a graph showing the logarithm of the number of local extrema as function of (the effective) scale. (c) For images of moderate complexity it will, however, usually not be possible to find such globally stable states. Even if there were a number of prominent plateaus corresponding to local structures at different scales, by adding up several such profiles one will anyway obtain a relatively uniformly decreasing curve. The graph in (c) shows the logarithm of the number of local extrema as function of scale for the Godthem Inn image.

10.2.3 Stable Scale is a Multi-Valued Function

Moreover, given some region in space there may in fact be several stable scales associated with that region corresponding to structures on different scale. Therefore, if one attempts to assign a property “stable scale” to every point in an image, one will obtain a “function” that in principle may assume an arbitrary number of values in each point. Therefore, the task of finding “the best scale” for treating a certain *point* in an image is in general an impossible problem, which cannot be solved except for very simple images, for which there is in fact only one such stable scale related to each point in the image (within the scale interval delimited by the inner scale and the outer scale).

10.2.4 Decreasing Amplitude of Feature Points

The behaviour of local extrema in scale-space has been studied also by Lifshitz and Pizer [Lif87, Piz88]. They link points across scales based on *iso-intensity* similar to the projection between scales described in [Koe84] and define blob regions in terms of watersheds. However, this leads to a serious problem of *non-containment*, which basically means that a point, which at one scale has been classified as belonging to a certain region (associated with a local maximum), can escape from that region when the scale parameter increases. More precisely, what can happen is the following: Assume that a point A is contained in a region associated with an extremum B at a certain scale and that we follow these points by iso-intensity linking to corresponding point A' and B' at a coarser scale. Then, we are not guaranteed that A' is contained in the same region as B' . Even worse, these paths can be intertwined in a rather complicated way, which means that the relations between extremum regions across scales can hardly be regarded as hierarchical.

These problems will not occur with our proposed way of linking blobs across scales, which is solely based on qualitative features. The main problem with the iso-intensity linking is that the grey-level value associated with a feature, say a critical point, will in general be

changed by the scale-space smoothing. By connecting a feature point existing at one scale to the nearest point at an adjacent scale having the same grey-level value, one makes a small error, which will accumulate, and cannot be neglected².

This is another illustration of the property that in general the magnitudes of grey-levels or derivatives cannot be trusted after scale-space smoothing, since the amplitude of a signal or its derivatives will in general decrease by this operation. Only qualitative features (or invariants) such as edges and local extrema, which can be defined in terms of singularities, can be used.

10.3 Relations to Previous Work

There are, of course, earlier attempts to derive similar representations of the grey-level landscape. Rosenfeld and his co-workers, see e.g. [Gro86, She86, She87] have studied blob detection in pyramids e.g. using relaxation methods. Blostein and Ahuja [Blo87] detect texture elements based on zero-crossings and use multiple scales and a significance measure based on a background noise assumption. There is also the wealth of literature on pyramids, see e.g. Levine [Lev80], Crowley and Parker [Cro84a], Crowley and Sanderson [Cro87] and Burt and Adelson [Bur83]. The texton theory proposed by Julesz, see [Jul83, Jul86] and [Vor87], essentially also treats the blob detection problem. There are finally a number of representations based on intensity changes, besides Marr [Mar82], Bergholm [Ber87, Ber89] and Watt [Wat88] and approaches working at higher levels like Saund's [Sau89, Sau90] token based symbolic grouping. Of interest is also the approach by Haralick et.al. [Har83], which allows a more detailed representation, but only at a single spatial scale.

Our approach differs from these in three important aspects. First, our representation can be seen as preceding e.g. the edge-based schemes in that it selects the appropriate scales and regions, intrinsically defined by the image itself, in a complementary data-driven manner. Secondly, it is a hierarchic representation of the structure at *all scales* in the image with explicit information about their significance and relations, and a competition between parts at different locations and scales. Finally, it is derived in a formal way using the well-defined notions of scale-space, which allow a precise analysis of the behaviour of structure. Hence, we can study how events at different scales can be related in a well-defined manner.

One can ask more generally what is the relation between our representation and zero-crossings of the second derivative. We suggest that our representation, with extrema and their extents, captures important structure. The zero-crossings will not always be localized in the same places and, therefore, not represent the same structure. Watt [Wat88], in fact argues that the extrema of the second derivative, and not the zero-crossings, should be used to pick up information about intensity discontinuities. We feel that this question should be investigated further.

10.4 Conclusions

We have presented a multi-scale representation of grey-level image structure similar to the primal sketch idea and shown that it can be used for detecting stable scales and extracting

²A similar problem arises with the motion constraint equation in optical flow, where it is usually assumed that the intensity value of a physical point is preserved under motion. In fact, Pentland [Pen90] has recently shown that under certain conditions the photometric distortions can be much greater than the geometric effects due to motion.

regions of interest from an image in a solely bottom-up data-driven way, without any a priori assumptions about the shape of the primitives. The representation, which is essentially free from tuning parameters and ad hoc error criteria gives a qualitative description of the grey-level landscape with information about

- approximate location,
- spatial extent and
- an appropriate scale

for important regions in the image. In other words, it generates coarse but safe segmentation cues, and can be used as a hypothesis generator for higher-level processes. We have demonstrated that this kind of information can serve as to control an edge detection scheme working at a proper level of scale and that it is useful for automatic cluster detection and modality analysis of histograms. More generally, we find this approach useful for

- guiding the focus-of-attention and
- tuning other low-level processes.

The representation is based on a well-defined notion of blob, which gives a natural geometric measure of significance. It is also based on scale-space theory, which means a well-founded treatment of structures at multiple scales. The principle we follow when extracting significant image structure from scale-space is based on transformational invariance and consists of the following steps:

- Vary the parameters systematically and try to detect locally stable states (intervals).
- Choose a representative descriptor as an abstraction of each stable interval and pass only this information on to the higher level modules.

In this specific case the parameter we vary is the scale parameter in the scale-space representation. However, we believe that methodology can be applicable also in other types of situations.

The computational aspects of scale-space are treated carefully. Particularly, the fact that realistic images are discrete is taken into account, and we use a scale-space concept specially designed for discrete signals. The evolution properties in scale-space of local extrema and blobs are analysed in detail. We also introduce the notion of effective scale, which is the natural unit for measurement of scale-space lifetime.

More detailed summaries and conclusions for the different subproblems have been given at the end of each independent treatment.

Part IV
Appendix

Appendix A

Technical Details and More Examples

A.1 From Chapter 2

A.1.1 Unimodality of the Fourier Transform in the Non-Circulant Case

We are to generalize the unimodality property in the frequency domain, derived for circulant convolution, to non-circulant convolution transformations. Consider a finite support kernel $K : Z \rightarrow R$, with $K(i) = 0$ if $|i| > M$, having a non-unimodal Fourier transform. Will show that this kernel cannot be a discrete scale-space kernel.

From the proof of Proposition 2.10 (the circulant case) it is apparent that if a kernel has a non-unimodal Fourier transform, then for some sufficiently large (odd) integer T there exists a T -periodic signal f_{per} ($f_{per}(x) = f_{per}(x+T)$), for which the number of zero-crossings in one period of $f_{out} = K * f_{per}$ is strictly greater than the number of zero-crossings in a corresponding period of f_{per} . The signal $f_{per} : Z \rightarrow R$ is constructed from the vector \mathbf{x} (of length $T = 2M + 1$) used in the proof of Proposition 2.10 by $f_{per}(i) = \mathbf{x}_i$ ($i = 0..T - 1$) and periodic extension. By this construction, the effect of K on one period of f_{per} will be the same as the effect on the vector \mathbf{x} by the associated circulant matrix $C_C^{(M)}$, which was used in the proof of Proposition 2.10. With the number of zero-crossings in one period of a signal f , we here mean the number of zero-crossings in the sequence $f(0), f(1), \dots, f(T - 1), f(0)$, including wrap-around.

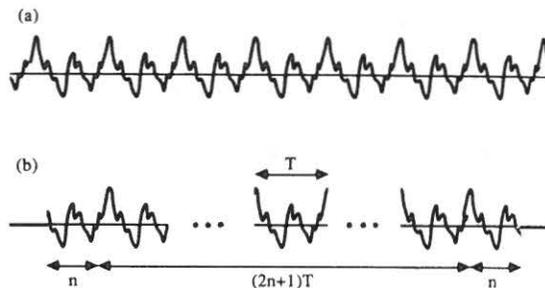


Figure A.1: Construction of the finite support signal f_{in} from the periodic function f_{per} .

Given this function f_{per} of period $T = 2M + 1$ let I_{2M+1} be an interval with $2M + 1$ such consecutive periods and construct a new function $f_{in} : Z \rightarrow R$, which is equal to f_{per} on I_{2M+1} and at the M nearest points outside each boundary of I_{2M+1} , see Figure A.1. At

all other points f_{in} should be zero. Due to the construction of f_{in} and the finiteness of K it follows that $K * f_{in}$ and $K * f_{per}$ will be equal on I_{2M+1} . Thus, provided that we only count the points in I_{2M+1} we have introduced at least $2M + 1$ additional zero-crossings.

Outside I_{2M+1} we might expect to find more zero-crossings in $K * f_{in}$. The support region of $K * f_{in}$ is in general larger than the support region of f_{in} . However, f_{in} cannot have more than a total of $2M$ additional zero-crossings since f_{in} is non-zero only at $2M$ points outside I_{2M+1} . Consequently, $K * f_{in}$ contains at least one zero-crossing more than f_{in} , which shows that K cannot be a scale-space kernel. This completes the proof of Proposition 2.11.

A.1.2 Positivity and Unimodality are Necessary but not Sufficient

We will show by counterexample that the positivity and unimodality properties in the spatial and frequency domains do not necessarily guarantee a kernel to be a discrete scale-space kernel. We will demonstrate this fact by considering symmetric five-kernels.

The case when at least one root is real and positive is not interesting, since then at least one filter coefficient would need to be negative. If one root is real and negative then also at least one more root must be real (since non-real roots occur pairwise) and can be assumed to be negative (because of the previous property). If the remaining second degree factor in the factorization of generating function has non-real roots, then the kernel cannot be positive in the Fourier domain (see the treatment in Section 2.3.3). The case we are interested in is when the generating function $\varphi(z)$ has only complex roots. Then $\varphi(z)$ can be written

$$\varphi(z) = \frac{(z + a + bi)(z + a - bi)(z + c + di)(z + c - di)}{z^2} = \quad (\text{A.1})$$

$$\frac{z^4 + 2(a + c)z^3 + (a^2 + b^2 + c^2 + d^2 + 4ac)z^2 + (2a(c^2 + d^2) + 2c(a^2 + b^2))z + (a^2 + b^2)(c^2 + d^2)}{z^2}$$

for some real a, b, c and d . To obtain a symmetric kernel we require $(a^2 + b^2)(c^2 + d^2) = 1$. Introducing polar coordinates

$$a = r \cos \alpha ; \quad b = r \sin \alpha \quad (\text{A.2})$$

$$c = \frac{1}{r} \cos \beta ; \quad d = \frac{1}{r} \sin \beta \quad (\text{A.3})$$

we have that $\varphi(z)$ can be rewritten as

$$\varphi(z) = z^2 + 2\left(r \cos \alpha + \frac{\cos \beta}{r}\right)z + \left(r^2 + \frac{1}{r^2} + 4 \cos \alpha \cos \beta\right) + 2\left(\frac{\cos \alpha}{r} + r \cos \beta\right)z^{-1} + z^{-2} \quad (\text{A.4})$$

If this kernel is to be symmetric then it is necessary that

$$r \cos \alpha + \frac{\cos \beta}{r} = \frac{\cos \alpha}{r} + r \cos \beta \quad (\text{A.5})$$

In other words, we get two cases from

$$(\cos \alpha - \cos \beta)\left(r - \frac{1}{r}\right) = 0 \quad (\text{A.6})$$

Case I: Consider first $\cos \alpha = \cos \beta$, which gives

$$\varphi(z) = z^2 + 2 \cos \alpha \left(r + \frac{1}{r}\right)z + \left(r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha\right) + 2 \cos \alpha \left(r + \frac{1}{r}\right)z^{-1} + z^{-2} \quad (\text{A.7})$$

The filter coefficients are positive if and only if $\cos \alpha \geq 0$. The unimodality requirement in the spatial domain

$$r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha \geq 2 \cos \alpha \left(r + \frac{1}{r}\right) \geq 1 \quad (\text{A.8})$$

can always be satisfied for sufficiently large r if $\cos \alpha \neq 0$. The Fourier transform is

$$\psi(\theta) = \left(r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha\right) + 4 \cos \alpha \left(r + \frac{1}{r}\right) \cos \theta + 2 \cos 2\theta \quad (\text{A.9})$$

Trivially we have $\psi(0) \geq 0$. Positivity in the other end point of the interval $[0, \pi]$ gives

$$\psi(\pi) = \left(r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha\right) - 4 \cos \alpha \left(r + \frac{1}{r}\right) + 2 \geq 0 \quad (\text{A.10})$$

The unimodality requirement implies that the equation

$$\psi'(\theta) = -4 \sin \theta \left(\cos \alpha \left(r + \frac{1}{r}\right) + 2 \cos \theta\right) = 0 \quad (\text{A.11})$$

must not have any real roots in the interior of $[0, \pi]$. Hence, one observes that ψ is unimodal if and only if

$$\cos \alpha \left(r + \frac{1}{r}\right) + 2 \cos \theta \neq 0 \quad (\text{A.12})$$

for all $\theta \in]0, \pi[$. A necessary and sufficient condition for unimodality is hence given by

$$\frac{|\cos \alpha|}{2} \left(r + \frac{1}{r}\right) \geq 1 \quad (\text{A.13})$$

To summarize, we have that the kernel is positive and unimodal both in the spatial and the frequency domains if these five inequalities are satisfied

$$\cos \alpha \geq 0 \quad (\text{A.14})$$

$$r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha - 2 \cos \alpha \left(r + \frac{1}{r}\right) \geq 0 \quad (\text{A.15})$$

$$2 \cos \alpha \left(r + \frac{1}{r}\right) - 1 \geq 0 \quad (\text{A.16})$$

$$\frac{|\cos \alpha|}{2} \left(r + \frac{1}{r}\right) - 1 \geq 0 \quad (\text{A.17})$$

$$r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha - 4 \cos \alpha \left(r + \frac{1}{r}\right) + 2 \geq 0 \quad (\text{A.18})$$

Obviously, (A.16) is comprised by (A.17) and can be omitted since $\cos \alpha > 0$. Similarly, by writing (A.18) on the form

$$r^2 + \frac{1}{r^2} + 4 \cos^2 \alpha - 2 \cos \alpha \left(r + \frac{1}{r}\right) - 2 \left(\cos \alpha \left(r + \frac{1}{r}\right) - 1\right) \geq 0 \quad (\text{A.19})$$

one observes that, since the rightmost parenthesis is strictly positive (due to A.17) we have that (A.18) holds whenever (A.15) is satisfied. Moreover, (A.15) can be rewritten as

$$\left(r + \frac{1}{r} + 2 \cos \alpha\right)^2 + 1 \geq 0 \quad (\text{A.20})$$

which shows that this inequality will always be satisfied. We conclude that a necessary and sufficient condition for such a kernel to be positive and unimodal both in the spatial and the frequency domains is that

$$\frac{\cos \alpha}{2} \left(r + \frac{1}{r}\right) \geq 1 \quad (\text{A.21})$$

Taking e.g. $\alpha = \frac{\pi}{4}$ and $r = 2\sqrt{2}$ we get

$$a + bi = 2(1 + i); \quad c + di = \frac{1}{4}(1 + i) \quad (\text{A.22})$$

and

$$\varphi(z) = (z + 2 + 2i)(z + 2 - 2i)(z + \frac{1}{4} + \frac{i}{4})(z + \frac{1}{4} - \frac{i}{4})z^{-2} = \quad (\text{A.23})$$

$$z^2 + \frac{36}{8}z + \frac{81}{8} + \frac{36}{8}z^{-1} + z^{-2} \quad (\text{A.24})$$

which obviously corresponds to a kernel that is positive and unimodal in the spatial domain. From the Fourier transform

$$\psi(\theta) = \frac{1}{8}(81 + 72 \cos \theta + 16 \cos 2\theta) \quad (\text{A.25})$$

and its derivative

$$\psi'(\theta) = -\sin \theta(9 + 4 \cos \theta) \quad (\text{A.26})$$

one can verify that the kernel is positive and unimodal in the frequency domain as well. However, from the characterization of discrete scale-space kernels in Section 2.4.2 we have that this kernel cannot possess scale-space properties, since its generating function has non-real roots.

Case II: When $r = 1$ we have

$$\varphi(z) = z^2 + 2(\cos \alpha + \cos \beta)z^1 + (2 + 4 \cos \alpha \cos \beta) + 2(\cos \alpha + \cos \beta) + z^{-2} \quad (\text{A.27})$$

The positivity and unimodality requirements in the spatial domain lead to the inequalities

$$\cos \alpha + \cos \beta \geq 0 \quad (\text{A.28})$$

$$2 + 4 \cos \alpha \cos \beta \geq 0 \quad (\text{A.29})$$

$$2(\cos \alpha + \cos \beta) \geq 1 \quad (\text{A.30})$$

$$2 + 4 \cos \alpha \cos \beta \geq 2(\cos \alpha + \cos \beta) \quad (\text{A.31})$$

From the Fourier transform

$$\psi(\theta) = (2 + 4 \cos \alpha \cos \beta) + 4(\cos \alpha + \cos \beta) \cos \theta + 2 \cos 2\theta \quad (\text{A.32})$$

and its derivative

$$\psi'(\theta) = -4 \sin \theta(\cos \alpha + \cos \beta - 2 \cos \theta) \quad (\text{A.33})$$

we observe that the kernel is positive and unimodal in the Fourier domain if and only if

$$\psi(0) = 2 + 4 \cos \alpha \cos \beta + 4(\cos \alpha + \cos \beta) + 2 \geq 0 \quad (\text{A.34})$$

$$\psi(\theta) = 4(\cos \alpha \cos \beta - \cos \alpha - \cos \beta) \geq 0 \quad (\text{A.35})$$

$$\frac{|\cos \alpha + \cos \beta|}{2} \geq 1 \quad (\text{A.36})$$

One easily verifies that these systems of inequalities can be reduced to

$$\cos \alpha \cos \beta - \cos \alpha - \cos \beta \geq 0 \quad (\text{A.37})$$

$$\frac{\cos \alpha + \cos \beta}{2} \geq 1 \quad (\text{A.38})$$

Obviously, the last equation implies that $\cos \alpha$ and $\cos \beta$ must be equal to one, which also satisfies the second equation. Then, the kernel will equal to the binomial kernel (1, 4, 6, 4, 1).

A.2 From Chapter 3

A.2.1 Separated Convolution with $T(n; t)$ Satisfies the Diffusion Equation

Consider the possible scale-space representation of a two-dimensional signal given by separated convolution with the one-dimensional discrete analogue of the Gaussian kernel.

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T_{2D}(m, n; t) f(x - m, y - n) \quad (t > 0) \quad (\text{A.39})$$

where

$$T_{2D}(m, n; t) = T(m; t)T(n; t) \quad (\text{A.40})$$

and T is the one-dimensional discrete analogue of the Gaussian kernel given by $T(n; t) = e^{-t}I_n(t)$, where $I_n(t)$ are the modified Bessel functions of integer order. We will show that this representation satisfies a discretized version of the two-dimensional diffusion equation.

$$\frac{\partial L}{\partial t} = \frac{1}{2} \nabla_5^2 L \quad (\text{A.41})$$

by considering

$$\frac{\partial}{\partial t} T_{2D}(m, n; t) = \frac{\partial T}{\partial t}(m; t)T(n; t) + T(m; t)\frac{\partial T}{\partial t}(n; t) = \quad (\text{A.42})$$

$$= \{\text{Eq. (2.78)}\} = \frac{1}{2} (T(m-1; t) - 2T(m; t) + T(m+1; t))T(n; t) + \\ T(m; t)\frac{1}{2} (T(n-1; t) - 2T(n; t) + T(n+1; t)) = \quad (\text{A.43})$$

$$\frac{1}{2} (T(m-1, n; t) + T(m+1, n; t) + T(m, n-1; t) + T(m, n+1; t) - 4T(m, n; t)) = \\ \frac{1}{2} (\nabla_5^2 T)(m, n; t) \quad (\text{A.44})$$

Provided that the differentiation and infinite summation operators commute we have that the same relation holds for L , compare with the proof of Theorem 2.26.

A.2.2 Equivalent 1-D Formulation of the 2-D Discrete Scale-Space

For the sake of clarity, we state the definitions that are necessary for the one-dimensional equivalent formulation of the two-dimensional discrete scale-space concept given in Theorems 3.4-3.5.

Definition A.1 (Discrete local maximum (1D))

A point x is said to be a (weak) local maximum point for a function $g : Z \rightarrow R$ if $g(x) \geq g(x-1)$ and $g(x) \geq g(x+1)$.

Definition A.2 (Discrete local minimum (1D))

A point x is said to be a (weak) local minimum point for a function $g : Z \rightarrow R$ if $g(x) \leq g(x-1)$ and $g(x) \leq g(x+1)$.

Definition A.3 (Pre-scale-space family of kernels (1D))

A one-parameter family of kernels $T : Z \times R_+ \rightarrow R$ is said to be a pre-scale-space family of kernels if it satisfies

- $T(\cdot; 0) = \delta(\cdot)$
- the semi-group property $T(\cdot; s) * T(\cdot; t) = T(\cdot; s + t)$
- the symmetry constraint $T(-x; t) = T(x; t)$ for all $x \in Z$
- the continuity requirement $\|T(\cdot; t) - \delta(\cdot)\|_1 \rightarrow 0$ when $t \downarrow 0$

Definition A.4 (Pre-scale-space representation (1D))

Let $f : Z \rightarrow R$ be a discrete signal and $T : Z \times R \rightarrow R$ a pre-scale-space family of kernels. Then the one-parameter family of signals $L : Z \times R \rightarrow R$ given by

$$L(x; t) = \sum_{n=-\infty}^{\infty} T(n; t)f(x - n) \quad (\text{A.45})$$

is said to be the pre-scale-space representation of f generated by T .

Definition A.5 (Scale-space property: Non-enhancement of local extrema (1D))

A differentiable one-parameter family of signals $L : Z \times R \rightarrow R$ is said to possess pre-scale-space properties, or equivalently, not enhance local extrema if for every value of the scale parameter $t_0 \in R_+$ it holds that if $x_0 \in Z$ is a local extremum point for the mapping $x \mapsto L(x; t_0)$ then the partial derivative of L with respect to t in this point satisfies

$$\frac{\partial L}{\partial t}(x_0; t_0) \leq 0 \quad \text{if } x_0 \text{ is a local maximum point} \quad (\text{A.46})$$

$$\frac{\partial L}{\partial t}(x_0; t_0) \geq 0 \quad \text{if } x_0 \text{ is a local minimum point} \quad (\text{A.47})$$

Definition A.6 (Scale-space family of kernels (1D))

A one-parameter family of pre-scale-space kernels $T : Z \times R \rightarrow R$ is said to be a scale-space family of kernels if for any signal $g : Z \rightarrow R$ the pre-scale-space representation of g generated by T possesses pre-scale-space properties, i.e. if for any signal local extrema are never enhanced.

Definition A.7 (Scale-space representation (1D))

A pre-scale-space representation $L : Z \times R_+ \rightarrow R$ of a signal $f : Z \rightarrow R$ generated by a family of kernels $T : Z \times R \rightarrow R$, which are scale-space kernels, is said to be a scale-space representation of f .

A.2.3 Derivation of the MacLaurin Expansion of the Fourier Transform

Given the expression for the Fourier transform on polar form (3.41) (3.42)

$$\psi_T(u, v) = e^{h(\omega \cos \phi, \omega \sin \phi)t} \quad (\text{A.48})$$

where

$$\begin{aligned}\bar{h}(\omega, \phi) = h(\omega \cos \phi, \omega \sin \phi) = & -(2 - \gamma) + \\ & (1 - \gamma)(\cos(\omega \cos \phi) + \cos(\omega \sin \phi)) + \\ & \gamma \cos(\omega \cos \phi) \cos(\omega \sin \phi)\end{aligned}\quad (\text{A.49})$$

we Taylor expand for small values of ω

$$\begin{aligned}\bar{h}(\omega, \phi) = & -(2 - \gamma) + \\ & (1 - \gamma)\left(1 - \frac{\omega^2 \cos^2 \phi}{2} + \frac{\omega^4 \cos^4 \phi}{24} + 1 - \frac{\omega^2 \sin^2 \phi}{2} + \frac{\omega^4 \sin^4 \phi}{24} + O(\omega^6)\right) \\ & \gamma\left(1 - \frac{\omega^2 \cos^2 \phi}{2} + \frac{\omega^4 \cos^4 \phi}{24} + O(\omega^6)\right)\left(1 - \frac{\omega^2 \sin^2 \phi}{2} + \frac{\omega^4 \sin^4 \phi}{24} + O(\omega^6)\right)\end{aligned}$$

and simplify

$$\begin{aligned}\bar{h}(\omega, \phi) = & -(2 - \gamma) + \\ & (1 - \gamma)\left(2 - \frac{\omega^2}{2} + \frac{\omega^4(\cos^4 \phi + \sin^4 \phi)}{24} + O(\omega^6)\right) \\ & \gamma\left(1 - \frac{\omega^2}{2} + \frac{\omega^4(\cos^4 \phi + \sin^4 \phi)}{24} + \frac{\omega^4 \cos^2 \phi \sin^2 \phi}{4} + O(\omega^6)\right)\end{aligned}\quad (\text{A.50})$$

to

$$\bar{h}(\omega, \phi) = -\frac{\omega^2}{2} + \frac{\omega^4(\cos^4 \phi + \sin^4 \phi)}{24} + \gamma \frac{6\omega^4 \cos^2 \phi \sin^2 \phi}{24} + O(\omega^6)\quad (\text{A.51})$$

which in turn rewritten as

$$\bar{h}(\omega, \phi) = -\frac{\omega^2}{2} + \frac{\omega^4(\cos^2 \phi + \sin^2 \phi)^2 + (6\gamma - 2) \cos^2 \phi \sin^2 \phi}{24} + O(\omega^6)\quad (\text{A.52})$$

to give the desired result in (3.44).

A.3 From Chapter 4

A.3.1 The l_1 Norms of the Difference between Various Discrete Implementations of the Scale-Space Theory

In this appendix we have tabulated the l_1 norms of the difference between various discrete kernels used for implementing the one-dimensional scale-space theory. The comparison comprises

- differences of the discrete analogue of the Gaussian kernel, denoted T , $\delta_x T$ and $\nabla_3^2 T$
- sampled derivatives of the Gaussian kernel, denoted G , $\frac{\partial}{\partial x} G$ and $\frac{\partial^2}{\partial x^2} G$
- differences of the sampled Gaussian, denoted G , $\delta_x G$ and $\nabla_3^2 G$
- integrated derivatives of the Gaussian kernel, denoted $\tilde{J}G$, $\tilde{J} \frac{\partial}{\partial x} G$ and $\tilde{J} \frac{\partial^2}{\partial x^2} G$
- differences of the integrated Gaussian kernel, denoted $\tilde{J}G$, $\delta_x \tilde{J}G$, and $\nabla_3^2 \tilde{J}G$

for orders 0, 1 and 2 of the differences and the derivatives.

t	$T - G$	$T - \int G$	$G - \int G$
0.01	$3.01 \cdot 10^{\pm 0}$	$1.99 \cdot 10^{-2}$	$2.99 \cdot 10^{\pm 0}$
0.1	$4.30 \cdot 10^{-1}$	$4.65 \cdot 10^{-2}$	$4.72 \cdot 10^{-1}$
1	$1.52 \cdot 10^{-1}$	$1.78 \cdot 10^{-1}$	$3.30 \cdot 10^{-2}$
10	$1.20 \cdot 10^{-2}$	$1.50 \cdot 10^{-2}$	$4.01 \cdot 10^{-3}$
100	$1.17 \cdot 10^{-3}$	$1.46 \cdot 10^{-3}$	$4.03 \cdot 10^{-4}$
1000	$1.17 \cdot 10^{-4}$	$1.46 \cdot 10^{-4}$	$4.04 \cdot 10^{-5}$
10000	$1.17 \cdot 10^{-5}$	$1.46 \cdot 10^{-5}$	$4.03 \cdot 10^{-6}$

t	$\delta_x T - \frac{\partial}{\partial x} G$	$\delta_x G - \frac{\partial}{\partial x} G$	$\delta_x T - \delta_x G$	$\delta_x T - \int \frac{\partial}{\partial x} G$	$\frac{\partial}{\partial x} G - \int \frac{\partial}{\partial x} G$	$\delta_x G - \delta_x \int G$
0.01	$9.95 \cdot 10^{-1}$	$3.98 \cdot 10^{\pm 0}$	$3.00 \cdot 10^{\pm 0}$	$9.95 \cdot 10^{-1}$	$2.98 \cdot 10^{-5}$	$2.99 \cdot 10^{\pm 0}$
0.1	$7.82 \cdot 10^{-1}$	$1.10 \cdot 10^{\pm 0}$	$3.94 \cdot 10^{-1}$	$2.30 \cdot 10^{-1}$	$5.53 \cdot 10^{-1}$	$4.24 \cdot 10^{-1}$
1	$1.15 \cdot 10^{-1}$	$1.91 \cdot 10^{-1}$	$1.18 \cdot 10^{-2}$	$7.65 \cdot 10^{-2}$	$5.42 \cdot 10^{-2}$	$3.26 \cdot 10^{-2}$
10	$4.27 \cdot 10^{-3}$	$7.78 \cdot 10^{-3}$	$7.30 \cdot 10^{-3}$	$4.24 \cdot 10^{-3}$	$1.97 \cdot 10^{-3}$	$1.87 \cdot 10^{-3}$
100	$1.35 \cdot 10^{-4}$	$2.51 \cdot 10^{-4}$	$2.45 \cdot 10^{-4}$	$1.40 \cdot 10^{-4}$	$6.28 \cdot 10^{-5}$	$6.25 \cdot 10^{-5}$
1000	$4.24 \cdot 10^{-6}$	$7.97 \cdot 10^{-6}$	$7.78 \cdot 10^{-6}$	$4.43 \cdot 10^{-6}$	$1.99 \cdot 10^{-6}$	$1.99 \cdot 10^{-6}$
10000	$1.35 \cdot 10^{-7}$	$2.52 \cdot 10^{-7}$	$2.46 \cdot 10^{-7}$	$1.40 \cdot 10^{-7}$	$6.28 \cdot 10^{-8}$	$6.27 \cdot 10^{-8}$

0.01	$1.00 \cdot 10^{\pm 0}$	$4.01 \cdot 10^{\pm 0}$	$3.02 \cdot 10^{\pm 0}$	$1.00 \cdot 10^{\pm 0}$	$2.99 \cdot 10^{-5}$	$3.00 \cdot 10^{\pm 0}$
0.1	$8.21 \cdot 10^{-1}$	$1.16 \cdot 10^{\pm 0}$	$4.13 \cdot 10^{-1}$	$2.41 \cdot 10^{-1}$	$5.81 \cdot 10^{-1}$	$4.45 \cdot 10^{-1}$
1	$1.70 \cdot 10^{-1}$	$2.84 \cdot 10^{-1}$	$1.75 \cdot 10^{-1}$	$1.14 \cdot 10^{-1}$	$8.05 \cdot 10^{-2}$	$4.83 \cdot 10^{-2}$
10	$1.71 \cdot 10^{-2}$	$3.12 \cdot 10^{-2}$	$2.93 \cdot 10^{-2}$	$1.70 \cdot 10^{-2}$	$7.90 \cdot 10^{-3}$	$7.52 \cdot 10^{-3}$
100	$1.69 \cdot 10^{-3}$	$3.15 \cdot 10^{-3}$	$3.07 \cdot 10^{-3}$	$1.75 \cdot 10^{-3}$	$7.88 \cdot 10^{-4}$	$7.84 \cdot 10^{-4}$
1000	$1.69 \cdot 10^{-4}$	$3.15 \cdot 10^{-4}$	$3.08 \cdot 10^{-4}$	$1.76 \cdot 10^{-4}$	$7.89 \cdot 10^{-5}$	$7.88 \cdot 10^{-5}$
10000	$1.69 \cdot 10^{-5}$	$3.15 \cdot 10^{-5}$	$3.09 \cdot 10^{-5}$	$1.76 \cdot 10^{-5}$	$7.88 \cdot 10^{-6}$	$7.88 \cdot 10^{-6}$

t	$\nabla_3^2 T - \frac{\partial^2}{\partial x^2} G$	$\nabla_3^2 G - \frac{\partial^2}{\partial x^2} G$	$\nabla_3^2 T - \nabla_3^2 G$	$\nabla_3^2 T - \int \frac{\partial^2}{\partial x^2} G$	$\frac{\partial^2}{\partial x^2} G - \int \frac{\partial^2}{\partial x^2} G$	$\nabla_3^2 G - \nabla_3^2 \int G$
0.01	$3.99 \cdot 10^{+2}$	$3.99 \cdot 10^{+2}$	$1.20 \cdot 10^{+1}$	$3.94 \cdot 10^{\pm 0}$	$3.99 \cdot 10^{+2}$	$1.20 \cdot 10^{+1}$
0.1	$1.11 \cdot 10^{+1}$	$1.11 \cdot 10^{+1}$	$1.71 \cdot 10^{\pm 0}$	$3.96 \cdot 10^{\pm 0}$	$1.11 \cdot 10^{+1}$	$1.89 \cdot 10^{\pm 0}$
1	$4.20 \cdot 10^{-1}$	$2.18 \cdot 10^{-1}$	$5.35 \cdot 10^{-1}$	$4.89 \cdot 10^{-1}$	$1.19 \cdot 10^{-1}$	$8.34 \cdot 10^{-2}$
10	$4.30 \cdot 10^{-3}$	$2.33 \cdot 10^{-3}$	$5.70 \cdot 10^{-3}$	$4.99 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$1.14 \cdot 10^{-3}$
100	$4.07 \cdot 10^{-5}$	$2.33 \cdot 10^{-5}$	$5.75 \cdot 10^{-5}$	$4.86 \cdot 10^{-5}$	$1.17 \cdot 10^{-5}$	$1.16 \cdot 10^{-5}$
1000	$4.05 \cdot 10^{-7}$	$2.33 \cdot 10^{-7}$	$5.76 \cdot 10^{-7}$	$4.86 \cdot 10^{-7}$	$1.17 \cdot 10^{-7}$	$1.17 \cdot 10^{-7}$
10000	$4.05 \cdot 10^{-9}$	$2.33 \cdot 10^{-9}$	$5.76 \cdot 10^{-9}$	$4.86 \cdot 10^{-9}$	$1.17 \cdot 10^{-9}$	$1.17 \cdot 10^{-9}$

0.01	$1.01 \cdot 10^{+2}$	$1.01 \cdot 10^{+2}$	$3.05 \cdot 10^{\pm 0}$	$1.00 \cdot 10^{\pm 0}$	$1.01 \cdot 10^{+2}$	$3.03 \cdot 10^{\pm 0}$
0.1	$3.21 \cdot 10^{+1}$	$3.21 \cdot 10^{+1}$	$4.95 \cdot 10^{-1}$	$1.15 \cdot 10^{\pm 0}$	$3.21 \cdot 10^{\pm 0}$	$5.48 \cdot 10^{-1}$
1	$4.07 \cdot 10^{-1}$	$2.12 \cdot 10^{-1}$	$5.19 \cdot 10^{-1}$	$4.74 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$	$8.08 \cdot 10^{-2}$
10	$4.45 \cdot 10^{-2}$	$2.41 \cdot 10^{-2}$	$5.90 \cdot 10^{-2}$	$5.17 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$1.18 \cdot 10^{-2}$
100	$4.21 \cdot 10^{-3}$	$2.41 \cdot 10^{-3}$	$5.95 \cdot 10^{-3}$	$5.02 \cdot 10^{-3}$	$1.21 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$
1000	$4.19 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$	$5.95 \cdot 10^{-4}$	$5.02 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$
10000	$4.19 \cdot 10^{-5}$	$2.41 \cdot 10^{-5}$	$5.95 \cdot 10^{-5}$	$5.02 \cdot 10^{-5}$	$1.21 \cdot 10^{-5}$	$1.21 \cdot 10^{-5}$

Table A.1: The l_1 norm of the difference between various discrete implementations of the zero, first and second order derivatives of the Gaussian kernel in one dimension. Two data sets are given for the first and second order derivative approximations; the upper table giving the l_1 norm of the difference, and the lower table giving the same norm divided by the l_1 norm of the same order difference of the discrete analogue of the Gaussian kernel.

A.4 From Chapter 5

In order to give a further intuitive idea of the effects of extracting grey-level blobs at different levels of scale, Figures A.2-A.3 show some more examples in addition to those given in Figures 5.8-5.9.

A.5 From Chapter 6

A.5.1 Polynomials Satisfying the Diffusion Equation.

This appendix lists a set of polynomials satisfying the diffusion equation which are used in Section 6. Each polynomial $p_{m,n}(x, y)$ has been generated from the monomial $x^m y^n$ by adding suitable lower order terms containing powers of t , and if necessary x and y as well, such that $p_{m,n}(x, y)$ satisfies the two-dimensional diffusion equation.

$$\begin{aligned} p_{0,0}(x, y; t) &= 1 \\ p_{1,0}(x, y; t) &= x \\ p_{0,1}(x, y; t) &= y \\ p_{2,0}(x, y; t) &= x^2 + t \\ p_{1,1}(x, y; t) &= xy \\ p_{0,2}(x, y; t) &= y^2 + t \\ p_{3,0}(x, y; t) &= x^3 + 3xt \\ p_{2,1}(x, y; t) &= x^2y + yt \\ p_{1,2}(x, y; t) &= xy^2 + xt \\ p_{0,3}(x, y; t) &= y^3 + 3yt \\ p_{4,0}(x, y; t) &= x^4 + 6x^2t + 3t^2 \\ p_{3,1}(x, y; t) &= x^3y + 3xyt \\ p_{2,2}(x, y; t) &= x^2y^2 + x^2t + y^2t + t^2 \\ p_{1,3}(x, y; t) &= xy^3 + 3xyt \\ p_{0,4}(x, y; t) &= y^4 + 6y^2t + 3t^2 \\ &\vdots \end{aligned}$$

A.5.2 Investigation about the Roots to $4x^3 + 12tx + v = 0$

Given a value of v , we will investigate for which t -values the equation $h(x) = 4x^3 + 12tx + v = 0$ has real roots (in x). Consider the derivative

$$\frac{\partial h}{\partial x}(x) = 12(x^2 + t) \tag{A.53}$$

If $t > 0$ then this expression is > 0 for all x and the function is monotone, which means that the equation has one unique real root.

On the other hand if $t < 0$ then this function has two critical points, $\xi_1 = -\sqrt{-t}$ and $\xi_2 = +\sqrt{-t}$, where ξ_1 is a local maximum and ξ_2 is a local minimum. The equation has three

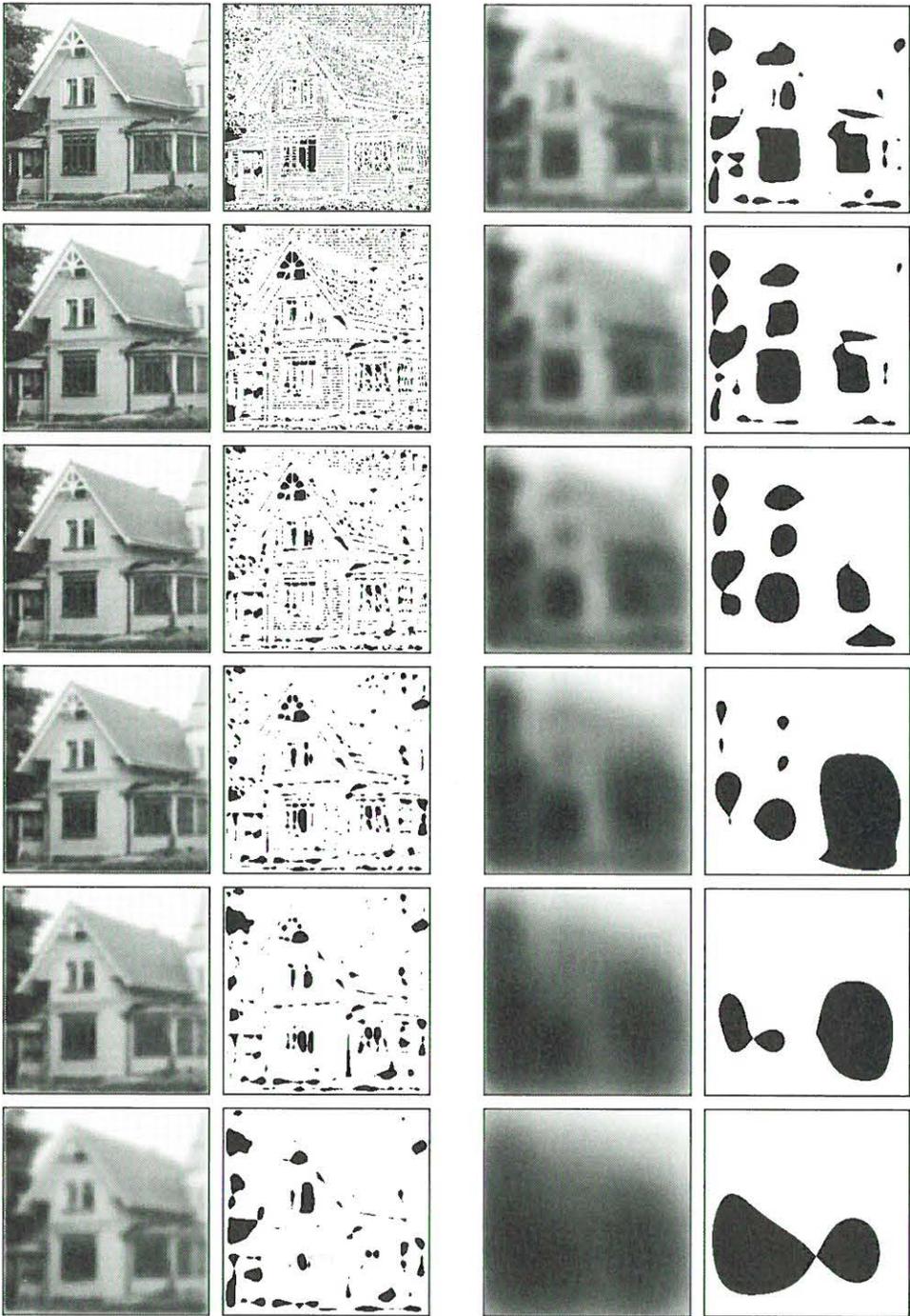


Figure A.2: Grey-level and dark grey-level blob images of the Godthem image at scale levels $t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 (from top left to bottom right).

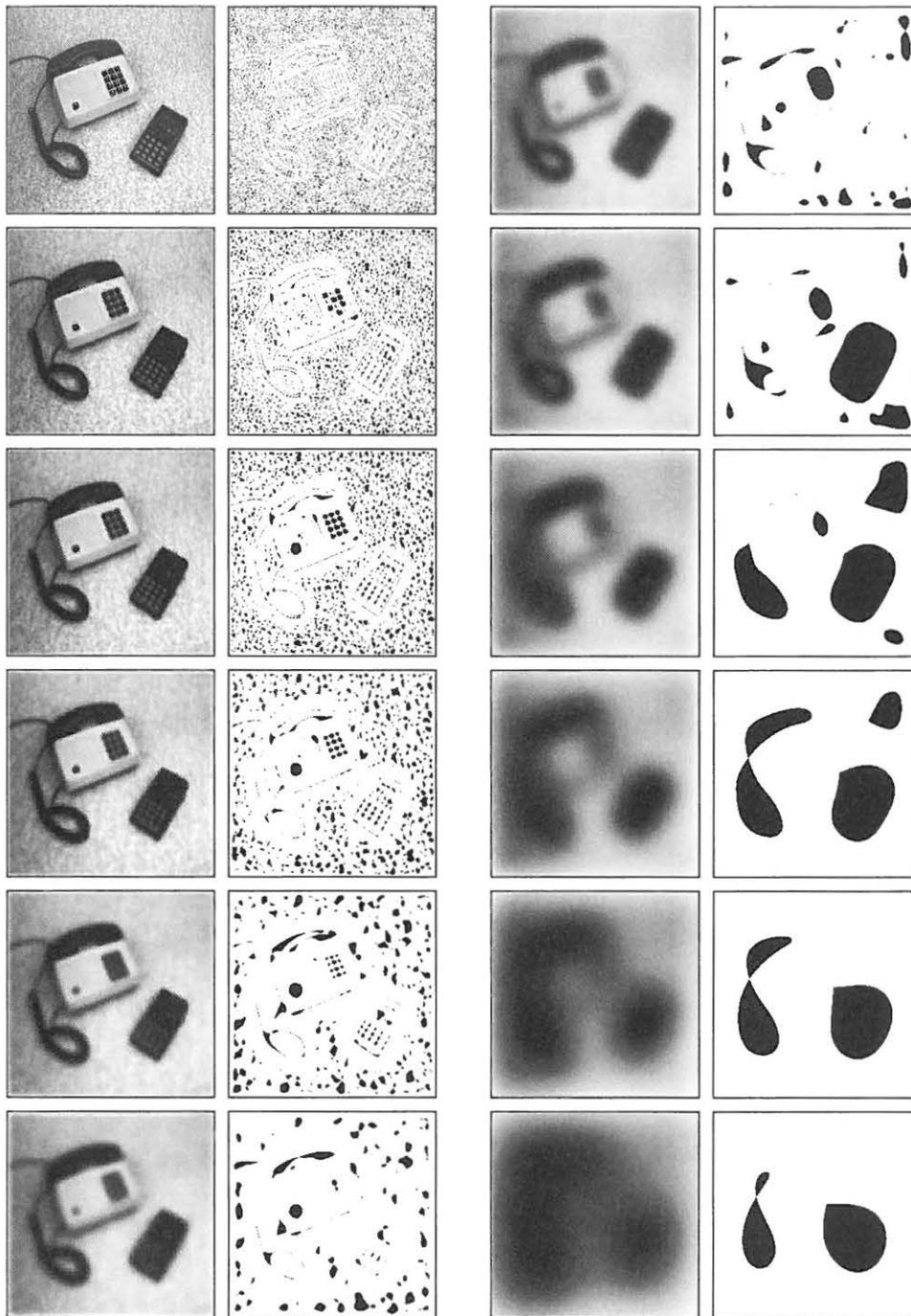


Figure A.3: Grey-level and dark grey-level blob images of a telephone and calculator image at scale levels $t = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 (from top left to bottom right).

real roots if and only if $f(\xi_1)$ and $f(\xi_2)$ have different signs, i.e., if and only if $f(\xi_1)f(\xi_2) < 0$. We have

$$f(\xi_1)f(\xi_2) = \dots = 64t^3 + v^2 \quad (\text{A.54})$$

Thus, the equation has three different real roots if and only if

$$t < -\left(\frac{v}{8}\right)^2 \quad (\text{A.55})$$

One easily verifies that if $t = -\left(\frac{v}{8}\right)^2$ and $v \neq 0$ then all roots are real and exactly two roots are equal. If $v = 0$ then for $t = 0$ the equation trivially has a root of multiplicity 3 at $x = 0$.

In the cases when the equation has three real roots x_1 , x_2 and x_3 the roots will be delimited by the critical points. Hence, we have

$$x_1 \leq \xi_1 \leq x_2 \leq \xi_2 \leq x_3 \quad (\text{A.56})$$

When $t < -\left(\frac{v}{8}\right)^2$ strict inequalities hold in this set of relations.

A.5.3 Detailed Investigation of the Singularity Set for the Elliptic Umbilic

We are to investigate the singularity set of the elliptic umbilic unfolding modified to satisfy the diffusion equation. From Section 6.4.3 we have that it is given by the solutions to

$$\begin{cases} \frac{\partial L}{\partial x} = 2x(y+w) + u = 0 \\ \frac{\partial L}{\partial y} = x^2 - 3y^2 - 2t = 0 \end{cases} \quad (\text{A.57})$$

and that the types of the critical points determined by

$$\begin{cases} (\mathcal{A}L) = \frac{\partial^2 L}{\partial x^2} = 2(y+w) \\ \frac{\partial^2 L}{\partial x \partial y} = 2x \\ \frac{\partial^2 L}{\partial y^2} = -6y \\ (\mathcal{H}L) = \frac{\partial^2 L}{\partial x^2} \frac{\partial^2 L}{\partial y^2} - \frac{\partial^2 L}{\partial x \partial y} \frac{\partial^2 L}{\partial y \partial x} = -4(3y(y+w) + x^2) \end{cases} \quad (\text{A.58})$$

Consider first the case when $u < 0$, set $u' = -u > 0$ and introduce new variables ξ and η by

$$\begin{cases} \xi = (y+w)/x \\ \eta = x(y+w) \end{cases} \quad (\text{A.59})$$

Since $u < 0$ we have from (6.115) that x and $y+w$ will always have the same sign. This means that ξ and η will both be non-negative. However, the mapping $(x, y) \mapsto (\xi, \eta)$ is not globally bijective since the two points $(x_1, y_1) = (a, -w+b)$ and $(x_2, y_2) = (-a, -w-b)$ will both be mapped onto the same (ξ, η) value. This means that the cases $(x > 0, y > -w)$ and $(x < 0, y < -w)$ must be treated separately. In these new coordinates (6.115) can be written

$$\begin{cases} 2\eta - u' = 0 \\ \eta/\xi - 3(-w \pm \sqrt{\eta\xi})^2 - 2t = 0 \end{cases} \quad (\text{A.60})$$

and simplified to

$$\frac{u'}{2} \left(3\xi - \frac{1}{\xi} \right) + 3w^2 + 2t = \pm 6w \sqrt{\frac{u'\xi}{2}} \quad (\text{A.61})$$

after observing that $\eta = u'/2$. This equation describes the relation between ξ and t for each one of the two arcs. The sign of the \pm term is the same as the sign of x . For every $\xi > 0$ there are two corresponding points (x_1, y_1) and (x_2, y_2) given by

$$(x_1, y_1) = \left(+\sqrt{\frac{u'}{2\xi}}, -w + \sqrt{\frac{u'\xi}{2}} \right) \quad (\text{A.62})$$

$$(x_2, y_2) = \left(-\sqrt{\frac{u'}{2\xi}}, -w - \sqrt{\frac{u'\xi}{2}} \right) \quad (\text{A.63})$$

Introduce t' by $2t' = 2t + 3w^2$. Then

$$\frac{u'}{2} \left(3\xi - \frac{1}{\xi} \right) + 2t' = \pm 6w \sqrt{\frac{u'\xi}{2}} \quad (\text{A.64})$$

Solving for t' yields

$$t' = -\frac{u'}{4} \left(3\xi - \frac{1}{\xi} \right) \pm 3w \sqrt{\frac{u'\xi}{2}} \quad (\text{A.65})$$

which by introduction of $\tilde{t} = t'/u'$ can be written

$$\tilde{t} = \frac{1}{4\xi} - \frac{3\xi}{4} \pm 3w \sqrt{\frac{\xi}{2u'}} \quad (\text{A.66})$$

If $w > 0$ we let $w = \sqrt{u'\tilde{w}}$

$$\tilde{t}_1 = \frac{1}{4\xi} - \frac{3\xi}{4} + \frac{3}{2} \sqrt{2\tilde{w}\xi} \quad (w > 0) \quad (\text{A.67})$$

$$\tilde{t}_2 = \frac{1}{4\xi} - \frac{3\xi}{4} - \frac{3}{2} \sqrt{2\tilde{w}\xi} \quad (w > 0) \quad (\text{A.68})$$

Else if $w < 0$ we let $w = -\sqrt{u'\tilde{w}}$. Then

$$\tilde{t}_1 = \frac{1}{4\xi} - \frac{3\xi}{4} - \frac{3}{2} \sqrt{2\tilde{w}\xi} \quad (w < 0) \quad (\text{A.69})$$

$$\tilde{t}_2 = \frac{1}{4\xi} - \frac{3\xi}{4} + \frac{3}{2} \sqrt{2\tilde{w}\xi} \quad (w < 0) \quad (\text{A.70})$$

These functions describe how \tilde{t} depends on ξ . The two arcs \tilde{t}_1 and \tilde{t}_2 correspond to the two cases, (x_1, y_1) and (x_2, y_2) respectively. Since $\tilde{w} > 0$ the curves are defined only for $\xi > 0$. The critical points of these mappings, which are the bifurcation points of L , are given by

$$\frac{d\tilde{t}}{d\xi} = -\frac{1}{4\xi^2} - \frac{3}{4} \pm \frac{3}{4} \sqrt{\frac{2\tilde{w}}{\xi}} = 0 \quad (\text{A.71})$$

which can also be summarized into

$$h(\xi) = 9\xi^4 - 18\tilde{w}\xi^3 + 6\xi^2 + 1 = 0 \quad (\text{A.72})$$

In order to find the number of roots to this equation we differentiate and set the derivative

$$\frac{dh}{d\xi} = 36\xi^3 - 54\bar{w}\xi^2 + 12\xi \quad (\text{A.73})$$

to zero. This yields three roots

$$\xi_{h,1} = 0 \quad (\text{A.74})$$

$$\xi_{h,2} = \frac{3}{4} \left(\bar{w} + \sqrt{\bar{w}^2 - \frac{16}{27}} \right) \quad (\text{A.75})$$

$$\xi_{h,3} = \frac{3}{4} \left(\bar{w} - \sqrt{\bar{w}^2 - \frac{16}{27}} \right) \quad (\text{A.76})$$

with

$$h(\xi_{h,1}) = 1 \quad (\text{A.77})$$

$$h(\xi_{h,2}) = \frac{27\bar{w}^2}{4} - \frac{243\bar{w}^4}{32} + \left(\frac{\bar{w}}{2} - \frac{27\bar{w}^3}{32} \right) \sqrt{81\bar{w}^2 - 48} \quad (\text{A.78})$$

$$h(\xi_{h,3}) = \frac{27\bar{w}^2}{4} - \frac{243\bar{w}^4}{32} - \left(\frac{\bar{w}}{2} - \frac{27\bar{w}^3}{32} \right) \sqrt{81\bar{w}^2 - 48} \quad (\text{A.79})$$

$\xi_{h,2}$ and $\xi_{h,3}$ only exist when $\bar{w} \geq \frac{4}{3\sqrt{3}}$. Since $h(\xi)$ is positive both when ξ tends to zero and when ξ tends to infinity, the number of roots to the equation $h = 0$ is given by the signs of $h(\xi_{h,2})$ and $h(\xi_{h,3})$. Setting $h(\xi_{h,2}) = 0$ and $h(\xi_{h,3}) = 0$ gives two equations, which can be summarized into the relation

$$\frac{3w^2}{16}(-8 + 9\bar{w})(8 + 9\bar{w}) = 0 \quad (\text{A.80})$$

One easily shows that

- if $\bar{w} < 8/9$ then both $h(\xi_{h,2})$ and $h(\xi_{h,3})$ will be positive and the equation $h(\xi) = 0$ will have no real roots. This means that $\tilde{t}(\xi)$ has no critical points, and accordingly no bifurcations can take place (for positive ξ). Therefore the type of critical points will remain the same when the scale parameter t (or equivalently \tilde{t}) increases from $-\infty$ to ∞ .

In the limit cases $\xi \rightarrow 0^+$ and $\xi \rightarrow +\infty$ the behaviours of the critical points (x_1, y_1) and (x_2, y_2) (according to (A.62)), \tilde{t}_1 and \tilde{t}_2 (from (A.67)) and the Hessian $\mathcal{H}L$ (see (6.116)) are as stated in Table A.2. By continuity it follows that (x_1, y_1) and (x_2, y_2) will always be saddle point on their trajectories, see also Figure 6.16 for an illustration.

- if $\bar{w} > 8/9$ then $h(\xi_{h,2})$ will be negative and $h(\xi_{h,3})$ positive. Thus, the equation $h(\xi) = 0$ will have exactly two real roots, $\xi_{\text{bifurc},1}$ and $\xi_{\text{bifurc},2}$, delimited by $\xi_{h,2}$

$$\xi_{\text{bifurc},1} < \frac{3}{4} \left(\bar{w} + \sqrt{\bar{w}^2 - \frac{16}{27}} \right) < \xi_{\text{bifurc},2} \quad (\text{A.81})$$

At those points bifurcations take place. There are two different types of behaviours depending on the sign of w .

ξ	t_1	x_1	y_1	$(\mathcal{HL})(x_1, y_1)$	$(\mathcal{AL})(x_1, y_1)$	type
$+\infty$	$-\infty$	0^+	$+\infty$	$-$	$+$	saddle
0^+	$+\infty$	$+\infty$	$-w^+$	$-$	$+$	saddle
ξ	t_2	x_2	y_2	$(\mathcal{HL})(x_2, y_2)$	$(\mathcal{AL})(x_2, y_2)$	type
$+\infty$	$-\infty$	0^-	$-\infty$	$-$	$-$	saddle
0^+	$+\infty$	$-\infty$	$-w^-$	$-$	$-$	saddle

Table A.2: The behaviour of the critical points (x_1, y_1) and (x_2, y_2) in the limit cases $\xi \rightarrow 0^+$ and $\xi \rightarrow +\infty$. Observe that t decreases when ξ increases.

- Consider first the case when $w > 0$. For the arc corresponding to (x_2, y_2) it then holds that

$$\frac{d\tilde{t}_2}{d\xi} = -\frac{1}{4\xi^2} - \frac{3}{4} - \frac{3}{4}\sqrt{\frac{2\tilde{w}}{\xi}} \quad (\text{A.82})$$

This expression will always be strictly negative. Therefore, the two bifurcations must occur on the arc with (x_1, y_1) where

$$\frac{d\tilde{t}_1}{d\xi} = -\frac{1}{4\xi^2} - \frac{3}{4} + \frac{3}{4}\sqrt{\frac{2\tilde{w}}{\xi}} \quad (\text{A.83})$$

When ξ tends to zero and when ξ tends to infinity this expression will be strictly negative. Moreover, since we know that the equation $\frac{d\tilde{t}_1}{d\xi} = 0$ has exactly two roots of multiplicity one, it follows that $\frac{d\tilde{t}_1}{d\xi}$ will undergo the sign sequence $\{-, 0, +, 0, -\}$ when ξ increases. Hence, for small ξ and large ξ we have that \tilde{t}_1 decreases with ξ while in an intermediate interval $]\xi_{\text{bifurc},1}, \xi_{\text{bifurc},2}[$ it holds that \tilde{t}_1 increases with ξ . Figure A.4 shows the graph of \tilde{t}_1 and \tilde{t}_2 as function of ξ .

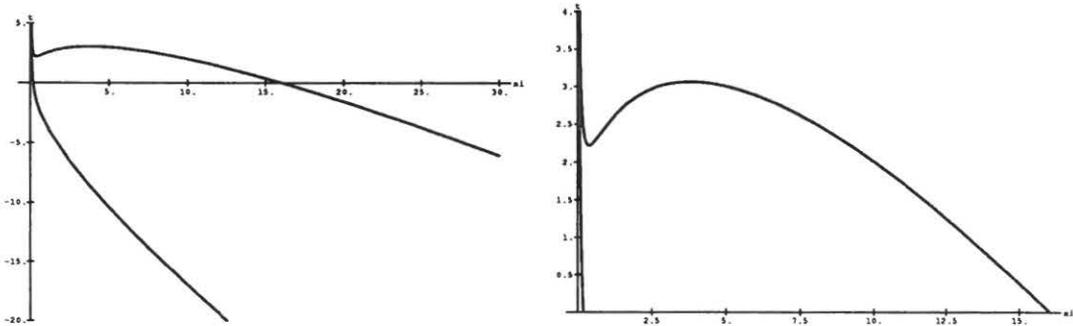


Figure A.4: \tilde{t}_1 and \tilde{t}_2 as functions of ξ in the case when $\tilde{w} > 8/9$ (and $w > 0$). (a) Graph for $w = 2$, $\xi \in [0, 30]$. (b) Enlargement of the region around the critical points for $\tilde{t}_1(\xi)$.

To analyse the sign of the Hessian we observe that $\mathcal{H}L$ will change sign for the same¹ values of ξ as $\frac{d\tilde{t}}{d\xi}$.

$$(\mathcal{H}L) = -4(3y(y+w) + x^2) \quad (\text{A.84})$$

When ξ tends to either 0 or ∞ the behaviour of the critical points will be the same as in the case when $\tilde{w} < 8/9$. However, when ξ increases from 0 to infinity two sign changes for $\mathcal{H}L$ will occur on the arc given by (x_1, y_1) . One verifies that $(\mathcal{H}L)(x_1, y_1)$ will undergo the sign sequence $\{-, 0, +, 0, -\}$ when ξ increases. Moreover, since $\mathcal{A}L = 2(y+w) > 0$ it follows that the critical point given by (x_1, y_1) changes from a saddle into a minimum and then back into a saddle again when ξ increases.

If we re-interpret this result in terms of increasing t (or equivalently increasing \tilde{t}) it means that for small values of t there are two saddle points in L . Then, at a certain scale a minimum-saddle pair is suddenly created. Later, the minimum point and the other saddle point on the same trajectory come together and annihilate, which means that at coarse scales there will again be two saddle points in L . Interpreted in terms of blobs this corresponds to the creation of a dark blob, which is then followed by an annihilation of the same dark blob (provided that the saddle points involved in the process are both non-shared). Note, however, that the minimum point will not have its delimiting saddle point on the same saddle path throughout the process.

– On the other hand if $w < 0$ we have that

$$\frac{d\tilde{t}_1}{d\xi} = -\frac{1}{4\xi^2} - \frac{3}{4} - \frac{3}{4}\sqrt{\frac{2\tilde{w}}{\xi}} < 0 \quad (\text{A.85})$$

and the bifurcations must occur on the arc with (x_2, y_2) where

$$\frac{d\tilde{t}_2}{d\xi} = -\frac{1}{4\xi^2} - \frac{3}{4} + \frac{3}{4}\sqrt{\frac{2\tilde{w}}{\xi}} \quad (\text{A.86})$$

With similar arguments as above one shows that the Hessian will undergo the sign sequence $\{-, 0, +, 0, -\}$ when ξ increases. However, here $\mathcal{A}L = 2(y+w)$ will be negative, which means that when ξ increases the critical point changes from a saddle into a maximum and then back into saddle.

Interpreted in terms of increasing t this corresponds to the creation of maximum-saddle pair with increasing scale followed by the annihilation of another maximum-saddle pair, or equivalently, the creation of a bright blob followed by the annihilation of a bright blob.

For the case with $u > 0$ the analog of (A.60) holds

$$\begin{cases} 2\eta + u = 0 \\ \eta/\xi - 3(-w \pm \sqrt{\eta\xi})^2 - 2t = 0 \end{cases} \quad (\text{A.87})$$

¹This property can also be shown algebraically by inserting the expressions for (x_1, y_1) and (x_2, y_2) into the expression for $\mathcal{H}L$.

which can be simplified to

$$\frac{u}{2} \left(3\xi' - \frac{1}{\xi'} \right) + 3w^2 + 2t = \pm 6w \sqrt{\frac{u\xi'}{2}} \quad (\text{A.88})$$

after observing that $\eta = -u/2$ and introducing $\xi' = -\xi$. As $\xi < 0$ we have $\xi' > 0$. We observe that this equation is similar to (A.61) although the new version of (A.62) holds

$$(x_1, y_1) = \left(-\sqrt{\frac{u'}{2\xi}}, -w + \sqrt{\frac{u'\xi}{2}} \right) \quad (\text{A.89})$$

$$(x_2, y_2) = \left(+\sqrt{\frac{u'}{2\xi}}, -w - \sqrt{\frac{u'\xi}{2}} \right) \quad (\text{A.90})$$

A positive sign of the \pm operator in (A.88) corresponds to the (x_1, y_1) point and a negative sign to the (x_2, y_2) point. Similar calculations as in the case when $u < 0$ show that bifurcations occur only when $\bar{w} > 8/9$ and that the sign of w determines whether they take place on (x_1, y_1) or (x_2, y_2) . To summarize, for $u \neq 0$ we have that an extremum-saddle pair can be created if $|\bar{w}| > 8/9$, that is if

$$|w| > w_0 = \frac{2}{3} \sqrt{\frac{2|u|}{3}} \quad (\text{A.91})$$

If $w > 0$ the extremum point is a minimum and if $w < 0$ the extremum point is a maximum.

A.5.4 Derivation of $p_d(t)$ in the Discrete Case

We are to solve the integral

$$p_d(t) = \int \int_{\{\eta=(\eta_1, \eta_2): (\eta_1 \geq 0) \wedge (\eta_2 \geq 0)\}} \frac{1}{\sqrt{(2\pi)^2 |C_{2D}|}} e^{-\frac{1}{2}\eta^T C_{2D}^{-1} \eta} d\eta_1 d\eta_2 \quad (\text{A.92})$$

where

$$C_{2D} = \begin{pmatrix} a_0(t) & a_1(t) \\ a_0(t) & a_1(t) \end{pmatrix} \quad (\text{A.93})$$

To simplify the notation $a_0(t)$ and $a_1(t)$ will from now on be denoted just by a_0 and a_1 respectively. Using

$$\begin{pmatrix} a_0 & a_1 \\ a_1 & a_0 \end{pmatrix}^{-1} = \frac{1}{a_0^2 - a_1^2} \begin{pmatrix} a_0 & -a_1 \\ -a_1 & a_0 \end{pmatrix} \quad (\text{A.94})$$

we obtain

$$p_d(t) = \int \int_{\{\eta=(\eta_1, \eta_2): (\eta_1 \geq 0) \wedge (\eta_2 \geq 0)\}} \frac{1}{2\pi \sqrt{a_0^2 - a_1^2}} e^{-\frac{1}{2(a_0^2 - a_1^2)} (a_0 \eta_1^2 + a_0 \eta_2^2 - 2a_1 \eta_1 \eta_2)} d\eta_1 d\eta_2 \quad (\text{A.95})$$

Introducing

$$b_0 = \frac{a_0}{2(a_0^2 - a_1^2)} \quad b_1 = \frac{a_1}{2(a_0^2 - a_1^2)} \quad (\text{A.96})$$

this expression can be written

$$p_d(t) = \int \int_{\{\eta=(\eta_1, \eta_2): (\eta_1 \geq 0) \wedge (\eta_2 \geq 0)\}} \frac{1}{2\pi \sqrt{a_0^2 - a_1^2}} e^{-(b_0 \eta_1^2 + b_0 \eta_2^2 - 2b_1 \eta_1 \eta_2)} d\eta_1 d\eta_2 \quad (\text{A.97})$$

Observing that the argument to the exponential function is

$$b_0 \left(\eta_1^2 + \eta_2^2 - 2 \frac{b_1}{b_0} \eta_1 \eta_2 \right) = b_0 \left(\left(\eta_1 - \frac{b_1}{b_0} \eta_2 \right)^2 + \left(1 - \frac{b_1^2}{b_0^2} \right) \eta_2^2 \right) \quad (\text{A.98})$$

and introducing new variables ν_1 and ν_2 by

$$\nu_1 = \sqrt{b_0} \sqrt{1 - \frac{b_1^2}{b_0^2}} \eta_1 \quad \nu_2 = \sqrt{b_0} \left(\eta_2 - \frac{b_1}{b_0} \eta_1 \right) \quad (\text{A.99})$$

which leads to

$$d\nu_1 d\nu_2 = \left| \frac{\partial(\nu_1, \nu_2)}{\partial(\eta_1, \eta_2)} \right| d\eta_1 d\eta_2 = \left\| \begin{array}{cc} \sqrt{b_0} \sqrt{1 - \frac{b_1^2}{b_0^2}} & 0 \\ -\sqrt{b_0} \frac{b_1}{b_0} & \sqrt{b_0} \end{array} \right\| d\eta_1 d\eta_2 = \sqrt{b_0^2 - b_1^2} d\eta_1 d\eta_2 \quad (\text{A.100})$$

we have that $p_d(t)$ can be written

$$p_d(t) = \frac{1}{2\pi \sqrt{a_0^2 - a_1^2} \sqrt{b_0^2 - b_1^2}} \int \int_{D_{\nu_1 \nu_2}} e^{-(\nu_1^2 + \nu_2^2)} d\nu_1 d\nu_2 \quad (\text{A.101})$$

where $D_{\nu_1 \nu_2}$ is the region

$$D_{\nu_1 \nu_2} = \{(\nu_1, \nu_2) : (\nu_1 \geq 0) \wedge (\nu_1 \geq -\frac{a_1}{\sqrt{a_0^2 - a_1^2}} \nu_2)\} \quad (\text{A.102})$$

Using

$$b_0^2 - b_1^2 = \frac{a_0^2}{4(a_0^2 - a_1^2)^2} - \frac{a_1^2}{4(a_0^2 - a_1^2)^2} = \frac{1}{4(a_0^2 - a_1^2)} \quad (\text{A.103})$$

this expression can be simplified further to

$$p_d(t) = \frac{1}{\pi} \int \int_{D_{\nu_1 \nu_2}} e^{-(\nu_1^2 + \nu_2^2)} d\nu_1 d\nu_2 \quad (\text{A.104})$$

By introducing polar coordinates $\nu_1 = r \cos \phi$ and $\nu_2 = r \sin \phi$ we can rewrite the $p_d(t)$ as

$$p_d(t) = \frac{1}{\pi} \int_{r=0}^{\infty} \int_{\phi=\phi_1}^{\pi/2} e^{-r^2} r dr d\phi \quad (\text{A.105})$$

where

$$\phi_1 = \arctan \left(-\frac{a_1}{\sqrt{a_0^2 - a_1^2}} \right) \quad (\text{A.106})$$

This integral is easily solved

$$p_d(t) = \frac{1}{\pi} \left[-\frac{e^{-r^2}}{2} \right]_{r=0}^{\infty} [\phi]_{\phi=\phi_1}^{\pi/2} = \dots = \frac{1}{4} - \frac{\phi_1}{2\pi} \quad (\text{A.107})$$

which shows that

$$p_d(t) = \frac{1}{4} + \frac{1}{2\pi} \arctan \left(\frac{a_1}{\sqrt{a_0^2 - a_1^2}} \right) \quad (\text{A.108})$$

In the special case $t = 0$ we get

$$a_0|_{t=0} = 2(T(0; 0) - T(1; 0)) = 2(1 - 0) = 2 \quad (\text{A.109})$$

$$a_1|_{t=0} = T(0; 0) - 2T(1; 0) + T(2; 0) = 1 - 0 + 0 = 1 \quad (\text{A.110})$$

and

$$p_d(0) = \frac{1}{4} + \frac{1}{2\pi} \arctan\left(\frac{1}{\sqrt{3}}\right) = \frac{1}{3} \quad (\text{A.111})$$

A.5.5 Asymptotic Expression for $p_d(t)$ at Fine Scales

In order to obtain a Taylor expansions of $T(0; t)$, $T(1; t)$ and $T(2; t)$ for small values of T we first observe that every $T(n; t)$ satisfies, see (2.78),

$$\frac{\partial T}{\partial t}(n; t) = \frac{1}{2}(T(n-1; t) - 2T(n; t) + T(n+1; t)) \quad (\text{A.112})$$

Moreover, the kernels are symmetric $T(-n; t) = T(n; t)$ and we have $T(n; 0) = \delta(n)$ where δ denotes the discrete delta function. From these relations one easily shows that the first order derivatives of $T(n; t)$ are

$$\frac{\partial T}{\partial t}(0; 0) = \frac{1}{2}(T(-1; 0) - 2T(0; 0) + T(1; 0)) = \frac{1}{2}(0 - 2 \cdot 1 + 0) = -1 \quad (\text{A.113})$$

$$\frac{\partial T}{\partial t}(1; 0) = \frac{1}{2}(T(0; 0) - 2T(1; 0) + T(2; 0)) = \frac{1}{2}(1 - 0 + 0) = \frac{1}{2} \quad (\text{A.114})$$

$$\frac{\partial T}{\partial t}(n; 0) = 0 \quad \text{if } |n| \geq 2 \quad (\text{A.115})$$

and the second order ones

$$\frac{\partial^2 T}{\partial t^2}(0; 0) = \frac{1}{2}\left(\frac{\partial T}{\partial t}(-1; 0) - 2\frac{\partial T}{\partial t}(0; 0) + \frac{\partial T}{\partial t}(1; 0)\right) = \frac{1}{2}\left(\frac{1}{2} - 2 \cdot (-1) + \frac{1}{2}\right) = \frac{3}{2} \quad (\text{A.116})$$

$$\frac{\partial^2 T}{\partial t^2}(1; 0) = \frac{1}{2}\left(\frac{\partial T}{\partial t}(0; 0) - 2\frac{\partial T}{\partial t}(1; 0) + \frac{\partial T}{\partial t}(2; 0)\right) = \frac{1}{2}\left(-1 - 2 \cdot \frac{1}{2} + 0\right) = -1 \quad (\text{A.117})$$

$$\frac{\partial^2 T}{\partial t^2}(2; 0) = \frac{1}{2}\left(\frac{\partial T}{\partial t}(0; 0) - 2\frac{\partial T}{\partial t}(1; 0) + \frac{\partial T}{\partial t}(2; 0)\right) = \frac{1}{2}\left(\frac{1}{2} - 0 + 0\right) = \frac{1}{4} \quad (\text{A.118})$$

$$\frac{\partial^2 T}{\partial t^2}(n; 0) = 0 \quad \text{if } |n| \geq 3 \quad (\text{A.119})$$

which means that the second order Taylor expansions of $T(0; t)$, $T(1; t)$ and $T(2; t)$ are

$$T(0; t) = 1 - t + \frac{3}{4}t^2 + O(t^3) \quad (\text{A.120})$$

$$T(1; t) = \frac{1}{2}t - \frac{1}{2}t^2 + O(t^3) \quad (\text{A.121})$$

$$T(2; t) = \frac{1}{8}t^2 + O(t^3) \quad (\text{A.122})$$

By inserting these results into the expressions for $a_0(t)$ and $a_1(t)$ we get

$$a_0(t) = 2(T(0; 2t) - T(1; 2t)) = 2 - 6t + 10t^2 + O(t^3) \quad (\text{A.123})$$

$$a_1(t) = T(0; 2t) - 2T(1; 2t) + T(2; 2t) = 1 - 4t + \frac{15}{2}t^2 + O(t^3) \quad (\text{A.124})$$

$$a_0(t) + a_1(t) = 3 - 10t + \frac{35}{2}t^2 + O(t^3) \quad (\text{A.125})$$

$$a_0(t) - a_1(t) = 1 - 2t + \frac{5}{2}t^2 + O(t^3) \quad (\text{A.126})$$

$$(a_0(t))^2 - (a_1(t))^2 = (a_0(t) + a_1(t))(a_0(t) - a_1(t)) = 3 - 16t + 45t^2 + O(t^3) \quad (\text{A.127})$$

and

$$\frac{a_1(t)}{\sqrt{(a_0(t))^2 - (a_1(t))^2}} = \frac{1}{\sqrt{3}} \frac{1 - 4t + \frac{15}{2}t^2 + O(t^3)}{\sqrt{1 - \frac{16}{3}t + \frac{45}{3}t^3 + O(t^3)}} \quad (\text{A.128})$$

Using the well-known MacLaurin expansion

$$(1 + x)^{-\frac{1}{2}} = 1 - \frac{1}{2}x + \frac{3}{8}x^2 + O(x^3) \quad (\text{A.129})$$

this expression can be simplified to

$$\frac{a_1(t)}{\sqrt{(a_0(t))^2 - (a_1(t))^2}} = \frac{1}{\sqrt{3}} \left(1 - \frac{4}{3}t + O(t^3)\right) \quad (\text{A.130})$$

By differentiation one easily shows that the second order MacLaurin expansion of $\arctan(a + x)$ is

$$\arctan(a + x) = \arctan(a) + \frac{1}{1 + a^2}x - \frac{a}{(1 + a^2)^2}x^2 + O(x^3) \quad (\text{A.131})$$

In the special case $a = \frac{1}{\sqrt{3}}$ this implies

$$\arctan\left(\frac{1}{\sqrt{3}} + x\right) = \frac{\pi}{6} + \frac{3}{4}x - \frac{3\sqrt{3}}{16}x^2 + O(x^3) \quad (\text{A.132})$$

and

$$\arctan\left(\frac{a_1(t)}{\sqrt{(a_0(t))^2 - (a_1(t))^2}}\right) = \frac{\pi}{6} - \frac{1}{\sqrt{3}}t + \frac{1}{3\sqrt{3}}t^2 + O(t^3) \quad (\text{A.133})$$

By inserting this result into the expression for $p_d(t)$ we obtain

$$p_d(t) = \frac{1}{4} + \frac{1}{2\pi} \arctan\left(\frac{a_1(t)}{\sqrt{(a_0(t))^2 - (a_1(t))^2}}\right) = \frac{1}{3} - \frac{1}{2\sqrt{3}\pi}t + \frac{1}{6\sqrt{3}\pi}t^2 + O(t^3) \quad (\text{A.134})$$

which means that we can write down the MacLaurin expansion for the effective scale²

$$\tau(t) = \log\left(\frac{p_d(0)}{p_d(t)}\right) = \dots = -\log\left(1 - \frac{\sqrt{3}}{2\pi}t + \frac{1}{2\sqrt{3}\pi}t^2 + O(t^3)\right) \quad (\text{A.135})$$

Using

$$\log(1 + x) = x - \frac{1}{2}x^2 + O(x^3) \quad (\text{A.136})$$

this expression can be simplified to

$$\tau(t) = \frac{\sqrt{3}}{2\pi}t + \left(\frac{1}{2\sqrt{3}\pi} + \frac{3}{8\pi^2}\right)t^2 + O(t^3) \quad (\text{A.137})$$

²We have selected values of the constants A and B such that $t = 0$ corresponds to $\tau = 0$ and the coefficient of the logarithmic term is one. Above it has been shown that $p_d(0) = \frac{1}{3}$.

A.5.6 Asymptotic Expression for $p_d(t)$ at Coarse Scales

At coarse scales it holds that $a_1(t) < 0$ (see below). In that case it is convenient to use the relation

$$\arctan(x) + \arctan\left(\frac{1}{x}\right) = -\frac{\pi}{2} \quad (x < 0) \quad (\text{A.138})$$

for rewriting the expression for

$$p_d(t) = \frac{1}{4} + \frac{1}{2\pi} \arctan\left(\frac{a_1(t)}{\sqrt{(a_0(t))^2 - (a_1(t))^2}}\right) \quad (\text{A.139})$$

into

$$p_d(t) = \frac{1}{2\pi} \arctan\left(\frac{\sqrt{(a_0(t))^2 - (a_1(t))^2}}{-a_1(t)}\right) \quad (a_1(t) < 0) \quad (\text{A.140})$$

According to Abramowitz and Stegun [Abr64] (9.7.1) for fixed n and large t it holds that

$$I_n(t) = \frac{e^t}{\sqrt{2\pi t}} \left(1 - \frac{4n^2 - 1}{8t} + \frac{(4n^2 - 1)(4n^2 - 9)}{2!(8t)^2} - \frac{(4n^2 - 1)(4n^2 - 9)(4n^2 - 25)}{3!(8t)^3} + \dots\right) \quad (\text{A.141})$$

which implies that

$$T(0; t) = e^{-t} I_0(t) = \frac{1}{\sqrt{2\pi t}} \left(1 + \frac{1}{8t} + \frac{9}{128t^2} + \frac{225}{3072t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.142})$$

$$T(1; t) = e^{-t} I_1(t) = \frac{1}{\sqrt{2\pi t}} \left(1 - \frac{3}{8t} - \frac{15}{128t^2} - \frac{315}{3072t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.143})$$

$$T(2; t) = e^{-t} I_2(t) = \frac{1}{\sqrt{2\pi t}} \left(1 - \frac{15}{8t} + \frac{105}{128t^2} + \frac{945}{3072t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.144})$$

and

$$T(0; 2t) = e^{-2t} I_0(2t) = \frac{1}{\sqrt{4\pi t}} \left(1 + \frac{1}{16t} + \frac{9}{512t^2} + \frac{225}{24576t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.145})$$

$$T(1; 2t) = e^{-2t} I_1(2t) = \frac{1}{\sqrt{4\pi t}} \left(1 - \frac{3}{16t} - \frac{15}{512t^2} - \frac{315}{24576t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.146})$$

$$T(2; 2t) = e^{-2t} I_2(2t) = \frac{1}{\sqrt{4\pi t}} \left(1 - \frac{15}{16t} + \frac{105}{512t^2} + \frac{945}{24576t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.147})$$

From these expressions one easily concludes that

$$a_0(t) = 2(T(0; 2t) - T(1; 2t)) = \frac{1}{\sqrt{4\pi t}} \left(\frac{1}{2t} + \frac{3}{32t^2} + \frac{45}{1024t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.148})$$

$$a_1(t) = T(0; 2t) - 2T(1; 2t) + T(2; 2t) = \frac{1}{\sqrt{4\pi t}} \left(-\frac{1}{2t} + \frac{9}{32t^2} + \frac{75}{1024t^3} + O\left(\frac{1}{t^4}\right)\right) \quad (\text{A.149})$$

$$a_0(t) + a_1(t) = \frac{1}{\sqrt{4\pi t}} \left(3 + \frac{15}{16t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.150})$$

$$a_0(t) - a_1(t) = \frac{1}{\sqrt{4\pi t}} \left(1 - \frac{3}{16t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.151})$$

$$(a_0(t))^2 - (a_1(t))^2 = \frac{3}{32\pi t^4} \left(1 + \frac{1}{8t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.152})$$

which in turn leads to

$$\sqrt{(a_0(t))^2 - (a_1(t))^2} = \sqrt{\frac{3}{2\pi}} \frac{1}{4t^2} \left(1 + \frac{1}{16t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.153})$$

$$(a_1(t))^{-1} = -2t\sqrt{4\pi t} \left(1 + \frac{9}{16t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.154})$$

and

$$\frac{\sqrt{(a_0(t))^2 - (a_1(t))^2}}{a_1(t)} = -\sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} \left(1 + \frac{5}{8t} + O\left(\frac{1}{t^2}\right)\right) < 0 \quad (\text{A.155})$$

which gives

$$p_d(t) = \frac{1}{2\pi} \arctan \left(\sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} \left(1 + \frac{5}{8t} + O\left(\frac{1}{t^2}\right)\right) \right) \quad (\text{A.156})$$

From the MacLaurin expansion

$$\arctan(x) = x - \frac{1}{3}x^3 + O(x^5) \quad (\text{A.157})$$

we finally get

$$p_d(t) = \frac{1}{2\pi} \sqrt{\frac{3}{2}} \frac{1}{\sqrt{t}} \left(1 + \frac{1}{8t} + O\left(\frac{1}{t^2}\right)\right) \quad (\text{A.158})$$

which asymptotically agrees with result from the previous continuous analysis.

A.6 From Chapter 7

A.6.1 Algorithmic Performance

In order to give a coarse estimate of the complexity of the algorithm it can be mentioned that on a Sun4 computer (Sparc Station 1) our non-optimized implementation of the grey-level blob detection algorithm (handling non-generic cases) takes about 5 seconds for an integer $256 * 256$ image and about 9 seconds for a floating point $256 * 256$ image. There are several optimizations that could be made if it is known for sure that the algorithm only needs to handle generic signals. The time required to compute the full scale-space representation (covering the scale range up to $t = 1024$) is about 30 minutes with the smoothing operation implemented as floating point convolutions in the spatial domain (with the truncation error ε set to 0.0005). These numbers were valid at the time of implementation (1989).

A.6.2 Bifurcation Statistics

The number of registered bifurcations can of course vary substantially from one image to another. However, in order to give a coarse indication of how many blob events can be expected to take place, we can mention that for a $256 * 256$ image treated in the scale interval $t \in [1, 1024]$ the following numbers can serve as guidelines:

- blob annihilations: 1000 (250-2000)
- blob merges: 300 (100-800)
- blob splits: 50 (20-100)
- blob creations: 30 (20-100)

A.6.3 Data Structure

In order to give a rough idea of what information can be available in a data structure representing the scale-space primal sketch, we briefly describe what kind of objects could be defined in an actual implementation of this concept and also what types of data can be stored in those, see also Figure 5.5.4.

grey-level blob :

polarity: bright or dark

scale level: pointer

extremum point: pointer

delimiting saddle point: pointer (Observe that in degenerate situations the extent of a grey-level blob could in fact be delimited by more than one saddle point.)

support region: pointer

grey-level blob volume

extremum point :

position: pixel coordinates (possibly several pixels for degenerate signals)

grey-level value

grey-level blob: pointer to the grey-level blob to which this saddle point serves as the seed.

saddle point :

position: pixel coordinates (possibly several pixels for degenerate signals)

grey-level value

grey-level blobs: pointers to the grey-level blobs to which this saddle point serves as a delimiting saddle point.

support region :

extreme coordinates: the minimum and maximum coordinate values along the x - and y -axes as well as the skewed 45-degree directions x' and y' .

blob area: number of pixels in the region.

first order moments: giving the center of gravity.

second order moments: allowing for an ellipse approximation giving the major and minor axes, which in turn give the orientation of the blob.

pixel representation: can be encoded either as a bit map in a blob image with label data or in a more compact form as e.g. run-length coding row by row.

boundary flag: telling whether the region belongs to the image boundary or not.

scale-space blob :

polarity: bright or dark

significance: normalized scale-space blob volume

bifurcation event at the appearance scale: pointer

bifurcation event at the disappearance scale: pointer

grey-level blobs: pointers to all the grey-level blobs the scale-space blob consists of.

appropriate scale level: pointer

grey-level blob at the appropriate scale level: pointer

boundary flag: telling whether there is a grey-level blobs belonging to the image boundary or not.

bifurcation event :

type of bifurcation: can be either

- one of the generic bifurcation situations: annihilation, merge, split, creation
- a non-generic complex bifurcation, with more than three scale-space blobs involved, that cannot be resolved into primitive transformations of the previously listed types
- a flag indicating that the minimum or the maximum scale of the analysis has been reached

participants from above: pointers to the scale-space blobs at the coarser scale that are involved in the bifurcation

participants from below: pointers to the scale-space blobs at the finer scale that are involved in the bifurcation

spatial position

bifurcation scale

scale level :

scale value

smoothed grey-level image: pointer

bright blob image: with all the bright grey-level blobs coded in a label image

dark blob image: with all the dark grey-level blobs coded in a label image

bright grey-level blobs: pointers to all the bright grey-level blobs at this scale

dark grey-level blobs: pointers to all the dark grey-level blobs at this scale

next coarser scale level: pointer

next finer scale level: pointer

Finally, it is convenient to create an object that can serve as a handle to all these subobjects:

scale-space primal sketch :

scale levels: it can be useful to represent all the scale levels accessed by the refinement algorithm both as a linked list and as a refinement tree.

bright scale-space blobs: pointers to all the bright scale-space blobs.

dark scale-space blobs: pointers to all the scale-space blobs.

bright bifurcations: pointers to all the bifurcations in which bright scale-space blobs are involved.

dark bifurcations: pointers to all the bifurcations in which dark scale-space blobs are involved.

Of course, depending on the actual application it will in some situations be computationally more efficient not to compute all this information when building the data structure and due to memory considerations, pieces of information may have to be thrown away during the process. For instance, regarding grey-level blobs it is in general sufficient to save only those blobs who correspond to the appropriate scale of a scale-space blob.

A.7 Test Images

In order to give a more detailed reproduction, Figure A.5 and Figure A.6 show larger size copies of some of the images used for the experiments in the thesis.

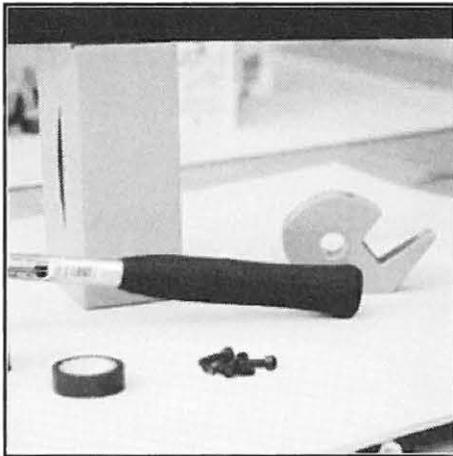
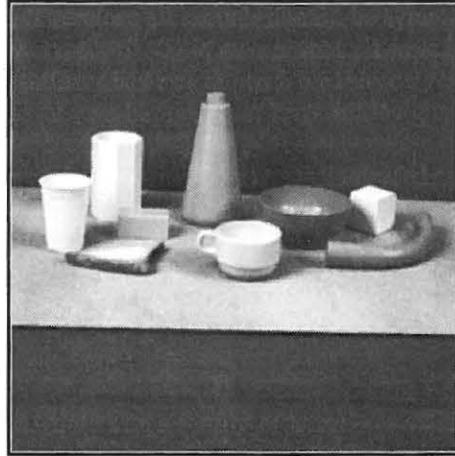
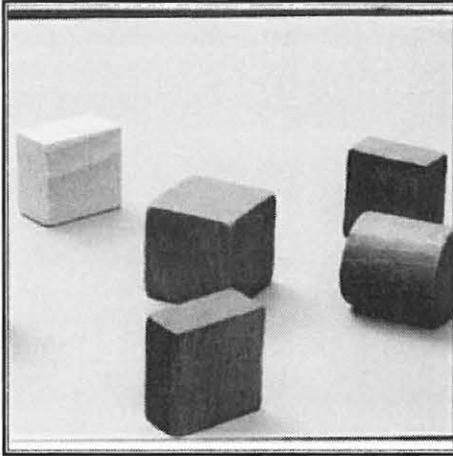


Figure A.5: Larger size copies of some of the original grey-level images used for the experiments.

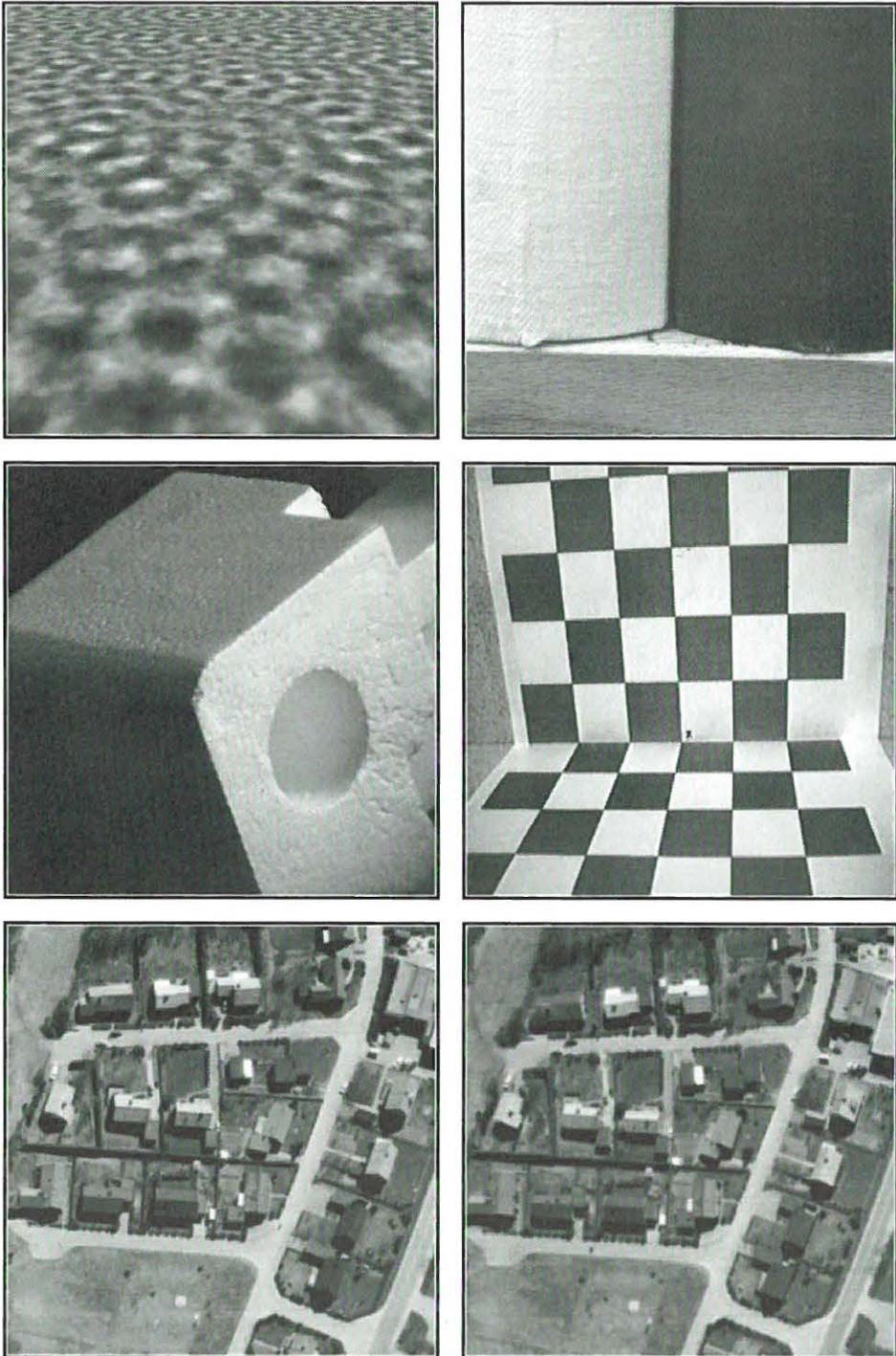


Figure A.6: Larger size copies of some of the original grey-level images used for the experiments.

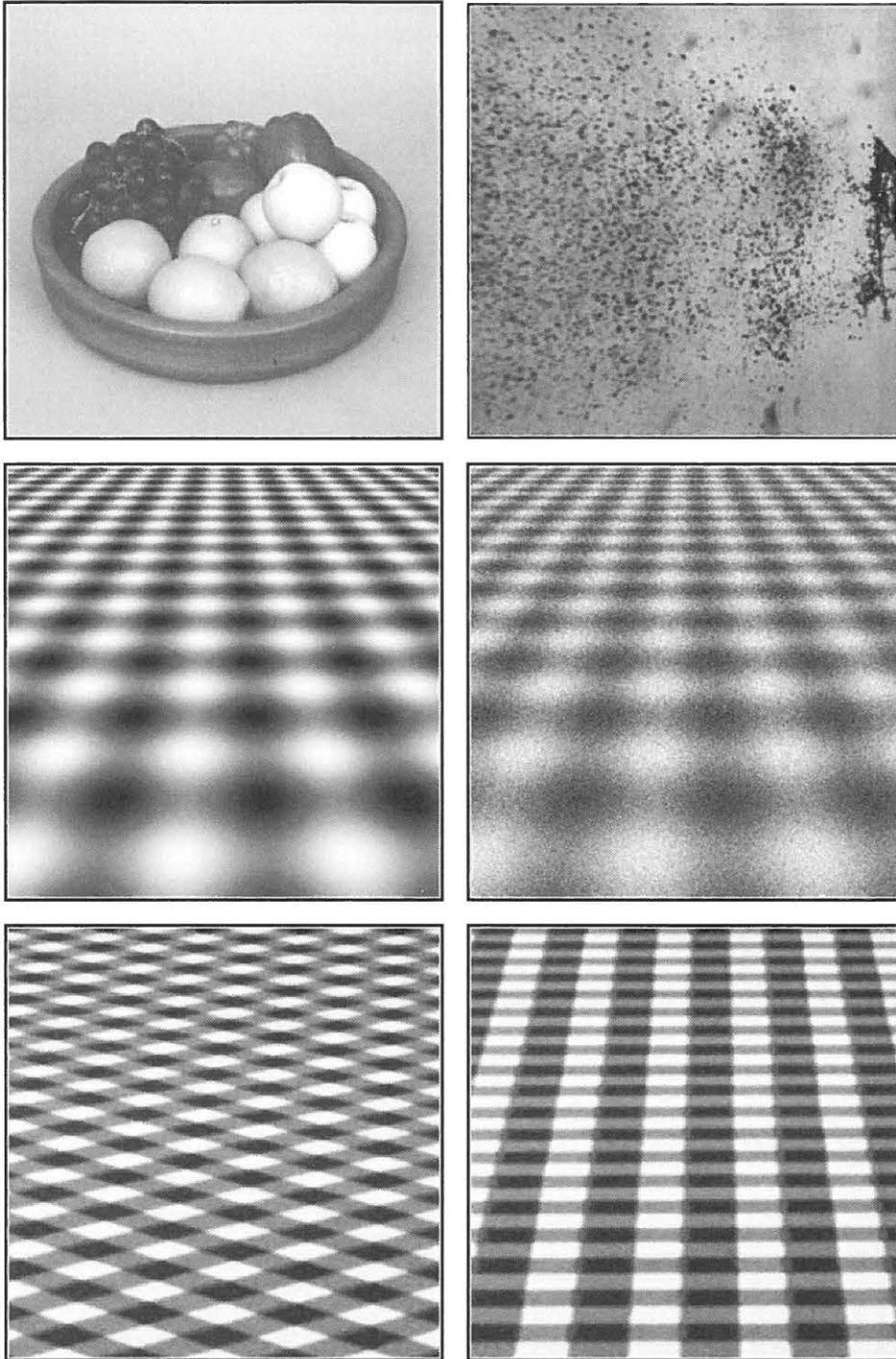


Figure A.7: Larger size copies of some of the original grey-level images used for the experiments.

Bibliography

- [Abr64] Abramowitz M., Stegun I.A. (1964) *Handbook of Mathematical Functions*, Applied Mathematics Series, **55**, National Bureau of Standards.
- [Ahu89] Ahuja N., Tuceryan M. (1989) "Extraction of Early Perceptual Structure in Dot Patterns: Integrating Region, Boundary and Component Gestalt", *Computer Vision, Graphics and Image Processing*, Vol. 27, pp304-356.
- [Alo89] Aloimonos J.Y., Schulman D. (1989) *Integration of Visual Modules: An Extension of the Marr Paradigm*, Academic Press, San Diego, California.
- [Alo90] Aloimonos, Y. (1990) "Purposive and Qualitative Active Vision", *Proceedings of the DARPA Image Understanding Workshop*, Pittsburgh, pp816-828.
- [And87] Ando T. (1987) "Totally Positive Matrices", *Linear Algebra and its Applications*, **90**, pp165-219.
- [Arn81] Arnold V.I. (1981) *Singularity Theory, Selected papers*, London Mathematical Society Lecture Note Series, **53**, Cambridge University Press, Cambridge.
- [Arn85] Arnold V.I., Gusein-Zade S.M., Varchenko A.N. (1985) *Singularities of Smooth Maps, Volume I*, Birkhäuser, Boston.
- [Arn88] Arnold V.I., Gusein-Zade S.M., Varchenko A.N. (1988) *Singularities of Smooth Maps, Volume II*, Birkhäuser, Boston.
- [Asa86] Asada H., Brady M. (1986) "The Curvature Primal Sketch", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp2-14.
- [Bab86] Babaud J., Witkin A.P., Baudin M., Duda R.O. (1986) "Uniqueness of the Gaussian Kernel for Scale-Space Filtering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 1, pp26-33.
- [Baj80] Bajcsy R., Rosenthal D.A. (1980) "Visual and Conceptual Focus of Attention", in Tanimoto S., Klinger A. (Eds), *Structured Computer Vision*, Academic Press, New York, pp133-147.
- [Baj85] Bajcsy R. (1985) "Active Perception vs Passive Perception", *Third Workshop on Computer Vision: Representation and Control*, Computer Society Press, Bellaire, MI.
- [Bak88] Baker H.H. (1988) "Surface Reconstruction from Image Sequences", *Proceedings of the 2:nd International Conference on Computer Vision*, Tampa, Florida, December 5-8, pp334-343.

- [Bal82] Ballard D., Brown C. (1982) *Computer Vision*, Prentice-Hall, New Jersey.
- [Bal90] Ballard D. (1990) *Animate Vision, Technical Report 329*, Computer Science, University of Rochester, Rochester, New York.
- [Bar88] Barnsley M.F., Devaney R.L., Mandelbrot B.B., Peitgen H-O., Saupe D., Voss R.F. (1988) *The Science of Fractals*, Springer-Verlag, New York, USA.
- [Bar78] Barrow H.G., Tenenbaum J.M. (1978) "Recovering Intrinsic Scene Characteristics from Images", in Hanson A.R., Riseman E.M. (Eds), *Computer Vision Systems*, Academic Press, New York.
- [Bar81] Barrow H.G., Tenenbaum J.M. (1981) "Computational Vision", *Proceedings of the IEEE*, Vol. 69, No. 5, pp572-595.
- [Ben86] Bengtsson A., Eklundh J.O., Howako J. (1986) "Shape Representation by Multiscale Contour Representation", *Technical Report TRITA-NA-8607, CVAP-27*, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Ben90] Bengtsson A., Eklundh J.O. (1990) "Contour Representation by Multiscale Approximation", *To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence*. Also available as *Technical Report TRITA-NA-9025, CVAP-76*, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Ber87] Bergholm F. (1987) "Edge Focusing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 6, pp726-741.
- [Ber89] Bergholm F. (1989) "On the Content of Information in Edges and Optical Flow", *Dissertation TRITA-NA-P8904*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Ber90] Bergholm F. (1990) *Personal communication*.
- [Ber84] Berzins V. (1984) "Accuracy of Laplacian Edge Detectors", *Computer Vision, Graphics and Image Processing*, **27**, 195-210.
- [Bie85] Biedermann I. (1985) "Human Image Understanding: Recent Research and a Theory", in Rosenfeld A. (Ed) *Human and Machine Vision II*, Academic Press, Boston, pp13-57.
- [Bil82] Billmeyer F., Saltzman M. (1982) *Principles of Colour Technology*, John Wiley and Sons.
- [Bin81] Binford T.O. (1981) "Inferring Surfaces from Images", *Artificial Intelligence*, pp205-244.
- [Bis88] Bischof W.F., Caelli T. (1988) "Parsing Scale-Space and Spatial Stability Analysis", *Computer Vision, Graphics and Image Processing*, **42**, pp192-205.
- [Bla87] Blake A., Zisserman A. (1987) *Visual Reconstruction*, MIT Press, Boston.

- [Blo87] Blostein D., Ahuja N. (1987) "Representation and Three-Dimensional Interpretation of Image Texture: An Integrated Approach", *Proceedings of the 1:st International Conference on Computer Vision*, London, England, pp444-449.
- [Blu90] Blum H., Florac L. (1990) *Personal Communication*.
- [Bol89] Boldt M., Weiss R., Riseman E. (1989) "Token-Based Extraction of Straight Lines", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 6, pp1581-?.
- [Boo78] de Boor C. (1978) *A Practical Guide to Splines*, Applied Mathematical Sciences, Vol. 27, Springer-Verlag, New York.
- [Bro81] Brooks R.A. (1981) "Symbolic Reasoning among 3-D models and 2-D Images", *Artificial Intelligence*, Vol. 17, pp285-348.
- [Bru84] Bruce J.W., Giblin P.J. (1984) *Curves and Singularities*, Cambridge University Press, Cambridge.
- [Bru89] Brunnström, K., Eklundh, J.O., Kakimoto, A., (1989), "On Focus-of-Attention by Active Focusing", *Proceedings of the NATO ASI on Robotics and Active Computer Vision*, Springer Verlag, New York, in press. Also available as *Technical Report TRITA-NA-P9004, CVAP-67*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Bru90a] Brunnström K., Eklundh J.O., Lindeberg T.P. (1990) "On Scale and Resolution in the Analysis of Local Image Structure", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp3-12. Also available as *Technical Report TRITA-NA-P9006, CVAP-69*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Bru90b] Brunnström K., Eklundh J.O., Lindeberg T.P. (1990) "On Scale and Resolution in Active Analysis of Local Image Structure", *Image and Vision Computing*, Vol. 8, No. 4, pp289-296. Also available as *Technical Report TRITA-NA-P9024, CVAP-75*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Bur81] Burt P.J. (1981) "Fast Filter Transforms for Image Processing", *Computer Vision, Graphics and Image Processing*, Vol. 16, pp20-51.
- [Bur83] Burt P.J., Adelson E.H. (1983) "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on Communications*, Vol. COM-9, No. 4, pp532-540.
- [Cam77] Campbell F.W.C., Robson J. (1977) "Application of Fourier Analysis to the Visibility of Gratings", *J. Physiol. (London)*, Vol. 197, pp417-424.
- [Can86] Canny J. (1986) "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp679-698.
- [Can86] Cantoni V., Levialdi S. (Eds) (1986) *Pyramidal Systems for Computer Vision*, Springer-Verlag, Berlin.

- [Car87] Carlotto M.J. (1987) "Histogram Analysis using a Scale-Space Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, pp121-129.
- [Chi76] Chillingworth D.R.J. (1976) *Differential Topology with a View to Applications*, Pitman Publishing, London.
- [Cho82] Chow S.-N., Hale J.K. (1982) *Methods of Bifurcation Theory*, Grundlehren der mathematischen Wissenschaften, **251**, Springer-Verlag, New York.
- [Com74] Comtet L. (1974) *Advanced Combinatorics*, Reidel, Dordrecht/Boston.
- [Cla88] Clark J.J. (1988) "Singularity Theory and Phantom Edges in Scale-Space", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-10, No. 5, pp720-727.
- [Cra67] Cramer H., Leadbetter M.R. (1967) *Stationary and Related Stochastic Processes*, John Wiley and Sons, New York.
- [Cro84a] Crowley J.L., Parker A.C. (1984) "A Representation of Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 2, pp156-169.
- [Cro84b] Crowley J.L., Stern R.M. (1984) "Fast Computation of the Difference of Low Pass Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, pp212-222.
- [Cro87] Crowley J.L., Sanderson A.C. (1987) "Multiple Resolution Representation and Probabilistic Matching of 2-D Gray-Scale Shape", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 1, pp113-121.
- [Dah74] Dahlquist G., Björk Å., Anderson N. (1974) *Numerical Methods*, Prentice-Hall, London.
- [Dan81] Danker A.J., Rosenfeld A. (1981) "Blob Detection by Relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 1, pp79-92.
- [Der87a] Deriche R. (1987) "Separable Recursive Filtering for Efficient Multi-Scale Edge Detection", *Proceedings International Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, Seiken Symposium, Tokyo, Japan, February 2-5.
- [Der87b] Deriche R. (1987) "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *International Journal of Computer Vision*, pp167-187.
- [Der90] Deriche R., Giraudon G. (1990) "Accurate Corner Detection: An Analytical Study", *Proceedings of the 3:rd International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp66-70.
- [Ehr78] Ehrich R.W., Lai P.F. (1978) "Elements of a Structural Model of Texture", *Proceedings of PRIP*, IEEE CS Press, pp319-326.

- [Fis86] Fischler M.A., Bolles R.C. (1986) "Perceptual Organization and Curve Partitioning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 1, pp100-105.
- [For89a] Forsyth D., Zissermann A. (1989) "Mutual Illumination", Proceedings of the CVPR-89, San Diego, California, June 4-6, pp466-473.
- [For89b] Forsyth D., Zissermann A. (1989) "Shape from Shading in the Light of Mutual Illumination", *Proceedings of the Fifth Alvey Vision Conference*, Reading, September 25-28, pp193-198.
- [For90] Forsyth D., Mundy J.L., Zisserman A., Brown C.M. (1990) "Projectively Invariant Representations Using Implicit Algebraic Curves", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp427-436.
- [Gab46] Gabor D. (1946) "Theory of Communication", *J. Inst. Elect. Eng.*, Vol. 93, pp429-457.
- [Gau88] Gauch J., Pizer S.M. (1988) "Image Description via the Multiresolution Intensity Axis of Symmetry", *Proceedings of the 2:nd International Conference on Computer Vision*, Tampa, Florida, December 5-8, pp269-274.
- [Gan59] Gantmacher F.R. (1959) *The Theory of Matrices*, Vols. I-II, Chelsea Publishing, New York.
- [Gib50] Gibson J.J. (1950) *The Perception of the Visual World*, Houghton Mifflin, Boston, Massachusetts.
- [Gib79] Gibson C.G. (1979) *Singular Points of Smooth Mappings*, Research Notes in Mathematics 25, Pitman Publishing, London.
- [Gol83] Golub G.H., van Loan C.F (1983) *Matrix Computations*, North Oxford Academic, Oxford
- [Gol85] Golubitsky M., Schaeffer D.G. (1985) *Singularities and Groups in Bifurcation Theory, Vol. I*, Applied Mathematical Sciences, 51, Springer-Verlag, New York.
- [Gra72] Gray R.M. (1972) "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices", *IEEE Transactions on Information Theory*, Vol. IT-18, No. 6, pp725-730.
- [Gre58] Grenander U., Szegö G. (1958), *Toeplitz Forms and Their Applications*, University of California Press, Los Angeles, California.
- [Gri85] Grimson W.E.L., Hildreth E.C. (1985) "Comments on Digital Step Edges from Zero Crossings of Second Directional Derivatives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 1, pp121-127.
- [Gro86] Gross A.D. (1986) "Multiresolution Object Detection and Delineation", *Technical Report TR-1613*, Computer Vision Laboratory, University of Maryland.
- [Går88] Gårding J. (1988) "Properties of Fractal Intensity Surfaces", *Pattern Recognition Letters*, Vol. 8, pp319-324.

- [Gär90] Gårding J., Lindeberg T.P. (1990) "CanApp — The Candela Application Library", *Technical Note CVAP-TN02*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.
- [Gär91] Gårding J. (1991) "Shape from Surface Markings", Ph.D. Dissertation, *In preparation*.
- [Hac85] Hackbush W. (1985) *Multi-Grid Methods and Applications*, Springer-Verlag, New York.
- [Har83] Haralick R.M., Watson L.T., Laffey T.J. (1983) "The Topographic Primal Sketch", *Int. J. Rob. Res.*, **2**, pp50-72.
- [Hil57] Hille E., Phillips R.S. (1957) *Functional Analysis and Semi-Groups*, American Mathematical Society Colloquium Publications, Vol. XXXI.
- [Hir55] Hirschmann I.I., Widder D.V. (1955) *The Convolution Transform*, Princeton University Press, Princeton, New Jersey.
- [Hor86] Horn B.K.P. (1986) *Robot Vision*, MIT Press, Cambridge, Massachusetts.
- [Hub88] Hubel D.H. (1988) *Eye, Brain and Vision*, Scientific American Library, Freeman, New York.
- [Hum86] Hummel R.A. (1986) "Representations Based on Zero-crossings in Scale Space", *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, June 1986, pp204-209.
- [Joh86] Johansen P., Skelboe S., Grue K., Andersen J.D. (1986) "Representing Signals by Their Top Points in Scale-Space", *Proceedings of the 8:th International Conference on Pattern Recognition*, Paris, France, October 27-31, pp215-217.
- [Jul83] Julesz B., Bergen J.R. (1983) "Textons, The Fundamental Elements in Preattentive Vision and Perception of Textures", *The Bell System Technical Journal*, Vol. 62, No. 6, pp1619-1645.
- [Jul86] Julesz B. (1986) "Texton Gradients: The Texton Theory Revisited", *Biological Cybernetics*, **54**, pp245-251.
- [Kak88] Kakimoto A. (1988) "An Algorithm for Finding Interest Points", *Technical Report TRITA-NA-P8807, CVAP-53*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Kar68] Karlin S. (1968) *Total Positivity*, Vol.I, Stanford University Press.
- [Kas87] Kass M., Witkin A., Terzopoulos D. (1987) "Snakes: Active Contour Models", *International Journal of Computer Vision*, Vol. 1, pp321-331.
- [Kim90] Kimia B.B., Tannenbaum A., Zucker S.W. (1990) "Toward a Computational Theory of Shape: An Overview", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp402-407.
- [Kit82] Kitchen, L., Rosenfeld, R., (1982), "Gray-Level Corner Detection", *Pattern Recognition Letters*, 1:2, pp95-102.

- [Kli71] Klinger A. (1971) "Pattern and Search Statistics", in Rustagi J.S. (Ed) *Optimizing Methods in Statistics*, Academic Press, New York.
- [Koe79] Koenderink J.J., van Doorn A.J. (1979) "The Structure of Two-Dimensional Scalar Fields with Applications to Vision", *Biological Cybernetics*, Vol. 33, pp151-158.
- [Koe84] Koenderink J.J., van Doorn A.J. (1984) "The Structure of Images", *Biological Cybernetics*, Vol. 50, pp363-370.
- [Koe86] Koenderink J.J., van Doorn A.J. (1986) "Dynamic Shape", *Biological Cybernetics*, Vol. 53, pp383-396.
- [Koe87] Koenderink J.J., van Doorn A.J. (1987) "Representation of Local Geometry in the Visual System", *Biological Cybernetics*, Vol. 55, pp367-375.
- [Koe88] Koenderink J.J., Richards W. (1988) "Two-Dimensional Curvature Operators", *Journal of the Optical Society of America*, Vol. 5, No. 7, pp1136-1141.
- [Koe90a] Koenderink J.J. (1990) *Solid Shape*, MIT Press, Boston.
- [Koe90b] Koenderink J.J., van Doorn A.J. (1990) "Generic Neighbourhood Operators", *Submitted*.
- [Koe90c] Koenderink J.J., van Doorn A.J. (1990) "Receptive Field Families", *Submitted*.
- [Kof35] Koffka K. (1935) *Principles of Gestalt Psychology*, Harcourt Brace, New York.
- [Kol90] Koller D., Nagel H.-H. (1990) "Ein robustes Verfahren zur Detection und Verfolgung bewegter Objekte in Bildfolgen", 12. *DAGM-Symposium*, Aalen, September, *Informatik-Fachberichte*, 254, pp625-633.
- [Kor88] Korn A.F. (1988) "Toward a Symbolic Representation of Intensity Changes in Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-10, No. 5, pp610-625.
- [Lev80] Levine M.D. (1980) "Region Analysis Using a Pyramid Data Structure", in Tanimoto S., Klinger A. (Eds), *Structured Computer Vision*, Academic Press, New York, pp57-100.
- [Ley84] Leyton M. (1984) "Perceptual Organization as Nested Control", *Biological Cybernetics*, Vol. 51, pp141-153.
- [Ley88] Leyton M. (1988) "A Process Grammar for Shape", *Artificial Intelligence*, Vol. 34, pp213-247.
- [Lif87] Lifshitz L.M., Pizer S.M. (1987) "A Multiresolution Hierarchical Approach to Image Segmentation Based on Intensity Extrema", *Internal report*, Departments of Computer Science and Radiology, University of North Carolina, Chapel Hill, N.C., U.S.A.

- [Lin88] Lindeberg T.P. (1988) "On the Construction of a Scale-Space for Discrete Images", *Technical Report TRITA-NA-P8808, CVAP-55*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.
- [Lin89] Lindeberg T.P. (1989) "Scale-Space for Discrete Signals", *Proceedings of the 6:th Scandinavian Conference on Image Analysis*, Oulu, Finland, June 19-22, pp1098-1107.
- [Lin90a] Lindeberg T.P. (1990) "Scale-Space for Discrete Signals", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 3, pp234-254.
- [Lin90b] Lindeberg T.P., Eklundh J.O. (1990) "Construction of a Scale-Space Primal Sketch", *Proceedings of the BMVC90 — British Machine Vision Conference*, Oxford, England, September 24-27, pp97-102.
- [Lin90c] Lindeberg T.P., Eklundh J.O. (1990) "Scale Detection and Region Extraction from a Scale-Space Primal Sketch", *Proceedings of the 3:rd International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp416-426.
- [Lin90d] Lindeberg T.P., Eklundh J.O. (1990) "The Scale-Space Primal Sketch: Construction and Experiments", To appear in *Image and Vision Computing*.
- [Lin90e] Lindeberg T.P., Eklundh J.O. (1990) "Guiding Early Visual Processing with a Scale-Space Primal Sketch", *Submitted*. Also available as *Technical Report TRITA-NA-P9013, CVAP-72*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Lin90f] Lindeberg T.P., Eklundh J.O. (1990) "Analysis of Aerosol Images Using the Scale-Space Primal Sketch", To appear in *Machine Vision and Applications*, Also available as *Technical Report TRITA-NA-P8920, CVAP-64*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Lin91a] Lindeberg T.P., Eklundh J.O. (1991) "On the Computation of a Scale-Space Primal Sketch", *Journal of Visual Communication and Image Representation*, Vol. 2, No. 1, pp55-78, 1991. Also available as *Technical Report TRITA-NA-9005, CVAP-68*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.
- [Lin91b] Lindeberg T.P. (1991) "On the Behaviour in Scale-Space of Local Extrema and Blobs", To appear in *The 7:th Scandinavian Conference on Image Analysis*, Aalborg, Denmark, August 13-16.
- [Lin91c] Lindeberg T.P. (1991) "On the Behaviour in Scale-Space of Local Extrema and Blobs", *Submitted*.
- [Lin91d] Lindeberg T.P. (1991) "Two-Dimensional Discrete Scale-Space Formulation", *Technical Report*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

- [Low85] Lowe D.G. (1985) *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, Boston.
- [Low88] Lowe, D. (1988) "Organization of Smooth Image Curves at Multiple Scales", *Proceedings of the 2:nd International Conference on Computer Vision*, Tampa, Florida, December 5-8, pp558-567.
- [Lu76] Lu Y.-C. (1976) *Singularity Theory and an Introduction to Catastrophe Theory*, Springer-Verlag, New York.
- [Mac88] Mackworth A.K., Mokhtarian, F. (1988) "The Renormalized Curvature Scale Space and the Evolution Properties of Planar Curves", *Proc. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, pp318-326.
- [Mal88] Mallat S.G. (1988) "Multiresolution Representations and Wavelets", *Dissertation*, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.
- [Mal89] Mallat S.G. (1989) "A Theory of Multiresolution Signal Processing: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, No. 6, pp674-693.
- [Mar76] Marr D. (1976) "Early Processing of Visual Information", *Phil. Trans. of the Royal Society (B)*, 27S, pp483-524.
- [Mar79] Marr D., Poggio T. (1979) "A Computational Theory for Human Stereo Vision", *Proceedings Royal Society London (B)*, pp301.
- [Mar80] Marr D., Hildreth E. (1980) "Theory of Edge Detection", *Proceedings of the Royal Society of London*, ser. B, Vol. 207, pp187-217.
- [Mar82] Marr D. (1982) *Vision*, Freeman, San Francisco.
- [Mey88] Meyer Y. (1988) *Ondlettes et Operateurs*, Hermann, 1988.
- [Mee87] Meer P., Baugher A.S., Rosenfeld A. (1987) "Frequency Domain Analysis and Synthesis of Image Pyramid Generating Kernels", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 6, pp512-522.
- [Mok86] Mokhtarian F., Mackworth A. (1986) "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp34-43.
- [Mok88] Mokhtarian, F. (1988) "Multi-Scale Representation of Space Curves and Three-Dimensional Objects", *Proc. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, pp298-303.
- [Mor77] Moravec, H.P. (1977) "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", *Stanford AIM-340*.
- [Mum83] Mumford D. (1983) *Tata Lectures on Theta I*, Birkhäuser, Boston.

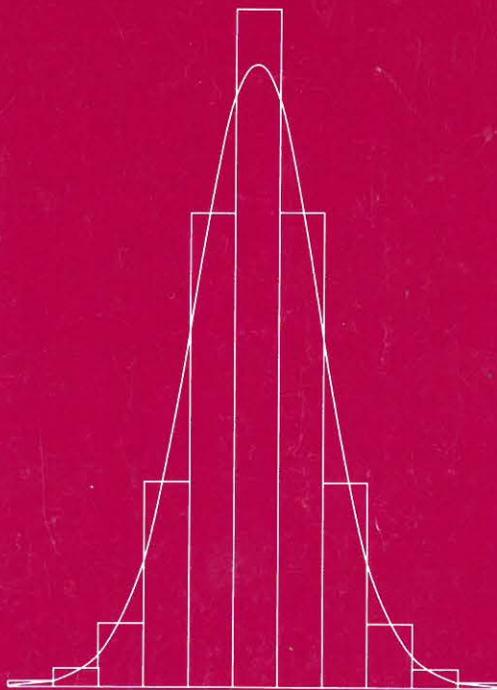
- [Müs89] Müssigmann U. (1989) "Texture Analysis, Fractals and Scale-Space Filtering", *Proceedings of the 6:th Scandinavian Conference on Image Analysis*, Oulu, Finland, June 19-22, pp987-993.
- [Nal86] Nalwa V.S., Binford T.O. (1986) "On Detecting Edges", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp699-714.
- [Nay90] Nayar S.K., Ikeuchi K., Kanade T. (1990) "Shape from Interreflections", *Proceedings of the 3:rd International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp2-11.
- [Nob88] Noble J.A. (1988) "Finding Corners", *Image and Vision Computing*, Vol. 6, No. 2, pp121-128.
- [Nor90] Nordström N. (1990) "Biased Anisotropic Diffusion — A Unified Regularization and Diffusion Approach to Edge Detection", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp18-27.
- [Nor60] Norman E. (1960) "A Discrete Analogue of the Weierstrass Transform", *Proceedings of the American Mathematical Society*, 11, pp596-604.
- [Pah90] Pahlavan K., Eklundh J.O. (1990) "A Head-Eye System for Active, Purposive Computer Vision", *Technical Report TRITA-NA-P9031, CVAP-80*, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Pap72] Papoulis A. (1972) *Probability, Random Variables and Stochastic Processes*, McGraw-Hill.
- [Pow81] Powell M.J.D. (1981) *Approximation Theory and Methods*, Cambridge University Press, Cambridge.
- [Pel78] Peleg, S., (1978), "Iterative Histogram Modification 2", *IEEE Transactions on Systems Man and Cybernetics*, Vol. 8, pp555-556.
- [Pen87] Pentland A. (1987) "Recognition by Parts", *Proceedings of the 1:st International Conference on Computer Vision*, London, England, June 8-11, pp612-620.
- [Pen88] Pentland A. (1988) "Automatic Extraction of Deformable Part Models", *Technical Report 104*, Vision Sciences Group, Media Lab, M.I.T., Massachusetts.
- [Pen90] Pentland A. (1990) "Extraction of Deformable Part Models", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp397-401.
- [Pen90] Pentland A. (1990) "Photometric Motion", *Proceedings of the 3:rd International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp178-187.
- [Per90] Perona P., Malik J. (1990) "Scale-Space and Edge Detection using Anisotropic Diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 7, pp629-639.

- [Piz87] Pizer S.M., Oliver W.R., Bloomberg S.H. (1987) "Hierarchical Shape Description Via the Multiresolution Symmetry Axis Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp505-511.
- [Piz88] Pizer S.M., Gauch J.M., Lifshitz L.M., Oliver W.R. (1988) "Image Description via Annihilation of Essential Structures", *Technical report 88-001*, Departments of Computer Science and Radiology, University of North Carolina, Chapel Hill, N.C., U.S.A.
- [Pos78] Poston T., Stewart I. (1978) *Catastrophe Theory and its Applications*, Pitman Publishing, London.
- [Pre86] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., (1986) *Numerical Recipes*, Cambridge University Press.
- [Ren90] Rendahl M., Gårding J., Lindeberg T.P. (1990) "Candela 1.0 — An Overview", *Technical Note CVAP-TN07*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.
- [Ren91] Rendahl M. (1991) "Candela 1.0 Programmer's Manual", Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, *In preparation*.
- [Rey87] Reynolds G., Beveridge J.R. (1987) "Searching for Geometric Structure in Images of Natural Scenes", *Proceedings of the Image Understanding Workshop.???*
- [Ric45] Rice S.O. (1945) "Mathematical Analysis of Random Noise", *Bell System Technical Journal*, Vol. XXIV, No. 1, pp46-156.
- [Roh90] Rohr K. (1990) "Über die Modellierung und Identifikation charakteristischer Grauwertverläufe in Realweltbildern", *12. DAGM Symposium Mustererkennung*, September 24-26, Oberkochen-Aalen, Germany.
- [Ros71] Rosenfeld A., Thurston M. (1971) "Edge and Curve Detection for Visual Scene Analysis", *IEEE Transactions on Computers*, Vol. 20, No. 5, pp562-569.
- [Ros77] Rosenfeld, A. and Davis L. S., (1977), "Iterative Histogram Modification", *TR-519*, University of Maryland Computer Science Center, Maryland.
- [Ros82] Rosenfeld A., Kak A.C. (1982) *Digital Picture Processing*, Academic Press, New York.
- [Ros84] Rosenfeld A. (1984) *Multiresolution Image Processing*, Springer-Verlag, New York.
- [Rud76] Rudin, W. (1976) *Principles of Mathematical Analysis*, McGraw-Hill.
- [San90] Sander P.T., Zucker S.W. (1990) "Charting Surface Structure", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp418-426.
- [San90] Sander P.T., Zucker S.W. (1990) "Inferring Surface Trace and Differential Structure from 3-D images", To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

- [Sau89] Saund E. (1989) "Adding Scale to the Primal Sketch", Proceedings of the CVPR-89, San Diego, California, June 4-6, pp70-78.
- [Sau90] Saund E. (1990) "Symbolic Construction of a 2-D Scale-Space Image", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 8, pp817-831.
- [Sch30] Schoenberg I.J. (1930) "Über Variationsvermindernde Lineare Transformationen", *Mathematische Zeitschrift*, **32**, pp321-328.
- [Sch46] Schoenberg I.J. (1946) "Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions", *Quarterly of Applied Mathematics*, **4**, pp45-99.
- [Sch47] Schoenberg I.J. (1947) "On Totally Positive Functions, Laplace Integrals and Entire Functions of the Laguerre-Pölya-Schur type", *Proceedings of the National Academy of Sciences*, Vol. 33, pp11-17.
- [Sch48] Schoenberg I.J. (1948) "Some Analytical Aspects of the Problem of Smoothing", Courant Anniversary Volume, *Studies and Essays*, New York, pp351-370.
- [Sch50] Schoenberg I.J. (1950) "On Pölya frequency functions. II. Variation-diminishing integral operators of the convolution type", *Acta Sci. Math. (Szeged)*, **12**, pp97-106.
- [Sch53] Schoenberg I.J. (1953) "On Smoothing Operations and their Generating Functions", *Bull. Amer. Math. Soc.*, **59**, pp199-230.
- [Sch88] Schulmann D., Aloimonos J. (1988) "Boundary Preserving Regularization: Theory Part I", *Technical Report CS-TR-2011*, Computer Vision Laboratory, University of Maryland.
- [She86] Sher A.C., Rosenfeld A. (1986) "Detection and Delineation of Compact Objects Using Intensity Pyramids", *Technical Report TR-1729*, Computer Vision Laboratory, University of Maryland, Maryland.
- [She87] Sher A.C., Rosenfeld A. (1987) "Detecting and Extracting Compact Textured Objects Using Pyramids", *Technical Report TR-1789*, Computer Vision Laboratory, University of Maryland, Maryland.
- [Sjö88] Sjöberg F., Bergholm F. (1988) "Extraction of Diffuse Edges by Edge Focusing", *Pattern Recognition Letters*, Vol. 7, pp181-190.
- [Spi68] Spiegel M.R. (1968) *Mathematical Handbook of Formulas and Tables*, Schaum's Outline Series in Mathematics, McGraw-Hill.
- [Str86] Strang G. (1986) *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Massachusetts.
- [Str83] Strömberg J.O. (1983) "A Modified Franklin System and Higher Order Splines as Unconditional Basis for Hardy Spaces", in Beckner W. et al. (Eds), *Proceedings of the Conference in Harmonic Analysis in Honor of Antoni Zygmund*, Vol. II, Wadworth Mathematical Series.

- [Sze90] Szeliski R. (1990) "Fast Shape from Shading", *Proceedings of the 1:st European Conference on Computer Vision*, Antibes, France, April 23-37, pp359-368.
- [Ter86] Terzopoulos D. (1986) "Regularization of Inverse Visual Problems Involving Discontinuities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp413-424.
- [Ter87] Terzopoulos D., Witkin A., Kass M. (1987) "Symmetry-Seeking Models for 3-D Object Reconstruction", *Proceedings of the 1:st International Conference on Computer Vision*, London, England, pp269-276.
- [Ter88] Terzopoulos D., Witkin A., Kass M. (1988) "Constraints on Deformable Models: Recovering 3-D Shape and Nonrigid Motion", *Artificial Intelligence*, Vol. 36, pp91-123.
- [Toe11] Toeplitz O. (1911) "Zur Theorie der quadratischen und bilinearen Formen von unendlichvielen Veränderlichen. I. Teil: Theorie der L-formen", *Mathematische Annalen*, **70**, pp351-376.
- [Tor86] Torre V., Poggio T.A. (1986) "On Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 2, pp147-163.
- [Tso90] Tsotsos J.K. (1990) "Analyzing Vision at the Complexity Level", *Behavioural and Brain Sciences*, Vol. 13, pp423-469.
- [Ull84] Ullman S. (1984) "Visual Routines", *Cognition*, Vol. 18, pp97-159.
- [Val89a] Valdsöo T. (1989) "The Jet Structure from Pneumatic Atomization Nozzles", *Technical Note*, Department of Aeronautics, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Val89b] Valdsöo T. (1989) "The Spray Structure from Typical Fuel Injectors: with Different Liquids and Varying Air-Assist", *Technical Note*, Department of Aeronautics, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Val89c] Valdsöo T. (1989) "Some Characteristics of Liquid Jet Disintegration Processes in High Velocity Air Flow", *Technical Note*, Department of Aeronautics, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Vor87] Voorhees H., Poggio T. (1987) "Detecting Textons and Texture Boundaries in Natural Images", *Proceedings of the 1:st International Conference on Computer Vision*, London, England, pp250-258.
- [Wat88] Watt R. (1988) *Visual Processing: Computational, Psychophysical and Cognitive Research*, Lawrence Erlbaum Associates, London.
- [Whi52] Whittaker E.T., Watson G.N. (1952) *Modern Analysis*, Cambridge University Press.
- [Wil83] Wilson, H.R., (1983), "Psychophysical Evidence for Spatial Channels", in Brad-dick, O.J., Sleigh, A.C. (Eds), *Physical and Biological Processing of Images*, Springer Verlag, New York.

- [Wit83a] Witkin A.P. (1983) "Scale-Space Filtering", *Proceedings of the 8:th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 8-12, pp1019-1022.
- [Wit83b] Witkin A.P., Tenenbaum J.M. (1983) "On the Role of Structure in Vision", in Beck J., Hope B., Rosenfeld A. (Eds), *Human and Machine Vision*, Academic Press, New York.
- [Wit87] Witkin A., Terzopoulos D., Kass M. (1987) "Signal Matching through Scale-Space", *International Journal on Computer Vision*, Vol. 1, pp133-144.
- [You85] Young R.A. (1985) "The Gaussian Derivative Theory of Spatial Vision: Analysis of Cortical Cell Receptive Field Line-Weighting Profiles", *Technical report GMR-4920*, Computer Science Department, General Motors Research Laboratories, Warren, Michigan.
- [You86] Young R.A. (1986) "Simulation of the Human Retinal Function with the Gaussian Derivative Model", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June22-26, pp 564-569.
- [Yse86] Yserentant H. (1986) "On the Multi-Level Splitting of Finite Element Spaces", *Numerische Mathematik*, Vol. 49, pp379-412.
- [Yui86] Yuille A., Poggio T. (1986) "Scaling Theorems for Zero-Crossings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 1, pp15-25.
- [Yui88] Yuille A.L. (1988) "The Creation of Structure in Dynamic Shape", *Proceedings of the 2:nd International Conference on Computer Vision*, Tampa, Florida, December 5-8, pp685-689.
- [Zha91] Zhang W., Bergholm F. (1990) "An Extension to Marr's Signature Based Edge Classification", to appear in *The 7:th Scandinavian Conference on Image Analysis*, Aalborg, Denmark, August 13-16. Also available as *Technical Note CVAP-TN-04*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.
- [Zhu86] Zhuang X., Huang T.S. (1986) "Multi-Scale Edge Detection with Gaussian Filters", *International Conference on Acoustics, Speech and Signal Processing*, Tokyo, Japan, pp2047-2050.
- [Zuc83] Zucker S.W. (1983) "Computational and Psychophysical Experiments in Grouping: Early Orientation Selection", in Beck J., Hope B., Rosenfeld A. (Eds), *Human and Machine Vision*, Academic Press, New York.



Computational Vision and Active Perception Laboratory (CVAP)