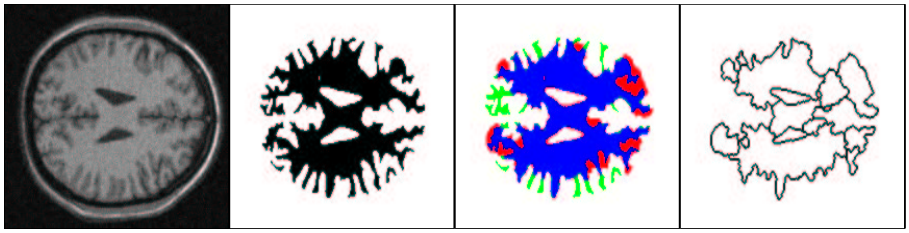
Evaluation of Diffusion Schemes for Multi-scale Watershed Segmentation

Erik B. Dam

May 19, 2000



Abstract

In [Olsen, 1996], [Olsen, 1997], and [Olsen & Nielsen, 1997] Ole Fogh Olsen and Mads Nielsen design a promising semi-automatic segmentation method. The method is theoretically founded in watershed segmentation, linear scale-space theory, and singularity theory.

The method is appealing since it provides the user with building blocks which are fitted to the geometry of the relevant image. These building blocks allow the user to construct the desired objects in an intuitive manner, that requires nothing other from the user than knowledge of the segmentation task objects.

This thesis has three major contributions:

Evaluation
: An evaluation methodology is designed. The evaluation methods analyse how well the generated building blocks generated from the segmentation method are suited for construction of a given object. The reference objects used in the evaluation are ground truth segmentations made by experts in the field of radiology. The ground truth segmentations are performed on MR brain scans. The evaluation method gives a quantitative measure of the applicability of the segmentation method with respect to data similar to the test data.

Performance of Diffusion Schemes
: The evaluation method forms a solid foundation for improvement of the segmentation method. Empirical studies are performed where the underlying diffusion scheme of the segmentation method is replaced by various other schemes. Central alternative schemes are Joachim Weickerts anisotropic nonlinear diffusion methods [Weickert, 1998a]. However, the best scheme is a new scheme called GAN.

Diffusion
: The performances of the different variants of the diffusion schemes offer insight in the nature of diffusion. For the specific segmentation task, it is found that the effect of anisotropy in the diffusion scheme is insignificant compared to the effect of nonlinearity in the diffusivity function.

Preface

This work constitutes Erik Dam's Master's Thesis. The thesis is a mandatory part of the M.Sc. (Cand.Scient.) degree in computer science from the Department of Computer Science at the University of Copenhagen, Denmark.

The thesis is written in the period from august 1999 to february 2000 under the supervision of associate professor Mads Nielsen from the IT University of Copenhagen.

Address

The author of this thesis can be contacted at the following address:

University of Copenhagen
Department of Computer Science
Universitetsparken 1
DK-2100 Copenhagen
Denmark
Att: Erik Dam
Email: `erikdam@diku.dk`

Acknowledgements

A number of persons have been helpful in the process of writing this thesis:

Associate professor Ph.D. Mads Nielsen (IT University of Copenhagen)
Supervision and in-expendable inspiration.

Phd. stud. Ole Fogh Olsen (University of Copenhagen, Department of Computer Science)
Original matlab watershed segmentation code and answers to numerous silly questions.

Associate professor Ph.D. Joachim Weickert (University of Mannheim, Dep. of Mathematics and Computer Science)
Original implementation of edge enhancing filter and very useful comments on the details of the anisotropic nonlinear filters.

Ph.D. Jon Sporring (ICS-FORTH, Science and Technology Park of Crete)
Matlab implementation of linear Gaussian scale-space.

Assistant professor Ph.D. Stephen Alstrup (IT University of Copenhagen)
Inspiration for efficient algorithms for evaluation methods.

Cand. scient Martin Koch
Careful Proof-reading.

Phd. stud. Martin Lillholm (IT University of Copenhagen)
Computational resources and support for the optimisation programs.

Phd. stud. Theo Engell-Nielsen (IT University of Copenhagen)
Computational resources.

The evaluation is dependent of the availability of ground truth data. The following institutions have provided these:

Brain Web:
The *McConnell Brain Image Centre* at *Montreal Neurological Institute and Hospital* provides simulated MR brain scans at `http://www.bic.mni.mcgill.ca/brainweb/`

Internet Brain Segmentation Repository:
MR brain data sets 788_6_m, 1320_2_max, and 20$Normals_T1$ and their manual segmentations are provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at `http://neuro-www.mgh.harvard.edu/cma/ibsr`

Contents

List of Figures

Truth rests in the eye of the beholder
Proverb

Chapter 1

Introduction

Computer vision is the research field which attempts to extend a computer system with the ability to see. The human visual system is extremely capable — in order to make an automated visually oriented computer system all we need to do is therefore to simulate the human visual system.

A central part of a any visual system is segmentation. This allows the image of the physical scene to be split into individual objects. The separated objects can then be analysed further for visualization, measurement, tracking, navigation, and various other tasks.

a *b* *c*

Figure 1.1: *Three images which are used to demonstrate some of the difficulties in simulating the human visual system. Figure a is "Holger", figure b is "Lena", and figure c is some trees.*

So how does the human visual system perform segmentation?

The examples in figure 1.1 offer some inspiration.

A quick glance at the image starring *Holger* suggests that the segmentation task is quite easy: The salient objects correspond to areas with approximately uniform intensity. However, this principle is far from adequate for the image of *Lena*. A differentiation of the objects based on intensity alone would make it hard to separate the chin from the shoulder and the hat from the background. As a contrast, segmentation based on the intensity alone would split the feather boa, the hair, and last but not least the nose. These problems demonstrate that texture as well as notions such as "connectedness" are taken into account when the images are partitioned. Furthermore, the human visual system "cheats" and recognizes a "person". This allows use of abstract a priori knowledge of what a person is supposed to look like. The final image is another example of how the visual system recognizes the objects and then extrapolates the perception of the rest of the image. Even though it is virtually impossible to see this due to shadow, we are able to "see" where the stem of the tree reaches the ground. We are able to separate the foreground tree from the right background tree (even though the small branches from the foreground tree have approximately same pattern, size, and intensity as the large branches from the background tree). Furthermore we also have an approximate sense of the distance to the background trees. And not only can the human visual system have a clear perception of areas where it is hard to extract precise information — the visual system has no problem seeing things that are nearly invisible. Does *Holger* have two arms or a hump on his back?

Obviously, the examples reveal little about how the human visual system actually performs segmentation — it is not even clear that segmentation is a separate step in the process. However, the examples do suggest that a lot of information is needed in order to simulate the abilities of the human visual system. We need to model local attributes as colour, intensity, and texture; as well as more global properties that allow perceptual grouping of connected areas. On top of this we must model the vast amount of acquired knowledge that allows that human visual system to extrapolate from the image and form a full perception of the scene.

Even though this is a very short listing of some of the problems with simulation of the human visual system, it would appear that this simulation is virtually impossible in any near future (an educated guess from an "expert" is in around 2040 [Moravec, 1999]).

Another key factor, that makes the design of a general automatic segmentation system even more difficult, is the fact that the "true" segmentation cannot be derived from an image alone. The desired segmentation depends on a fiducial task as well. Different tasks require different output from the same image. Truth rests in the eye of the beholder — not in the image.

Instead of attempting to handle the general task of automatic segmentation in one giant leap we must therefore approach the problem in smaller steps. The segmentation problem can be simplified in a number of ways:

Constrained environment
: A warehouse robot can navigate flawlessly under the assumption that the "world" consists entirely of a white floor and shelves with red supports.

Reduction of the required output
: A burglar alarm only needs to determine whether there is substantial movement in the scene — a detailed description of the scene is irrelevant.

User interaction
: A system that performs the tedious part of a task and allows a user to perform the problem specific choices can achieve the best of both worlds: semi-automation of the task combined with the quality performance of the expert user.

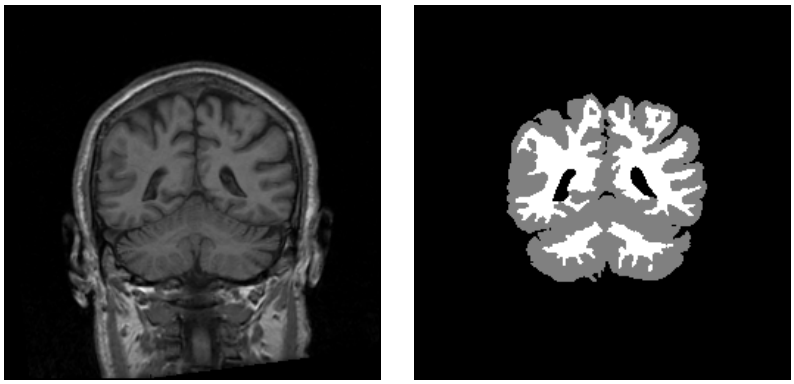
Quantitative a priori knowledge
: The knowledge of a ball's approximate size, colour, and shape combined with the fact that it must obey the newtonian laws of motion, allows a sports analysis program to track the ball.

1.1 The Focus of this Thesis

The data used in this thesis are images acquired through medical scanning. This is an example of a mildly controlled environment, in the sense that we know more or less what to expect from images obtained from the different medical scanning modalities.

The complexity of the segmentation task varies a lot with medical images. Segmentation of bone from CT scannings is a relatively trivial task, since the intensity of the areas corresponding to bone is distinct from the intensities of the other tissue types. As a contrast, segmentation of tiny blood vessels from PET or SPECT scannings is a difficult task since the vessels are quite thin and twisted. Images acquired via MR scans are also quite complicated to segment. This is due to the low intensity contrast between the areas corresponding to different types of soft tissue.

The majority of the images used in this thesis are MR brain scans. The MR images offer challenging data sets and the different types of brain tissue provide objects with qualitatively different shapes. An example MR brain scan is displayed in figure 1.2.



a *b*

Figure 1.2: *Slice from MR brain scan from the Internet Brain Segmentation Repository. These images are introduced in section 8.2. It is not trivial to segment the MR scan in figure a into the regions corresponding to the grey matter and white matter tissue areas of the brain. The regions in figure b are the result of a manual segmentation by an expert in the field of radiology.*

The segmentation method that this thesis investigates is a semi-automatic method developed by Ole Fogh Olsen ([Olsen, 1996], [Olsen, 1997], and [Olsen & Nielsen, 1997]). The method requires the interaction of a user in order to perform the segmentation.

Automatic methods for segmentation of the brain exist as well. An example is the method from [Atkins & Mackiewich, 1996] which locates the entire brain from a MR scan. However, this method will not work for segmentation of, for instance, the grey matter tissue only.

In general, the user interaction of the semi-automatic methods allows the methods to be non-committed in the sense that they can be used for segmentation of any type of images. Other examples of semi-automatic methods are:

- The *Hyperstack* [Vincken, 1995]. This is actually an automatic method with an interactive correction phase after the actual segmentation.
- The image processing toolbox system from [Chung & Sapiro, 1999]. This is a versatile user guided semi-automatic method. However, the toolbox-nature of the system requires the user to have image processing knowledge.
- Recently, another method that resembles the one from [Olsen, 1996] has been published [Gauch, 1999].

The approach by Ole Fogh Olsen is unique in the sense that it allows an application area expert with no image processing knowledge to handle the segmentation in a very intuitive way. But even though this approach is intuitively appealing, it is not clear how good it actually is.

In order to establish this, a proper evaluation is necessary. This is much in the line of the work performed on the hyperstack by Koster [Koster, 1995].

1.2 The Contributions of this Thesis

Evaluation methods are developed that allow evaluation of the segmentation method from [Olsen, 1996].

The evaluation is performed mainly with MR brain scans, where the focus is segmentation of the white matter and grey matter tissues of the brain. The evaluation is based on comparison with ground truth segmentations of the data sets.

Once an evaluation method is established, it is obvious to attempt to improve the method. This is the second major purpose of the thesis. This is done by exploring the use of other diffusion schemes than linear Gaussian in the multi-scale setting of the method. The central alternatives are different anisotropic nonlinear diffusion schemes [Weickert, 1998a]. Furthermore, some new diffusion schemes are designed.

The performance of the different diffusion schemes offers a perspective on the significance of the different characteristic features of the diffusion schemes. The significances of nonlinearity and anisotropy are investigated.

1.3 How to Read this Thesis

The thesis is split into four main parts:

Part One: *Multi-scale Watershed Segmentation by Linear Diffusion*
This part introduces the theoretical foundation for the basic segmentation method. These chapters are quite "light" reading but serve as a presentation of the fundamental concepts and problems that must be considered in the later parts. An exception is chapter 4: *Catastrophe Theory Perspective*. The theory presented here is far more rigorous than needed — it is written in order to satisfy "advanced readers" and the author himself.

Part Two: *Evaluation of Segmentation Methods*
A comprehensive treatment of the development of the evaluation methods. This includes efficient algorithms for implementation.

Part Three: *Alternative Diffusion Schemes*
The effect of changing the underlying scale-space concept is discussed. A number of different diffusion schemes are presented with focus on the anisotropic nonlinear diffusion of Joachim Weickert. Three new schemes are designed as well.

Part Four: *Results & Conclusion*
The evaluation is presented and discussed.

Part I

Multi-scale Watershed Segmentation by Linear Diffusion

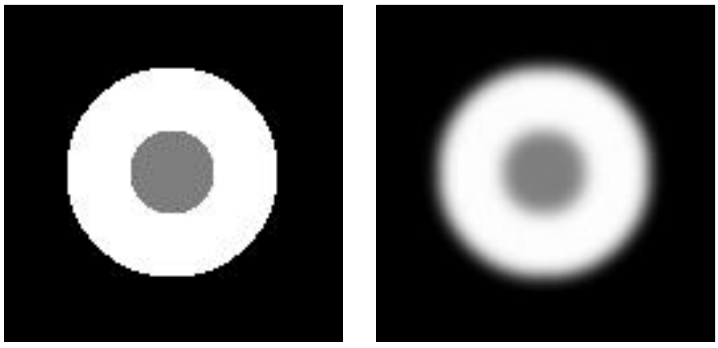
The tendency of man's nature to good, is like the tendency of water to flow downward.
Meng-Tse

Chapter 2

Introduction to Watershed Segmentation

Imagine rain pouring down a hilly landscape. After hitting the ground the individual drops run downhill and gather in pools. Every time a drop hits a certain spot the drop will run into the same pool. This implicitly partitions the landscape into regions of support for each pool.

A part of the landscape that leads water to a specific pool belongs to the *catchment basin* for this pool. The borders between the catchment basins are the ridges of the landscape — the *watersheds*.



a *b*

Figure 2.1: *The image used for the watershed segmentation example. Figure a is the original 128 by 128 image. Figure b is smoothed with a Gaussian filter of scale 3— a volcano!*

How can this be used for segmentation? Obviously the landscape is divided into a number of regions — the catchments basins. To see whether this partitioning is suitable for segmentation we look at a simple example. We begin with the wheel of figure 2.1*a*. In order to make it look a bit more like a landscape we smooth it. If we perceive the image as a height map (bright areas are high) the result is a volcano-like landscape (figure 2.1*b*).

To see how the watersheds partition this example, we look at the profile (figure 2.2*b*) across the center of the landscape. The water will either flow into the middle or out into the exterior. The watersheds form a circle at the top of the volcano (figure 2.2*c*). This divides the image into two segments: the center circle containing part of the wheel; and the exterior containing the rest of the wheel and the background. Certainly not a very good segmentation. What we really would like is one segment for the interior circle, one for the ring, and one for the exterior.

a *b* *c*

Figure 2.2: *The smoothed volcano (figure a) from figure 2.1b with a plot of the intensities of the center line (figure b). The watersheds form a circle in the center of the volcano ring (figure c). Note that c is merely an illustration of how the watersheds would be ideally. Since the smoothing is done with a finite kernel, the image is constant in large areas. Watersheds are not well-defined in constant areas.*

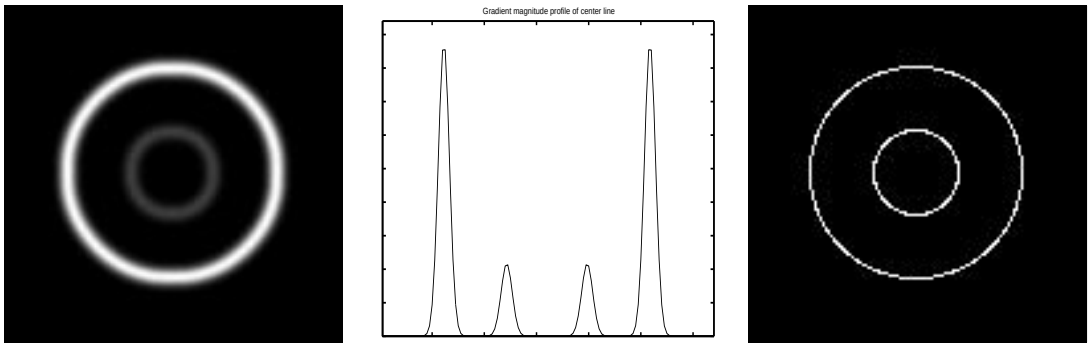
2.1 Dissimilarity Measure

The problem with this segmentation is that the water flows away from parts of the image with high intensity. Therefore the image is divided in the middle of the volcano ridge rather than at the edges. Instead the water should flow away from the parts of the image where there is a large "change".

The simple solution is to let the water flow in an image of a dissimilarity measure. Where the original image changes the most, there will be a watershed in the dissimilarity image.

A number of dissimilarity measures are possible. In simple grey-scale images we perceive areas with similar intensity to be connected regions. A possible dissimilarity measure is therefore a rate of change in the intensity — the gradient magnitude is a reasonable choice.

Figure 2.3 illustrates the use of the gradient magnitude as dissimilarity measure. We can see that the gradient magnitude image has two circular watersheds approximately at the edges of the top of the volcano. The corresponding catchment basins therefore provide a reasonable segmentation of the original image.



a *b* *c*

Figure 2.3: *The gradient magnitude of the example image. Figure a shows the gradient magnitude of the image in 2.1b. Figure b is a plot of the center line of the gradient magnitude image. Figure c shows the watersheds for the gradient magnitude. Again the watersheds in figure c are illustrations of they would ideally be.*

2.2 Innate Problems

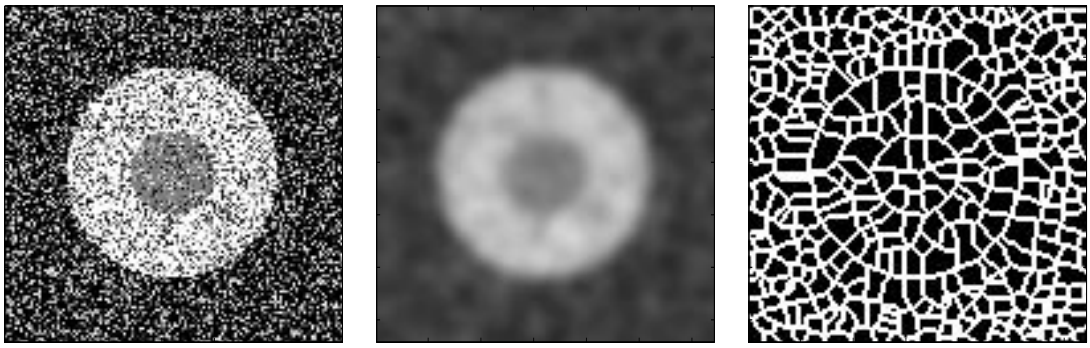
The first example was extremely simple. In order to reveal some of the problems with watershed segmentation we make it a bit more difficult by adding some noise.

The segmentation process requires calculation of the gradient magnitude. In a mathematical sense, the gradient is not well-defined for a discrete image. And a simple discrete approximation will not handle a noisy image very well. We therefore smooth the image in order to overcome this problem.

Figure 2.4 shows the result of a watershed segmentation on the smoothed version (figure 2.4*b*) of the noisy image (figure 2.4*a*). The segmentation consists of a lot of small regions (figure 2.4*c*). Even though the ring is clearly visible for the human eye in figure 2.4*b*, the watershed algorithm results in a plethora of superfluous catchment basins. This *over-segmentation* is a well-known problem with the watershed segmentation method.

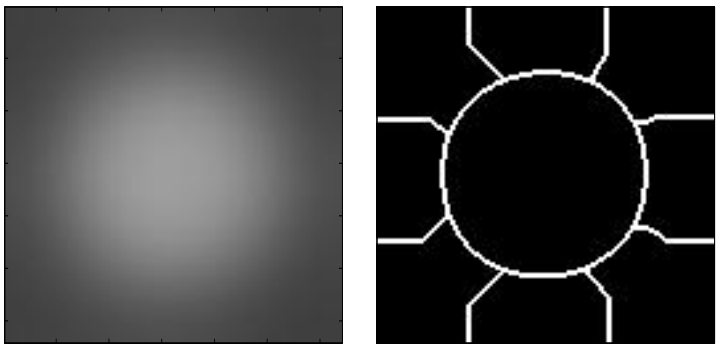
The small catchment basins in figure 2.4*c* are due to the small fluctuations caused by smoothed noise. Therefore it is natural to smooth the image even further to see if the fluctuations can be smoothed away. Figure 2.5 shows the result of smoothing until the volcano can be identified as a single catchment basin.

However, this segmentation is not quite satisfying. First, the inside of the volcano has melted together with the surrounding ring due to the smoothing. It is not possible to preserve the small inside region while smoothing enough to capture the larger surrounding ring as a single region. Secondly, the outside border of the volcano has been dislocated due to the high degree of smoothing.



a *b* *c*

Figure 2.4: *Watersheds for the example image with added noise. In figure a 50% of the original pixels have been replaced by a random intensity value. Figure b shows this image blurred with a Gaussian filter of scale 3. The resulting watersheds are shown in figure c.*



a *b*

Figure 2.5: *Watersheds for the noisy volcano image at high scale. Figure a shows the image from figure 2.4a blurred with a Gaussian filter of scale 18. The resulting watersheds are shown in figure b.*

2.3 More on Watersheds

The original concept from geography was introduced in mathematics in the nineteenth century by Cayley and Maxwell[1].

A more recent treatment is [Lopez et al., 1999].

By the way, according to Encyclopedia Britannica the term watershed is actually a drainage basin. Furthermore, "*the term has also been used synonymously with drainage divide, but this use is discouraged*". Apparently, the computer vision community is not discouraged.

[1] Watersheds were introduced in [Cayley, 1859] and [Maxwell, 1870] according to [Koenderink, 1984] — I have not been able to obtain these papers.

2.4 How is this ever going to work?

The watershed segmentation method is intuitively simple and has some appealing features:

- The watersheds form closed curves by definition. This leaves no need for a post-processing step where border elements are connected into complete borders.
- The formulation is very general and encompasses a number of segmentation methods depending on the dissimilarity measure.

However, there are a number of problems — some of which were introced in this chapter. Some of these are general image analysis considerations while others are specific for watershed segmentation.

In the following chapters we will try to deal with some of these problems:

1. How is the gradient calculated in a well-defined manner?
2. How is the over-segmentation problem handled?
3. How do regions develop as the image is smoothed?
4. What do we mean by "smoothing" the image?
5. What is the best way to smooth the image?
6. How is the dislocation of borders due to smoothing handled?
7. How are regions of different sizes handled simultaneously?
8. How do we know if the resulting segmentation is "good"?

For those with an eye for the finer detail — we salute you!

Frankie Goes To Hollywood

Chapter 3

Appetizers from Scale-Space Theory

In the previous chapter, the basics of watershed segmentation were introduced. The method had a number of innate problems. The concept of *Scale-Space* gives theoretically well-founded solutions to some of these problems:

- What do we mean by "smoothing" the image?
- How is the gradient calculated in a well-defined manner?
- How are regions of different sizes handled simultaneously?

This chapter gives a short introduction to Scale-Space theory. This field of research is in itself quite extensive. The purpose of this introduction is not to give an overview of the area but merely to emphasize some concepts (and the notation) that will be used in the rest of this thesis.

3.1 The General Concept

The n-dimensional intensity image I regarded as a function is generalised into a one-parameter family of n-dimensional images $L \in C^\infty(\mathbb{R}^{1+n}, \mathbb{R})$:

$$I(\vec{x}) \quad \curvearrowright \quad L(\sigma; \vec{x}) \tag{3.1}$$

The parameter σ is called the *scale*. The family includes the original image at scale 0:

$$L(0; \vec{x}) = I(\vec{x}) \tag{3.2}$$

A number of axioms are generally associated with this family of images. Among the most important ones are the *causality principle*, *average grey level invariance*, and the *semi-group property*.

The first two deal with the simplifying properties of the family. The causality principle states that no maximum or minimum is enhanced with increasing scale. Combined with the average grey level invariance this gives a notion of the image becoming gradually smoother tending to a constant function as the scale tends to infinity.

The semi-group property states that any member of the family can be regarded as the generating image for the rest of the family.

3.2 Linear Gaussian Scale-Space

Different authors use their own set of axioms. Common for these axiomatics is that if the mapping into the family of images is required to be *linear*, the scale-space is uniquely defined.

In this case, each member of the family of functions for the n-dimensional image I can be calculated explicitly by a convolution with a Gaussian kernel with standard deviation σ:

$$G(\sigma;\vec{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}\, exp(-\frac{\vec{x}^T\vec{x}}{2\sigma^2}) \tag{3.3}$$

$$L(\sigma;\vec{x}) = I(\vec{x}) * G(\sigma;\vec{x}) \tag{3.4}$$

$$\equiv \int_{\mathbb{R}^N} G(\sigma;\vec{x}-\vec{y})I(\vec{y})\, d\vec{y} \tag{3.5}$$

The scale-space images $L(\sigma;\vec{x})$ are analytical for $\sigma > 0$ if the original image function is integrable.

Since the family can be constructed using a convolution kernel, the process is translationally invariant. Since the kernel is symmetric, it is also rotationally invariant. A process which is rotationally invariant is also called *isotropic*.

Here translational and rotational invariance are stated as properties of the scale-space. A number of authors state them explicitly as axioms. The table in figure 3.1 summarizes common axiomatics.

3.3 Heat Diffusion

Another formulation of the linear Gaussian scale-space is through a PDE (partial differential equation). The *heat diffusion equation* describes the linear Gaussian scale-space[1]:

$$\frac{\partial L(t;\vec{x})}{\partial t} = \Delta L(t;\vec{x}) = L_{ii}(t;\vec{x}) \tag{3.6}$$

The PDE has the original image I as initial condition: $L(0;\vec{x}) = I(\vec{x})$

The Gaussian convolution kernel is the *Green's function* for the heat diffusion equation. Note however, that the "time" parameter t is different from the standard deviation parameter σ in equation 3.5. More precisely $t = \frac{1}{2}\sigma^2$

Due to the relationship with the heat diffusion equation, the smoothing process is often referred to as a diffusion process.

[1] In equation 3.6 the Einstein summation convention defines $L_{ii}(t;\vec{x}) \equiv L_{x_1x_1}(t;\vec{x}) + I_{x_2x_2}(t;\vec{x}) + \cdots + I_{x_nx_n}(t;\vec{x})$. Note that only spatial coordinates are affected in the summation.

	K	L	A	F	N
Convolution kernel		.	.	.	
Semigroup property		.	.	.	.
Locality			.		
Regularity		.	.	.	
Causality	.	.			
Nonnegativity			.	.	
Tikhonov regularisation					.
Average grey level invariance		.	.		
Isometry invariance		.	.	.	.
Homogeneity & Isotropy	.				
Scale invariance				.	.
Valid for dimension	1,2	1	n	n	n

Figure 3.1: Overview of Gaussian scale-space axiomatics.
The acronyms are: K = [Koenderink, 1984], L = [Lindeberg, 1990], A = [Alvarez et al., 1993], F = [Florack, 1996], and N = [Nielsen et al., 1997].
The table is part of a table appearing in [Weickert, 1998a].

3.4 Differentiability

Differentiation of an image function is not a well-defined operation since the image function is not differentiable in general (and typically unknown as well). Linear scale-space theory solves this elegantly. Given an arbitrary differential operator D the operation is performed:

$$\begin{aligned} DL(\sigma;\vec{x}) &= D\left(I(\vec{x}) * G(\sigma;\vec{x})\right) \\ &= I(\vec{x}) \;*\; DG(\sigma;\vec{x}) \end{aligned} \tag{3.7}$$

The differential operator is applied to the infinitely differentiable Gaussian kernel. Thereby the differentation becomes well-defined[2].

An important aspect is that the differentiation has a context — the scale. Applying a differential operator directly on the image is ill-defined. A *Gaussian derivative* not only delivers a result — it also states/requires the scale at which the derivative is observed.

3.5 The Scale Concept

The scale-parameter σ has a nice intuitive meaning in the linear setting. At a particular scale σ, a given point is represented by a weighted average of the point and its neighbouring points from the original image. The bell-shaped Gaussian is the weighting function.

[2]The bright reader will argue that the ill-defined differentiation is simply replaced by an ill-defined convolution over an unknown function. However, since convolution is simply integration, this operation can be discretized in a satisfying manner by a summation over the image pixels. So this is merely a problem of numerical approximation accuracy rather than of the operation being ill-defined.

Since σ is the standard deviation of the Gaussian, it gives a straight-forward interpretation. Informally, the scale can be perceived as the "width" of the support for a point.

In an image the highest meaningful scale is the image size — this is called the *outer scale* . Conversely, the *inner scale* is defined by the resolution of the image. Typically this is the pixel size.

3.6 The Superficial and Deep Structure

It is important to stress that the original image is not simply replaced by another image at a certain scale. Instead the focus is the entire scale-space of images with scale ranging from the inner to the outer scale.

As an example, a feature detector will typically locate features of a certain size at a specific scale. In order to analyze the presence of these features in the image, it is not sufficient to locate them at a fiducial scale — they must be located at all scales. Different representatives of the feature will be located at different scales depending on their size.

In order to fully understand the significance of a feature, the development of the feature must be tracked through the full range of scales where it is detected. In some cases, the longevity over scale of a feature can be regarded as a measure of the importance of the feature ([Witkin, 1983] and [Lindeberg, 1994]).

The development of features over scale is termed the *deep structure*, whereas the features present at a particular scale is called *superficial structure*.

3.7 More Detailed Treatment of Scale-Space Theory

This chapter introduces only the few concepts from scale-space theory that are central for the chapters to come. Obviously, a mere three pages is insufficient for a proper presentation of scale-space theory. For this purpose a few pointers are given:

- The basis of scale-space theory can be studied in the classics [Koenderink, 1984] and [Witkin, 1983]. Another perspective of the early work is [Weickert et al., 1997].
- Overviews of the different axiomatic foundations for scale-space theory are given in both [Weickert, 1998a] and [Sporring et al., 1997].
- The concept of derivatives in the setting of linear scale-space has been studied carefully by Lindeberg (for example in [Lindeberg, 1996a], [Almansa & Lindeberg, 1997], and [Lindeberg, 1996b]). Another approach is given in [Pedersen & Nielsen, 1999].
- Comprehensive introductions to scale-space theory are [ter Haar Romeny, 1994] and [Sporring et al., 1997

3.8 Possibly there is Hope?

Chapter 2 ended with quite a lot of open problems. Scale-space theory solves some of these and offers inspiration for others.

The concept of *smoothing* is given a solid theoretical framework that yields a well-defined derivation operation as well. Thereby the *gradient* can be calculated. This allows us to do what we implicitly did in chapter 2: Establish watersheds in blurred versions of the original image.

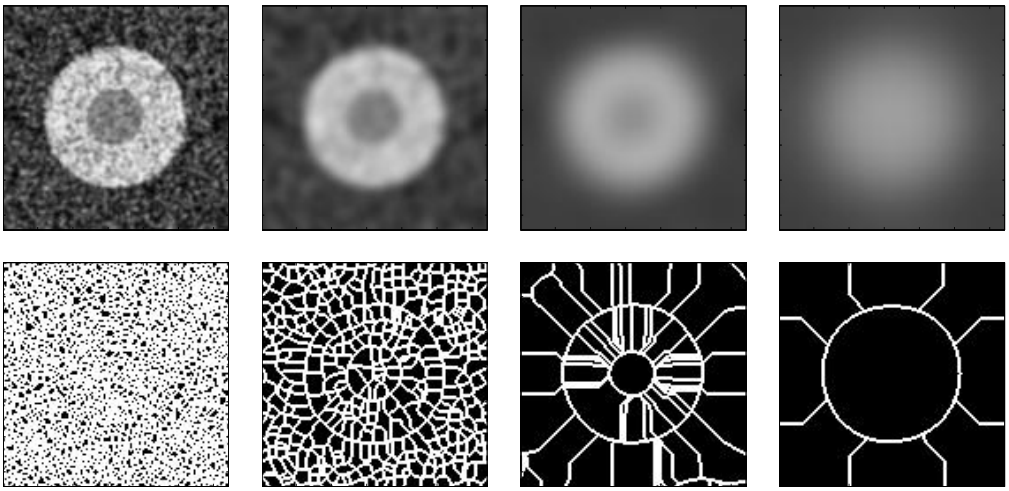


Figure 3.2: *The image from figure 2.1a with 50 % of the pixel replaced by random intensities. In the top row the image is shown at four different scales: 1, 3, 10, and 18. The bottom row shows the corresponding watersheds.*

Figure 3.2 shows the noisy example image from chapter 2 on four different scales with the corresponding watersheds. It is apparent that the image is simplified as the scale increases. On the lowest scale an abundance of watersheds is present due to the noise in the image. The next scale reveals the structure of the inner and outer ring, but there is still a lot of spurious watersheds. The inner ring is present as a single region on the third scale — but apparently the watersheds do not correspond exactly to the location of the ring in the original image. On the largest scale the outer ring is present as a single region.

Even though scale-space theory presented a nice framework that handles the smoothing concept and the derivation problem, the examples in figure 3.2 pose new questions:

- Obviously watersheds disappear with increasing scale. It is not clear from the example whether new watersheds are ever created?
- Watersheds corresponding to the inner and outer rings are visually present on different scales. However, it is not clear how to describe the concept of "correspondance". Theory formalizing the notions of *creation*, *longevity*, and *annihilation* is needed.
- If corresponding watersheds can be identified at different scales, their location apparently changes. How is this dislocation (due to blurring) handled?

The time you enjoy wasting is not wasting time.
T.S. Elliot

Chapter 4

Catastrophe Theory Perspective

Scale-space theory offers insight on how to handle objects of different sizes while diminishing the effects of noise. Furthermore, instead of simply viewing diffusion as a noise reduction process, the theory suggests investigating the deep structure — in this case the deep structure of the catchment basins. However, in order to look at the deep structure, it is necessary to handle some new problems:

- How is the notion of correspondance of regions across scale formalized?
- How do catchment basins behave as the scale increases — apparently regions disappear but are new regions ever created?
- When do these disappearances (and possibly creations) occur?

The appropriate theoretical framework for handling these questions is *catastrophe theory*. Catastrophe theory is the study of the development of the structure of the functions in families of functions. The structure of a function can be described in many ways — a possible choice is by the singularities. Therefore, catastrophe theory is also known as *singularity theory*.

A scale-space is clearly a family of functions, but a catchment basin is not a singularity. In order to apply catastrophe theory, it is therefore simply necessary to realize that each catchment basin contains exactly one minimum for the dissimilarity measure. The catchment basins can therefore be identified by the corresponding minima.

So the aim is to establish whether minima for the gradient magnitude squared (or another dissimilarity measure) are created or annihilated as scale increases. Before getting that far, an introduction to catastrophe theory is in place.

The introduction will contain known results from catastrophe theory with a few simple additional propositions. The propositions are special cases of more general theorems and are only added because the simple proofs yield useful intuition.

A little warning is in order. This chapter is longer and more rigorous than strictly required for this thesis. However, the reader must rest assured that writing it in this detail provided the author with useful insight.

4.1 Introduction to Catastrophe Theory

The focus of attention is a m parameter family of a n parameter real function. The m parameters that generates the family are called *control parameters*. The remaining n parameters are called *state parameters*.

A *singularity* is defined as a singularity with respect to the state parameters for some fixed set of control parameters.

A *catastrophe* is the creation or annihilation of a singularity as the control parameters change.

Let's formalize this. Let $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$. The functional f is written $f(c_1, \ldots, c_m; x_1, \ldots, x_n)$ dividing the parameters of f into the m control parameters $c_1, \ldots, c_m$ and the n state parameters $x_1, \ldots, x_n$.

The *gradient* ∇f is defined by the partial derivatives of the state parameter only:

$$\nabla f(\vec{c}; \vec{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right) \equiv (f_{x_1}, \ldots, f_{x_n}) \tag{4.1}$$

The *hessian* H is defined accordingly:

$$H(f, \vec{c}, \vec{x}) = \begin{pmatrix} f_{x_1 x_1} & \cdots & f_{x_1 x_n} \\ \vdots & \ddots & \vdots \\ f_{x_n x_1} & \cdots & f_{x_n x_n} \end{pmatrix} \tag{4.2}$$

Definition 1. (Singularity Set)
A *singularity* for the function $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$ occurs at $(\vec{c}; \vec{x}) \in \mathbb{R}^{m+n}$ if and only if:

$$\nabla f(\vec{c}; \vec{x}) = 0$$

The *singularity set* for the function $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$ is defined:

$$\mathcal{S}_f = \{(\vec{c}, \vec{x}) \in \mathbb{R}^{m+n} \mid \nabla f(\vec{c}; \vec{x}) = 0\}$$

A singularity is also called a *critical point*.

Definition 2. (Catastrophe Set)
A *catastrophe* for the function $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$ occurs at $(\vec{c}; \vec{x}) \in \mathbb{R}^{m+n}$ if and only if:

$$\nabla f(\vec{c}; \vec{x}) = 0 \quad \text{and} \quad Det(H(f, \vec{c}, \vec{x})) = 0$$

The *catastrophe set* for the function $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$ is defined:

$$\mathcal{C}_f = \{(\vec{c}, \vec{x}) \in \mathcal{S}_f \mid Det(H(f, \vec{c}, \vec{x})) = 0\}$$

The singularity set $\mathcal{S}_f$ and the catastrophe set $\mathcal{C}_f$ are central in the description of the structure of a function. In order to understand this, it is necessary to look more closely at the notion of "structure" of a function.

4.1.1 The Topology of Functions

In the following we will investigate how the structure of a function changes when the function is changed "a little". In order to do this, we need a formal definition of the distance between functions. A distance function induces a metric which in turn induces a topology. The topology used here is defined in terms of the taylor series coefficients.

Given an analytical function $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$ the taylor series expansion (at $\vec{0}$ for simplicity) is defined (using the Einstein summation convention):

$$f(\vec{x}) = f(\vec{0}) + x_i f_i + \frac{1}{2!} x_i x_j f_{ij} + \frac{1}{3!} x_i x_j x_k f_{ijk} + \dots$$

An analytical function $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$ can then locally, at a given $\vec{x}_0 \in \mathbb{R}^n$, be represented by the coefficients $(f, f_i, f_{ij}, \dots)$. If the coordinate set is truncated to terms of degree k, this gives a set of coordinates in a D-dimensional space (where $D = \frac{(n+k)!}{n!k!}$). These truncated coordinates are called the *k-jet* and denoted $j^k f(x_0)$.

Now a topology can be defined via one of the regular distance functions on $\mathbb{R}^D$. We will use the ∞-norm $\|\cdot\|_\infty$ and denote the corresponding topology $T^\infty(\mathbb{R}^n, \mathbb{R})$.

The topology allows us to introduce the notion of a *pertubation*, i.e. a slight change of the function.

4.1.2 The Structure of Functions

If a function is perturbed ever so slightly, it becomes a different function. The two functions will be almost identical but not equal. When looking at the qualitative nature of functions, the strict notion of equality is not applicable. A weaker equivalence relation is more appropriate for describing the structure of a function.

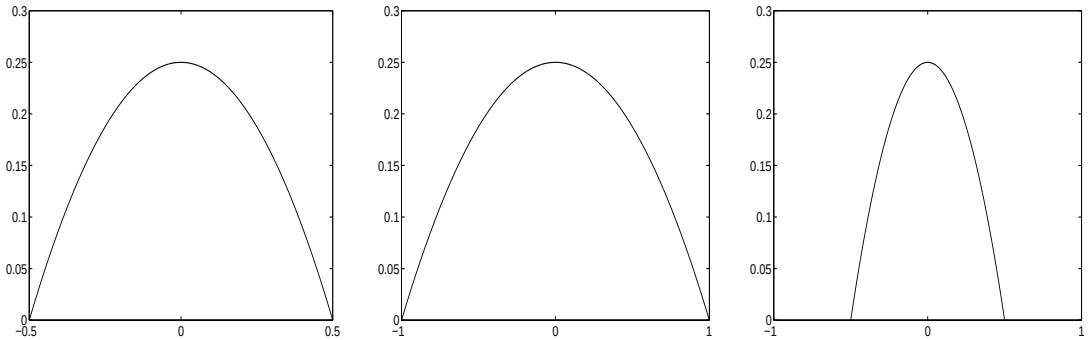


Figure 4.1: *Three structurally equivalent function all belonging to the class of convex parabola.*

Figure 4.1 shows three functions. At first glance the first two look equal. This is due to the simple trick of using different coordinate axes. Actually the functions in figure 4.1*a* and 4.1*c* are equal. However, the function in figure 4.1*b* is obviously "similar". An equivalence relation that describes the qualitative structure of functions should place the three functions in the same equivalence class. This inspires the following definitions.

Definition 3. (Local Structural Equivalence)
Let $f, g \in C^\infty(\mathbb{R}^n, \mathbb{R})$. The two functions are *locally structurally equivalent* at a point $x_0 \in \mathbb{R}^n$ if there exists a smooth, invertible (with a smooth inverse) change in coordinates $\vec{x}'(\vec{x})$ such that $f(\vec{x}) = g(\vec{x}'(\vec{x}))$ in a neigborhood around x_0.
That two functions f, g are locally structurally equivalent is denoted: $f \doteq g$

Definition 4. (Global Structural Equivalence)
Two functions are *globally structurally equivalent* if they are locally structurally equivalent at all points.

The definitions require a few remarks:

- Strictly speaking, the definition should state $f(\vec{x}) = g(\vec{x}'(\vec{x})) + constant$. It is implicit that the value of the function does not affect the structure.
- Generally in this thesis, "smooth" means *infinitely* differentiable.
- The requirement that the coordinate change is invertible ensures that the equivalence relation is symmetrical (and generally sensible).
- A smooth, invertible function with a smooth inverse is also called a *diffeomorphism*.
- The definition for global structural equivalence does not require the coordinate change to be the same at all points.
- The coordinate change is a diffeomorphism if and only if it has a non-zero Jacobian (the determinant of the Jacobi-matrix with elements $\frac{\partial x_i'}{\partial x_j}$)

The concept of structural equivalence can be used to categorise the space of functions. This is done in section 4.1.5. At this point we will only guide the intuition by an example and a simple proposition.

Using definition 3 it is now possible to formalize the "similarity" between the two functions in figure 4.1*a* and 4.1*b*. The coordinate change function is simply $x'(x) = 2x$ with the inverse $x(x') = \frac{1}{2}x'$ — both functions are smooth. By definition, the two functions are therefore globally structurally equivalent.

The following simple proposition demonstrates the significance of singularities in connection with structural equivalence.

Proposition 1. (Local Structure of Singularities)
Let $f, g \in C^\infty(\mathbb{R}^n, \mathbb{R})$. Assume that g has a singularity at $\vec{x}_0$: $\nabla g(\vec{x}_0) = 0$.
If f and g are locally structurally equivalent f has a corresponding singularity.

Proof of Proposition 1
Let $\vec{x}'(\vec{x})$ be a coordinate change function such that $f(\vec{x}'(\vec{x}_0)) = g(\vec{x}_0)$. First $\nabla(f(\vec{x}'(\vec{x}))) = (\nabla(f(\vec{x}'(\vec{x}))))J(\vec{x}'(\vec{x}))$ where J is the Jacobi matrix $J(\vec{x}'(\vec{x}))$ with elements $\frac{\partial x_i'}{\partial x_j}$. Since $\nabla g(\vec{x}_0) = 0$ we have that $\nabla(f(\vec{x}'(\vec{x}_0))) = 0$. This means that either $(\nabla f)(\vec{x}'(\vec{x}_0)) = 0$ or that the Jacobian of the coordinate change function is zero.
If the Jacobian is zero f are g are not locally structurally equivalent. If $\nabla(f(\vec{x}'(\vec{x}_0))) = 0$ the function f has a singularity at the point $\vec{x}'(\vec{x}_0)$.

This simple proposition implies that if two functions are structurally equivalent then for each singularity in one function there is a corresponding singularity in the other. This justifies that the singularity set is relevant in the description of the basic structure of a function.

4.1.3 Stability

The structure of a function can be described by the set of singularities. This is why a structural change in this set is considered a "catastrophe". This is also captured in the concept of stability. A stable function is a function where a small pertubation (in terms of the $T^\infty(\mathbb{R}^n, \mathbb{R})$ topology) will not change the structure of the function.

Definition 5. (Structural Stability)
A function $F \in C^\infty(\mathbb{R}^n, \mathbb{R})$ is *locally stable* at $\vec{x_0} \in \mathbb{R}^n$ if the effect of a small pertubation $\epsilon f(\vec{x})$ (where $\epsilon \in \mathbb{R}$ and $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$) does not change the structure of the function. In terms of definition 3: $F + \epsilon f \doteq F$
A function is *structurally stable* if it is locally stable at all points.

A function will be locally stable in a neighbourhood around a singularity unless the hessian is degenerate. If the hessian is degenerate, a small pertubation can remove or change the singularity. The hessian is therefore also called the *stability matrix*. Otherwise the pertubation will only move the singularity and change its value slightly — but the structure of the function in a neighbourhood around the singularity will remain unchanged.

A function f with an empty catastrophe set $\mathcal{C}_f$ will be structurally stable.

Proposition 2. (Stability Matrix)
A singularity is stable unless the hessian is degenerate

4.1.4 An Example

As an illustration of the concepts, lets look at the family $f \in C^\infty(\mathbb{R}^{1+2}, \mathbb{R})$:

$$f(c; x, y) = x^3 + cx - 24y^2.$$

The singularity set for the family is:

$$\mathcal{S}_f = \{(c; x, y) \in \mathbb{R}^3 \mid 3x^2 = -c \ \wedge \ y = 0\}$$

The catastrophe set is:

$$\begin{aligned} \mathcal{C}_f &= \{(c;x,y) \in \mathbb{R}^3 \mid 3x^2 = -c \ \wedge \ y = 0 \ \wedge \ 12x = 0\} \\ &= \{(0;0,0)\} \end{aligned}$$

Figure 4.2 shows a function from the family at a pair of singularities. Both the saddle and the local maximum are stable (in the sense that the function is locally stable in a neighborhood around them) — therefore the function is structurally stable.

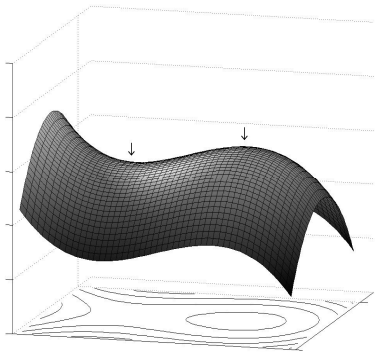


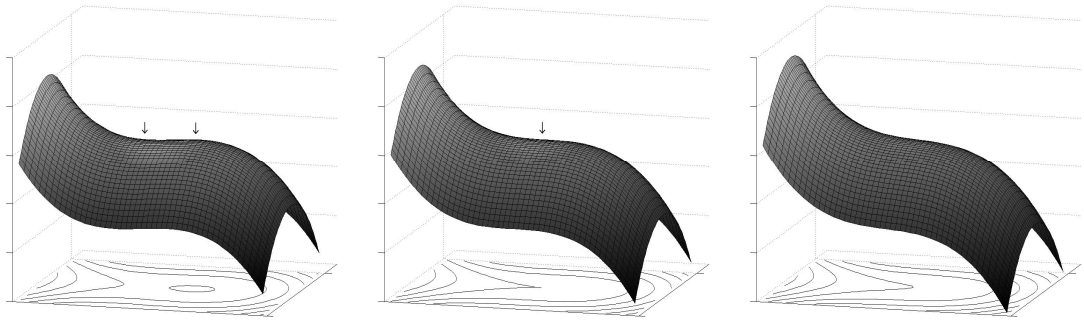
Figure 4.2: *The function $f(c,x,y) = x^3 + cx - 24y^2$ for $c = -300$. The singularity set consists of a saddle at $(x,y) = (10,0)$ (below the left arrow) and a local maximum at $(x,y) = (-10,0)$ (below the right arrow). The curves in the bottom of the figure are isophote contour curves (curves where the function has a constant value). At the saddle the isophote curves form a cross (not visible on the figure) — at the maximum the isophote curve is a single point.*

The family has one catastrophe point. This is illustrated in figure 4.3. In figure 4.3*b* the function for $c = 0$ is shown with the unstable singularity at $(x,y) = (0,0)$. In figure 4.3*a* and 4.3*c* the function is shown for different values of c. These illustrate the effects of pertubations on the structure of the function. When c is negative the structure of the function corresponds to the function in figure 4.2 with a saddle and a local maximum. When c is positive the function has no singularities.

The local structure of the functions in figure 4.3 are different. Obviously the function in 4.3*c* is not structurally equivalent with the others since it has no singularities (this follows from proposition 1). That the local structures of the functions in 4.3*a* and 4.3*b* are different is intuitively clear, since the two singularities in 4.3*a* cannot both correspond to the degenerate singularity in 4.3*b* (the coordinate change function could not be invertible).

4.1.5 The Local Character of a Function

The preceeding definitions allow us to state the following theorems that characterise the local structure of functions into a number of categories.



a *b* *c*

Figure 4.3: *The catastrophe of the family of functions* $f(c;x,y) = x^3 + cx - 24y^2$ *occur at* $(c;x,y) = (0;0,0)$. *This is illustrated in figure b where the catastrophe is located below the arrow at a saddle point. The saddle is unstable and an infinitesimal pertubation will (generically) transform the local structure of the function into either the structure in figure a (with a saddle and a maximum) or the structure in figure c (with no singularities).*

Theorem 1. (The Implicit Function Theorem)
Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$. If $\nabla f \neq \vec{0}$ at $\vec{x}_0 \in \mathbb{R}^n$ then $g \in C^\infty(\mathbb{R}^n, \mathbb{R})$ exists such that

$$f \doteq g \quad \text{where} \quad g(x_1, \ldots, x_n) = x_1$$

In this case the point is called a *regular point* or a *non-singular point*.

In general, the theorems are not proved. The exception is the following proof of the implicit function theorem — the proof is simple and it offers a bit of mathematical intuition.

Proof of theorem 1
The n parameters of f are split into $x \in \mathbb{R}$ and $\vec{y} \in \mathbb{R}^{n-1}$. The aim is to establish a smooth coordinate change function $(v(x,\vec{y}), \vec{w}(x,\vec{y}))$ such that $f(x,\vec{y}) = g(v,\vec{w})$ locally. Define $v(x,\vec{y}) = f(x,\vec{y})$ and let $w(x,\vec{y})$ be an arbitrary smooth function. The coordinate change function $(v,\vec{w})$ is smooth since f is smooth. The Jacobian $Det \begin{pmatrix} v_x & v_y \\ w_x & w_y \end{pmatrix} = f_x w_y - f_y w_x$ is only zero when $f_x = f_y = 0$ since $w(x,\vec{y})$ is an artitrary function (here the notation is quite sloppy since y and w are in fact $(n-1)$-dimensional). Since it is assumed that $\nabla f \neq \vec{0}$, this is not the case.
Now $g(v,\vec{w}) = v$ is locally structurally equivalent with $f(x,y)$ by definition 3 and its remarks.

Theorem 2. (The Morse Lemma)
Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$. If $\nabla f = 0$ and $Det(H(f)) \neq 0$ at $x_0 \in \mathbb{R}^n$ then $g \in C^\infty(\mathbb{R}^n, \mathbb{R})$ exists such that

$$f \doteq g \quad \text{where} \quad g(\vec{x}) = \sum_{i=1}^{n} \lambda_i x_i^2$$

where λ_i are the eigenvalues of the hessian matrix.
In this case the singularity is called a *Morse singularity*.

If a function only consists of regular points and Morse singularities, it is called a *Morse function*.

Name	Nickname	m	l	Catastrophe Germ	Pertubation
A_2	Fold	1	1	x^3	c_1x
$A_{\pm 3}$	Cusp	2	1	$\pm x^4$	$c_1x + c_2x^2$
A_4	Swallowtail	3	1	x^5	$c_1x + c_2x^2 + c_3x^3$
$A_{\pm 5}$	Butterfly	4	1	$\pm x^6$	$c_1x + c_2x^2 + c_3x^3 + c_4x^4$
A_6		5	1	x^7	$c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5$
D_{-4}	Elliptic Umbilic	3	2	$x^2y - y^3$	$c_1x + c_2y + c_3y^2$
D_{+4}	Hyperbolic Umbilic	3	2	$x^2y + y^3$	$c_1x + c_2y + c_3y^2$
D_5	Parabolic Umbilic	4	2	$x^2y + y^4$	$c_1x + c_2y + c_3x^2 + c_4y^2$
D_{-6}		5	2	$x^2y - y^5$	$c_1x + c_2y + c_3x^2 + c_4y^2 + c_5y^3$
D_{+6}		5	2	$x^2y + y^5$	$c_1x + c_2y + c_3x^2 + c_4y^2 + c_5y^3$
$E_{\pm 6}$		5	2	$x^3 \pm y^4$	$c_1x + c_2y + c_3xy + c_4y^2 + c_5xy^2$

Figure 4.4: *Elementary catastrophes of Thom. Thom's theorem (theorems 4 and 5) states the canonical structure of the non-morse part of a function. The number of control parameters s m and the number of degenerate directions l.*

Theorem 3. (The Splitting Lemma)
Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R})$. Assume $\nabla f = 0$ and $Det(H(f)) = 0$ at $\vec{x}_0 \in \mathbb{R}^n$. Let λ_i be the eigenvalues of $H(f)$ be ordered by absolute value with the first l eigenvalues being zero — corresponding to the degree of degeneracy. Then $g \in C^\infty(\mathbb{R}^n, \mathbb{R})$ exists such that

$$f \doteq g \quad \text{where} \quad g(\vec{x}) = g_{nm}(x_1, \ldots, x_l) + \sum_{i=l+1}^{n} \lambda_i x_i^2$$

where g_{nm} is the non-Morse part of the function.

At a degenerate singularity the function has a catastrophe point. The splitting lemma states that the local structure can be described by dividing the structure into a non-Morse part and a Morse function. However, the splitting lemma does not state anything about the non-morse part — except that is has no first and second order terms.

The local structure is explicitly stated in Thom's Theorem which describes the structure in two steps. First the structure of the function is described in a neighbourhood around the catastrophe point in $\mathbb{R}^n \otimes \mathbb{R}$ for fixed control parameters in theorem 4. In the second step (theorem 5) this is expanded into a description of the structure in a neighbourhood around the catastrophe point in $\mathbb{R}^{m+n} \otimes \mathbb{R}$ where the m control parameters are allowed to vary.

Theorem 4. (Thom's Theorem - Part One)
Let $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$. Assume $\nabla f = 0$ and $Det(H(f)) = 0$ at $\vec{x}_0 \in \mathbb{R}^n$ for a given $c_0 \in \mathbb{R}^m$. Let λ_i be the eigenvalues of $H(f)$ ordered by absolute value with the first l eigenvalues being zero — corresponding to the degree of degeneracy. Then $g \in C^\infty(\mathbb{R}^n, \mathbb{R})$ exists such that

$$f \doteq g \quad \text{where} \quad g(\vec{x}) = CG(l) + \sum_{i=l+1}^{n} \lambda_i x_i^2$$

where, if $m \leq 5$, $CG(l)$ is one of the *catastrophe germs* listed in figure 4.4.

Theorem 5. (Thom's Theorem - Part Two)
Let $f \in C^\infty(\mathbb{R}^{m+n}, \mathbb{R})$. Assume $\nabla f = 0$ and $Det(H(f)) = 0$ at $(c_0; x_0) \in \mathbb{R}^{m+n}$. Let λ_i be the eigenvalues of $H(f)$ ordered by absolute value with the first l eigenvalues being zero — corresponding to the degree of degeneracy. Then $g \in C^\infty(\mathbb{R}^n, \mathbb{R})$ exists such that

$$f \doteq g \quad \text{where} \quad g(\vec{c}; \vec{x}) = CG(l) + Pert(l, m) + \sum_{i=l+1}^{n} \lambda_i x_i^2$$

where, if $m \leq 5$, $CG(l)$ is one of the *catastrophe germs* listed in figure 4.4 and $Pert(l, m)$ is the corresponding pertubation.

The theorems that characterise the local structure of a function can be stated a bit more informally:

1. At a regular point the function has the local structure of a (hyper-)plane.
2. At a non-degenerate singularity it has the local structure of a Morse function.
3. At a degenerate singularity the local structure of the function is split into a Morse part and a Non-Morse part. If the dimension of the control parameter space is low ($k \leq 5$) the local structure of the Non-Morse part can be stated in terms of one of the the simple catastrophies of figure 4.4.

4.1.6 Generic Property

The concepts presented so far describe the local structure of a function. The concept of genericity can be used to describe properties for a system.

Definition 6. (Generic)
A property for a system is *generic* if an open, dense subset of the system possesses the property.

In probabilistic terms, a property is generic if it is posessed with probability one.

4.1.7 Transversality

The concept of transversality can be used to determine the dimension of various manifolds. This is useful for determination of the genericity of the properties defined via these manifolds.

Definition 7. (Local Transversality)
Two manifolds $P, Q \in \mathbb{R}^n$ are *locally transverse* (or intersect transversally) at $\vec{x}_0 \in \mathbb{R}^n$ if the direct sum of their tangent spaces at the point has dimension n.

Definition 8. (Global Transversality)
Two manifolds $P, Q \in \mathbb{R}^n$ are *globally transverse* at $\vec{x}_0 \in \mathbb{R}^n$ if they are locally transverse at all points of intersection or they do not intersect at all.

Theorem 6. (The Transversality Lemma)
Let P and Q be two manifolds of dimension p and q in $\mathbb{R}^D$. If P and Q are globally transverse the following statements hold:

a) If $p + q < D$ then P and Q do not intersect at all
b) If $p+q \geq D$ then either P and Q do not intersect or else intersect in a manifold of dimension $p + q - D \geq 0$.

The transversality lemma is particularly useful when looking at manifolds defined by differential expressions. In jet-space it is often trivial to determine transversality between such manifolds since the differential expressions are simply the coordinates.

Of course this only determines the dimension of the intersection of the manifolds in jet-space. That the corresponding manifolds in parameter space have the same dimension can be seen from the following theorem (from [Bruce & Giblin, 1992]):

Theorem 7. (Inverse Mapping from Transverse Intersection)
Let $f \in C^\infty(X, \mathbb{R}^D)$ and Y be a smooth manifold in $\mathbb{R}^D$. If $f(X)$ and Y are transverse then $f^{-1}(Y)$ is a smooth manifold.

The implication that $f^{-1}(Y)$ is a smooth manifold ensures the conservation of the dimensionality.

A simple example of the use of transversality is the following proposition.

Proposition 3. (Singularity Points in Scale-space)
Singularities are generically isolated points in an image and generically curves in scale-space.

Proof of proposition 3
Let $f \in C^\infty(\mathbb{R}^{1+n}, \mathbb{R})$ with $\mathcal{S}_f = \{(\sigma; \vec{x}) \in \mathbb{R}^{1+n} \mid \nabla f(\sigma; \vec{x}) = 0\}$. The manifolds defined by $f_{x_1} = 0, \ldots, f_{x_n} = 0$ are mutually transverse in jet-space. Therefore the co-dimension (defined as $(1 + n)$ minus the dimension) of their intersection is n by the transversality lemma (theorem 6). This determines the dimension of the singularity set to be zero for a given scale and one in scale-space. Therefore the singularities are generically isolated points in an image and generically curves in scale-space.

4.1.8 The Example Revisited

To illustrate the presented results from catastrophe theory, lets look at the family of functions from figure 4.3 once again:

- The Implicit Function Theorem (theorem 1) states that the local structure at all regular points for the example function $f(c; x, y) = x^3 + cx - 24y^2$ is equal to the local structure of its tangent plane.
- The Morse Lemma (theorem 2) states that the local structure of the function at the non-degenerate singularities is that of a Morse function. More explicitly, the eigenvalues of the hessian $H(f) = \begin{pmatrix} 6x & 0 \\ 0 & -48 \end{pmatrix}$ evaluated at the singularities ($3x^2 + c = 0$ and $y = 0$) are $\lambda_1 = \pm 6\sqrt{\frac{-c}{3}}$ and $\lambda_2 = -48$. Therefore the local structure of f at the nondegenerate singularities equals the structure of the functions $\pm 6\sqrt{\frac{-c}{3}}x^2 - 48y^2 \doteq \pm x^2 - y^2$.

- The Splitting Lemma (theorem 3) states that the local structure of the function at the degenerate singularity $(c; x, y) = (0; 0, 0)$ is equal to the structure of the function $f_{nm} - 48y^2$, where f_{nm} is the non-morse part of the function.
- The first part of Thoms Theorem (theorem 4) explicifies the non-Morse part to be a *fold catastrophe* so the local structure of f at the catastrophe point is equal to the local structure of the function $x^3 - 48y^2$ in a neighbourhood around $(0, 0) \in \mathbb{R}^2$ for $c = 0$.
- The second part of Thoms Theorem (theorem 5) extends this to a neighbourhood around $(0; 0, 0) \in \mathbb{R}^3$ to the local structure of the function $x^3 + cx - 48y^2$.

The last result does not really reveal much more about the structure of f than the original expression. This is simply because f is chosen to be the simplest non-trivial function that illustrates interesting aspects of the theorems. Higher order terms could have been added without affecting the structure of the function.

Note that the singularity and catastrophe sets for the specific family of functions have the generic dimensionalities. The singularity set is a curve (more precisely a parabola). Given a specific control parameter value c_0 the singularities are isolated points (defined by $(x, y) = (\pm\sqrt{-c_0/3}, 0)$ for $c \leq 0$). The catastrophe set contains a single isolated point: $(x, y, c) = (0, 0, 0)$

4.2 Catastrophe Theory in Computer Vision

When catastrophe theory is applied to computer vision problems the focus is often not pure mathematics. This has influenced a few specific "additions" to the theory.

4.2.1 An Image as a Function

Singularity theory treats the behaviour of smooth functions. A typical image is not a smooth function. That makes the application of singularity theory somewhat nonsense.

However, the embedding of the image into linear scale-space ensures that the scale-space image is analytical (for scales above zero).

When the use of linear scale-space is not relevant, a simple observation saves the day: Generically, a function is a Morse function. Thereby an arbitrary function can be approximated arbitrarily close by a Morse function. This enables the use of the theory on the approximated function with results that can typically be applied to the original function.

4.2.2 The Image Model

In the analysis of the local structure around catastrophes, an explicit model of the scale-space image is often useful. Instead of using a local Taylor series expansion we use an expansion in heat polymials (as proposed in [Florack, 1993]). This is simply the usual Taylor series expansion in the state variables with terms added in order to ensure that the scale-space function obeys the heat diffusion equation.

The scale-space image L is then modelled at $(t, \vec{x}) = (0, \vec{0})$ by $\tilde{L}$ in the following manner:

$$\begin{aligned}
\tilde{L}(t;\vec{x}) \;=\; & L \\
& + L_i x_i \\
& + \frac{1}{2} L_{ij} x_i x_j + L_{ii} t \\
& + \frac{1}{6} L_{ijk} x_i x_j x_k + L_{iij} x_j t \\
& + \frac{1}{24} L_{ijkl} x_i x_j x_k x_l + \frac{1}{2} L_{iijk} x_j x_k t + \frac{1}{2} L_{iijj} x_j t^2 \\
& + \frac{1}{120} L_{ijklm} x_i x_j x_k x_l x_m + \frac{1}{6} L_{iiklm} x_k x_l x_m t + \frac{1}{2} L_{iijjl} x_l t^2 \\
& + \ldots
\end{aligned} \tag{4.3}$$

$L_{i\ldots m}$ are Gaussian image derivatives evaluated at $(t;\vec{x}) = (0;\vec{0})$ and $i, j, k, l, \ldots$ are indices running over the n spatial variables where the Einstein summation convention applies.

4.2.3 Gauge Coordinate System

Above we have used an arbitrary cartesian coordinate system. Expressions can sometimes be expressed more elegantly using a coordinate system defined by the local geometry. The local gauge coordinate system is defined by the direction of the local gradient.

Given a point in an image I, the gradient direction defines the direction of the w basis vector for the local gauge coordinate system:

$$\vec{w} \;=\; \frac{\nabla I}{\|\nabla I\|} \tag{4.4}$$

For an n-dimensional image the $n-1$ remaining mutually orthonormal basis vectors $\vec{v}_1, \ldots, \vec{v}_{n-1}$ are defined so they span the isophote tangent space at the fiducial point. This implies that their first order partial derivatives vanish: $I_{v_i} = 0$. However, this does not imply $I_{v_i v_j} = 0$.

Note also, that the spatial gauge coordinate system is undefined at $(t_0, \vec{x}_0)$ if $\nabla I = \vec{0}$.

4.3 Generic Events for the Gradient Squared

The purpose of this chapter is to analyse the development of catchment basins over scale. Since the catchment basins can be identified as minima in the dissimilarity measure this analysis is an excellent application for catastrophe theory.

The goal of this section is therefore to investigate the chosen dissimilarity measure — the gradient magnitude — and answer the following questions:

- Do catastrophes occur?
- Which elementary catastrophes do they correspond to?
- Do both creations and annihilations occur?
- Are the catastrophes generic/stable?

Since the conversion is monotone, the structure is equivalent for the gradient magnitude squared. The analysis is simpler for the gradient magnitude squared — therefore this function is analysed instead. At first we will look at singularities for the gradient magnitude squared in general. Later in the analysis we will restrict it to the relevant singularities corresponding to catchment basins — the minima.

In scale-space we have only one control parameter - the scale parameter. At first glance we would therefore only look for *fold*-catastrophes since Thom's theorem (theorem 4) states that this is the only generic catastrophe with only one control parameter. However, the structure induced by the specific function of interest can introduce higher order catastrophes (as we shall see in the following).

The function of interest is the gradient magnitude squared $G : \mathbb{R}^{1+n} \curvearrowright \mathbb{R}$:

$$G(t;\vec{x}) = L_1(t;\vec{x})^2 + \ldots + L_n(t;\vec{x})^2 = L_i^2 \tag{4.5}$$

This is a smooth function for $t > 0$. Note that L_1 is short for L_{x_1} — the partial derivative with respect to the variable x_1.

The spatial gradient and the hessian for expression 4.5 are:

$$\begin{aligned} \nabla G(t;\vec{x}) &= 2L_iL_{ij} \\ H(G,t,\vec{x}) &= 2L_{ik}L_{ij} + 2L_iL_{ijk} \end{aligned}$$

This is written in tensor notation. The gradient is an n-dimensional vector where j is the index. The hessian is a $n \times n$-matrix where j and k are the indices (in both cases the Einsteins summation convention "takes care of" the i indices).

The singularity and the catastrophe sets are by definition:

$$\mathcal{S}_G = \{(t;\vec{x}) \in \mathbb{R}^{1+n} \mid L_iL_{ij} = 0\} \tag{4.6}$$

$$\mathcal{C}_G = \{(t;\vec{x}) \in \mathcal{S}_G \mid Det(2L_{ik}L_{ij} + 2L_iL_{ijk}) = 0\} \tag{4.7}$$

The analysis of these expression will in the following be divided into two cases:

- Points where $L_i = 0$. These points are image singularities and global minima for the gradient magnitude squared.
- Points where $L_i \neq 0$. Here the gauge coordinate system is defined and can be used to simplify expressions.

4.3.1 The Dimensions of $\mathcal{S}_G$ and $\mathcal{C}_G$

First the singularity set is examined more carefully:

Points where $L_i = 0$
In this case $L_i L_{ij} = 0$ is trivially met. Therefore these points belong to $\mathcal{S}_G$.

Points where $L_i \neq 0$
In gauge coordinates $L_i L_{ij}$ can be simplified to $L_w L_{wj}$ since $L_v = 0$ for $v \neq w$ (where v is an arbitrary coordinate direction in the gauge coordinate system). Singularities occur when $L_{wj} = 0$

From this the co-dimension of the singularity set $\mathcal{S}_G$ can easily be determined. In both cases n equations ($L_i = 0$ and $L_{wj} = 0$) must be satisfied. The manifolds defined by the n equations are trivially transverse in jet-space. Therefore the co-dimension of the singularity set is n. Hereby the singularities for the gradient magnitude squared are generically isolated points in an image at a given scale. Note that this does not follow directly from proposition 3 (the gradient magnitude squared could induce a completely different structure on the singularities).

The singularity set $\mathcal{S}_G$ can now be specified more precisely:

$$\mathcal{S}_G = \{(t; \vec{x}) \in \mathbb{R}^{1+n} \mid L_i = 0 \vee L_{wj} = 0\}$$

The dimension of the catastrophe set can be determined by examining the eigenvalues for the hessian $H(G, t, \vec{x}) = L_{ik} L_{ij} + L_i L_{ijk}$:

Points where $L_i = 0$
A catastrophe occurs when the hessian $H(G, t, \vec{x})$ is degenerate. This requires one or more eigenvalues to be zero. For l eigenvalues to be zero $l(l+1)/2$ conditions are required (since $l(l+1)/2$ independent entries in the hessian must vanish).

The co-dimension of the manifold defined by $L_i = 0$ is n. The manifolds defined by conditions on the second order structure are trivially transverse with the manifold defined by $L_i = 0$ in jet-space. The transversality lemma therefore yields the co-dimension of catastrophes with l degenerate eigenvalues to be $n + l(l+1)/2$. Thereby only one degenerate eigenvalue is generic in the $n+1$ dimensional scale-space. These catastrophes will generically occur in a manifold of co-dimension $n+1$ — isolated points in scale-space.

The hessian is $H(G, t, \vec{x}) = L_{ik} L_{ij}$. In order to simplify the expressions, we use the local coordinate system that diagonalise the hessian for the image (not the hessian for the gradient magnitude squared $H(G, t, \vec{x})$). This imply $L_{ij} = 0$ for $i \neq j$. The n eigenvalues are then $L_{11} L_{11}, \ldots, L_{nn} L_{nn}$. If the degenerate direction of the hessian $H(G, t, \vec{x})$ is assumed to be the x_1 direction then the constraint for catastrophes to occur is $L_{11} = 0$.

Points where $L_i \neq 0$

In this case singularities for G occur where $L_{wj} = 0$. In the analysis of the catastrophes we will use the local gauge coordinate system with coordinates $(v_1, \ldots, v_{n-1}, w)$. The v_i directions are fixed so $L_{ij} = 0$ for $i \neq j$ (this is possible since $\vec{w}$ is an eigenvector for L_{ij} because $L_{wj} = 0$). Now the element with indices jk in $H(G, t, \vec{x})$ is

$$H_{(jk)} = \begin{cases} L_{ik}L_{lj} + L_w L_{wjk} & \text{for } i = j = k = l \\ L_i L_{ijk} & \text{otherwise} \end{cases}$$

It is not trivial to see that the determinant of this matrix is non-zero. Furthermore, it is not trivial to see how many constraints are required to force a specific number of eigenvalues to become zero.

I will simply allow myself to claim that, again, $l(l+1)/2$ conditions are required for l eigenvalues to be zero. Further, I also claim that the manifold defined by these conditions are tranverse with the n manifolds defined by $L_{wj} = 0$. These claims are shown for two spatial dimensions in [Olsen, 1996]. More importantly, Ole assures me that he has actually proven it for arbitrary dimensions.

Thereby, analogously with the case above, the catastrophes arising from this case will also generically occur only in a single direction in isolated points in scale-space.

If the single degenerate direction for $H(G, t, \vec{x})$ is assumed to be the x_1 direction then the constraint for catastrophes to occur can be derived anagously to the first case. The local coordinate system that diagonalises the hessian $H(G, t, \vec{x})$ is chosen. Thereby the n eigenvalues become $L_{i1}L_{i1} + L_i L_{i11}, \ldots, L_{in}L_{in} + L_n L_{inn}$. The constraint for a catastrophe to occur in the degenerate direction is then $L_{i1}L_{i1} + L_i L_{i11} = 0$.

In both cases catastrophes are generically isolated points in scale-space. Furthermore, these catastrophes will only have one degenerate direction. If this degenerate direction is assumed to be the x_1 direction, then the catastrophe set is:

$$\mathcal{C}_G = \{(t; \vec{x}) \in \mathbb{R}^{1+n} \mid (L_i = 0 \wedge L_{11} = 0) \vee (L_{wj} = 0 \wedge L_{i1}L_{i1} + L_i L_{i11} = 0)\}$$

It should be stressed that the expression above is a terrible abuse of notation. It implies that the catastrophes have the same degenerate direction in a global coordinate system. This is not the case — the x_1 direction is in the local coordinate systems.

4.3.2 The Local Structure at Catastrophes

The previous analysis shows that catastrophes occur generically at isolated points in scale-space. Furthermore, the catastrophes have only one degenerate direction. The set of conditions that implies a catastrophe can be used to investigate the local structure of the image and the gradient squared at the catastrophe points. In the analysis the image is replaced by the image model from equation 4.3.

Without loss of generality we assume that the catastrophes are located at $(t; \vec{x}) = (0; \vec{0})$ and that the degenerate direction is x_1 — this simplifies the expressions. So the aim is to express

$\tilde{G} = \tilde{L}_i\tilde{L}_i$ as a function of t and x_1. We start with $\tilde{L}_m$ where the image model is truncated to fourth order:

$$\tilde{L}_m(t;\vec{x}) = L_m + L_{im}x_i + \frac{1}{2}L_{ijm}x_ix_j + L_{iim}t + \frac{1}{6}L_{ijkm}x_ix_jx_k + L_{iijm}x_jt + \frac{1}{2}L_{iimm}t^2$$

For $x_2 = \ldots = x_n = 0$ this simplifies to:

$$\tilde{L}_m(t;x_1,0,\ldots,0) = L_m + L_{1m}x_1 + \frac{1}{2}L_{11m}x_1^2 + L_{iim}t + \frac{1}{6}L_{111m}x_1^3 + L_{ii1m}x_1t + \frac{1}{2}L_{iimm}t^2$$

The expression for $\tilde{L}_m$ can be simplified even further by choosing the coordinate system that diagonalises the hessian so that mixed second order derivatives are zero: $L_{ij} = 0$ for $i \neq j$.

$$\tilde{L}_m(t;x_1,0,\ldots,0) = L_m + L_{11}x_1 + \frac{1}{2}L_{11m}x_1^2 + L_{iim}t + \frac{1}{6}L_{111m}x_1^3 + L_{ii1m}x_1t + \frac{1}{2}L_{iimm}t^2 \quad (4.8)$$

We then examine $\tilde{G} = \tilde{L}_i\tilde{L}_i$ in the cases that lead to catastrophes:

Points where $L_i = 0$

The catastrophes occur when the second order image derivative in the degenerate direction is zero. Since we assume the degenerate direction to be x_1 the constraint is $L_{11} = 0$, where L_{11} is short for $L_{x_1x_1}$.

The first and second order structure vanish; this implies that the local structure of the image at these catastrophe points for G is that of a *fold* catastrophe.

The local structure of G can be derived from equation 4.8:

$$\begin{aligned}\tilde{G}(t;x_1,0,\ldots,0) &= \tilde{L}_i\tilde{L}_i \\ &= L_{11j}L_{iij}tx_1^2 + (L_{11j}L_{ii1j} + \frac{1}{3}L_{111j}L_{iij})tx_1^3 \\ &\quad + (\frac{1}{4}L_{11j}L_{11j} + \frac{1}{3}L_{111j}L_{ii1j}t)x_1^4 + O(x_1^5) + O(t^2)\end{aligned}$$

For $t = 0$ this expression reveals the catastrophe germ $\frac{1}{4}L_{11j}L_{11j}x_1^4$. This is a cusp catastrophe. The sign in front of the tx_1^2 term determines whether it is an annihilation or a creation. In 1D the sign is the sign of L_{111}^2 — this positive sign signifies an annihilation. In higher dimensions creations are possible.

Points where $L_i \neq 0$

In this case singularities for G occur where $L_{wj} = 0$. The condition for a catastrophe to occur in the x_1 direction is $L_{i1}L_{i1} + L_iL_{i11} = 0$.

The local structure of the image itself at the catastrophe points is simply the structure implied by the implicit function theorem (theorem 1).

The local structure of G is again derived from equation 4.8:

$$\begin{aligned}\tilde{G}(t;x_1,0,\ldots,0) &= \tilde{L}_i\tilde{L}_i \\ &= L_j^2 + 2L_{iij}L_jt + 2(L_{11}L_{iij} + L_{ii1j}L_j)tx \\ &\quad +(2L_{11}L_{ii1j} + L_{11j}L_{iij})tx^2 \\ &\quad +(L_{11}L_{11j} + \frac{1}{3}L_{111j}L_j + (L_{11j}L_{ii1j} + \frac{1}{3}L_{111j}L_{iij})t)x^3 \\ &\quad +O(t^2) + O(x^4)\end{aligned}$$

For $t = 0$ the catastrophe germ $(L_{11}L_{11j} + \frac{1}{3}L_{111j}L_j)x^3$ appears. This is a fold catastrophe. The sign in front of the tx and the x^3 terms determines whether the catastrophe is an annihilation or a creation (if the signs differ, it is a creation). These signs can be both equal and opposite for any image dimension.

4.3.3 Generic and Stable Catastrophes

It has been shown that catastrophes of both the fold and the cusp type are generically isolated points in scale-space.

Does this imply that catastrophes are generically present in a scale-space image?

No – this is not the case.

This is equivalent to singularities in a one-dimensional function. These are generically present in isolated points. This does not imply that they are generically present. In order to see this, think of the strictly monotone functions (function without singularities). If a strictly monotone function is perturbed slightly it is still a strictly monotone function (due to proposition 1) — therefore they form an open set. Since the set of functions without singularities form an open set the complementary set cannot be dense. Therefore, it is not generic for a function to belong to the complementary set. Thus, it is not generic for a function to have a singularity[1].

Where does this leave creations and annihilations?

Catastrophes can be divided by specific inequalities that determine whether the catastrophe is an annihilation or a creation (as done in section 4.3.2). Unfortunately, it is a theoretically unsolved problem how to treat inequalities that "divide" the jet-space (as opposed to equalities that correspond to manifolds in jet-space).

For this reason it is not theoretically sound to propose statements about genericity of the annihilation or creation events individually. However, it is possible to prove if they are stable. In order to prove that the the presence of annihilations (respectively creations) is stable, we only need to find an open subset of the catastrophe set which contains annihilations. The is simply done by finding an instance (expressed in terms of a set of coefficients for the local image model from section 4.3) that produces an annihilation catastrophe. Since the inequalities which divide annihilations from creations define open sets, the presence of a single annihilation/creation will imply the presence of an open set of them. This implies that the presence is stable.

The conclusion is that catastrophes are generically located in isolated points and that the presence of the various events in an image is simply stable.

4.3.4 The Generic Set

The term generic has an unambiguous definition. Nevertheless, it is often used with a slightly different meaning in computer vision. The derived "generic events" for the gradient squared are actually not generic — they are only stable.

[1] This simple line of reasoning is due to Ole Fogh Olsen.

The use of "generic" really implies that the event belongs to a *generic set*. A generic set is a list of properties, that as a whole is a full description of the system. As an example, a generic set for a function can be described: "The points are generically either regular or singular points". As shown in the previous section, the presence of a singular point in a function is not generic. They are, however, stable. This means that they are relevant in a complete description of a function. If only regular points were considered the description would not cover a dense set of the points.

The properties of the points can be described in a number of ways. The properties included in a generic set are chosen so they meet the following requirements:

- The generic set must be minimal.
- The properties must be as low-order descriptions as possible.

That the generic set must be minimal implies that only stable properties are included. Low-order descriptions of the properties are preferred since these descriptors are simplest.

The generic set for the gradient magnitude squared is as follows. For a scale-space image, the points are generically one of the following:

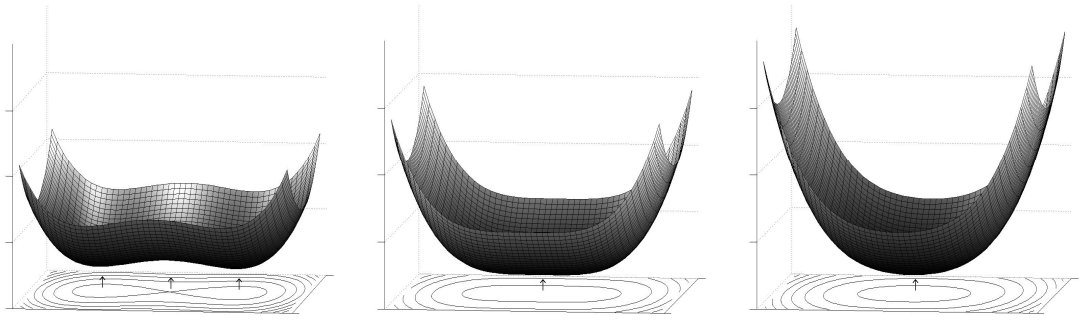
- Regular points
- Singular points
- Points with a Fold Creation
- Points with a Fold Annihilation
- Points with a Cusp Creation
- Points with a Cusp Annihilation

4.3.5 Generic Events for the Catchment Basins

The treatment of the generic events for the gradient magnitude squared has dealt with singularities in general, even though only the minima correspond to catchment basins from the watershed segmentation.

In order to see the effect of the catastrophes on the catchment basins, it is necessary to examine the relevant catastrophes a bit further.

The *fold* annihilation catastrophe transforms a maximum and a minimum into a saddle point at the catastrophe point. After the catastrophe (meaning at increased control parameter values) no singularities are present. The *fold* creation catastrophe has the opposite effect. These events are illustrated in figure 4.3. The corresponding events for the catchment basins are denoted *annihilation* and *creation*.



a *b* *c*

Figure 4.5: *Illustration of a cusp catastrophe. The family of functions* $f(c;x,y) = x^4 + cx^2 + 300y^2$. *The catastrophe occurs at* $(c;x,y) = (0;0,0)$. *This is illustrated in figure b where the catastrophe is located above the arrow at a minimum. The minimum is unstable and an infinitesimal pertubation will transform the local structure of the function into either the structure in figure a (with a saddle and two minima) or the structure in figure c (with a stable minimum).*

Catastrophe for gradient magnitude squared	Event for catchment basin
Fold Creation	Creation
Fold Annihilation	Annihilation
Cusp Creation	Split
Cusp Annihilation	Merge

Figure 4.6: *Stable events for the catchments basins.*

The *cusp* annihilation catastrophe transforms a minimum, a maximum, and a minimum into a double degenerate minimum at the catastrophe point. After the catastrophe one minimum is present. The *cusp* creation event has the reverse effect. These events are illustrated in figure 4.5. The corresponding events for the catchment basins are denoted *merge* and *split*.

The minima mentioned here are only minima in the degenerate direction. In order to correspond to catchment basins they have to be minima with respect to all spatial directions. This adds constraints in terms of inequalities on the second order derivatives in all spatial directions. That the minima are stable is analogous with the argument regarding the stability of creations and annihilations in the previous section. These stable events are summarized in figure 4.6.

4.4 Catastrophe Theory under Linear Diffusion

The transversality lemma is very useful for determining the dimensions of manifolds. This is particularly applicable since it is quite simple to determine transversality of manifolds defined by differential expressions in jet-space — the coordinate axes in jet-space are the partial derivatives.

However, in the analysis of scale-space images there is an important aber-dabei. The linear diffusion ensures that the coordinate axes are no longer independent. A simple example will

illustrate this. The manifolds defined by $L_x = 0$ and $L_{xx} = 0$ are trivially transverse. But the manifolds defined by $L_{xx} = 0$ and $L_t = 0$ are not transverse even though they are defined by partial derivatives with respect to different coordinates. In fact they are identical.

The example is somewhat artificial since jet-space is only defined on the spatial coordinates. But the implicit dependencies defined by the diffusion equation could induce other specific structures than those covered by Thom's theorem (theorem 4 and 5).

That the approach in this chapter is actually correct is therefore not obvious. For further studies along this line read [Damon, 1995] and [Damon, 1997] until your head turns blue.

4.5 More on Catastrophes

For early work on critical points the reader is referred to [Morse, 1931].

A more detailed treatment of modern catastrophe theory can be found in [Gilmore, 1993] and [Bruce & Giblin, 1992].

The original treatments of the generic events for the gradient magnitude squared are [Olsen, 1996], [Olsen & Nielsen, 1997], and [Olsen, 1997].

4.6 Finally!

The purpose of this chapter is twofold:

- Introduce catastrophe theory.
- Apply catastrophe theory to analyse the behaviour of the watershed catchment basins over scale.

The introduction to catastrophe theory is somewhat more rigorous than necessary. This is due to my personal desire to present this subject thorougly. You simply don't know whether you actually understand catastrophe theory until you write it down in sufficient detail.

The application of the theory with respect to the watershed catchment basins is more directly relevant for this thesis. The findings were:

- Both creations and annihilations of catchment basins are to be expected as the scale increases.
- The analysis revealed the generic structure of the dissimilarity function where the creations and annihilations occur.

These conclusions should be useful in the following chapter.

The end to all our exploring will be to arrive at where we started — and to know the place for the first time.
T.S. Elliot

Chapter 5

Multi-scale Watershed Segmentation

We have seen how watershed segmentation can partition an image into regions. The chosen dissimilarity measure for this process requires calculation of the gradient. This implies the need for a well-posed differentiation of an image. The theory of linear Gaussian scale-space provides this differentiation.

This means that the watersheds are only well-defined in the context of a given scale. This might seem like a undesired limitation. However, any feature calculation on an image has an implicit notion of range. Scale-space makes this notion explicit in an intuitively and theoretically "nice" framework.

It would appear that the choice of the correct scale is all there is left. This is however not feasible. A specific fixed scale would limit a segmentation tool to be applicable to a very limited group of images. It is therefore necessary to look at the entire range of scales from the inner to the outer scale of an image. Furthermore, Koenderink suggests looking not only at all scales but at the development of features across scale — the deep structure.

Singularity theory allows us to analyse this development and reveal the events that we must anticipate for the catchment basins as scale increases. The stable events are *annihilation, creation, merge*, and *split*.

After this recap, we are now ready to look at how this knowledge can be applied in a segmentation application.

5.1 Discrete Linking

In the previous chapter we derived the stable events for the gradient magnitude squared and determined the corresponding events for the catchment basins.

An implicit requirement in these derivations was that the scale-space image was continuous in both spatial and scale dimensions. Discrete implementation of the Gaussian filter is beyond the scope of this thesis. However, the discrete sampling of scales requires a few considerations. In determining how closely the scale levels should be sampled, there are three main considerations:

- The scales needed to provide the regions relevant for the segmentation task should be available.
- The discrete sampling should be close enough to facilitate robust linking of catchment basins across scale.
- The number of scale levels should not be excessive and thereby require too much computation.

The evolution of the image across scales implies that the discrete scales should be distributed logarithmically (with the same relative increment between each scale step). This achieves the approximate simplification between each scale step. This thesis does not cover these topics theoretically any further. Practical experiments show that the applicability of an application is not very sensitive to these choices. A simple example of this is shown in section 9.3.

However, the algorithm for performing the linking of catchment basins across scale is worth looking into. In [Gauch & Pizer, 1993] the minima for the dissimilarity measure were tracked across scale. The position of the minimum at one scale was used as starting point for a gradient descent search for the minimum at the next scale. In [Olsen, 1996] an alternative method is proposed. Here a catchment basin at one scale is matched with the basin on the following scale with maximum overlap. This region based method is simpler and more robust than the point based method.

The linking method simplifies the task of establishing corresponding catchment basins across scale a great deal. The events following from the catastrophes can all be treated in the same framework. Figure 5.1 illustrates this. Notice how the *Annihilation* and *Merge* events are indistinguishable (and the *Creation* and *Split* likewise).

The linking results in a tree (or a forest if more than one region is present at the highest scale). Informally, most leaves will be a the lowest scale — the creation and split events do however create new leaves higher up in the tree.

It is worth noticing that the simple linking algorithm does not rely on our knowledge of the stable events for the catchment basins. This will prove very useful in the following chapters.

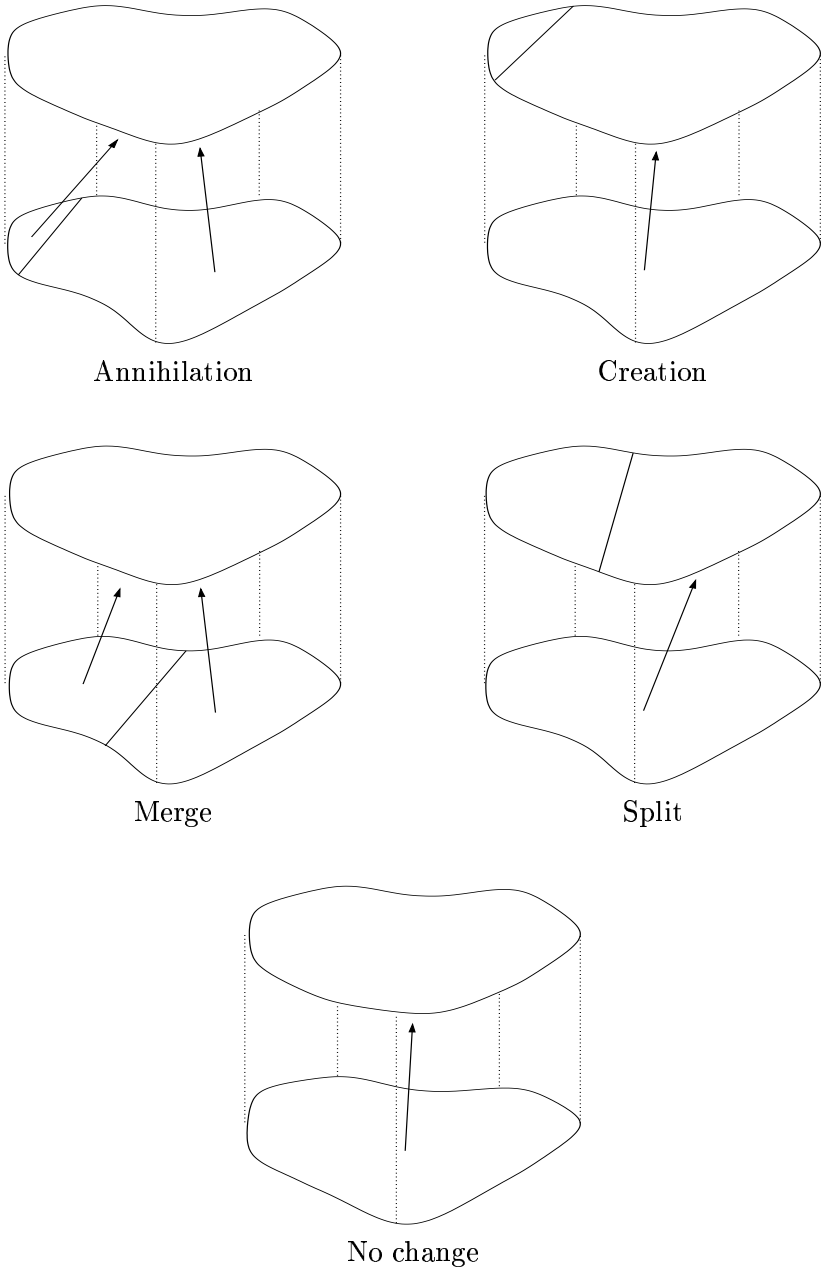


Figure 5.1: *Linking of catchment basins across scale. The figure shows the five categories of events for the catchment basins: Annihilation, Creation, Merge, Split, and No change. Each category is illustrated with a figure showing the relevant catchment basins at two adjacent scales. From the low scale to the high scale each region is linked to the region at high scale with maximum overlap.*

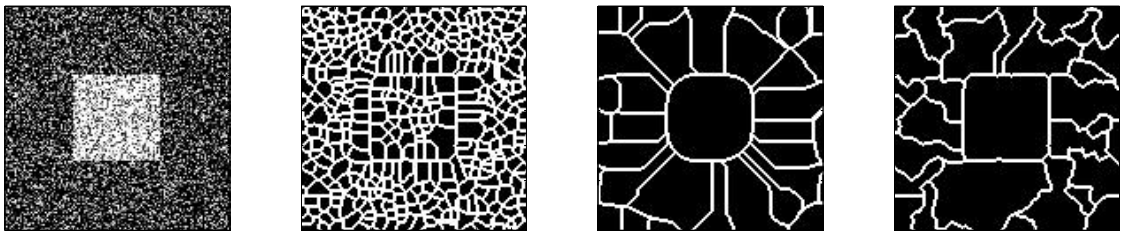
5.2 Localisation Scale & Detection Scale

As mentioned in the opening chapters on watershed segmentation there are two conflicting considerations when choosing an appropriate scale for segmentation of a specific object:

- A low scale is desirable in order to get accurate region border corresponding with the geometry of the image.
- A higher scale is necessary to avoid excessive over-segmentation caused by noise and spurious details in the image.

The linking of the catchment basins allows the best of both worlds. Regions at high scale can be linked down to a lower scale. Thereby it is possible to display a region at high scale using the precise borders from the corresponding regions at low scale. This is illustrated in figure 5.2.

The example shows how the desired region can be detected at high scale — the *detection scale* — using the precision from the low scale — the *localisation scale* — to place to region boundaries. The resulting segmentation is quite nice.



Original image | Watersheds at localisation scale | Watersheds at detection scale | Watersheds at detection scale linked to localisation scale

Figure 5.2: *Linking of watersheds from detection scale to localisation scale. The scale-space is generated in 10 scale levels from localisation scale 3 to detection scale 10. The original image is a square where 50 % of the pixels have been replaced by random noise.*

Notice that the *localised watersheds* (watersheds at detection scale linked to localisation scale) are a subset of the watersheds at the localisation scale. Even though creation or split events can create new catchment basins they will not be present among the localised watersheds. A catchment basin which results from a creation has no representation beneath the creation scale.

A key observation is that a very low localisation scale is not necessarily optimal. In noisy images it is advantageous to raise the localisation scale in order to suppress noise — you dont want to look for more precision than the image provides. However, a relatively high localisation scale makes the minimal segments accordingly large. This has inspired the addition of an extra "artificial" bottom level, where the segments are the individual pixels (or voxels in 3D). Thereby the localisation scale can be large enough to suppress noise without the risk of preventing the user from reaching the desired segmentation.

5.3 Dimensionality of the Approach: 2D vs. 3D

The theory covered so far have all been general for functions with an arbitrary spatial dimension. However, all the illustrations that will be shown are in 2D — illustrations of higher dimensional images simply don't agree well with paper.

5.4 A Development Interface

A multi-scale algorithm takes care of some of the problems inherent in the general segmentation task. However, using a method inspired by the previous chapters the user is still required to supply knowledge of the specific task that allows:

- Selection of the relevant regions that correspond to the desired objects in the images.
- Selection of a localisation scale that diminishes the effect of noise in the image.
- Selection of a detection scale that is suitable for the desired objects.

If some a priori knowledge of the images is available the localisation scale can be provided by the program. However, for a general system to function, the other two must be supplied through user interaction.

The purpose of this thesis is definitely not to create an optimal user interface for a segmentation program. The presented interface is designed to present enough detail to enable the user (in this case me) to observe how different algorithms and specific choices of parameters affect the segmentation task. Therefore the term *development interface* is more appropriate than the traditional term user interface.

The interface has been developed in Matlab. Some of the code is modifications of code by Ole Fogh Olsen (used in his masters thesis [Olsen, 1996]) and Jon Sporring (a Fourier domain implementation of linear Gaussian scale-space).

In this interface the example from figure 5.2 is as illustrated in figure 5.3. This is obviously a very simple example. In general the localisation and detection scales are not necessarily the lowest and the highest scales, repectively. Furthermore, segmentation of the objects usually requires selecting more than one region. A typical segmentation process consists of selection of a few regions at coarse scale followed by refinement by selection or deselection of a few regions at a lower scale.

As mentioned, the interface is not suited for a "real" user. However, the principles are well suited for a segmentation tool. The selection of building blocks directly corresponding to geometrical objects is very intuitive. The development of an efficient segmentation tool using these principles is a goal for the research project *Computing Natural Shape* (see [Johansen et al., 1999]). Note that the "artificial" bottom level pixel segments are not available in the development interface. However, this is the case in the "real" user interface in the Natural Shape implementation. The Natural Shape implementation has been tested for clinical use with promising results [Dam et al., 2000].

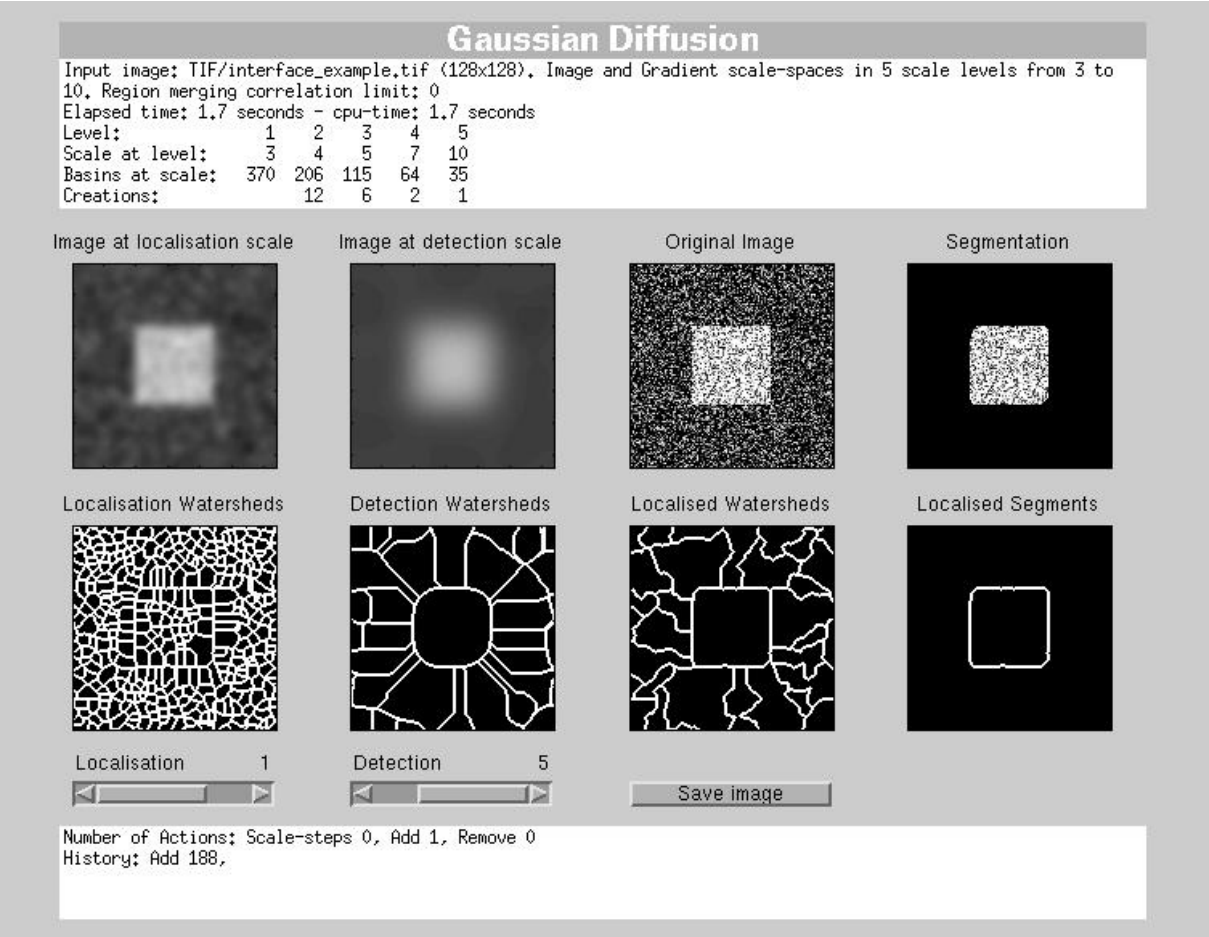


Figure 5.3: *The example from figure 5.2 presented in the development interface.*
The top window contains information on the generated watershed scale-space. Of most interest is the number of regions at each scale level.
The center 8 images illustrate the segmentation process. The four images in the left half show the watersheds at the chosen localisation and detection scales. These scales are selected using the sliders. The effect of the choices are also displayed in the most important image — the one showing the localised watersheds. To the right is displayed the chosen regions. Localised catctment basins are selected (or deselected) by clicking in any of the images.
The bottom window displays the action history revealing the number and type of actions required to perform the segmentation task.
A typical segmentation process consists of selection of a few regions at coarse scale followed by refinement by selection or deselection of a few regions at a lower scale.

5.5 A Segmentation Example

The understanding of the segmentation process is vital for the appreciation of the remaining chapters of this thesis. Thus, an extra example is provided in order to make sure that the concept of detection scale and the notion of building blocks is absolutely clear.

In figure 5.4 the segmentation of the volcano ring from figure 3.2 is performed in three steps:

1. The circumsphere of the volcano including the center is selected at coarse detection scale.
2. The detection scale is lowered to display building blocks corresponding in size to the volcano center.
3. The segment corresponding to the volcano center is deselected and thereby the middle of the segmentation is removed. The resulting area corresponds well with the volcano ring.

5.6 More on Multi-scale Watershed Segmentation

Multi-scale watershed segmentation is by no means invented in this thesis. Early work using some of the theory described in these first chapters can be studied through [Lifshitz & Pizer, 1990].

However, the principle of providing the user with geometrical building blocks in [Olsen, 1996] is a significant improvement. This approach has later been adopted in [Gauch, 1999].

Parallel work using some of the same principles in another theoretical framework — the *Hyperstack* — can also been studied in [Vincken, 1995] and [Koster, 1995].

5.7 The Purpose of this Thesis

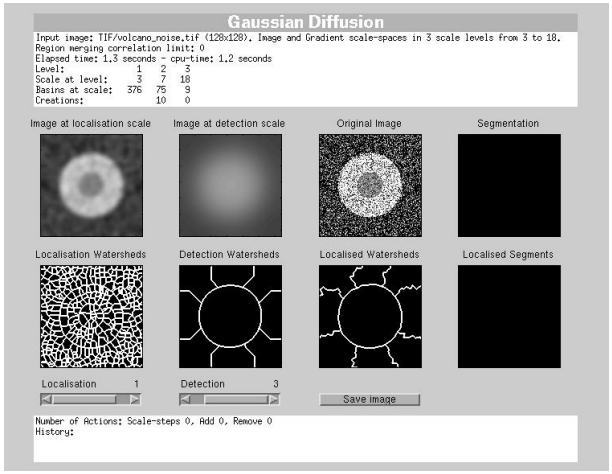
The sound theory and the nice examples provided in the preceeding chapters indicate that multi-scale watershed segmentation is an attractive method. However, well-founded theory and colourful illustrations are not enough to ensure that a method is applicable in real world applications.

The applicability of the method relies mainly on two points:

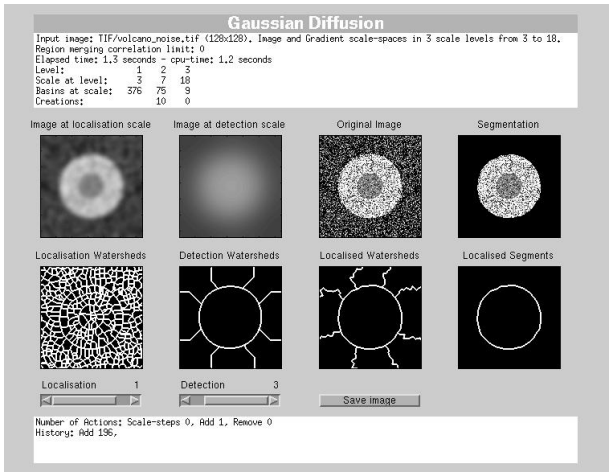
- Do the borders of the building blocks coincide with the borders of the desired objects? Since the size of the building blocks in the limit is reduced to pixel size the question is rather: Are the building blocks that correspond well with the desired objects sufficiently large to enable an easy segmentation?
- Since the method requires the user to select and deselect the building blocks the effect of the questions above is: How much user interaction is required to perform a segmentation?

Part Two of this thesis contains the design of an evaluation method that answers these questions.

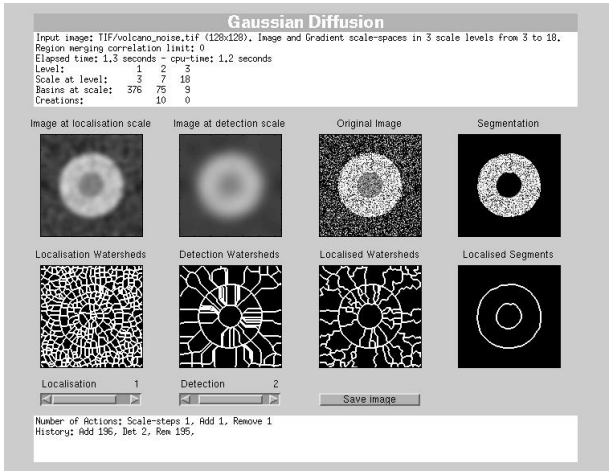
Once a proper evaluation is established it is possible to measure the effect of changes to the segmentation method. The goal of Part Three is to design alternative diffusion schemes for the underlying scale-space. In Part Four the evaluation method is used to determine whether these new diffusion schemes improve the segmentation method.



a



b



c

Figure 5.4: *Segmentation of the volcano from the noisy image of the volcano (from figure 3.2) is performed by a selection and a deselection:*
a) The generated gradient watershed scale space without any selections.
b) The outher rim of the volcano is added by selecting a segment at coarse detection scale.
c) The center of the volcano is removed by deselecting a smaller segment at finer detection scale.

Part II

Evaluation of Segmentation Methods

Some men see things as they are and ask why?
Others dream things that never were and ask why not?
George Bernard Shaw

Chapter 6

The Evaluation Methodology

"The resulting segmentation is quite nice."

Possibly, this conclusion from section 5.2 (page 40) would be more convincing with a few praises: "The resulting segmentation is certainly excellent."

Statements like the above mentioned are common in the concluding sections of papers concerning segmentation methods. The evaluations are often unfounded and not as objective as desired. A few examples from work closely related to this thesis are:

- "The inaccurate segmentations are frequently close enough to the desired result so that a simple interactive postprocessing step might produce a correct segmentation." [Lifshitz & Pizer, 1990]
- "Segmentations correspond well with the visual perceived object boundaries." [Olsen, 1996]
- "Visually sensible objects in an image can be easily constructed." [Gauch, 1999]

Admittedly, it is not fair to exhibit these quotes without the context. The authors of each of the papers mentioned are very well aware that further work is necessary in order to evaluate their methods properly. However, the quotes exemplify that proper evaluation is most often beyond the scope of the work.

6.1 Evaluation Objective

The evaluation should establish whether the segmentation program works. This decision needs to be task driven. For a method that is part of a medical diagnosis system, the segmentation works if it allows the patient to get the proper treatment. Therefore, an optimal evaluation of the segmentation method cannot be based solely on the medical scan and the result of the segmentation. The performances of the methods are affected by factors far beyond the geometry of the input image.

The evaluation for a general-purpose segmentation method cannot be as specific as an optimal evaluation. Instead of focusing on the end-result of the entire system, the evaluation must rely on the geometry of the data.

A first step towards an actual evaluation is the approach in [Lopez et al., 1999]. A number of objective goals is to be met by each method. However, the actual evaluation is performed by visual inspection by the authors. There a two major problems with this: The results are not quantitative and the evaluation is far from objective.

This inspires the following requirements for an evaluation method:

General
: It must not be restricted to a specific segmentation method, a particular segmentation task, or a specific dimensionality (2D or 3D).

Objective
: No human factor must be involved — this can possibly be relaxed to allow test persons.

Quantitative
: The result of the evaluation must be quantitative in nature — not qualitative.

Specific
: The evaluation must be performed with data sets that resemble the desired segmentation task.

The final item emphasises that no segmentation method will work equally well for different segmentation tasks. Therefore, an evaluation will only predict the behaviour of the method on data resembling the test data.

The bottom level pixel-sized building blocks of the segmentation method of this thesis allow the user to reach an arbitrary segmentation (as explained in section 5.2). Thereby the method can always produce the desired result. The overall goal of the evaluation method is therefore to determine: *How much user effort is required to reach a correct segmentation?*

This leaves two open problems:

- What is "a correct segmentation"?
- How is "user effort" measured?

6.1.1 Ground Truth

The notion of "correct segmentation" only makes sense in the presence of a specific segmentation task. There are then a number of ways to obtain a correct segmentation:

- Manual segmentation by experts of the relevant area of expertise on real data.
- *Cryosectioning* of the psychical objects after image acquisition (in the field of medical imaging).
- Images obtained from artificial, known scenes with *Phantom* objects.
- Simulation of the objects and acquisition method.

Real Data

The use of real data obviously gives the most realistic data set. The correct segmentation is here obtained through manual segmentation performed by segmentation task experts. Since a person is involved, this conflicts with the objectivity requirement. Different experts will provide slightly different — but still equally "correct" — segmentations. This problem can possibly be overcome by letting different experts perform segmentation on the same data set. This would even allow an estimate of the imprecision of the segmentations.

In medical imaging in particular, there is a possibly critical problem with the images obtained from scannings. Most medical scanning techniques include non-trivial reconstruction of the volume data from the actual measurements. These reconstructions have inherent artifacts that introduce irregularities (an example is the intensity non-uniformity of MR scannings). This means that the data available for the segmentation method is possibly flawed. Even a perfect segmentation of these data will therefore not obtain the correct location of the actual anatomical parts.

Since no impeccable method for obtaining one true segmentation from live data exists, a manual segmentation performed by an expert is the best possible solution. In medical imaging this requires meticulous inspection of scanning data by trained radiologists and/or clinicians.

Cryosectioning is performed in three steps. First the body of interest is frozen. Then the desired scannings are acquired. Finally the frozen body is sliced into thin slices. This gives "direct access" to the physical objects that are depicted through scanning. Measurements performed on the cryosections require an expert and therefore also suffer from the drawbacks concerning subjectivity mentioned above. Cryosectioning data are extremely scarce for obvious reasons (at least in the area of medical scannings on human beings).

An important use of cryosectioning data (apart from providing ground truth segmentation data) is evaluation of the other acquisition techniques in medical imaging. The physically measured data from a cryosection can be used to calibrate the reconstructions irregularities previously mentioned.

Simulated Data

Phantoms are physical models of the objects of interest. These models are then subjected to the actual image acquisition process. The segmentation data provided by the phantoms are the result of measurements on the physical phantom. Like the cryosection data, phantoms can be used to evaluate and calibrate the image acquisition technique.

With simulated data no physical objects or actual acquisitioning process are involved. The objects and the acquisition are modelled including noise and other artifacts. The construction of the base data delivers the actual segmentation data directly.

Both phantom and simulated data suffer from one possibly major drawback. In order to resemble real data well, the object model must be quite excellent. This requires comprehensive knowledge about the eigenshapes and variations of the real world data — and about which of these shapes and variations that are important and should be included in the simulation. This knowledge is typically not available for the objects of interest of a specific segmentation task.

Worth mentioning is that comprehensive knowledge about the shape and variation of the objects of interest would allow further specialisation of the segmentation method. In the realm of the segmentation method described in Part One, the dissimilarity measure could be adapted to the data set. The shape and variation knowledge would allow construction of a model of the profile of the edges in the images — this allows an optimal edge detection (along the lines of [Lorenz et al., 1997a] and [Lorenz et al., 1997b]) — thereby a dissimilarity measure for these specific type of images can be constructed. More general, the shape and variation knowledge could allow the construction of a specialised segmentation method that would perform optimal segmentation in a Bayesian sense.

Another source of imprecision for purely simulated data is that the image acquisition process must be modeled. This introduces possible variation from the real data as well.

No Ground Truth Data?

An evaluation of a segmentation method need not be based on ground truth data. The time-consuming task of producing manual ground truth segmentations can be replaced by manual inspection of the resulting segmentation by an expert. However, this is more appropriate for the development phase of a project. As an evaluation method it suffers from lack of objectivity and even more from the lack of a quantitative assessment of the segmentation.

As a contrast, several sources of ground truth data could be available. This would allow a very comprehensive evaluation. The combination of different ground truth modalities could be used for evaluating the various steps in the process: the acquisitioning, the manual segmentation, and the segmentation method. For this thesis, no such abundance of ground truth data is available.

The Ground Truth Data

Segmentation data obtained through any of the above mentioned methods are called *ground truth* or *golden standard* data. It would appear that no method for acquiring ground truth data is perfect. Simulations and phantoms provide precise, objective data sets. However, the drawback is that they are not real data. Therefore, an evaluation performed with these data is not guaranteed to predict the performance of a segmentation method on real data.

The ground truth data that originates from real data obviously resembles real data well. Again, there are drawbacks however. The real problem is lack of expert manual segmentations available for evaluation. In medical imaging the data sets obtained from live patients are confidential and therefore not easily available. Furthermore, the expert radiologists have limited time for participation in research.

The ground truth data sets used in this thesis are described in chapter 8.

6.2 The Basic Evaluation Method

Once a ground truth segmentation have been obtained, the evaluation method shall measure the user effort required to reach the ground truth.

6.2.1 Processing Cost

A straightforward measure of the user effort required to reach the ground truth is the time spent. This has two obvious disadvantages:

- The test situation and the test examples must be carefully selected in order to avoid bias. For instance, if the same example was used in evaluation of two different segmentation methods, the test user would "know" the data second time, and therefore the second method would seem better.
- It requires a large number of realistic test persons in order to be reasonable objective.

However, the fundamental problem with this approach is that is will provide a usability study of the program user interface rather than an evaluation of the underlying segmentation methodology. A measure of the number of "canonical actions", that the user is needed to perform, is therefore preferable. These basic actions should be carefully chosen. An improvement of the segmentation method (resulting in fewer basic actions) should automatically be reflected in a decrease in the actual user effort.

The example from section 5.5 (page 42) reveals that there are two obvious basic actions: *Selection* and *deselection*. An improvement of the segmentation method will result in building blocks better suited for construction of the desired segment. This will be reflected in fewer user actions needed. The processing cost can therefore be defined accordingly simple.

Definition 9. (Processing Cost)
The processing cost PC for a segmentation made by S selection actions and D deselection actions is defined:

$$PC \equiv S + D$$

Actually, there is another possible basic action: The scale shift. This is not just a "user interface artifact" — the basic segmentation method requires the user to select the appropriate scales for selecting and deselecting segments. These scale shifts have been left out of the processing cost for three reasons:

- The processing cost is strikingly intuitive without the scale shift.
- It is not clear how a scale shift should be weighted against selection/deselection in the processing cost.
- The user effort required to perform scale shifts depends more on the user interface than the segmentation method.

None of the evaluation algorithms in the following chapter will become significantly more complicated by including the scale shift in the processing cost. It simply doesn't belong there. The quality of the segmentation method is measured by the number of building blocks that need to be handled in order to build the desired object. This is simple, intuitive, objective, general, and quantitative.

Another possible alteration is more gentle weighting of the cost of actions at the highest resolution level — the pixel level. Possibly, a brush-like selection interface would make the selection of neighbouring pixel faster. Again, I will refrain from any such thing. The evaluation is intended to evaluate the segmentation method — not the user interface.

6.2.2 Method 1: Minimal Processing Cost

Evaluation method 1 measures the quality of the building blocks produced by the segmentation method as the minimal processing cost required to reach the ground truth segmentation.

An efficient algorithm for the method is included in section 7.2.

6.3 Advanced Evaluation Methods

The minimal processing cost needed to reach ground truth objects would be sufficient to define a proper evaluation.

However, as mentioned above, the ground truth segmentations are not perfect. In addition to that, some segmentation task will not require perfect segmentations. This inspires a relaxed notion of "correctness". Instead of measuring the processing cost required for perfection it is

relevant to measure the effort required to reach a "sufficiently correct" segmentation. Since the threshold for being "sufficiently correct" will depend on the segmentation task, the method can measure the processing cost as a function of the desired quality of the segmentation.

In order to do this, a measure of quality is needed. As mentioned above, the applicability of a quality measure depends on the segmentation task. The aim is therefore to choose a simple and general measure.

6.3.1 A Measure of Correctness

Suitable measures of the error of a segmentation can be defined in various ways. As mentioned earlier, in order to apply to a general method the measures need to be geometrical.

An obvious error measure is the number of pixels (or voxels) that differ between the segmentation and the ground truth. This can possibly be refined slightly by distinguishing between falsely selected pixels and unselected true pixels.

A simple measure like this ignores that a segmentation is better understood as a dividing contour rather than just two sets of pixels (inside and outside). A simple measure taking this into account is the squared distance between the segmentation contour and the ground truth contour (integrated across the ground truth contour). This is used convincingly in [Bouma et al., 1996].

This immidiately inspires even more sophisticated measures that explore the shape of the contours. For a particular segmentation task, certain shapes can signify drastically different objects. The topology of the contour is also interesting. For instance, in segmentation of blood vessels, the quality of a segmentation should plunge enormously if a vessel is uncorrectly split into two unconnected vessels.

A measure, that incorporates shape and topology of the contour, could be defined on an embedding of the contour in a *Level Set* formulation [Sethian, 1996], where the contour is represented as a particular level. The distance measure between the segmentation and the ground truth would then be defined in the hyper-space containing the embedding. And the definition of the measure should be specific to the particular segmentation task.

Error measures can obviously be defined quite complicated. For a specific segmentation task, a sophisticated measure could be appropriate. However, this measure would typically be irrelevant for other segmentation tasks. For a general method a simple, general measure is the best choice. Most segmentation tasks only require segmentation into "object" and "background". A simple and general measure is therefore:

Definition 10. (Correctness of Segmentation)
Let O be on object and S a segment both defined as sets of pixels. The quality Q of the segment is defined:

$$Q = 1 - \frac{|O \cap \overline{S}| \; + \; |\overline{O} \cap S|}{|O|}$$

Here $|\cdot|$ is the number of elements in a set and $\overline{\cdot}$ is the complementary set.

In plain words, the error is simply the number of falsely segmented pixels (either "inside" and not selected, or "outside" and selected) relative to the number of pixels in the desired object. The quality is defined as one minus the error.

This measure is only well-defined for segmentation tasks where only one object is to be segmented. This allows the simple notions of "inside" and "outside". In [Koster, 1995] a measure for segmentations containing an arbitrary number of objects is presented. This generalisation is not applied here in order to keep the quality measure simple and intuitive. And actually, it does not make the method less general.

Consider a brain scan with a ground truth segmentation dividing the scan into "white matter", "grey matter", and "other matter". The quality measure seems to be too limited to handle this problem. However, an evaluation of the applicability of a scheme for creating building blocks is not optimally performed in a single step.

The "grey matter" of the brain is a thin, twisted layer. The shape of the "white matter" is more like a solid blob with short tentacles. The qualitative difference between the shapes of the objects of interest implies that the evaluation should be performed in two stages: one stage that investigates performance against thin structure, and one stage that investigates performance against blob-like structure. A single-stage evaluation will only report the "average performance" without really revealing the true applicability of the segmentation method.

6.3.2 Method 2: Optimal with Limited Number of Actions

Evaluation method 2 shows the quality of a segmentation by the graph of the relative part of the ground truth segmentation optimally recovered given a limited number of actions.

Figure 6.1 illustrates the evaluation method. Notice that the graph does not display the processing cost as a function of the quality of the segmentation as stated in section 6.3 (but instead quality as a function of the processing cost). This is obviously just the inverse function. Both graphs are used where appropriate.

Note also, that insensible behaviour by a user that make the segmentation error rise with an increasing number of actions. However, since the method is defined in terms of the optimal set of actions, the graphs are monotonically decreasing.

An efficient algorithm for the method is included in section 7.3.

6.3.3 Method 3: Top-Down User-friendly Strategy

The two previous evaluation methods are based on optimal segmentations. However, a typical user will not apply the optimal set of actions in a segmentation task. Instead, the user will gradually develop an informal strategy that seems to work good.

If the user does not experience the applicability of the building blocks in the same way as the evaluation methods, the results from the first two evaluation methods are somewhat irrelevant.

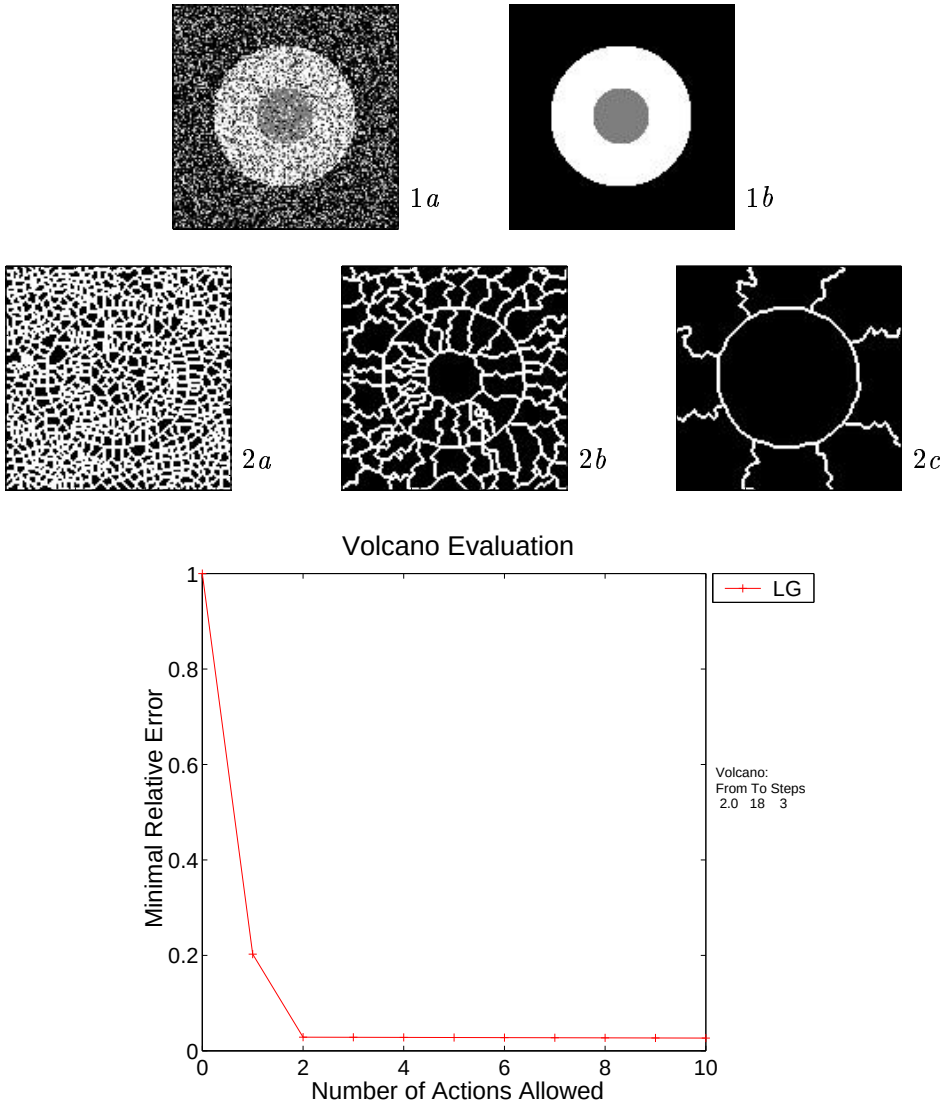


Figure 6.1: *Illustration of evaluation method 2 on segmentation of the volcano image.*
Figure 1a is the example image. It originates from figure 1b, which serves as ground truth segmentation for this example. Approximately 50% *of the pixels in figure 1a have been replaced with random noise. The white ring is the object of interest.*
Figures 2a, 2b, and 2c show the catchment basin scale-space generated in only three levels from scale 2 to scale 18. The shown regions are the available building blocks for the segmentation.
The graph shows the result of evaluation method 2 performed on the example. With zero actions the relative error of the segmentation is 1. With one action allowed the error drops to 0.20. This is due to the selection of the large center region in figure 2c. With two actions allowed the center region of figure 2b is deselected. Now the relative error is 0.03. This means that 97% *of the ground truth object was segmented in two actions. The remaining* 3% *corresponds to the small fluctuations around the object border due to the high degree of noise. These remaining pixels will have to be selected/deselected pixel-wise.*

The purpose of this evaluation method is therefore to evaluate the applicability of the building blocks when the user follows a simple, reasonable heuristic strategy.

Evaluation with this method will reveal two things:

- How good is the heuristic strategy compared to the optimal strategy in method 1?
- Are there the same relative differences between the applicability of the building blocks from the different segmentation schemes in this evaluation method as in the other?

Obviously, the naive hope is that this evaluation method will show that it is possible to teach the user a simple, "near-optimal" strategy.

The strategy is as follows:

Go through the scale levels from top to bottom. At each level each region is

- *selected if it is not currently selected and more inside than outside the ground truth object.*
- *deselected if is is currently selected and more outside than inside the ground truth object.*

Evaluation method 3 measures the value of a segmentation by the processing cost resulting from the strategy above.

An efficient algorithm that implements the method is included in section 7.4.

6.3.4 Tolerance Area

As mentioned earlier, ground truth data are not uniquely perfect. The notion of a perfect contour enclosing the ground truth object should therefore be replaced by a contour with a specific imprecision.

For real data, both the image acquisition process and the manual segmentations performed by experts are sources of imprecision. For both, the imprecision contributions could be established through statistical analysis. For simulations the accuracy is an (possibly implicit) attribute of the model. For phantom data the imprecision sources are the acquisition process as well as the measurements of the phantom object.

In addition, the pixel/voxel grid discretisation imposes an extra source of imprecision. Typically object boundaries are not perfectly aligned with the pixel/voxel grid. Therefore, a pixel/voxel should not ideally be classified exclusively as "inside" or "outside". This is called the *partial volume effect*.

The evaluation methods are neither designed to account for imprecision nor the partial volume effect. A simple way to incorporate this is through a *tolerance area*. The tolerance area is defined as an area (or volume in 3D) around the ground truth contour where the quality measure is relaxed.

Since the imprecision of the contour is typically unknown, the proper width of the tolerance area is also unknown. Analogously, the proper relaxation of the quality measure is also unknown.

In this thesis it is assumed that the comparative evaluation of the performance of the diffusion schemes is unaffected by the imprecision and partial volume effects. The assumption is, that two different segmentation methods will be punished equally by the arbitrary quirks of the ground truth data.

This assumption is subjected to a simple test in Part Four. The tolerance area is defined as the width of one pixel. The pixels inside the tolerance area are simply allowed to be arbitrarily segmented without punishment from the quality measure. The expectation is that the incorporation of a tolerance area does not affect the comparison between the different segmentation schemes.

However, the incorporation of the tolerance area will make the segmentation methods look better. With a small tolerance area the evaluation methods will possibly even give a more fair measure of the performance of the segmentation schemes. Whether or not this is actually the case depends on the (unknown) imprecision and partial volume effects in the ground truth data.

This described tolerance area requires a simple addition to the evaluation algorithms — details are in section 7.6.

6.4 The Evaluation Yardstick

The evaluation methods yield a measure of the quality of the different diffusion schemes for the segmentation method. This allows comparison of the different schemes and thereby the choice of the best scheme for a specific segmentation task.

However, this comparison does not explicitly give an intuitive sense of the absolute quality of each method. Comparison with another, possibly inferior, method only offers a relative measure. Ideally, the methods should be evaluated against the theoretically optimal method. Unfortunately the theoretically optimal method has a trivial processing cost: One action yields perfect match with ground truth. This does not add much insight on the performance of other methods. Instead, they can be put into perspective by a comparison with a "simplest reasonable linking scheme".

Quad trees (see [Samet, 1984] for a review, and figure 6.2 for an illustration) offer a suitable scheme for naive construction of building blocks for segmentation. The building blocks of the quad tree are not adapted to the geometry of the image. The image is recursively split into four blocks of equal size. This gives a hierarchy with pixels at the lowest level and the entire image at top level. The linking of the individual building blocks in this tree is trivial.

The performance of the segmentation method based on quad trees offers a suitable "zero point" for the measure of the performance of the other schemes.

It is worth mentioning that quad trees are not just a "silly" linking scheme. The number of quad tree building blocks required to construct an object is a relevant measure of the complexity of the object. This is due to the relation to the dimension of the object.

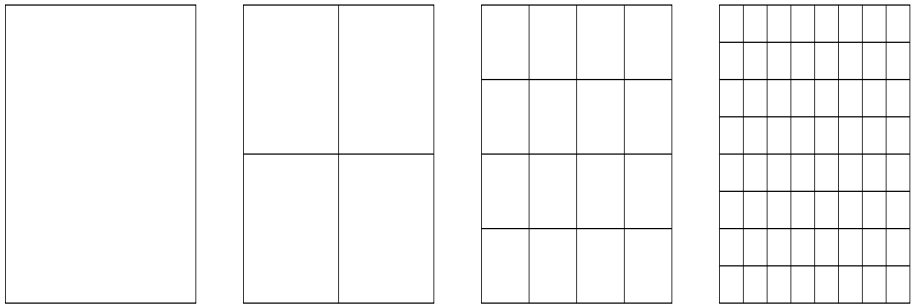


Figure 6.2: *The Quad Tree principle splits the image recursively into four blocks of equal size. Note that this does not require that the image is quadratic. If the image sizes are not a power of two, some minor irregularities is needed to reach regions of with only a single pixel. The hierarchy of regions can be used as the simplest reasonable building blocks for the segmentation method.*

A simple generalisation of the quad tree principle allows the blocks to have arbitrary size (instead of the sizes defined by the splitting scheme). The number of blocks required to provide a covering of the object then has a close relation to the *box-counting dimension*. The box-counting dimension is a definition of the fractal dimension of a given set — and a simplification of the Hausdorff dimension.

Therefore the "zero point" for the performance is defined in terms of the complexity of the objects of interest. This allows a notion of invariance with respect to the specific ground truth objects in the evalution [Ott, 1993].

6.5 More on Evaluation

In [Koster, 1995] a similar evaluation method is designed for a segmentation system called the *Hyperstack*. Since the hyperstack is an automatic segmentation method the evaluation is somewhat different.

The evaluation is based on a measure of the degree of correctness based on pixel correspondance with the ground truth segmentation like the one presented here. In general, an automatic method does not produce satisfying results — in realisation of this Koster adds a post-processing phase. The segmentation method is then evaluated in terms of the post-processing editing required to reach a sufficiently correct segmentation. Actually, the method can thereby be viewed as semi-automatic.

The post-processing editing actions are *merge* and *split*. A *merge* consists of selecting two segments and thereby merging them into one. A *split* is more complicated and consists of the definition of a new segment border by selection of the individual pixels of the border. This border splits an existing segment into two new segments. The post-processing editing cost is a weighted sum of the number of clicks necessary to perform the actions. Typically a *split* action is much more costly than a *merge* action.

The acceptable segmentation quality is either defined by a tolerance area around the object boundary or by imposing a limit on the number of pixels that are allowed to be incorrectly segmented. The limit on the number of incorrect pixels is defined relative to the number of boundary pixels in the ground truth objects. Thereby both quality requirements are somewhat independent of the object complexity.

6.6 Perspectives on the Evaluation

The concepts of information theory provide another perspective on the evaluation process.

A canonical user action can be replaced by the number of bits required to represent the selection/deselection of the region. Likewise, the error measure of the segmentation can be defined in terms of the a minimal description of the difference between the segmentation and the ground truth. This leads to a MDL approach (see [Leclerc, 1989] for an example of MDL segmentation).

The evaluation methods described here can be formulated as an MDL approach (with suitable representation cost of the actions and a suitable error model).

6.7 Alternative Evaluation Strategies

Typically, ground truth data sets are scarce. Therefore, evaluation methods that do not require ground truth data has been investigated. This has been used successfully for evaluation of stereo algorithms.

The general idea is to apply a method to more than one representation of the same physical data. In stereo this is done with pictures of the same scene from slightly different viewpoints. The evaluation checks whether the results from the different viewpoints are "self-consistent" (see [Leclerc et al., 1999]).

An interesting point is that a an evaluation using ground truth data have been compared with an evaluation on the same data set using the *self-consistency* principle. The results were similar (see [Szeliski & Zabih, 1999] for details).

The adaption of the self-consistency principle to segmentation is not obvious. An evaluation using several images of the same data would primarily reveal the robustness of the segmentation method. It could possible reveal if an algorithm was not invariant to rotations of the image. But it would not be able to indicate whether the segmentation actually produced the correct salient segments from the image. This is simply because the specific segmentation task defines what is to be perceived as objects in the image.

6.8 Summary

In this chapter three evaluation methods are designed. Each of them is based on:

- Ground truth segmentation data that defines the "correct segmentation".
- The processing cost defined as minimal number of selections and deselections of building block required to perform the segmentation.

The first method is simply based in the processing cost required to reach ground truth.

The second method refines this by looking at how good a segmentation can be achived with a limited number of actions. This requires a measure of the quality of a segmentation — this is defined through a measure that simply counts the number of falsely segmented pixels/voxels (either inside ground truth and not selected, or outside but selected).

The third method is designed to investigate whether a user will be able to take advantage of the segmentation method. This method is based on a simple — not optimal — selection strategy.

In the following chapters in Part Two the intricate details of the evaluation methods will be scrutinised. The results from the actual evaluation are not revealed until Part Four.

Per aspera ad astra
Latin proverb

Chapter 7

The Evaluation Algorithms

In chapter 6 three evaluation methods are designed. The methods are intuitively simple — efficient implementations are unfortunately a bit more complicated. The purpose of this chapter is to present algorithms that allow efficient implementation of the evaluation methods.

The linking structure of the catchments basins from the multi-scale watershed segmentation can be perceived as a forest since each region is linked to exactly one parent region at the scale level above. Therefore, the algorithms in the following are based on tree traversals.

7.1 The Warm Ups

The linked regions are perceived as a collection of trees. The leaves of the trees are the individual pixels/voxels. The inner nodes of the trees are the catchments basins from the multi-scale watershed segmentation. The roots of the trees correspond to the regions at the highest scale level.

The algorithms are simpler if a conceptual simplification is added. The individual trees in the forest is linked to one "conceptual parent" — the entire image. Thereby, it is not necessary to keep track of a list of trees in the algorithms. This simplification obviously has no influence on the result of the evaluation.

In chapter 4 it is shown that creations of regions can be expected as scale increases. It is worth noticing that these creations pose no problem to the evaluation algorithms. The regions at the detection level are represented by the regions at localisation level that they are linked to. Regions that are created at a higher level than the localisation level, are not linked to any regions at the localisation level. They are therefore "invisible" at the detection level — both to a user of a program and the evaluation algorithms.

A good, intuitive sense of the linking tree is necessary in the following. At lowest level each node is a pixel. For each level in the tree a number of regions are merged. This corresponds

to removing the region border between regions corresponding to nodes in the three. No region border ever moves as the scale is increased — the border elements are simply removed. All leaf nodes in the tree are the same number of levels from the root node (they have the same depth).

The algorithms have a few terms in common:

- Three choices are possible at each node in the tree: *Select*, *Deselect*, and *Ignore*. Only the first two count as actions. When a node is ignored, its descendants and predecessors in the tree determine whether the corresponding part of the image is included in that particular segmentation.
- Only two states are possible at each node: *In* or *Out*. This state determines whether the node contributes to the actual segmentation. The state of a node is inherited from the parent by default. The "artificial" root of the tree is considered *Out*. This requires a number of actions at lower nodes for anything in the image to become *In*.

The resulting segmentation is "read" from the states of the nodes — starting with the root and letting the children override the state from the parent for the subregion corresponding to the child.

7.2 Method 1: Minimal Processing Cost

The algorithm determines how many actions it requires to achieve the perfect segmentation according to the ground truth segmentation. This number can be established by exhaustive search of all combinations of *Select*, *Deselect*, and *Ignore* at each node. However, the computational complexity of this approach is not appealing.

The following algorithm provides the optimal action count:

1. For each leaf:
   - Determine whether the leaf pixel is inside or outside the ground truth.
   - Determine *In*-count (supposing the parent has state *In*):
     If the leaf is outside, it is necessary to *Deselect* it and the action count is one.
     If the leaf is inside, no action is necessary and the action count is zero.
   - Determine *Out*-count (supposing the parent has state *Out*):
     If the leaf is outside, no action is necessary and the action count is zero.
     If the leaf is inside, it is necessary to *Select* it and the action count is one.
   - Keep both of these conditional counts — denoted *In*-count and *Out*-count — as attributes for the leaf.
2. Visit each inner node in the tree in a bottom up order:
   Assume the node has m children. Let the *In*-counts for the children be denoted $I_1, \ldots, I_m$. Let the *Out*-counts be denoted $O_1, \ldots, O_m$.

- Determine *In*-count (supposing the parent has state *In*):
  We can either choose to keep the default state *In* or choose to *Deselect* the node. If we keep the state, we can simply add the *In*-counts of the children in order to get an *In*-count I^I for the node. If we *Deselect* the node we get an *In*-count I^A by adding one to the sum of the *Out*-counts of the children. In order to get the optimal count *In*-count I we choose a smallest of these two possible counts:

$$\begin{aligned} I^I &= \sum_{i=1}^{m} I_i \\ I^A &= 1 + \sum_{i=1}^{m} O_i \\ I &= min\{I^I, I^A\} \end{aligned}$$

  The superscripts in I^I and I^A are short for *Ignore* and *Action*.
- Determine *Out*-count (supposing the parent has state *Out*):
  The *Out*-count O is achieved in the same manner:

$$\begin{aligned} O^I &= \sum_{i=1}^{m} O_i \\ O^A &= 1 + \sum_{i=1}^{m} I_i \\ I &= min\{O^I, O^A\} \end{aligned}$$

3. At the root:
   - Since the root state is *Out* by definition, the optimal action count is the *Out*-count for the root node.

Note that we do not have to create the leaves explicitly. For each leaf-parent (corresponding to the regions at the localization scale level) the *In*-count and *Out*-count can simply be calculated from a count of the number of pixels inside and outside of the ground truth for each region.

It is quite simple to see that the algorithm provides the minimal processing cost by structural induction. At the leaf nodes the minimal processing is trivially performed. At an inner node, the minimal processing cost is obtained since the minimal of the two possibilities (corresponding to *Ignore* or *Select*/*Deselect*) is chosen.

7.3 Method 2: Optimal with Limited Number of Actions

The algorithm determines how good a segmentation can be achieved with a limited number of actions.

It is an extension of the algorithm from method 1. Instead of keeping *In*-count and *Out*-count at each node, tables *In*-table and *Out*-table are kept. Given a maximal allowed number of actions

for the node subtree, the table states the error count. The error count is the number of falsely segmented pixels (either inside ground truth but not selected, or outside but selected).

The key element of the algorithm is the way the *In*-tables and *Out*-tables of siblings are combined to produce the *In*-table and *Out*-table of the parent.

The following algorithm provides the error count given a limited number of actions:

1. Denote the given maximal number of actions k.
2. For each leaf:
   - Determine whether the leaf pixel/voxel is inside or outside the ground truth.
   - Determine *In*-table (supposing the parent has state *In*):
     If the leaf is outside, a *Deselect* action is necessary for reaching error count zero. This gives the following *In*-table I:

Number of actions:	0	1	2	$\cdots$	k
Minimal error count:	1	0	0	$\cdots$	0

     In short notation this is written $I = [[1, 0, \ldots 0]]$
     If the leaf is inside, the error count is zero even without any actions. This yields the *In*-table $I = [[0, 0, \ldots, 0]]$.
   - Determine *Out*-table (supposing the parent has state *Out*):
     If the leaf is outside, the *Out*-table becomes $[[0, 0, \ldots, 0]]$.
     If the leaf is inside, the *Out*-table becomes $[[1, 0, \ldots, 0]]$.
3. Visit each inner node in the tree in a bottom up order:
   - Determine *In*-table (supposing the parent has state *In*):
     We can either choose to keep the default state *In* or choose to *Deselect* the node.
     If we keep the state, we must combine the *In*-tables of the children (in an optimal manner) to produce the *In*-table I^I for the node:
     - Initialise $I^I = [[0, 0, \ldots, 0]]$
     - For each of the m children of the node, *Combine* their *In*-tables I_i successively with the *In*-table I^I:

$$\begin{aligned} 1) \quad I^I &= Combine(I^I, I_1) \\ &\vdots \\ \mathrm{m}) \quad I^I &= Combine(I^I, I_m) \end{aligned}$$

     The *Combine* operation destributes the available actions between the children in an optimal manner. The operation $T = Combine(T_1, T_2)$ that combines the table is defined:

$$\begin{aligned} T(0) &= T_1(0) + T_2(0) \\ T(1) &= min\{T_1(0) + T_2(1),\ T_1(1) + T_2(0)\} \\ \vdots &= \vdots \\ T(i) &= min\{T_1(n) + T_2(i - n) \mid n \in [0..i]\} \end{aligned}$$

Here $T(i)$ is the entry in the action table corresponding to the minimal error count with i actions allowed. The successive application of the *Combine* operation is written in short $I^I = Combine(I^I, I_1, \ldots, I_m)$.

If the state is changed by a *Deselect* action, the *Out*-tables of the children are combined to produce the *In*-table I^A for the node. However, since an action is performed the entry $I^A(0)$ makes no sense. Therefore it is assigned the error ∞:

- Initialise $I^A = [[\infty, 0, \ldots 0]]$
- For each of the node's m children, *Combine* their *Out*-tables O_i successively with the *In*-table I^A:
  $I^A = Combine(I^A, O_1, \ldots, O_m)$
  The *Combine* operation is previously defined.

The minimal entries from the two *In*-tables I^I and I^A now yield the optimal *In*-table I for the node (where $i \in [0..k]$):

$$I(i) = min\{I^I(i), I^A(i)\}$$

Note that $I^A(0) = \infty$ ensures that $I(0) = I^I(0)$.

- Determine *Out*-table (supposing the parent has state *Out*):
  We can either choose to keep the default state *Out* or choose to *Select* the node.
  If we keep the state, we must combine the *Out*-tables of the children to produce the *Out*-table O^I for the node:
  - Initialise $O^I = [[0, 0, \ldots, 0]]$
  - For each of the m children, *Combine* their *Out*-tables O_i successively with the *Out*-table O^I:
    $O^I = Combine(O^I, O_1, \ldots, O_m)$

  If the state is changed by a *Select* action, the *In*-tables of the children are combined to produce the *Out*-table O^A for the node. Note that $O^A(0)$ is initialised to ∞ since an action is performed.
  - Initialise $O^A = [[\infty, 0, \ldots 0]]$
  - For each of the m children, *Combine* their *In*-tables I_i successively with the *Out*-table O^A:
    $O^A = Combine(O^A, I_1, \ldots, I_m)$

  The minimal entries from the two *Out*-tables O^I and O^A yield the optimal *Out*-table O for the node (where $i \in [0..k]$):

$$O(i) = min\{O^I(i), O^A(i)\}$$

  Note that $O^A(0) = \infty$ ensures that $O(0) = O^I(0)$.

4. At the root:

   - Since the root state is *Out* by definition, the minimal error count with the number of actions limited to k can be inspected from the *Out*-table O^I for the root node as the entry $O^I(k)$.

Again, it is simple to see that the algorithm provides the minimal processing cost by structural induction. At the leaf nodes the entries in the tables are trivially minimal. At an inner node, the entries in table are minimal since the best among the possible distributions of the available number of actions are chosen.

Note that the $0, 1, \ldots, k-1$ entries in the *Out*-table of the root node reveal the minimal error count for the number of actions limited to their index. In particular, the entry for index zero will display an error count equal the area of the ground truth — since no actions are allowed, the entire area will be missing.

If k is chosen as the optimal action number from method 1 (or larger), this will result in an error count of zero.

Neither algorithm 1 nor 2 yield the actual actions that result in the segmentation. It is quite simple to extend the algorithms to deliver the list of actions as well, though. In fact, this is done in the implementation, but including it in the algorithms would clutter the presentation.

7.4 Method 3: Top-Down User-friendly Strategy

The simplicity of the strategy is reflected in the algorithm. Each node has a state as in the previous methods, but only a global count of the number of actions is kept:

1. Start at the root node.
   This node has state *Out* by definition. *Action*-count is zero.
2. Visit each of the remaining nodes in a top down order:
   - Determine whether the corresponding region is mostly inside or outside the ground truth segmentation.
   - Supposing the parent has state *In*:
     If the region is mostly outside, the state of the node is changed to *Out* by a *Deselect* action and the *Action*-count is incremented.
   - Supposing the parent has state *Out*:
     If the region is mostly inside, the state of the node is changed to *In* by a *Select* and the *Action*-count is incremented.
3. When all nodes have been visited, *Action*-count is the number of actions performed.

Note that the segmentation will be without error (since the leaf nodes correspond to the individual pixels).

7.5 Complexity

Let p be the number of pixels/voxels and n be the number of inner nodes in the tree.

Algorithm 1 has a complexity of $O(p + n)$. Each inner node is inspected once from its parent. This is optimal since an algorithm has to visit every node in the tree (or convert the tree into some other data structure — which would require a visit at each node anyway).

Let k be the action count limit.

Algorithm 2 has a complexity of $O(p + n * k)$. A *Combine* operation is performed once on the *In*-table and once on the *Out*-table of each inner node. The *Combine* operations are linear in the action count limit. It is not equally trivial to see that this complexity is optimal.

Algorithm 3 has a complexity of $O(p + n)$. This is optimal.

7.6 Addition of Tolerance Area

In section 6.3.4, the possibility of a tolerance area is presented. This is extremely simple to add to the previous algorithms.

If the tolerance area is larger than one pixel, the distance to the ground truth contour is calculated for each pixel/voxel. It the tolerance is the pixel-width, the tolerance area is simply the pixels/voxels with a neighbour on the other side of the ground truth contour.

In algorithm 1, pixels inside the tolerance area are simply ignored from both *In*-count and *Out*-count.

In algorithm 2, leaf nodes corresponding to pixels inside the tolerance area are given both *In*-table and *Out*-table equal $[[0, 0, \ldots, 0]]$.

In algorithm 3, *Action*-count is not incremented for actions performed on pixels inside the tolerance area.

Living with uncertainty is unavoidable, but don't ever let that paralyze you with hesitation.
Bruce J. Ketchum

Chapter 8

The Ground Truth Data Sets

A number of ground truth data sets have been obtained.

The focus has been put on MR brain scans. The limited contrast information, combined with the intricate shapes of the various brain tissue types, provide an excellent challenge for segmentation methods.

The reader is assumed to be familiar with brain scans and brain anatomy. However, readers without this knowledge can simply look at the illustrations and thereby get a good sense of the structures that are to be segmented.

A number of data sets are mentioned even though they have not been used in the evaluation. However, the data sources seem very well suited for future work, and could be valuable for other projects, as well.

8.1 2D versus 3D

The segmentation and evaluation methods described in the previous chapters have only been implemented in 2D. However, it should be stressed that both the segmentation and evaluation methods generalise straightforward to higher dimensions.

The ground truth data sets acquired are all 3D. This is both an advantage and a disadvantage.

A 3D data set can be split into a number of 2D slices. Since ground truth data are scarce, it is nice that a single 3D data set provides a lot of 2D test data.

However, when manual segmentation is performed directly in 3D, the resulting ground truth segmentation might not be optimal for 2D use. Small errors are more easily overlooked in the 3D volume. It is assumed that the effect of this is negligible. Since manual segmentation is most often performed slice-wise (where the segmentation from one slice is used as "initial guess" for the next slice), this is a fair assumption.

8.2 Internet Brain Segmentation Repository

The data at the *Internet Brain Segmentation Repository* is provided by the *Center for Morphometric Analysis at Massachusetts General Hospital* ([IBS, 1999]). Both real MR brain scan data and corresponding ground truth segmentations are available at the IBSR web site at `http://neuro-www.mgh.harvard.edu/cma/ibsr`

Two main data sets have been obtained:

Normal Male
Data set 788_6_m is a "typical" T1-weighted MR brain scan. The name 788_6_m means test subject 788, scan 6, segmentor "m". The test subject is a 55 year old male. The set contains 60 256x256 slices with slice thickness 3.0mm. The slices are coronal with a flip angle of 40 degrees.
Two segmentations are provided. One where the tissue is marked "grey matter", "white matter", or "other matter". The other contains outlines of further neuroanatomical structures.

Schizophrenic subject
Data set 1320_2_max is analogous to 788_6_m with the exception that the test subject is schizophrenic. The set contains 128 256x256 slices with slice thickness 1.0mm.

Figure 8.1 displays example slices from the normal brain with the corresponding manual ground truth segmentations. Example slices from the brain scan of the schizophrenic test subject are shown in figure 8.2.

Also available at the IBSR are ground truth segmentations of a collection of 20 normal subjects with various anatomical structures. They could be a relevant data set for other studies. They have been left out of this thesis since the interpretation of the data is somewhat ambiguous.

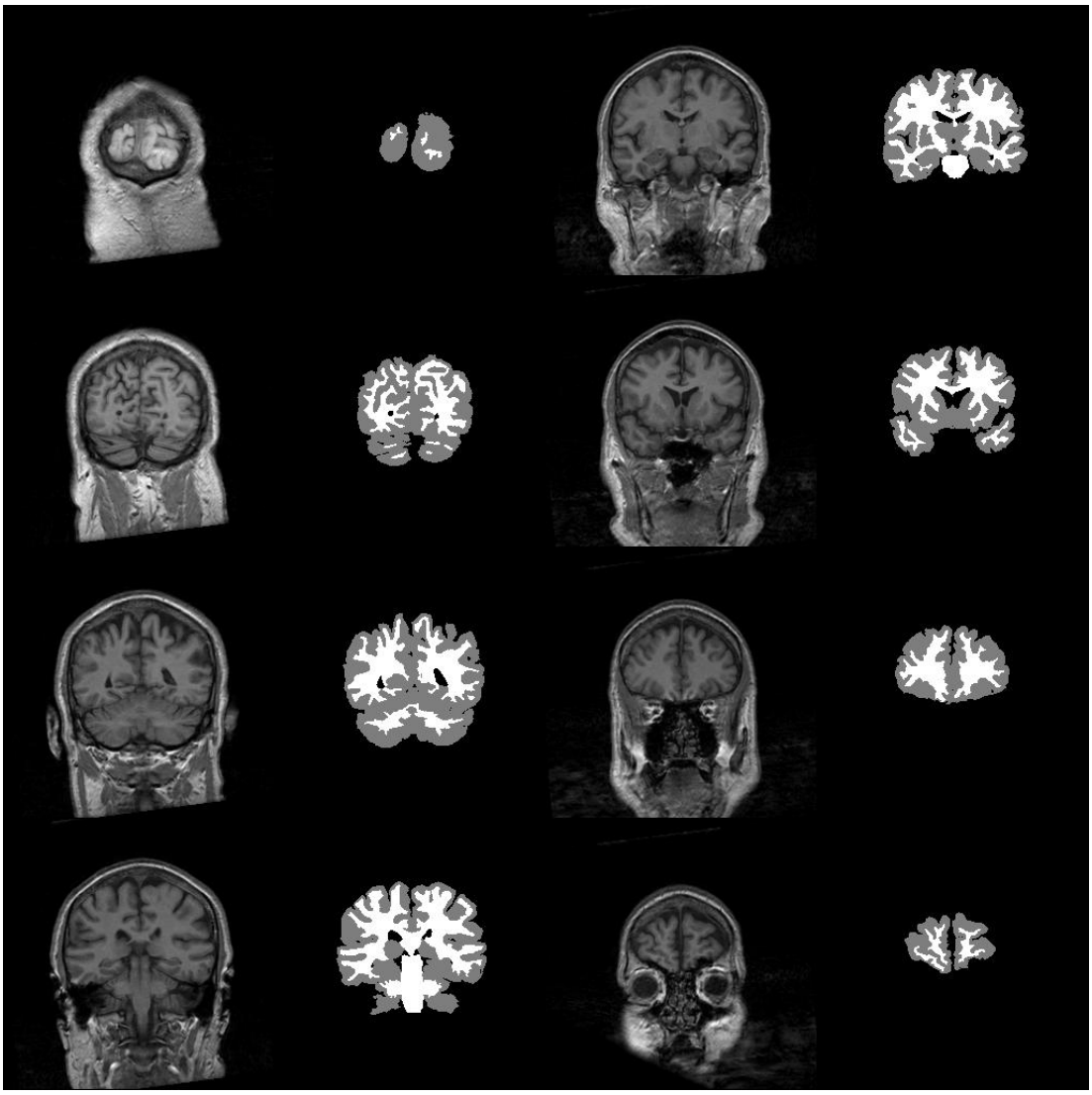


Figure 8.1: *Internet Brain Segmentation Repository example of real MR scan of normal brain. Every 7th slice from slices 4 to 53 are shown. For each slice the corresponding manual ground truth segmentation is displayed. The grey areas are "grey matter" — white areas are "white matter" tissue.*
The MR brain data set 788_6_m *and its manual segmentation was provided by the Center for Morphometric Analysis at Massachusetts General Hospital and is available at* `http://neuro-www.mgh.harvard.edu/cma/ibsr`

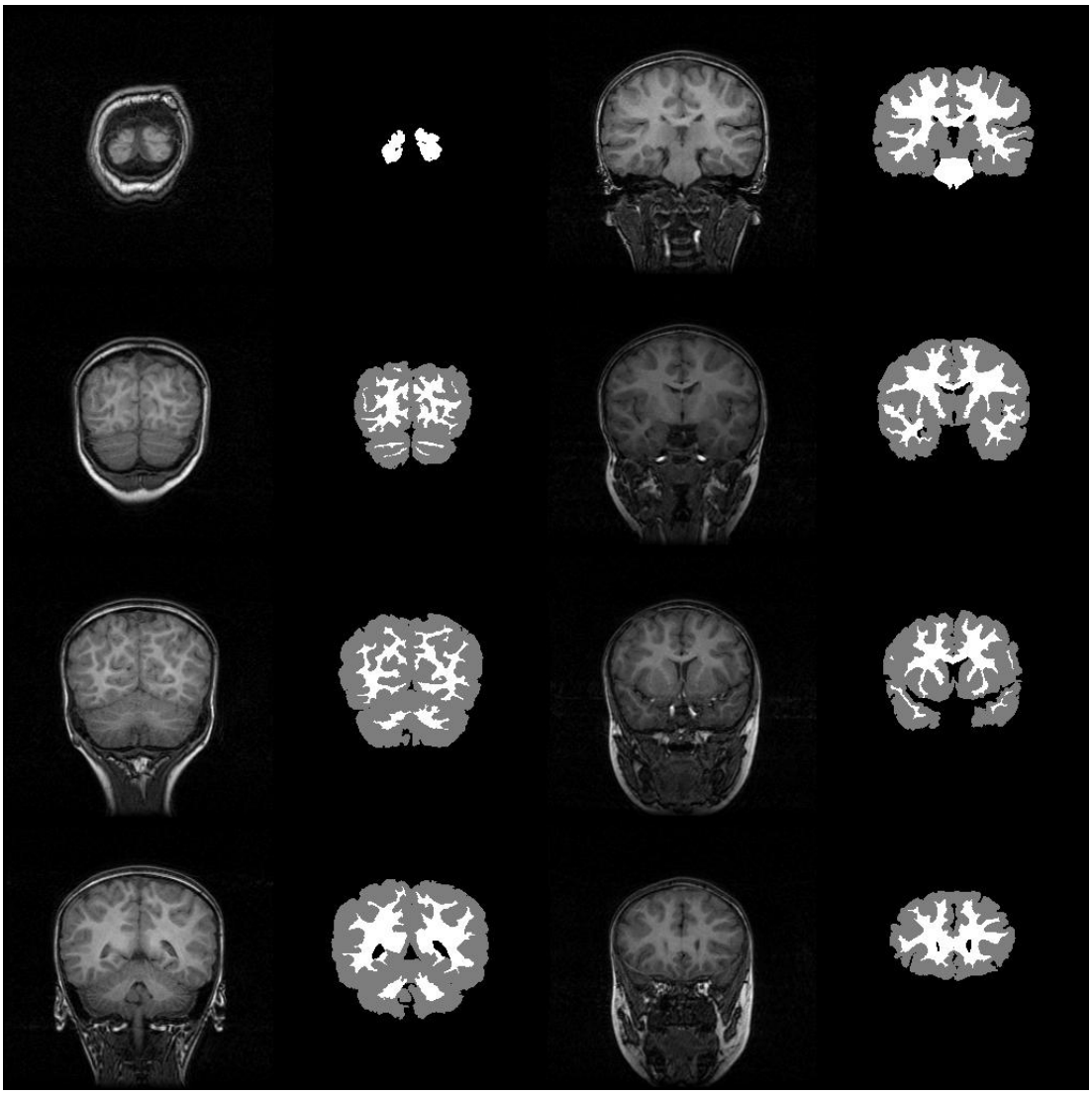


Figure 8.2: *Internet Brain Segmentation Repository example of real MR scan of brain from schizophrenic person. Every 10th slice from slices 17 to 87 are shown. For each slice the corresponding manual ground truth segmentation is displayed. The grey areas are "grey matter" — white areas are "white matter" tissue.*
The MR brain data set 1320_2_*max and its manual segmentation was provided by the Center for Morphometric Analysis at Massachusetts General Hospital and is available at* `http://neuro-www.mgh.harvard.edu/cma/ibsr`

8.3 Brain Web

The *McConnell Brain Image Centre* at *Montreal Neurological Institute and Hospital* is another source of ground truth data for segmentation evaluation. The centre runs a web site (`http://www.bic.mni.mcgill.ca/brainweb/`) where the data can be freely downloaded.

The data supplied at the *Brain Web* are simulated data. There are two main data sets available: a model of a normal brain, and a model of a brain with multiple schlerosis. From these anatomical models, brain scans can be generated using a MRI simulator. The simulator allows specification of modality (pulse sequences T1, T2, or proton-density-weighted), slice thickness, noise level, and intensity non-uniformity (denoted RF). For technical details on the models and the MRI simulator see [Cocosco et al., 1997], [Kwan et al., 1996], and [Collins et al., 1998].

Each model consists of two types of 181x217x181 3D volumes:

- A discrete volume with each voxel marked with the most dominant type of tissue. The tissue types are CSF, grey matter, white matter, fat, muscle, skin, skull, glial matter and connective. For the multiple schlerosis model an additional type for the MS lesion is present.
- A volume for each tissue type describing the percentage of the voxel populated by this tissue type.

The second data set allows analysis of partial volume effects. This is not done in this thesis.

Two data sets from the Brain Web have been used in this thesis. Both are from the normal brain phantom generated as T1 with slice thickness 1 mm and an intensity non-uniformity level of 20%. The pixel size in each plane is 1mm by 1mm. The two sets differ in the noise level which is 1% in the first set and 9% in the other. This noise level refers to the amount of Gaussian noise added to intermediary components of the simulation process. This results in rayleigh distributed statistics for the resulting image. However, the key observation is simply that 1% noise is virtually nothing and 9% is quite a lot.

Figure 8.3 shows example slices from the two data sets with the corresponding ground truth segments for grey matter and white matter.

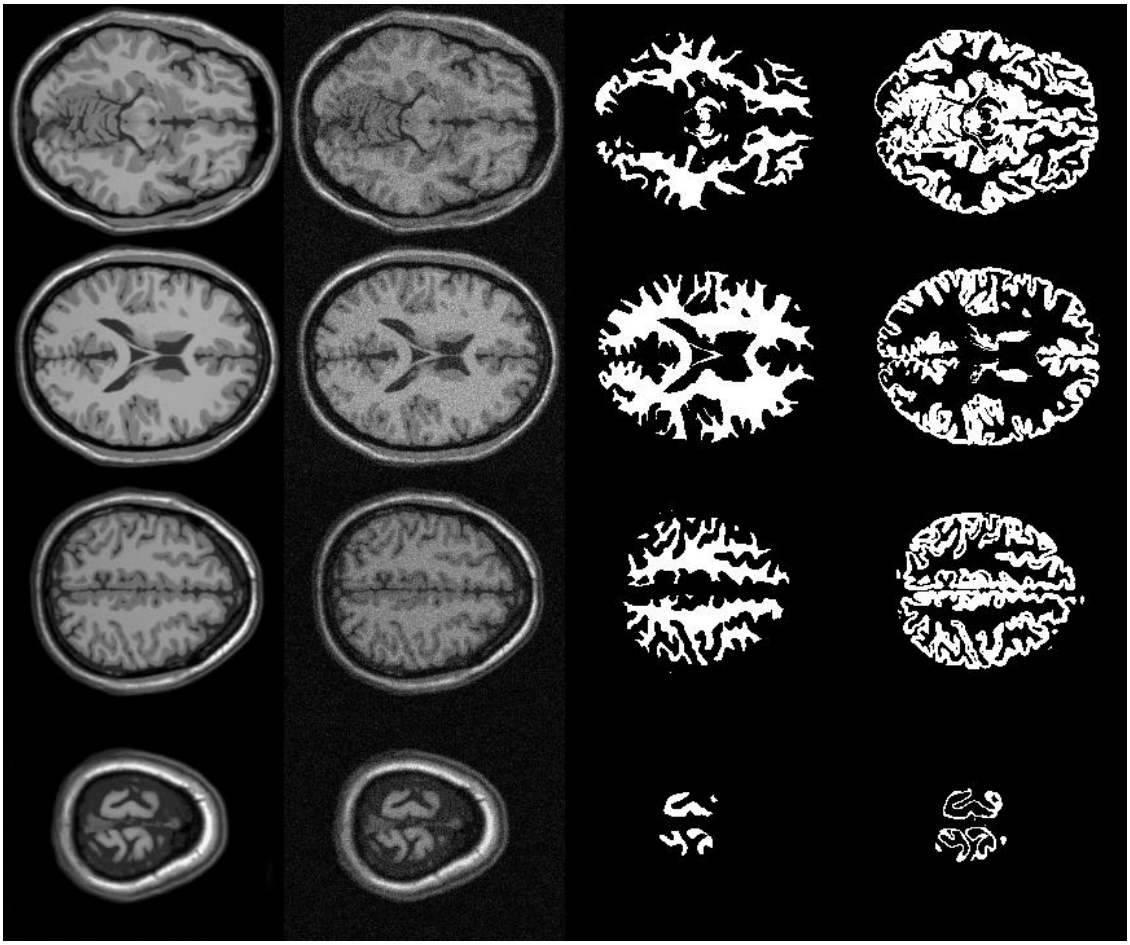


Figure 8.3: *Example of simulated MRI brain scan with ground truth segmentations obtained at Brain Web. The figure shows four coronal slices (slice 60, 90, 120, and 150) from the normal brain.*
The first column shows the four slices with a simulated noise level of 1%.
The second column has noise level 9%.
The third column shows the ground truth white matter segment.
The fourth column shows the ground truth grey matter segment.
Both simulated sets are T1 pulse sequence with slice thickness 1mm and 20% *RF.*
Data sets were acquired at the Brain Web: `http://www.bic.mni.mcgill.ca/brainweb/`

8.4 The Visible Humans Project

The visible humans project is a research project based on unique data. Two data sources, in the shape of a dead female and a dead male test subject, have been CT and MR scanned. Furthermore, the subjects have been subjected to cryosection.

These multi-modal data provide accurate data for numerous research activities. Of particular interest for this project is an ongoing reseach project with the intent of providing ground truth segmentations of large parts of the data.

The data were not available in time for this thesis. For more information, see [VHP, 1995].

8.5 The SPL & NSG Brain Tumor Database

Related to the visible humans data is the SPL & NSG Brain Tumor Database. Part of this research is centered on ground truth segmentations of the brain. This part is based partly on [Kaus et al., 1998].

Unfortunately, these data were not available in time for this thesis. For more information, see [SPL, 1999].

8.6 The Evaluation Sets

The evaluation sets are variations of MR brain scans. A total of 8 sets are selected for the evaluation:

Acronym	Description
BW1W	**B**rain **W**eb simulated MR data. Noise level **1**%.
	Ground truth segmentation of **W**hite matter tissue.
BW1G	**B**rain **W**eb simulated MR data. Noise level **1**%.
	Ground truth segmentation of **G**rey matter tissue.
BW9W	**B**rain **W**eb simulated MR data. Noise level **9**%.
	Ground truth segmentation of **W**hite matter tissue.
BW9G	**B**rain **W**eb simulated MR data. Noise level **9**%.
	Ground truth segmentation of **G**rey matter tissue.
IRNW	**IBSR** real MR data. **N**ormal brain.
	Manual ground truth segmentation of **W**hite matter tissue.
IRNG	**IBSR** real MR data. **N**ormal brain.
	Manual ground truth segmentation of **G**rey matter tissue.
IRSW	**IBSR** real MR data. **S**chizophrenic test subject.
	Manual ground truth segmentation of **W**hite matter tissue.
IRSG	**IBSR** real MR data. **S**chizophrenic test subject.
	Manual ground truth segmentation of **G**rey matter tissue.

All data sets are expected to provide any segmentation method with a real challenge. In a typical data slice, each ground truth object consists of several disconnected areas with rugged contours. Even further, there is relatively low contrast across the areas in the data that correspond to edges in the ground truth segmentation.

Apart from the general level of difficulty, the noisy data are expected to be more difficult to segment that the "clean" data (BW9W versus BW1W, and BW9G versus BW1G, respectively). For all data sets it is also expected that the grey matter will be harder to segment than the white matter.

It should be noted that the evaluation is not based on the entire data sets. Typically, only every 10'th or 20'th slice is used in the evaluation. This is simply due to the computational time required to analyse the large amounts of data. Each evaluation example will state the actual number of slices used.

I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection.
Charles Darwin

Chapter 9

Optimisation of the Implementations

The main purpose of the evaluation methods is to allow comparison of the performances of the segmentation method with different diffusion schemes. However, there is a nice side effect. The evaluation can equally well be used to fine-tune the parameters for the various diffusion schemes. The best parameter set is the one that performs best in the evaluation.

9.1 Optimisation Concept

Since the ground truth data sets are not parameterised, it is not feasible to derive the optimal parameters analytically.

The parameter choices for a given diffusion scheme can be improved by a simple manual optimisation. However, a proper optimisation process requires the evaluation of a large number of parameter sets. Manual optimisation is therefore not desirable since the generation of the scale-spaces is generally quite time-consuming. The optimisation process will stop when the person grows tired — not when the resulting parameters are optimal.

Unfortunately, automatic optimisation is troublesome as well. The parameter spaces are not continuous — discrete parameters (like the number of scales) are annoying in optimisation algorithms. Even further, the evaluation function is not very "nice" — it is not smooth and has a plethora of local minima. Therefore, simple gradient descent methods will not produce the optimal result. In fact, no method other than exhaustive search of the parameter space is guaranteed to produce the optimal parameter set.

Fortunately, an optimal parameter set is not necessary. It is not vital whether the algorithm performs 1% better or worse with regard to the evaluation test set. The interesting thing is the overall performance compared to the other diffusion schemes. A key point is that the method is quite robust with respect to parameter changes for the various diffusion schemes.

Therefore, a number of parameter sets with almost equal performance exist (this is illustrated in section 9.3). An end user the will not notice the difference in performance between these sets of parameters.

The effect of the many local minima is minimised by the use of several evaluation test slices. The average of the performances on each slice is more robust with respect to optimisation. And more relevant, as well — the evaluation in Part Four is based on the average performance against a number of test data slices.

9.2 The Optimisation Method

The method is embarrassingly simple:

- Define initial value and step value for each parameter.
- Repeat until no more improvements are found:
  - For each parameter:
    * Four parameter sets with different values for the specific parameter are evaluated. The four values are the current value where respectively 1 and 3 times the step value are added and subtracted.
    * If the best of these alternative parameter sets provides a better performance than the current performance, the current value of the specific parameter is changed to the corresponding value.

The effect of addition and subtraction of one times the step value resembles a regular gradient descent method. The large steps (addition and subtraction of tree times the step value) are intended to allow the algorithm to escape from a local minimum.

9.3 Optimisation of Linear Diffusion for BW1W

The properties of the optimisation algorithm are illustrated by an example. The parameters of the linear diffusion scheme (*From scale*, *To scale*, *Scale levels*) are "optimised" in five independent runs.

The BW1W ground truth data set is used (Brain Web simulated MR brain scan with 1% noise, ground truth segmentation of the white matter tissue) for these examples. Five slices from this data set is used (slices 60, 80, 100, 120, and 140).

In each example, an action limit of 15 actions is used. Evaluation algorithm 2 results in a table with 16 entries that display the minimal error with the specific number of actions from 0 to 15. The error is the number of pixels falsely segmented.

The "error value" in the table below is the average error count for the 16 entries in the table averaged over the 5 slices. The average area of the ground truth segments for the 5 slices is 6115 pixels.

	A	B	C	D	E
Inital parameters					
From Scale	0.50	0.50	1.50	1.50	4.00
To Scale	15.00	13.00	18.00	18.00	8.00
Scale Steps	30	25	25	25	20
Initial error value	3828	3872	3854	3854	4142
Initial step values					
From Scale	0.05	0.05	0.05	0.10	0.40
To Scale	0.50	0.50	0.50	1.00	1.00
Scale Steps	1	1	1	2	2
End parameters					
From Scale	0.45	0.55	1.45	1.20	1.20
To Scale	15.00	11.50	18.00	18.00	14.00
Scale Steps	30	27	28	25	34
End error value	3809	3810	3793	3791	3741

First, look at runs "A", "B", "C", and "D". Various slightly different initial parameter and step values produce quite different end parameters. The end error values are more or less equal, though. This would indicate that the optimisation method has "chosen" four local "good" minima — the expected behaviour. However, it is worth noticing that the end parameter values are quite close to the initial ones. Possibly, the optimisation algorithm simply can't improve the parameter choices sufficiently due to the many local minima?

Run "E" is designed to reveal this. The initial parameter values are somewhat silly. The relatively large step values are intended to allow the optimisation algorithm to cover a large part of the parameter space. The end error value for this run is slightly better than the error values for the previous runs. This indicates that the optimisation algorithm is able to escape bad parameter values and reach a "plateau" of good values.

The lessons are:

- A lot of equally "good" parameter sets exist — the segmentation method is quite robust with respect to changes in the parameter values for the diffusion schemes.
- Use relative large step values — this prevents the optimisation algorithms from getting stuck in a "bad" local minimum.

The end error values around 3800 means that the diffusion schemes allow the segmentation method to capture approximately 38% of the ground truth segment on average over 0 to 15 actions (the relative error is $3800/6115 \approx 0.62$). Figure 9.1 displays the distribution of this. It also shows that the performances of the different parameter sets are practically equal.

A reason for the slightly superior performance of the end parameter values from optimisation run "E" could simply be that the end number of scale levels is higher. On average, a higher number of scale levels will result in better performance. Therefore, an increase in the scale levels is a bit like "cheating". An increase does give a better performance, but for computational reasons the number of levels can't be increased beyond a certain point (due to computation time and

memory usage). In order to provide a fair comparison of the different diffusion schemes, the number of scales levels is fixed to 30 for all schemes.

The fixed number of scale levels at 30 means that the end parameter set from run "E" can't be used. Furhtermore, the above optimisation runs are performed with an action limit of 15 actions. A higher action limit is desirable for two reasons. First, a higher limit reveals more about the performance of the parameter set. Secondly, the error value is the average performance for the number of actions from 0 to the action limit — a higher limit makes the optimisation based on this average more robust. In the optimisation runs that form the basis of the evaluation, an action limit of 30 is used. Therefore, another parameter set than the end parameter set from "E" is used in the actual evaluation in Part Four.

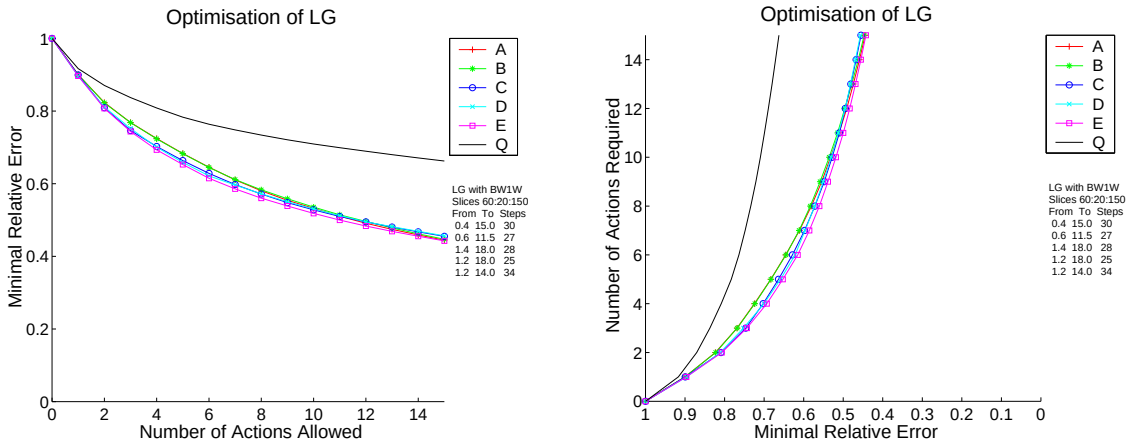


Figure 9.1: *The performance of the segmentation method based on linear Gaussian diffusion (LG) with five different parameter sets. The first four, "A" to "D", are the result of the optimisation algorithm, where the initial parameter sets are sensible. For the fifth parameter set, "E", the optimisation algorithm was initialised with silly parameter values. The approximately equal performances of the five parameter sets show the applicability of the optimisation algorithm and the robustness of the segmentation method. The last curve shows the performance of Quad tree linking.*
The left graph displays the minimal relative error as function of the number of actions allowed. The right graphs shows the same data where the number of actions required is plotted as a function of the minimal relative error.
The graph to the right gives another perspective on the comparison of the schemes. For an arbitrary minimal relative error the number of actions required differ by no more than one for the five linear Gaussian schemes. As a contrast the quad tree linking scheme requires a lot more actions. For a relative error of 0.8 the Gaussian schemes requires a little more than 2 actions — quad linking requires a little more than 4. For 0.7 Gaussian requires around 4 — quad requires 11. The user effort required to reach a specific quality threshold is less than half for Gaussian linking compared to quad tree linking in this example.

9.4 Consequences of the Optimisation Scheme

In the optimisation method, the average performance for the number of actions limited from 0 to 30 is used as the optimisation measure. The second evaluation method is chosen as the basis for the optimisation since the choice of maximal action limit allows an explicit choice of the type of performance that the diffusion schemes are optimised to produce. For a low maximal action limit, the schemes are optimised to produce reasonable good segmentation very fast. For a larger maximal action limit, the schemes do not produce as good segmentations in few actions — however for many actions the quality of the segmentation will be better compared to a scheme optimised with a low maximal action limit.

The choice of action limit 30 bias the diffusion schemes towards a good segmentation fast. The effects of the choice of maximal action limit is investigated in section 15.9.

Evaluation methods 1 or 3 could also be used as the basis for the optimisation method. Even though these methods measure the processing cost required to reach perfect segmentation, it is possibly to have an implicit choice to the bias of the optimisation similar to the choice of maximal action limits. This could be done through the assignment of an appropriate tolerance area. This is not explored further in this thesis.

Another specific choice in the optimisation methods is the fixation of the number of scale levels. In section 15.8 the effect of increasing the number of scale levels is analysed further.

9.5 Alternative Optimisation Methods

A number of different optimisation schemes have been tested. The difficulty with the presence of numerous local minima is well-known in optimisation. A standard gradient descent method will often perform well if the step sizes are changed during the process. A common remedy is large step sizes during the first few iterations of optimisation followed by smaller step sizes in the later fine-tuning optimisation iterations. This procedure does not produce satisfying results with the present optimisation problem. Obviously, the proposed optimisation followed by repeated use with smaller step sizes could possible produce better results. However, this is not practically possible due to the computational requirements of the optimisation method.

The proposed method is quite slow and meticulous compared to regular gradient descent methods. The approach sacrifices speed in order to get the desired robustness.

It should be noted that speed is literally "sacrificed". A method with four parameters requires the generation and evaluation of 16 sets of gradient scale-spaces for each optimisation round. A typical optimisation requires 3 rounds — approximately 50 scale-spaces are generated. This is done for 5 slices from each of the 8 chosen ground truth data sets with a total of 2000 scale-spaces for the optimisation of one diffusion scheme with respect to all data sets. The generation of one scale-space with 30 levels requires between 5 for linear diffusion and 15 minutes for true anisotropic nonlinear diffusion (on a HP9000/K460 or a HP9000/J2240).

This means that the optimisation of the parameters for one single diffusion method requires approximately 2 two weeks of computation time on average!

9.6 The Importance of Good Parameter Choices

It is virtually impossible to choose the optimal parameters for a diffusion scheme with respect to a given evaluation data set. The evaluation function is not suited for efficient optimisation schemes and the computational time required is too large for an exhaustive search of the parameter space.

However, the possibility for generation of good parameters is important for a number of reasons:

- The diffusion schemes must perform well in order to make the evaluation of their performance fair.
- The comparison between the diffusion schemes is not valid if the different schemes do not perform well.
- A segmentation tool will only perform well if the underlying method can be tuned to the relevant type of image data.

Fortunately, the stability of the method with respect to the parameter choices saves the day. The differences in performance with different reasonable parameters sets are insignificant.

Part III

Alternative Diffusion Schemes

Everybody thinks of changing humanity and nobody thinks of changing himself.
Leo Tolstoy

Chapter 10

Change of Scale-Space

Part One introduces a multi-scale watershed segmentation method. The method supplies the user with building blocks of different sizes that can be used for construction of segments. The constructed segments are supposed to model the objects in the images. The method is based on linear diffusion of the image followed by gradient magnitude watershed segmentation. The resulting catchment basins are linked across scale. This allows the regions at detection scale to be replaced by the regions they originate from at localisation scale.

The linear diffusion simplifies the image — this implies an implicit ranking of the watersheds. The "weak" watersheds are annihilated as scale increases, and the corresponding catchment basins are merged. At very low scale the individual regions are typically too small to be very useful for construction of the desired objects. At a specific level the regions are supposed to match well with image structures of the corresponding size. Whether the merging due to diffusion results in good regions at higher levels depends on the diffusion of the image.

The theoretical properties of linear diffusion are very nice. However, this does not imply that the corresponding ranking of the watersheds lead to applicable building blocks at the various levels in the scale-space hierarchy. It is therefore tempting to investigate the effect of changing the diffusion scheme.

It is not obvious that the method will work properly with another diffusion scheme:

- The calculation of the gradient depends on the Gaussian derivatives of linear diffusion.
- The catchment basins may have other generic events with other diffusion schemes.
- The simplifying properties of the linear diffusion are needed in order to achieve the effect of increasing size in building blocks with increasing scale level.

10.1 Calculation of the Gradient

Gradient magnitude watershed segmentation requires calculation of the gradient. Gaussian derivatives provide a theoretically founded gradient defined with respect to a specific scale.

Other diffusion schemes may not have an equally nice derivative operation. However, this problem can easily be solved. Gaussian derivatives can simply still be used.

An alternative diffusion scheme results in a scale-space of the original image. A gradient scale-space can be calculated with Gaussian derivatives at each scale level. This requires some fixed scale for the derivative operation — which implies a regularisation (in terms of a linear diffusion) of the alternative scale-space.

Therefore, an extra parameter is introduced to each diffusion scheme: the regularisation scale.

10.2 Linking Scheme

In chapter 4 the generic events for the catchment basins are derived. Section 5.1 shows the linking of catchment basins across scale that corresponds to these events. However, the linking scheme does not take advantage of this knowledge. The robust region matching is independent of the actual *merge*, *split*, *annihilation*, and *creation* events.

Therefore, it is not strictly needed to derive the generic events for the catchment basins for each alternative diffusion scheme. Nevertheless, the generic events for the catchment basins provide insight in the underlying diffusion process. Other diffusion schemes with simpler evolution possibly allow more efficient implementations of the linking. In contrast, the catastrophe theory analysis of an alternative diffusion scheme could show that the resulting catchment basins were unstable and therefore unsuitable.

However, the bottom line is that the existing method will create intuitive linkings for any reasonable diffusion scheme (as explained in section 5.1).

10.3 Scale-space Properties

A central property of linear scale-space is the general simplification of the image (theoretically founded in the causality axiom). This ensures that regions are in general merged into larger regions as scale increases. Thereby, it supports the intuitive notion of building blocks of different sizes.

In general, alternative diffusion schemes do not have a well-defined "scale" concept. Therefore the sizes of the building blocks will not change according to the scale level.

Whether this is a problem or not depends on the behaviour of the specific diffusion scheme. The theoretically founded scale concept is not necessary, but the overall simplification with increasing scale level is essential.

10.4 The Referee

The theoretical considerations regarding the applicability of a linking scheme and the necessity for a scale-space-like simplication behaviour are, to some extent, irrelevant. The modularity of the general segmentation method allows the incorporation of any scheme that provides a partitioning of the image into regions. Just about any mechanism that provides a hierarchy of regions can be used.

However, it is a completely different question whether or not a particular scheme is any good. The evaluation is the referee that rules regarding the applicability of the diffusion schemes.

The following chapters present a number of diffusion schemes. The evaluation of these schemes is the purpose of Part Four.

Ethvert ordentligt speciale med respekt for sig selv har et ubrugeligt citat før hvert kapitel.
Erik Dam

Chapter 11

Recursive Filter Linear Scale-Space

With respect to computational requirements, the generation of the gradient magnitude scale-space is by far the most demanding part of the multi-scale watershed segmentation method. The original implementation in [Olsen, 1996] use a fourier domain Gaussian filter implementation.

The Gaussian filter can also be implemented with a recursive filter algorithm. In general, A n'th order recursive filter uses the values calculated at the n previous points to calculate the value at the current point. Practical experiments show that this implementation reduces the computational time to approximately one tenth compared to the original implementation.

Two versions of the recursive algorithm are implemented: a second order and a fourth order. Informal visual inspection offers the general impression that especially the fourth order recursive filter performs as well as the fourier domain implementation. The evaluation method of Part Two allows proper comparative evaluation of the implementations.

The theoretical foundation for the recursive filtering technique is beyond the scope of this thesis. The following presentation is therefore somewhat superficial with the focus on implementation — for more detail see [Deriche, 1990] and [Deriche, 1993].

The goal of this treatment is to arrive at a recursive filter algorithm for the calculation of the gradient magnitude at a given scale using Gaussian derivatives.

11.1 1D Recursive Filter Basics

For a discrete signal $x(i)$ (where $i = 1, \ldots, N$) the convolution with a causal filter h of width W is defined:

$$y(i) = \sum_{k=0}^{W-1} h(k)\ x(i-k) \qquad \text{for } i = 1, \ldots, N \tag{11.1}$$

The convolution filter is causal since the convolution at a point only depends on previous values.

This convolution filter can be approximated by a recursive system of order n:

$$y(i) = \sum_{k=0}^{m-1} \beta_k \; x(i-k) \; - \; \sum_{k=1}^{n} \alpha_k \; y(i-k) \qquad \text{for } i = 1, \ldots, N \tag{11.2}$$

For a n order system, the value $y(i)$ is determined by the n previous y values and the m previous x values. For the filters presented here $m = n$.

The constants $\alpha_1, \ldots, \alpha_n$ and $\beta_0, \ldots, \beta_{m-1}$ are determined such that the least square error of the filter response is minimised (for filter design details see [Deriche, 1993]).

11.1.1 Causal and Anti-Causal Filter

In general, convolution filters are not causal — $y(i)$ depends on the x values both to the left and the right. Therefore the convolution filter h is split into two — one with the causal part h^+ and one with the anti-causal part h^-:

$$\begin{aligned} h &= h^+ + h^- \\ h^+(k) &= \begin{cases} h(k) & for k \geq 0 \\ 0 & otherwise \end{cases} \\ h^-(k) &= \begin{cases} h(k) & for k < 0 \\ 0 & otherwise \end{cases} \end{aligned}$$

A recursive filter is then designed for each of these convolution filters. The anti-causal part is transformed into a causal filter in the process. Where the recursive filter corresponding to the causal part is applied left to right; the recursive filter corresponding to the anti-causal part is applied right to left. The result of applying the causal part is denoted y^+ and the result of applying the anti-causal part is denoted y^-.

11.1.2 The Recursive Filter Algorithm

The complete recursive filter algorithm can then be written for a fourth order filter:

For $k = 1, \ldots, N$:

$$\begin{aligned} y^+(k) &= n_0^+ x(k) + n_1^+ x(k-1) + n_2^+ x(k-2) + n_3^+ x(k-3) \\ &\quad -d_1^+ y(k-1) - d_2^+ y(k-2) - d_3^+ y(k-3) - d_4^+ y(k-4) \end{aligned}$$

For $k = N, \ldots, 1$:

$$\begin{aligned} y^-(k) &= n_1^- x(k-1) + n_2^- x(k-2) + n_3^- x(k-3) + n_4^- x(k-4) \\ &\quad -d_1^- y(k-1) - d_2^- y(k-2) - d_3^- y(k-3) - d_4^- y(k-4) \end{aligned}$$

For $k = 1, \ldots, N$:

$$y(k) = y^+(k) + y^-(k)$$

11.1.3 Filter Constants

The definitions for the constants $n_0^+, \ldots, n_3^+, d_1^+, \ldots, d_4^+, n_1^-, \ldots, n_4^-, d_1^-, \ldots, d_4^-$ are needed in the algorithm above. Due to the intricate details of the filter design they are written in terms of another set of constants $a_0, a_1, b_0, b_1, c_0, c_1, w_0, w_1, \sigma$ in the following manner. First the constants for the causal part of the filter:

$$
\begin{aligned}
n_0^+ &= a_0 + c_0 \\
n_1^+ &= \exp\left(\frac{-b_1}{\sigma}\right)\left(c_1 \sin\left(\frac{w_1}{\sigma}\right) - (c_0 + 2a_0)\cos\left(\frac{w_1}{\sigma}\right)\right) \\
&\quad + \exp\left(\frac{-b_0}{\sigma}\right)\left(a_1 \sin\left(\frac{w_0}{\sigma}\right) - (2c_0 + a_0)\cos\left(\frac{w_0}{\sigma}\right)\right) \\
n_2^+ &= 2\exp\left(\frac{-b_0 - b_1}{\sigma}\right)\left((a_0 + c_0)\cos\left(\frac{w_1}{\sigma}\right)\cos\left(\frac{w_0}{\sigma}\right) - \cos\left(\frac{w_1}{\sigma}\right) a1 \sin\left(\frac{w_0}{\sigma}\right)\right. \\
&\quad \left. - \cos\left(\frac{w_0}{\sigma}\right) c_1 \sin\left(\frac{w_1}{\sigma}\right)\right) + c_0 \exp\left(\frac{-2b_0}{\sigma}\right) + a_0 \exp\left(\frac{-2b_1}{\sigma}\right) \\
n_3^+ &= \exp\left(\frac{-b_1 - 2b_0}{\sigma}\right)\left(c_1 \sin\left(\frac{w_1}{\sigma}\right) - \cos\left(\frac{w_1}{\sigma}\right) c_0\right) \\
&\quad + \exp\left(\frac{-b_0 - 2b_1}{\sigma}\right)\left(a_1 \sin\left(\frac{w_0}{\sigma}\right) - \cos\left(\frac{w_0}{\sigma}\right) a_0\right) \\
d_1^+ &= -2\exp\left(\frac{-b_1}{\sigma}\right)\cos\left(\frac{w_1}{\sigma}\right) - 2\exp\left(\frac{-b_0}{\sigma}\right)\cos\left(\frac{w_0}{\sigma}\right) \\
d_2^+ &= 4\cos\left(\frac{w_1}{\sigma}\right)\cos\left(\frac{w_0}{\sigma}\right)\exp\left(\frac{-b_0 - b_1}{\sigma}\right) + \exp\left(\frac{-2b_1}{\sigma}\right) + \exp\left(\frac{-2b_0}{\sigma}\right) \\
d_3^+ &= -2\cos\left(\frac{w_0}{\sigma}\right)\exp\left(\frac{-b_0 - 2b_1}{\sigma}\right) - 2\cos\left(\frac{w_1}{\sigma}\right)\exp\left(\frac{-b_1 - 2b_0}{\sigma}\right) \\
d_4^+ &= \exp\left(\frac{-2b_0 - 2b_1}{\sigma}\right)
\end{aligned}
\tag{11.3}
$$

For a symmetrical convolution filter (such as the Gaussian), the constants for the anti-causal part of the filter can be derived from the constants for the causal part of the filter:

$$
\begin{aligned}
d_1^- &= d_1^+ \\
d_2^- &= d_2^+ \\
d_3^- &= d_3^+ \\
d_4^- &= d_4^+ \\
n_1^- &= n_1^+ - d_1^+ n_0^+ \\
n_2^- &= n_2^+ - d_2^+ n_0^+ \\
n_3^- &= n_3^+ - d_3^+ n_0^+ \\
n_4^- &= -d_4^+ n_0^+
\end{aligned}
\tag{11.4}
$$

In the description above, the only requirement on the convolution filter is that it is symmetrical. The definitions of the constants $a_0, a_1, b_0, b_1, c_0, c_1, w_0, w_1, and \sigma$ determine the specific filter.

11.2 1D Recursive Filter for Gaussian

The following set of constants defines a fourth order recursive filter that approximates a Gaussian convolution filter:

$$\begin{aligned} a_0 &= 1.68 \\ a_1 &= 3.735 \\ b_0 &= 1.783 \\ b_1 &= 1.723 \\ c_0 &= -0.6803 \\ c_1 &= -0.2598 \\ w_0 &= 0.6318 \\ w_1 &= 1.997 \end{aligned} \tag{11.5}$$

The second order recursive filter is a simplified version of the fourth order filter where some of the terms are removed. The following set of constants takes care of this and defines a second order recursive filter that approximates a Gaussian convolution filter:

$$\begin{aligned} a_0 &= 0.9629 \\ a_1 &= 1.942 \\ b_0 &= 1.26 \\ w_0 &= 0.8448 \\ b_1 &= 0 \\ c_0 &= 0 \\ c_1 &= 0 \\ w_1 &= 0 \end{aligned} \tag{11.6}$$

This leaves one symbol in the filter definition undefined. This σ is actually the desired standard deviation of the Gausssian. Note that the filter implementation is independent of σ. Therefore, a large scale filter does not require more computation.

11.3 1D Recursive Filter for First Derivative of Gaussian

The first derivative of the Gaussian is another convolution filter that can be implemented by the general recursive filter. However, the first derivative of the Gaussian is not symmetrical but anti-symmetrical. This influences the constants for the anti-causal part of the filter as follows:

$$
\begin{aligned}
d_1^- &= d_1^+ \\
d_2^- &= d_2^+ \\
d_3^- &= d_3^+ \\
d_4^- &= d_4^+ \\
n_1^- &= -\left(n_1^+ - d_1^+ n_0^+\right) \\
n_2^- &= -\left(n_2^+ - d_2^+ n_0^+\right) \\
n_3^- &= -\left(n_3^+ - d_3^+ n_0^+\right) \\
n_4^- &= d_4^+ n_0^+
\end{aligned} \tag{11.7}
$$

The following set of constants defines a fourth order recursive filter that approximates a convolution filter for the first derivative of a Gaussian:

$$
\begin{aligned}
a_0 &= -0.6472 \\
a_1 &= -4.531 \\
b_0 &= 1.527 \\
b_1 &= 1.516 \\
c_0 &= 0.6494 \\
c_1 &= 0.9557 \\
w_0 &= 0.6719 \\
w_1 &= 2.072
\end{aligned} \tag{11.8}
$$

The following set of constants defines a second order recursive filter that approximates a convolution filter for the first derivative of a Gaussian:

$$
\begin{aligned}
a_0 &= -0.1051 \\
a_1 &= 1.898 \\
b_0 &= 0.9338 \\
w_0 &= 0.9459 \\
b_1 &= 0 \\
c_0 &= 0 \\
c_1 &= 0 \\
w_1 &= 0
\end{aligned} \tag{11.9}
$$

Again, the symbol σ in the recursive filter implementation is the scale at which the Gaussian derivative is to be calculated.

11.4 N-Dimensional Generalisation

In the treatment above, the recursive filter is one-dimensional. The extension to higher dimensions is quite simple due to the seperability of the Gaussian. In the following the algorithm for the two-dimensional case is stated explicitly. The generalisation to higher dimensions is straightforward.

Since the convolution operation is separable, the two-dimensional filtering can be performed as a filtering operation in the first coordinate direction followed by a filtering operation in the second coordinate direction. The convolution of the image I can therefore be separated (where the two-dimensional convolution filter is denoted F and the one-dimensional filters in the first and second coordinate directions are denoted F^i and F^j): $F * I = F^i * F^j * I$

For the discrete two-dimensional image $x(i,j)$ (where $i = 1,\ldots,N$ and $j = 1,\ldots,M$) the convolution with a Gaussian filter is performed as follows, where the constants $n_0^+,\ldots,n_3^+$, $d_1^+,\ldots,d_4^+$, $n_1^-,\ldots,n_4^-$, $d_1^-,\ldots,d_4^-$ are defined by equations 11.3, 11.4, the desired scale σ, and 11.6 for a second order filter or 11.5 for at fourth order filter:

- For each $j = 1,\ldots,M$ perform filtering in the first coordinate direction:

  For $i = 1,\ldots,N$:

$$\begin{aligned} y^+(i,j) &= n_0^+ x(i,j) + n_1^+ x(i-1,j) + n_2^+ x(i-2,j) + n_3^+ x(i-3,j) \\ &\quad - d_1^+ y(i-1,j) - d_2^+ y(i-2,j) - d_3^+ y(i-3,j) - d_4^+ y(i-4,j) \end{aligned}$$

  For $i = N,\ldots,1$:

$$\begin{aligned} y^-(i,j) &= n_1^- x(i-1,j) + n_2^- x(i-2,j) + n_3^- x(i-3,j) + n_4^- x(i-4,j) \\ &\quad - d_1^- y(i-1,j) - d_2^- y(i-2,j) - d_3^- y(i-3,j) - d_4^- y(i-4,j) \end{aligned}$$

  For $i = 1,\ldots,N$:

$$y(i,j) = y^+(i,j) + y^-(i,j)$$

- For each $i = 1,\ldots,N$ perform filtering in the second coordinate direction on the result of the first filtering operation:

  For $j = 1,\ldots,M$:

$$\begin{aligned} z^+(i,j) &= n_0^+ y(i,j) + n_1^+ y(i,j-1) + n_2^+ y(i,j-2) + n_3^+ y(i,j-3) \\ &\quad - d_1^+ z(i,j-1) - d_2^+ z(i,j-2) - d_3^+ z(i,j-3) - d_4^+ z(i,j-4) \end{aligned}$$

  For $j = M,\ldots,1$:

$$\begin{aligned} z^-(i,j) &= n_1^- y(i,j-1) + n_2^- y(i,j-2) + n_3^- y(i,j-3) + n_4^- y(i,j-4) \\ &\quad - d_1^- z(i,j-1) - d_2^- z(i,j-2) - d_3^- z(i,j-3) - d_4^- z(i,j-4) \end{aligned}$$

  For $j = 1,\ldots,M$:

$$z(i,j) = z^+(i,j) + z^-(i,j)$$

The filters that approximate the two-dimensional partial derivatives of the Gaussian filter are equally simple. The derivation rules for convolution allow the derivation to be isolated to the relevant one-dimensional convolution filter (where F, F^1, F^2, and I are defined as previously):

$$\frac{\partial F * I(i,j)}{\partial i} = \frac{\partial F^i}{\partial i} * F^j * I \tag{11.10}$$

$$\frac{\partial F * I(i,j)}{\partial j} = F^i * \frac{\partial F^j}{\partial j} * I \tag{11.11}$$

This means that the recursive filter algorithm that calculates the partial derivatives of the Gaussian for a two-dimensional image is the same as the algorithm for the calculation of the two-dimensional Gaussian. The only difference is that the constants $n_0^+, \ldots, d_4^-$ are defined according to the derivative filter.

For instance, in the calculation of the partial derivative with respect to the first coordinate the constants in the first part of the algorithm (that performs the filtering in the first coordinate direction) are defined by equations 11.3, 11.7 and 11.9 for a second order filter or 11.8 for at fourth order filter. In the second part of the algorithm the constants are defined by 11.3, 11.4, and 11.6 for a second order filter or 11.5 for at fourth order filter.

11.5 Boundary Condition

In the algorithms above, the indices are allowed to index points outside the image. These outside points are simply defined as zero in [Deriche, 1990] and [Deriche, 1993] . However, this leads to noticeable boundary effects. The filter response tends to be quite erratic near the borders.

A simple heuristic is used to minimise this problem. Before the recursive filter is applied, the image is enlarged with a margin of 8 pixels. The boundary values of the image are extended into this margin. The filtering algorithm is then performed on the enlarged image. A substantial part of the erratic behaviour is then located in the margin (which is removed again before the result is used for further operations).

It should be noted that a certain boundary effect is unavoidable, and that there is no "correct" behaviour at the boundaries. The physical properties of the heat diffusion equation can inspire different boundary conditions to be imposed on the algorithm. A natural condition is that the boundary is completely insulating — this means that there is no flux across the borders. The heuristic above is a crude approximation of this boundary condition.

Another common boundary condition implies that the image is perceived as continuing symmetrically across the border. This is a natural boundary condition for a fourier domain algorithm.

The heuristic above is simply chosen because the resulting watersheds are visually more appealing than the ones from the original algorithm.

11.6 Normalisation

The recursive filters above implement an approximation of the Gaussian defined as:

$$G(i) = \exp\left(\frac{-i^2}{2\sigma^2}\right)$$

This definition is not normalised (compare with 3.3 on page 13). In [Deriche, 1993] the normalisation process is treated.

However, in this context the recursive filters are used as preparation for the watershed segmentation. Since the watershed segmentation is invariant to a scaling of the image, the normalisation is not necessary. Therefore, the unnormalised recursive filter is used.

11.7 Summary

The purpose of this chapter is to present a recursive filter algorithm that can be used to approximate the Gaussian gradient calculation. The presentation above describes how the partial derivatives are calculated. Thereby the gradient magnitude can be calculated. The algorithm can be extended to any image dimension.

The computational complexity of the algorithm is not treated. The strength of the algorithm is that the number of computations is independent of the scale. For a given image, a direct convolution would require $O(scale)$ operations — the recursive filter algorithm requires $O(order)$ operations.

Practical experiments show that the recursive filter implementation is approximately ten times faster than a fourier domain convolution implementation.

The evaluation of the recursive implementation has practical implications since the segmentation method described in Part One is realised under the research project "Computing Natural Shape" [Johansen et al., 1999].

In the following the second order recursive filter diffusion scheme is denoted R2; the fourth order scheme is denoted R4.

"schwarmerei" : excessive or unwholesome sentiment
In 1845, the editors of the Edinburgh Review felt compelled to use the German "schwarmerei" to describe fanatical enthusiasm because the concept seemed so foreign to them. In commenting on the writings of German critic and dramatist Gotthold Lessing, they declared "schwarmerei" to be "untranslatable, because the thing itself is un-English."
Merriam-Webster's Dictionary, Word of the day, January 14., 2000

Chapter 12

Anisotropic Nonlinear Diffusion

Linear scale-space has appealing noise-reduction and simplification properties. However, the linearity requires all parts of the image to undergo the same smoothing. Therefore the salient features of the image are blurred and dislocated as well.

Nonlinear diffusion allows the diffusion to depend locally on the underlying image. Thereby desired features can be preserved through the diffusion. This is used for edge detection in [Perona & Malik, 1990]. The approach is extended in [Nitzberg & Shiota, 1992] to an anisotropic diffusion scheme with improved results at corners.

Another important application for nonlinear diffusion is preprocessing for segmentation (examples are [Chung & Sapiro, 1999], [Niessen et al., 1997], [Weickert, 1998c], [Vincken, 1995], and [Weickert, 1998b]).

Weickert has generalized these approaches as *Anisotropic Nonlinear Diffusion* ([Weickert, 1998a]).

12.1 Diffusion

In the realm of physics, diffusion can be perceived as the random walk of an ensemble of particles from regions of high concentration to regions of lower concentration. If the concentration of particles is regarded as a function, this process causes a smoothing of the function.

With regards to watershed segmentation, a diffusion process can be explained quite intuitively. A function can be viewed as a height map of some landscape. Diffusion is then analogous to an erosion process where the soil flows downhill over time. This erosion process eventually transforms any hilly landscape into a flat plain.

The central notions in these processes are the laws that define the flow and the preservation of matter. These can be formulated mathematically in the following manner.

A way to describe a flow function is by a generalisation of *Fick's law*[1]. This law defines the flow of matter — the *flux* j — in terms of a *diffusion tensor* D and the gradient ∇U (where U is a function that defines the concentration of matter, or the height in the landscape analogy):

$$j = -\ D\ \nabla U \tag{12.1}$$

The preservation constraint can be formulated through[2]:

$$\frac{\partial U}{\partial t} = -\ div\ j \tag{12.2}$$

Intuitively, this equation simply states that the local height change is the difference in flow to and from the point. The general anisotropic diffusion equation emerges from substitution of the expression for the flux (equation 12.1) into the preservation equation (equation 12.2):

$$\frac{\partial U}{\partial t} = div(D\ \nabla U) \tag{12.3}$$

This diffusion equation is worth studying in detail. The nature of the diffusion depends on the diffusion tensor. The following examples begin with the simplest diffusion tensor and end with the most general:

1. The scalar value 1 (or the unity matrix) is the simplest diffusion tensor. With this diffusion tensor, the diffusion equation becomes the ordinary heat diffusion equation $\frac{\partial U}{\partial t} = \Delta U$ (equivalent with equation 3.6). This yields *isotropic linear* diffusion. The diffusion is *isotropic* if the flux is parallel with the gradient, and *linear* if the flux is independent of the underlying image.

2. A scalar valued constant diffusion tensor (or the unity matrix times a scalar value) yields a similar diffusion. The diffusion is the same as in the ordinary Gaussian diffusion, however the pace of the evolution is changed. In the analogy of erosion in a landscape, the scalar valued diffusion tensor intuitively plays the role of the gravity acceleration constant.

3. A scalar valued diffusion tensor function (or the unity matrix times a diffusivity function) results in isotropic *nonlinear* diffusion. Depending on the chosen function this can reverse, fixate or speed up the diffusion in areas with certain properties. Sections 12.5.2 and 12.5.3 contain examples of diffusivity functions.

[1] Fick's law is equivalent to the first law of diffusion. Here the diffusion tensor D is a diffusion constant determined by the characteristics of the dissolved material.

[2] The divergence $div\ j$ for $j \in C(\mathbb{R}^n, \mathbb{R})$ is defined as:
$div\ j = \nabla \bullet j = \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix} \bullet j = \frac{\partial j_1}{\partial x_1} + \cdots + \frac{\partial j_n}{\partial x_n}$
For comparison the gradient ∇j for $j \in C(\mathbb{R}^n, \mathbb{R})$ is defined as:
$\nabla j = \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix} j = \begin{pmatrix} \frac{\partial j}{\partial x_1} & \cdots & \frac{\partial j}{\partial x_n} \end{pmatrix}$

4. In the general case the diffusion tensor is a matrix function. This results in *anisotropic nonlinear* diffusion.

The presentation above is somewhat intuitive. In section 12.4, the general anisotropic diffusion equation is defined properly.

12.2 The Diffusion Tensor

The effect of the diffusion tensor in the general case is best understood through the flux function $j = -D\,\nabla U$. Intuitively, the flux is a vector that describes the direction and the size of the flow of matter. The diffusion tensor is a mapping from the gradient to the flux. The diffusion tensor is required to be a positive definite symmetric matrix. This requirement ensures the properties following in section 12.4.

However at this point, the positive definiteness simply implies that the diffusion tensor can be characterized by its eigenvectors and corresponding eigenvalues. Let $\mathbf{v}_1, \ldots, \mathbf{v}_i$ be the eigenvectors and $\lambda_1, \ldots, \lambda_i$ be the corresponding eigenvalues ordered such that $0 < \lambda_1 \leq \ldots \leq \lambda_i \leq 1$. The following sections reveal that a sensible design choice is $\mathbf{v}_1 \parallel \nabla U_\sigma$. Here ∇U_σ is the gradient evaluated at Gaussian scale σ.

For $\sigma = 0$ the effect of this diffusion tensor is quite simple. Then $j = -D\,\nabla U = -\lambda_1\,\nabla U$ since $\nabla U = \nabla U_0$ is an eigenvector with eigenvalue λ_1. This is isotropic diffusion with diffusivity function λ_1.

For $\sigma > 0$ the diffusion tensor affects the direction of the flux and the diffusion becomes anisotropic. Since this behaviour is crucial it is investigated in full detail in two dimensions. The eigenvectors for the diffusion tensor are then $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$. In this orthogonal basis, the gradient ∇U is written as a linear combination $\nabla U = c_1\mathbf{v}_1 + c_2\mathbf{v}_2$. The flux is then

$$j = -D\,\nabla U = -(\lambda_1 c_1 \mathbf{v}_1 + \lambda_2 c_2 \mathbf{v}_2)$$

The direction and size of the flux is determined by two factors:

- How well aligned ∇U is with ∇U_σ
- The design of the eigenvalues λ_1 and λ_2

The first factor is determined by the geometry of the image in combination with the choice of σ. For small σ, ∇U and ∇U_σ is approximately aligned and the diffusion is close to isotropic. In noisy parts of the image, ∇U and ∇U_σ are poorly aligned and the diffusion is largely determined by λ_2.

The design of λ_1 and λ_2 decides the degree of anisotropy in the diffusion scheme. For $\lambda_1 = \lambda_2$ the diffusion is isotropic. The diffusion becomes more and more anisotropic the larger λ_2 is relative to λ_1.

In the general formulation, the diffusion is tailored by assignment of suitable pairs of eigenvectors and eigenvalues, that steer the diffusion in the desired directions. Isotropic linear diffusion is simply diffusion straight downhill. This is achieved by restricting the diffusion to the direction of the gradient as shown above. Obviously, more advanced diffusion schemes can be designed — several examples are shown in section 12.5.

In the following, all images considered are two-dimensional.

12.3 The Structure Tensor

The contents of the diffusion tensor is quite unrestricted. However, in order to provide a sensible diffusion, the diffusion tensor must adapt to the local structure of the image. A means of specifying this local structure is the *structure tensor* J_ρ (where M^T is the matrix M transposed):

$$J_\rho(\nabla U_\sigma) = G_\rho * (\nabla U_\sigma \nabla U_\sigma^T) \tag{12.4}$$

The structure tensor uses the gradient ∇U_σ to describe the local structure of the image. The gradient is evaluated at Gaussian scale σ in order to reduce effects of noise — therefore σ is denoted the *noise scale* or the *regularisation scale*. The eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of the matrix $\nabla U_\sigma \nabla U_\sigma^T$ preserve the orientation of the gradient: $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$. The corresponding eigenvalues are $|\nabla U_\sigma|^2$ and 0. Thereby the matrix describes the orientation of the local structure — not the direction.

The matrix $\nabla U_\sigma \nabla U_\sigma^T$ is "averaged" over an area of the image by means of a convolution with a Gaussian with standard deviation ρ — called the *integration scale*. This allows a calibration of the desired size of the image structures that steer the diffusion proces.

This blurring of the matrix $\nabla U_\sigma \nabla U_\sigma^T$ over an area explains the need for orientation information instead of direction information. In an image containing line-like structure, the gradients on each side of the line are approximately opposite. These would cancel each other in the blurring of the matrix $\nabla U_\sigma \nabla U_\sigma^T$ if directions were used instead of orientations. This is discussed further in section 12.5.5.

With the structure tensor included, the anisotropic diffusion equation becomes:

$$\frac{\partial U}{\partial t} = div(\ D(J_\rho(\nabla U_\sigma))\ \nabla U\ ) \tag{12.5}$$

The diffusion tensor can take advantage of the structure information of the structure tensor in various ways. The way that the diffusion tensor preserves or changes the eigenvectors and eigenvalues of the structure tensor is central. The eigenvectors of the structure tensor are denoted $\mathbf{v}_1$ and $\mathbf{v}_2$ and the corresponding eigenvalues are denoted μ_1 and μ_2. For $\rho = 0$, $\mathbf{v}_1 \parallel \nabla U_\sigma$, $\mathbf{v}_2 \perp \nabla U_\sigma$, and $\mu_1 \geq \mu_2$. For $\rho > 0$ it is only approximately true that $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$ since the structure tensor is "averaged" over the integration window.

In the following, the eigenvectors of the diffusion tensor are equivalent to the eigenvectors of the structure tensor. The corresponding eigenvalues are denoted λ_1 and λ_2.

12.4 Diffusion Properties

In the context of the multi-scale watershed segmentation method, two properties of the linear Gaussian scale-space are central:

- The well-defined differential operator.
- The simplification effect (theoretically founded in the causality principle, the average grey level principle, and/or the regularisation principles).

These properties are required of the anisotropic nonlinear diffusion scheme as well. The diffusion equation is properly stated in the following definition.

Definition 11. (Anisotropic Diffusion Equation)
Let $I \in L^\infty(\Omega)$ be a real-valued two-dimensional image defined on the image domain Ω with boundary Γ. The general anisotropic nonlinear scale-space $U(t, \vec{x}) \in C^\infty(\mathbb{R}_0 \times \Omega, \mathbb{R})$ is the solution of the initial boundary value problem

$$\frac{\partial U(t,\vec{x})}{\partial t} = div(\ D(J_\rho(\nabla U_\sigma))\ \nabla U\ ) \quad \text{on} \quad]0,\infty[\ \times\ \Omega \tag{12.6}$$

$$U(0,\vec{x}) = I(\vec{x}) \quad \text{on} \quad \Omega \tag{12.7}$$

$$(D\ \nabla U) \cdot \vec{n} = 0 \quad \text{on} \quad [0,\infty[\ \times\ \Gamma \tag{12.8}$$

Here $\vec{x} \in \Omega$ are the spatial variables and $t \in [0,\infty[$ is the temporal variable. The outer normal is denoted by $\vec{n}$ — thereby equation 12.8 requires that there is no flux across the image border. Furthermore, it is assumed that the diffusion tensor $D \in C^\infty(\mathbb{R}^{2\times 2}, \mathbb{R}^{2\times 2})$ is symmetric and uniform positive definite.

In [Weickert, 1998a] Weickert shows a number of appealing properties for diffusion schemes that meet the definition above. Among these are:

- Existence, uniqueness and regularity of solutions to the differential equations from definition 11.
- The solutions are infinitely differentiable.
- Average grey level invariance.
- Translation and isometry invariance.
- Nonenhancement of local extrema (a causality principle).
- Simplification: for $t \to \infty$ the diffusion process leads to a constant image.

This abundance of theoretically proven properties ensures that the anisotropic nonlinear diffusion scheme is applicable for the multi-scale watershed segmentation method.

It would appear that what we need is some examples of diffusion tensors and some nice illustrations.

12.5 Diffusion Schemes

In the following section some examples of diffusion schemes are presented. The first four schemes use a simplified version of the structure tensor, where the integration scale is zero. This is appropriate when segmentation-like diffusion is desired. The last scheme shows an example of the use of the general structure tensor.

In general, the schemes are formulated through the eigenvectors and eigenvalues for the diffusion tensor.

12.5.1 Isotropic Linear Diffusion

As stated previously, this is the simplest special case of the anisotropic nonlinear diffusion scheme. Linear diffusion is achieved by eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$, $\mathbf{v}_2 \perp \nabla U_\sigma$ and corresponding eigenvalues $\lambda_1 = \lambda_2 = 1$.

This diffusion is equivalent to the original Gaussian diffusion scheme from [Olsen, 1996]. However, the implementation is different. The original Gaussian diffusion used in Part One is implemented in the fourier domain using the Gaussian convolution kernel. This scheme is implemented as a special case of the general anisotropic diffusion scheme. Therefore, the implementation is an iterative numerical solution of the PDE.

12.5.2 Regularised Perona-Malik

In [Perona & Malik, 1990] the classical Perona-Malik nonlinear diffusion scheme is introduced. The diffusion is locally determined by the *diffusivity function* p. This is based on the PDE:

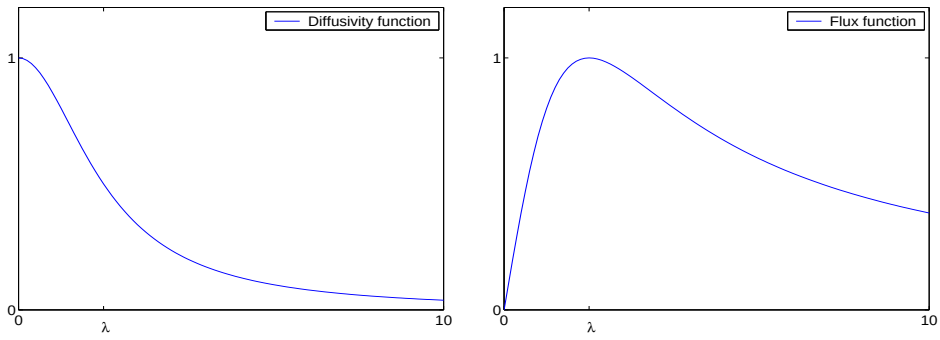
$$\begin{aligned} \frac{\partial U(t,\vec{x})}{\partial t} &= div(\ p(|\nabla U|^2)\ \nabla U\ ) \\ p(|\nabla U|^2) &= \frac{1}{1+\frac{|\nabla U|^2}{\lambda^2}} \end{aligned}$$

This is actually an ill-posed PDE. However, the implicit regularisation in the discretisation ensures robust implementations [Weickert & Benhamouda, 1997]. This regularisation is formalised in [Catté et al., 1992]. Here the gradient $|\nabla U|$ is replaced by a Gaussian smoothed version $|\nabla U_\sigma|$:

$$\begin{aligned} \frac{\partial U(t,\vec{x})}{\partial t} &= div(\ p(|\nabla U_\sigma|^2)\ \nabla U\ ) \\ p(|\nabla U_\sigma|^2) &= \frac{1}{1+\frac{|\nabla U_\sigma|^2}{\lambda^2}} \end{aligned} \tag{12.9}$$

The diffusion scheme defined by equation 12.9 is called Regularised Perona-Malik.

Here, an extra parameter λ is introduced. This is not to be confused with the eigenvalues. Intuitively, the parameter λ determines the gradient magnitude required to protect an edge from diffusion. In order to see this, it is useful to look at the flux magnitude function $j(\nabla U) = p(|\nabla U_\sigma|^2)\ |\nabla U|$ (compare with equation 12.2). The behaviour of the flux function is illustrated for a one-dimensional function in figures 12.1 and 12.2.



a *b*

Figure 12.1: *Diffusivity and flux function for the Perona-Malik isotropic nonlinear diffusion scheme. The flux function is maximal for* $|\nabla U_\sigma| = \lambda$*. This flux function enhances edges with a gradient larger than* λ*. For a comprehensive illustration of this, see figure 12.2.*

12.5.3 Isotropic Nonlinear Diffusion

Weickert replaces the diffusivity function from equation 12.9 with a function w_m that preserves edges better [Weickert, 1998a]. This gives the following diffusion scheme (where $exp(x)$ is e^x):

$$\begin{aligned} \frac{\partial U(t,\vec{x})}{\partial t} &= div(\ w_m(|\nabla U_\sigma|^2)\ \nabla U\ ) \\ w_m(|\nabla U_\sigma|^2) &= \begin{cases} 1 & |\nabla U_\sigma| = 0 \\ 1 - exp\left(\frac{-C_m}{\left(\frac{|\nabla U_\sigma|^2}{\lambda}\right)^m}\right) & |\nabla U_\sigma| > 0 \end{cases} \end{aligned} \tag{12.10}$$

This introduces extra parameters m and C_m. Actually, only one parameter is introduced since C_m is a function of m. The parameter C_m is determined such that the flux magnitude function $j(|\nabla U_\sigma|) = w_m(|\nabla U_\sigma|^2)\ |\nabla U|$ is increasing for $|\nabla U|^2 < \lambda$ and decreasing for $|\nabla U|^2 > \lambda$ — analogously to section 12.5.2.

More specifically, C_m is derived from m as follows. First the derivative of $j(s)$ is derived. The zero-crossing of this derivative reveals the flux magnitude maximum. Therefore the equation $\frac{\partial j(s)}{\partial s} = 0$ is solved where λ is substituted with s^2. This is equivalent to the equation:

$$1 = \exp(-C_m)\ (1 + 2\ C_m\ m) \tag{12.11}$$

The choice $m = 4$ (which implies $C_m = 3.31488$) is used in the examples in [Weickert, 1998a] since they yield visually appealing results. In this thesis $m = 2$ ($C_m = 2.33666$) and $m = 3$ ($C_m = 2.9183$) are used as well. Note that the trivial solution ($C_m = 0$) to equation 12.11 yields linear diffusion.

The diffusion scheme is designed to protect edges more agressively than the diffusion scheme of the regularised Perona-Malik diffusion equation. This effect is enhanced the most for larger m values.

Formulated in terms of eigenvectors and eigenvalues, the diffusion tensor has the usual eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$ with the corresponding eigenvalues $\lambda_1 = \lambda_2 = w_m(|\nabla U_\sigma|^2)$.

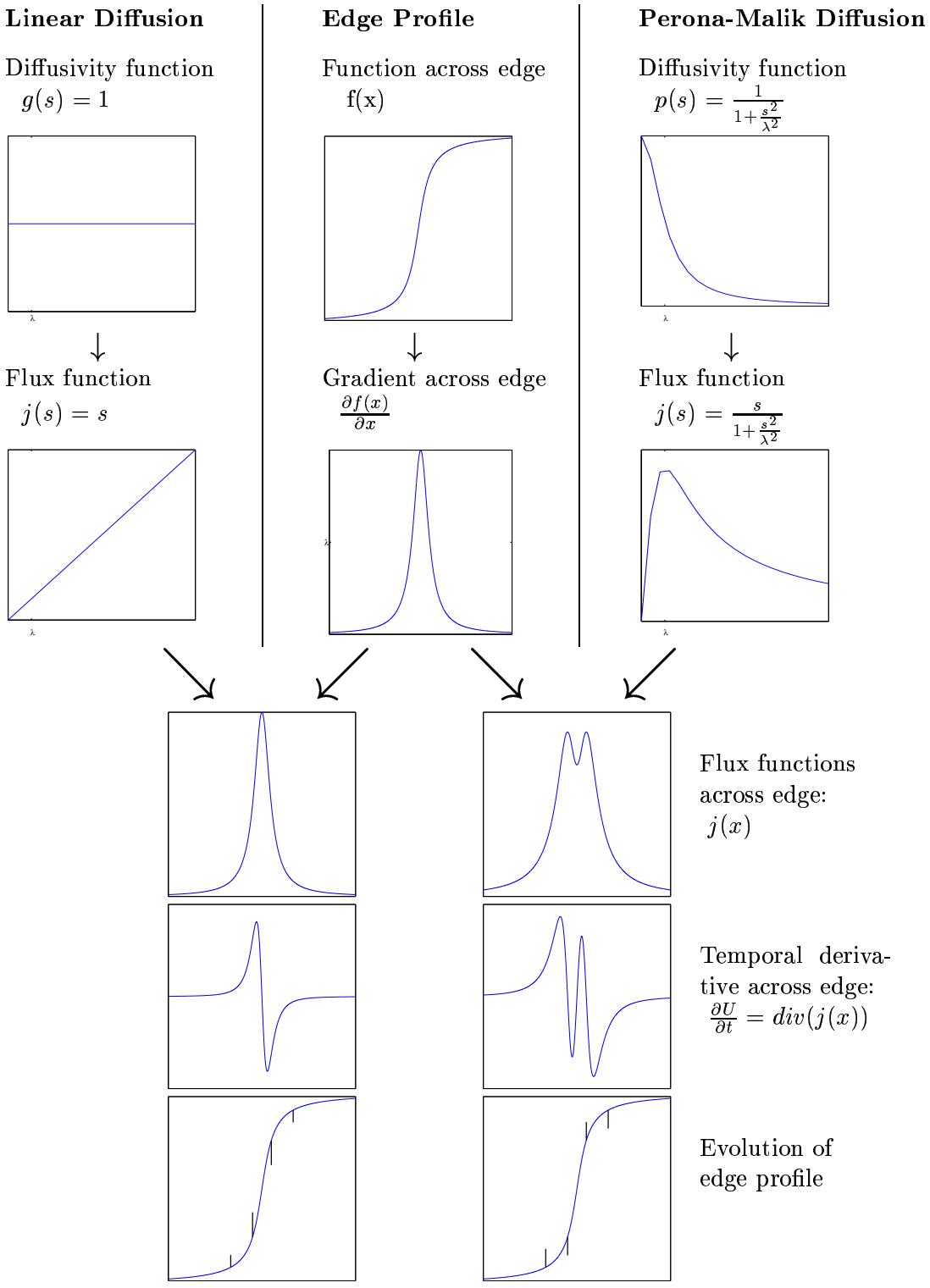


Figure 12.2: *Illustration of the difference between linear and nonlinear diffusion on an artificial function with a nice edge in 1-D (isotropic diffusion is essentially a 1-D process). Linear diffusion smoothes the edge in the ususal fashion. Due to the gradient above λ in the edge profile, the Perona-Malik process performs forward-backward diffusion, which actually enhances the edge. The vertical lines in the bottom row illustrate the direction and approximate size of the evolution at central points on the curve.*

12.5.4 Edge Enhancing Diffusion

The nonlinear schemes limit the diffusion in areas with a large gradient magnitude. This means that the diffusion around edges is minimal. Therefore the noise-reducing properties of diffusion are also lost at these edges. The result of this is illustrated in figure 12.3.

Anisotropic diffusion allows the diffusion to be different in the different directions defined by the local geometry of the image. Therefore, diffusion *across* the edges can be prevented while the diffusion *along* the edges is allowed.

In terms of the landscape analogy, the soil is prevented from eroding down a steep slopes but still allowed to smooth the ground along the edge of the slope. The result of this could be similar to the staircasing of fields known from hilly landscapes.

The formulation by eigenvectors and eigenvalues is similar to the one from the isotropic diffusion scheme in the previous section. The diffusion tensor has the same eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$. The corresponding eigenvalues are $\lambda_1 = w_m(|\nabla U_\sigma|^2)$ and $\lambda_2 = 1$. Here w_m is defined as in equation 12.10. The only difference is that full diffusion is allowed in the $\mathbf{v}_2$ direction — along the isophote.

These eigenvectors and eigenvalues define the diffusion tensor D in the general anisotropic nonlinear diffusion equation:

$$\frac{\partial U(t,\vec{x})}{\partial t} = div(\ D\ \nabla U\ )$$

Note, that for the noise scale σ tending to zero the eigenvector $\mathbf{v}_1$ tends to (be parallel with) ∇U. Therefore, for $\sigma = 0$, the expression $D\ \nabla U$ simply becomes $\lambda_1\ \nabla U$. The eigenvector λ_1 is equal to the diffusivity function from equation 12.10. Therefore, edge enhancing diffusion has the isotropic nonlinear diffusion of equation 12.10 as the limit for σ tending towards zero.

Figure 12.3 shows the convincing performance of the edge enhancing diffusion scheme compared to the previously presented schemes.

12.5.5 Coherence Enhancing Diffusion

The diffusion schemes above are designed to preserve contrast in the image during diffusion. Thereby they are well suited for robust edge detection and segmentation. The following diffusion scheme is specialised towards enhancement of coherent structures. The classical example of images with coherent structures are fingerprint images (see [Almansa & Lindeberg, 1997], [Weickert, 1998a], and [Weickert, 1999]). In these images the line-structures are often unconnected due to errors during the acquisition.

Like previously, the eigenvectors of the structure tensor are denoted $\mathbf{v}_1$ and $\mathbf{v}_2$ and the corresponding eigenvalues are denoted μ_1 and μ_2. Here $\mathbf{v}_1 \parallel \nabla U_\sigma$, $\mathbf{v}_2 \perp \nabla U_\sigma$, and $\mu_1 \geq \mu_2$. The eigenvectors of the diffusion tensor are equivalent to the eigenvectors of the structure tensor with the corresponding eigenvalues λ_1 and λ_2.

The structure tensor, $G_\rho * (\nabla U_\sigma \nabla U_\sigma^T)$, describes the orientation of the image over an area defined by the integration scale ρ. This window of interest is characterised by the eigenvalues. For areas

with little contrast $\mu_1 \approx \mu_2 \approx 0$, for sharp edges $\mu_1 \gg \mu_2 \approx 0$, for corners $\mu_1 \geq \mu_2 \gg 0$. The *local coherence* can be measured by the expression $(\mu_1 - \mu_2)^2$.

A coherence enhancing diffusion scheme is defined by the following eigenvalues for the diffusion tensor:

$$\begin{aligned} \lambda_1 &= \alpha \\ \lambda_2 &= \begin{cases} \alpha & \text{if } \mu_1 = \mu_2 \\ \alpha + (1-\alpha) exp\left(\frac{-C}{(\mu_1 - \mu_2)^{2m}}\right) & \text{otherwise} \end{cases} \end{aligned} \tag{12.12}$$

[Weickert, 1998a] reports satisfying fingerprint enhancement for $C = m = 1$, and $\alpha = 0.001$

As mentioned, coherence enhancing diffusion is not suited for segmentation-like purposes. Therefore it is not used further in this thesis — it is included in this chapter for the sake of completeness.

12.6 Examples

The effect of anisotropic nonlinear diffusion is illustrated for the first four diffusion tensors in figure 12.3:

- The isotropic linear diffusion results in a general blurring of the image where noise is reduced at the cost of severe blurring of the edges.
- This degrading and dislocation of the edges is much reduced in the Perona-Malik diffusion.
- The Weickert nonlinear scheme reduces diffusion at edges much more agressively. This has the disadvantage of preservation of the noise along the edges.
- Edge enhancing diffusion preserves the edges quite well and reduces the noise at the edges at the same time. The anisotropic diffusion along the edges allows the noise to be blurred into oblivion.

12.7 Analysis of the Diffusion Schemes

A lot of work is needed in order to match the analysis of the linear diffusion scheme of Part One.

The nature of the diffusion schemes and their effect on the gradient watersheds could be further explored along the lines of the catastrophe theory in chapter 4. As mentioned in chapter 10, the diffusion schemes can be used in the segmentation method regardless of this. The theoretical properties stated in section 12.4 ensure that the diffusion schemes behave reasonable well in the setting of the multi-scale watershed segmentation method. Whether they actually behave better or worse than the original linear Gaussian diffusion scheme is to be revealed by the evaluation in Part Four.

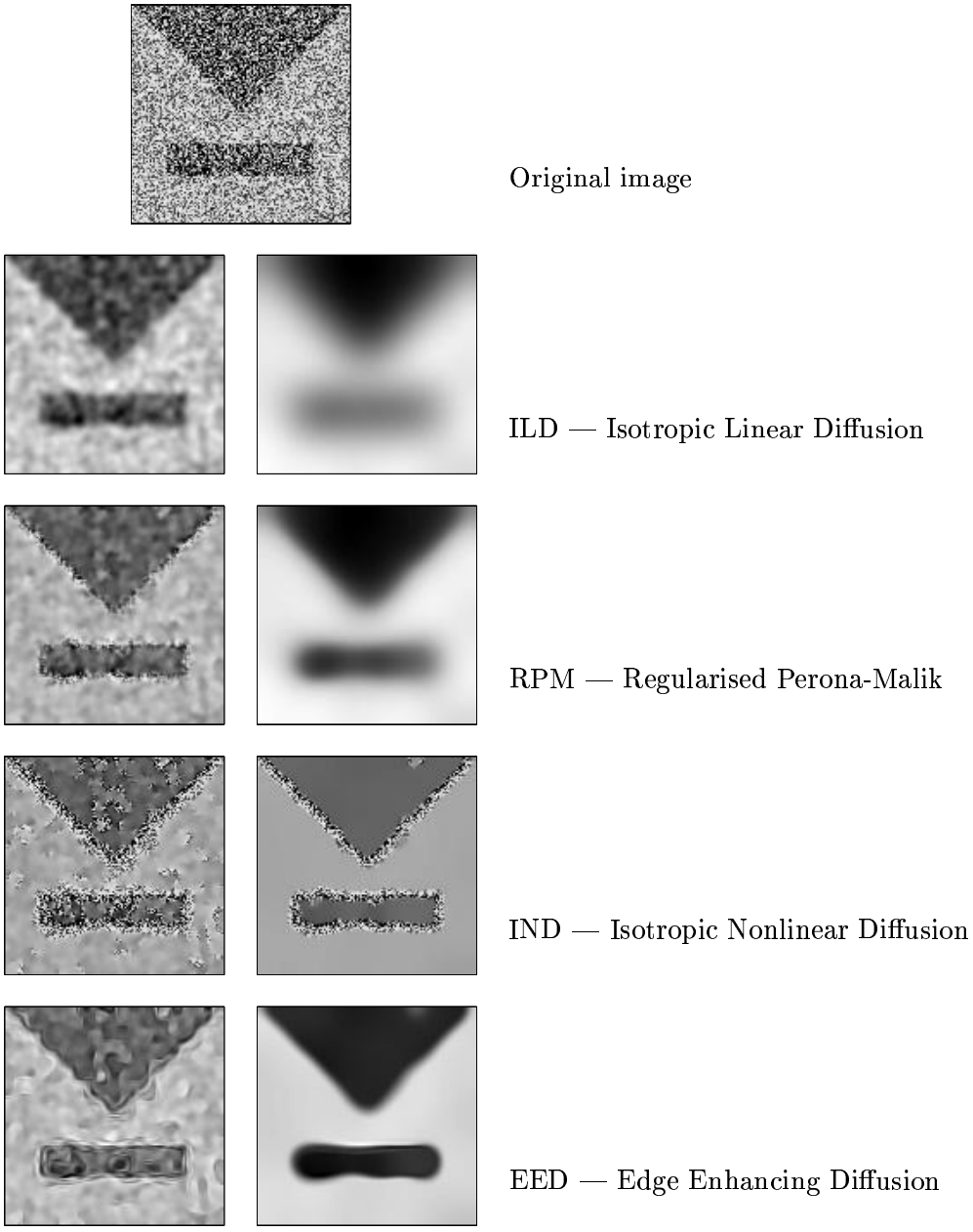


Figure 12.3: *The effect of the various diffusion tensors from [Weickert, 1998a] is demonstrated on an example image. The original image is a 128 by 128 image of a triangle and a rectangle where 70% of the pixels are degraded by noise. Each diffusion is illustrated at $t = 2$ and $t = 80$. The parameter choices are $\sigma = 3$ for all schemes and $\lambda = 3.5$ and $m = 4$ for IND and EED. An explicit scheme time discretisation with time step 0.2 is used for the illustrations.*
Note that the time t for the isotropic linear diffusion corresponds to the variance — not the standard deviation — of the Gaussian.
The original image is obtained from J. Weickert (but originates from the `MegaWawe` *package).*

12.8 The Scale Concept

The linear diffusion equation has the Gauss distribution as Green's function. Therefore, the standard deviation of the Gaussian gives a nice notion of a region of influence. The size of this region is the *scale*.

Nonlinear diffusion schemes do not have an equally intuitive notion of scale. In some areas the diffusion is large, in other areas the diffusion is minimal. However, it is a useful observation that linear diffusion gives an upper bound for the diffusion of the presented diffusion schemes (inspired by [Niessen et al., 1997]). Areas without much contrast will undergo a diffusion approximately equivalent to linear diffusion, whereas areas with higher contrast will experience less diffusion.

Note that the time t from the anisotropic nonlinear diffusion equation is not comparable with the Gaussian scale. In the special case of isotropic linear diffusion, the time t corresponds to the *variance* — not the standard deviation — of the Gaussian.

The "synchronisation of scale" across different diffusion schemes can also be inspired by information theory. In this approach "scale" is defined in terms of the amount of information in the image. Examples of this approach are [Oomes & Snoeren, 1996], [Sporring, 1996], and [Niessen et al., 1997]. In the multi-scale watershed segmentation context, the number of catchment basins could be a relevant indication of the amount of information left in the image at a given scale. However, this approach is also problematic since the number of catchment basins can increase with increasing diffusion.

Synchronisation of scale is not necessary for the evaluation of the diffusion schemes and therefore beyond the scope of this thesis.

12.9 Implementation

In section 12.4 some theoretically proven properties of anisotropic nonlinear diffusion are stated. It is worth mentioning that Weickert also proves analogous theoretical results for the discrete version of the anisotropic diffusion equation [Weickert, 1998a].

The implementations of the anisotropic diffusion schemes are modifications of code by Joachim Weickert. The implementations use the explicit scheme for temporal discretisation (a time step of 0.2 is used throughout this thesis). This implementation has been compared by an implementation based on an implicit scheme (made from scratch) with no significant differences.

12.10 Eureka?

The examples show that anisotropic nonlinear diffusion is capable of preserving the edges during diffusion better than linear Gaussian diffusion.

Edge enhancing diffusion, in particular, seems very well suited for a segmentation method. Whether this apparently appealing diffusion scheme performs better in this particular segmentation method is revealed in Part Four — what a cliff-hanger!

Kært barn har mange navne.
Danish proverb

Chapter 13

Curvature Motion

The presented diffusion processes smooth images — these schemes reduce noise and allow gradually larger structures to reveal themselves.

The goals of morphological processes are similar. However, the focus of the process is the shape of the objects instead of the image function. In the classical approach binary images are considered — the shape is then simply the contour that divides the white and black areas. These approaches have been generalised to cover greyscale images.

Diffusion processes and morphological methods are becoming more connected as research reveal similarities in axiomatics, properties and applications. A number of schemes have been shown to be equivalent [Alvarez & Morel, 1994].

A central example of a morphological process is the *Mean Curvature Motion* — this will be presented in the following.

13.1 Mean Curvature Motion

Let a simple, closed curve $\gamma \in C^\infty(\mathbb{R} \times \mathbb{R}, \mathbb{R}^n)$ be defined with the natural parameter (the arc length) as first parameter and an evolution parameter as second parameter. Let this curve be subjected to *Normal Curvature Flow*. Each point evolves in the normal direction with a velocity given by the signed curvature of the curve (where κ is the curvature and $\vec{n}$ is the inwards normal):

$$\frac{\partial \gamma(p,t)}{\partial t} = \kappa(p,t)\ \vec{n}(p,t) \tag{13.1}$$

This curve evolution has an extremely nice simplifying effect on the curve described by Grayson's theorem: The curve collapses smoothly to a point. In the process any nonconvex curve becomes convex. Even further, if the curve is enclosed by a circle with radius σ then the curve collapses to a point in time $t \leq \frac{1}{2}\sigma^2$.

This normal curvature flow is the fastest way to shrink the euclidean arc length of the curve $\int \left|\frac{\partial \gamma}{\partial p}\right| dp$. Therefore, the flow is also termed *Euclidean Shortening Flow*.

Yet another equivalent formulation of the flow is the *Geometric Heat Equation*:

$$\frac{\partial \gamma(p,t)}{\partial t} = \frac{\partial^2 \gamma(p,t)}{\partial p^2} \tag{13.2}$$

Due to the obvious resemblance to ordinary one-dimensional linear diffusion equation this flow is also named *Geometric Heat Diffusion*. However, since the arc-length parameter p is a function of the curve, it is in fact nonlinear.

13.1.1 PDE Formulation

Obviously, when segmentation is the goal, no curve outlining the object of interest is available beforehand. However, the above formulation can be generalised by letting the flow act on the isophote curves of an image.

In general, a curve evolution formulation can be stated in terms of an equivalent level set formulation. The isophotes in the image U undergo the curve evolution defined by the real-valued propagation function β:

$$\frac{\partial \gamma(p,t)}{\partial t} = \beta\, \vec{n}(p,t) \quad \text{is equivalent to} \quad \frac{\partial U}{\partial t} = -\beta\, |\nabla U|$$

This equivalence is derived in the following manner. At a given point, the isophote curve is denoted γ. Then $U(\gamma(p,t),t)$ is a constant function. This means that the partial derivative with respect to t is zero.

$$\begin{aligned}
\frac{\partial U(\gamma(p,t),t)}{\partial t} &= \nabla U \cdot \frac{\partial \gamma(p,t)}{\partial t} + \frac{\partial U}{\partial t} \\
&= \nabla U \cdot (\beta\, \vec{n}(p,t)) + \frac{\partial U}{\partial t} \\
&= \nabla U \cdot \left(\beta\, \frac{\nabla U}{|\nabla U|}\right) + \frac{\partial U}{\partial t} \\
&= \beta\, |\nabla U| + \frac{\partial U}{\partial t}
\end{aligned}$$

Since this partial derivative is zero: $\frac{\partial U}{\partial t} = -\beta\, |\nabla U|$

The normal curvature flow is then written as a PDE:

$$\frac{\partial U}{\partial t} = -\kappa(U)\, |\nabla U| \tag{13.3}$$

For a two-dimensional image I, the isophote curvature is defined by (written in cartesian and gauge coordinates):

$$\kappa(I) = \frac{2 I_x I_y I_{xy} - I_x^2 I_{yy} - I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{\frac{3}{2}}} = \frac{-I_{vv}}{I_w} \tag{13.4}$$

More general, for the image $U \in C(\mathbb{R}^n, \mathbb{R})$ the *mean curvature* is defined:

$$\kappa(U) = div\left(\frac{\nabla U}{|\nabla U|}\right) \tag{13.5}$$

The *Mean Curvature Motion* is then formulated:

$$\frac{\partial U}{\partial t} = -div\left(\frac{\nabla U}{|\nabla U|}\right) |\nabla U| \tag{13.6}$$

13.2 Implementation

The numerous perspectives on mean curvature motion implies that there are many possible implementations inspired by the different formulations of the flow.

The level set formulation of mean curvature motion does not qualify for use of the efficient implementation in [Sethian, 1996]. These narrow band, fast marching techniques are not directly applicable since they only apply to the evolution of a curve.

A simple implementation is possible due to the insight of [Guichard & Morel, 1995]. Here mean curvature motion is shown to be the limit of iterative application of the morphological median filter.

However, the simplest implementation is to use the anisotropic nonlinear diffusion scheme presented in chapter 12. Somewhat surprisingly, mean curvature motion is a special case of the general anisotropic nonlinear diffusion equation.

Let D be the usual diffusion tensor with eigenvectors $\mathbf{v}_1 \parallel \nabla U$, $\mathbf{v}_2 \perp \nabla U$ and corresponding eigenvalues $\lambda_1 = 0$ and $\lambda_2 = 1$. This diffusion is completely anisotropic with diffusion exclusively perpendicular to the gradient. Now the diffusion scheme from definition 11 becomes (where gauge coordinates are used for simplicity since the basis vectors are eigenvectors for the diffusion tensor):

$$\begin{aligned}
\frac{\partial U}{\partial t} &= div\,(D\ \nabla U) \\
&= div\left(D\begin{pmatrix} U_w \\ U_v \end{pmatrix}\right) \\
&= div\,(U_v) \\
&= U_{vv} \\
&= \frac{U_{vv}}{U_w}\, U_w \\
&= -\kappa(U)\ |\nabla U|
\end{aligned}$$

The last equality follows from equation 13.4.

Therefore, no additional implementation is necessary — mean curvature motion is a special case of the implementation developed for chapter 12.

However, it is worth noticing that the diffusion tensor defined by the eigenvalues 0 and 1 does not meet the requirements stated in definition 11. Specifically, this diffusion tensor is not positive definite. Therefore, this diffusion scheme does not necessarily possess the properties listed below definition 11.

The diffusion scheme is illustrated in figure refmcmexample. The effect can be compared with the other anisotropic nonlinear diffusion schemes in figure 12.3.

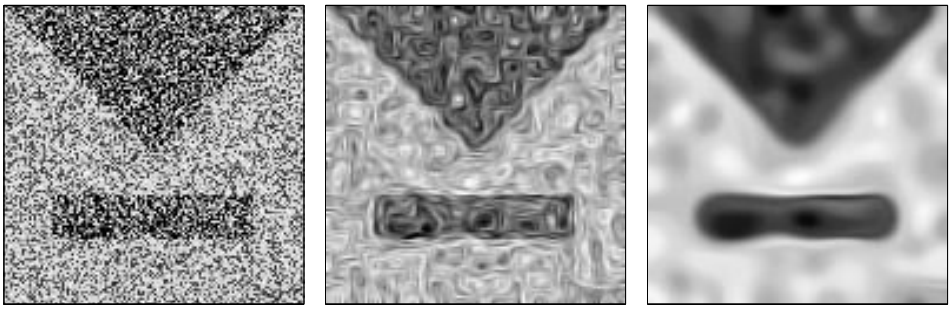


Figure 13.1: *The effect of the MCM diffusion scheme on the image used in figure 12.3.*
The original image is a 128 by 128 image of a triangle and a rectangle where 70% *of the pixels are degraded by noise.*
The diffusion is illustrated at $t = 2$ *and* $t = 80$*. An explicit scheme time discretisation with time step 0.2 is used for the illustrations.*
The original image is obtained from J. Weickert (but originates from the `MegaWawe` *package).*

13.3 The Perspective

Mean curvature motion simplifies the shape of the objects "as fast as possible". This is rather contrary to the anisotropic diffusion schemes presented in chapter 12. These schemes seek to preserve the location of the edges during diffusion while enhancing the contrast in the images.

Curvature motion schemes are used for extraction of the shape, for instance [Malladi et al., 1995] and [Kimia & Siddiqi, 1996]. Admittedly, the most successful schemes use variations of curvature motion — not the pure mean curvature motion.

In the context of the present multi-scale segmentation method, the salient shapes can possibly be extracted during the evolution while the exact location of the edges can be extracted through the linking down to lowest level. The evaluation of the diffusion scheme reveals this in Part Four.

However, the main purpose for including mean curvature motion in this thesis is the fact that it is the true anisotropic diffusion scheme. Thereby the performance of the diffusion scheme gives a nice perspective on the other diffusion schemes.

Perfection is not attainable, but if we chase perfection we can catch excellence.
Vince Lombardi

Chapter 14

The Realm of Diffusion Schemes

The purpose of this chapter is to offer an intuitive overview of the diffusion schemes that are presented in the previous chapters. The underlying motive is the search for a unifying diffusion scheme. This ambitious goal is not reached in this chapter. However, the presentation does collect the previously presented diffusion schemes in a common framework.

In chapter 13 it is shown that mean curvature motion is a special case of the anisotropic nonlinear diffusion equation defined in definition 11. Thereby, all the diffusion schemes are special cases of this diffusion equation. The eigenvalues corresponding to the eigenvectors in the gradient and isophote directions are the descriptors that characterise the schemes. An overview of the schemes is shown in figure 14.1. The exception is coherence enhancing diffusion (section 12.5.5), which is not treated in this chapter.

Diffusion Scheme	Acronyms	Eigenvalue for gradient direction	Eigenvalue for isophote direction
Isotropic Linear	LG, R2, R4 & ILD	1	1
Isotropic Nonlinear	IND	w_4	w_4
Regularised Perona-Malik	RPM	p	p
Edge Enhancing Diffusion	EE2	w_2	1
	EE3	w_3	1
	EE4	w_4	1
Mean Curvature Motion	MCM	0	1

Figure 14.1: *Overview of the eigenvalues for the previously presented diffusion scheme. The diffusivity function p is defined by equation 12.9 and the diffusivity functions w_2, w_3, and w_4 are defined by equation 12.10.*

As mentioned in chapter 12, the diffusivity function w_m is designed to preserve edges more aggressively than the Perona-Malik diffusivity function p. Furthermore, this edge enhancing effect is more pronounced for larger values of m. This notion of aggressiveness of the diffusivity functions allows a mapping of the diffusion schemes into the "space of diffusion schemes" in figure 14.2.

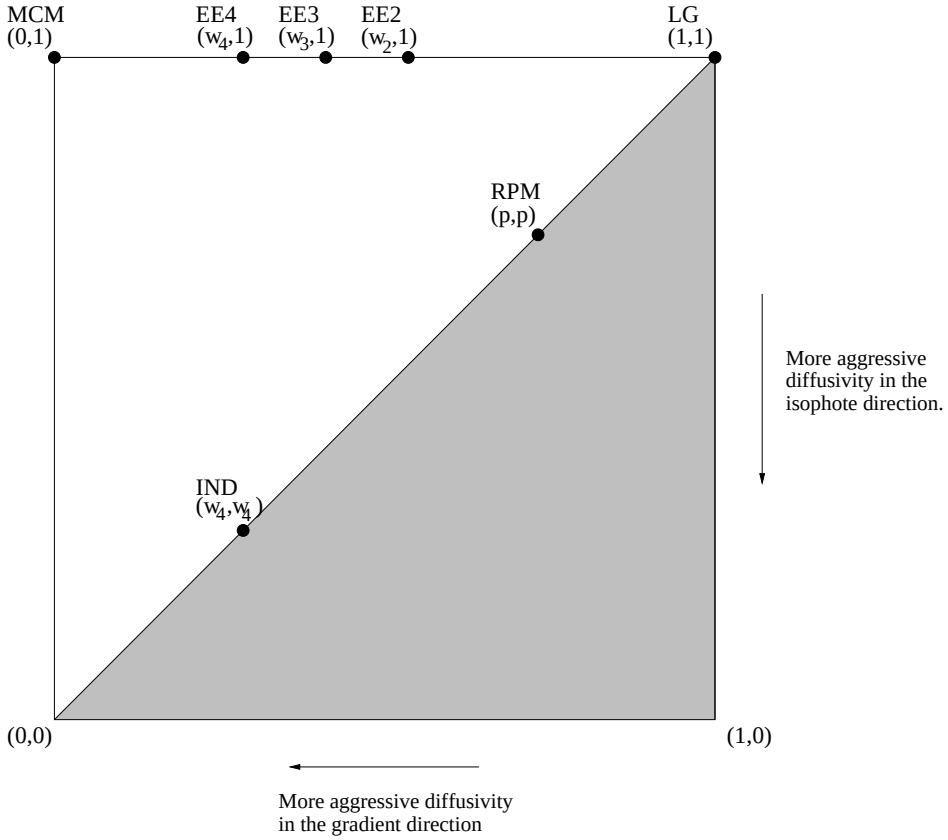


Figure 14.2: *The diffusion schemes of figure 14.1 mapped into the "space of diffusion schemes". The position on the horizontal axis is determined by the diffusivity function corresponding to the eigenvalue for the gradient direction. The position on the vertical axis is determined by the diffusivity in the isophote direction. The diffusion schemes with aggressive diffusivity functions are mapped closest to the lower, left corner.*
The grey area is populated by diffusion schemes that are not suited for segmentation-like purposes — they diffuse more across edges than along them. Thereby the catchment basins merge across possible object borders before merging inside the regions likely to correspond to the desired objects.

Note that the lower right half of figure 14.2 is shaded. This grey area corresponds to the diffusion schemes that have a higher diffusivity along the gradient than along the isophote. These diffusion schemes are fundamentally unsuited for the segmentation method treated in this thesis (and for most other segmentation methods as well). The purpose of the diffusion is to facilitate catchment basins inside the salient objects to merge into larger regions. The building blocks of the segmentation method become perfect if the diffusion scheme preserves the edges corresponding to object border while smoothing the inside of the object into one region.

Possibly, some area in the lower, left part of the figure should be shaded as well. This part contains diffusion schemes where the diffusivity in both directions protect edges very aggressively. This is likely to make the schemes quite sensitive to noise — much like the behaviour of the isotropic nonlinear diffusion scheme IND in figure 12.3.

The specific placement of the diffusion schemes in figure 14.2 is somewhat arbitrary. In order to place the schemes properly, the concept of aggressiveness of the diffusivity functions needs to be formally defined in a mathematical sense.

14.1 Aggressiveness of Diffusivity Functions

In the previous chapters, only four basic diffusivity functions are presented:

Linear diffusivity: $g(s) = 1$

Perona-Malik diffusivity: $p(|\nabla U_\sigma|^2) = \frac{1}{1+\frac{|\nabla U_\sigma|^2}{\lambda^2}}$

Weickert diffusivity: $$w_m(|\nabla U_\sigma|^2) = \begin{cases} 1 & |\nabla U_\sigma| = 0 \\ 1 - exp\left(\frac{-C_m}{\left(\frac{|\nabla U_\sigma|^2}{\lambda}\right)^m}\right) & |\nabla U_\sigma| > 0 \end{cases}$$

No diffusivity: $g(s) = 0$

However, the m parameter in the Weickert diffusivity function allows a broad spectrum of diffusivities for $m \in \ ]0.5, \infty[$.

The diffusivity functions and their corresponding flux functions are illustrated in figure 14.3.

It appears that the m parameter allows the Weickert diffusivity function to cover a large class of diffusivity functions — including Perona-Malik to a close approximation (for m around 0.75).

It is therefore tempting to replace the informal notion of aggressiveness from figure 14.2 with the m parameter. Thereby the family of diffusivity functions w_m is perceived as the entire space of diffusivity functions. However, this is too simplified.

The problem is not the other parameters in the diffusivity functions. For instance, the parameter λ does not, in general, change the qualitative nature of the diffusivity function. It merely changes where the flux function has its maximum — this affects the edges that the diffusion preserves but does not change how "aggressively" the edges are enhanced.

However, the space of diffusivity functions is very unrestricted. The functions illustrated in figure 14.3 share the natural requirement that the diffusivity function is a positive non-increasing function starting at one. Another natural requirement is that the function is smooth. These requirements do not limit the infinite dimensional function space much — and certainly no to a one-parameter space. The family w_m belongs to this much larger class.

Therefore, a complete parameterisation of the diffusivity function class is possibly not desirable due to the large number of parameters. Instead, the available diffusivity functions could be described by the qualitative properties of the diffusivity.

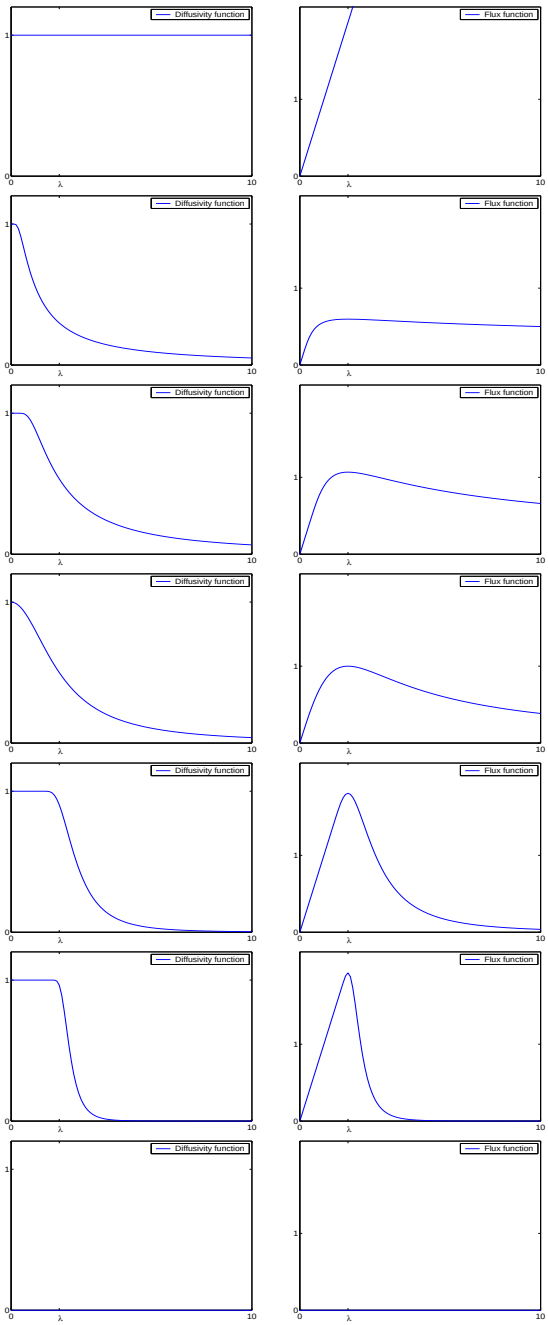


Figure 14.3: *The Spectrum of Diffusivity Functions.*
The limits are linear diffusivity where the diffusivity function is constant one, and no diffusivity where the diffusivity function is constant zero. The other diffusivity functions are constructed such that the corresponding flux function has a maximum for $|\nabla U_\sigma| = \lambda$. The more aggressive diffusivity functions have diffusivity close to 1 for $|\nabla U_\sigma| < \lambda$ and diffusivity decreasing fast to zero for $|\nabla U_\sigma| > \lambda$.
Note that Perona-Malik diffusivity is qualitatively included in the range of Weickert diffusivities.

An example of a possible quantitative property is the ratio between the intregral of the flux function before and after the maximum. The area under the function curve is predominantly after the maximum in the flux functions corresponding to "soft" enhancement of edges. The more aggressive diffusivity functions have a much larger part of the area under the flux curve before the maximum. This measure of "aggressiveness" is illustrated in the table in figure 14.4.

Diffusivity function	A_b	A_a	A_b/A_a
Linear	2	∞	0
Weickert, $m = 0.6$	0.96	∞	0
Perona-Malik	$\log(4) \approx 1.39$	∞	0
Weickert, $m = 0.75$	1.50	∞	0
Weickert, $m = 2$	1.97	∞	0
Weickert, $m = 4$	2.00	∞	0
No diffusivity	0	0	-

Figure 14.4: *A measure of aggressiveness of the diffusivity functions from figure 14.3. The areas under the flux curves before the maximum are denoted A_b — the areas after the maximum are denoted A_a. For all functions $\lambda = 2$ is used.*

This quantitative ratio is not a good measure of the qualitative notion of aggressiveness. It fails to distinguish any of the diffusivity function from the "soft" nonlinear diffusivity functions. The area under the flux curve before the maximum is not a good measure either — before the maximum, the linear diffusivity function is approximately equal to the w_4 diffusivity function.

Even if any of these areas appeared to be a good measure, a problem is that they depend of the choice of λ.

14.2 The Space of Diffusion Schemes

A comprehensive description that uses quantitative properties to describe the qualitative nature is not simple. The obvious difficulty is to determine the appropriate minimal set of descriptors. Another more fundamental problem is that the diffusivity functions do not necessarily depend exclusively on the gradient magnitude. For instance, the diffusion can be adapted to higher order structure (second order structure is used in [Lorenz et al., 1997a]) — this makes the search for a unifying description even more complicated.

Yet another complication is that the descriptors for the diffusivity functions in the gradient and isophote directions should probably not be identical.

The long-term goal is to parameterise the "space of diffusion schemes" in figure 14.2. However, the conclusion is that one parameter in each direction is insufficient. Furthermore, the diffusivity functions should be able to be based on more image information that the gradient magnitude exclusively.

Therefore, figure 14.2 is nothing more than an illustration — not a mapping of the diffusion schemes into the space of diffusion schemes.

14.3 Additional Diffusion Schemes

Apparently, the set of generating parameters for the unifying set of diffusivity functions is not readily available. This is somewhat disappointing, since it would allow the determination of the optimal diffusion scheme for a given class of images. The parameters would simply need to be optimised for the specific type of images through an optimisation process like the one presented in chapter 9.

Unfortunately, this unifying approach is beyond the scope of this thesis. Therefore, additional diffusion schemes must be explicitly designed with the desired properties of the diffusion in mind.

The performance of a diffusion scheme is to a large extent determined by the ability to enhance edges in the image during diffusion. Actually, edge preservation is not desirable in itself — the multi-scale scheme ensures that the locations of the edges are determined through the linking. However, the regions at low scale are merged into larger regions at higher scale. If edges are preserved through this process, regions inside objects are merged and allowed to form large building blocks that correspond to the objects of interest. If the edges are not preserved, the regions merge across region borders at lower scale — thereby the small building blocks must be used for the construction of the objects.

Another key factor is the implicit definition of the edges. The dissimilarity measure defines the edges in the segmentation method presented in this thesis. A diffusion scheme should be designed such that it protects these edges. Since the dissimilarity measure throughout this thesis is the gradient magnitude, the diffusion should enhance the contrast in the image. This is the design goal for all the nonlinear diffusion schemes presented in the previous chapters.

The following diffusion schemes are inspired by figure 14.2 — with the desire to investigate some of the gaps.

14.3.1 Anisotropic Perona-Malik

Inspired by the performances of the various diffusion schemes (which can be inspected in Part Four), it is natural to try the trivial anisotropic extension of the Perona-Malik diffusion scheme. This diffusion scheme is denoted APM for *Anisotropic Perona-Malik*.

The formulation in eigenvectors and eigenvalues is similar to the one from the isotropic Perona-Malik scheme in section 12.5.2. The diffusion tensor has the same eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$. The corresponding eigenvalues are $\lambda_1 = p(|\nabla U_\sigma|^2)$ and $\lambda_2 = 1$. Here p is defined by equation 12.9.

The only difference to the original regularised Perona-Malik scheme is that full diffusion is allowed in the $\mathbf{v}_2$ direction.

14.3.2 GAN — Generalised Anisotropic Nonlinear Diffusion

The anisotropic Perona-Malik is a simple combination of the diffusivity functions previously defined. Other variants of the diffusion schemes are also possible. For instance, in the Weickert diffusivity function w_m, other values for m than 4 could be used for the isotropic diffusion scheme in section 12.5.3.

However, more appealing is a generalisation that encompasses all the combinations in a single diffusion scheme. This is achieved by the following choices for the diffusivitiy functions λ_1 and λ_2 corresponding to the usual eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$.

$$w(m, \lambda, s^2) = \begin{cases} 1 & |\nabla U_\sigma| = 0 \\ 1 - exp\left(\frac{-C_m}{\left(\frac{s^2}{\lambda}\right)^m}\right) & |\nabla U_\sigma| > 0 \end{cases} \tag{14.1}$$

$$\lambda_1 = w(m, \lambda, |\nabla U_\sigma|^2) \tag{14.2}$$

$$\lambda_2 = \theta + (1 - \theta)\ \lambda_1 \tag{14.3}$$

Note that the Weickert diffusivity function is written $w(m, \lambda, s^2)$ instead of $w_m(\lambda, s^2)$. This emphasises that m is to be perceived as a regular parameter of the diffusivity function.

The corresponding diffusion tensor gives a diffusion scheme that is a straigthforward generalisation of the previously presented schemes.

The added parameter θ determines the degree of anisotropy in the diffusion. The previously defined diffusion schemes can be considered special cases of this diffusion scheme:

- Isotropic linear diffusion is defined by $\lambda \to \infty$.
- The isotropic nonlinear scheme (section 12.5.3) is achived for $\theta = 0$.
- The regularised Perona-Malik scheme is approximated by $\theta = 0$ and $m = 0.75$.
- The edge enhancing schemes (section 12.5.4) are defined by $\theta = 1$ and the relevant choice of m.
- Mean curvature motion is achieved for $\theta = 1$ and $\lambda \to 0$.

Even though the GAN diffusion scheme above is a generalisation of the previous diffusion schemes, it is not to be considered as a "unifying scheme". The generalisation is achieved through an additional parameter. Therefore, the scheme has no fundamental theoretical implications. However, the single generalised scheme does allow better overall intuition of the diffusion schemes.

The single generalised diffusion scheme has a practical implication as well. Since all the previously defined diffusion schemes can be achieved by suitable parameter choices (or at least closely approximated), this single diffusion scheme could replace the previously defined schemes in the search for the best diffusion scheme for the segmentation method. The parameters simply need to be optimised for the specific class of data.

In particular, the aggressiveness of the diffusivity function w is allowed to be optimised to the specific class of ground truth image by the optimisation method. Since C_m is not a trivial function of m (as stated in section 12.5.3), only finite precision in the choice of m is allowed — however, for practical purposes this does not differentiate it from the rest of the parameters (i.e. the step size used for m in the optimisation algorithm is 0.1).

It should be noted that the name *Generalised Anisotropic Nonlinear* diffusion is potentially misleading. The scheme does not offer a generalisation of the anisotropic nonlinear diffusion equation. It merely is a generalisation of the schemes evaluated in this thesis.

Expectations for GAN

The diffusion scheme GAN is expected to perform at least as well as the best of the original schemes in the evaluation — the scheme is a generalisation. Furthermore, the extra parameter gives the scheme better potential for adaption to the specific ground truth data set.

However, no fundamentally different diffusion is introduced compared to the previous schemes. Therefore the diffusion scheme is not expected to be drastically better than the best of the remaining schemes.

14.3.3 CID — Curvature Isophote Diffusion

The diffusivity functions previously presented are designed to regulate the diffusion in the gradient direction in order to enhance the edges. Further, the pure anisotropic versions use full diffusion in the isophote direction to smooth the noise close to the enhanced edges.

None of the anisotropic diffusion schemes investigate nontrivial diffusivities in the isophote direction. Since the isophote direction is perpendicular to the edge, it seems sensible to have diffusivity equal constant one is this direction. However, the corners of objects are blurred considerably by these diffusion schemes. A possible extension of the edge enhancing schemes is therefore to apply a diffusivity function in the isophote direction that decreases at corners.

Corners can be defined in a number of ways. The generic evolution have been analysed for some of these corners measures ([Sporring et al., 1998] and [Dam & Lillholm, 1999]).

The following diffusion scheme uses the isophote curvature to steer the isophote diffusivity function. It seems intuitively sensible to let the diffusion in the isophote direction decrease where the isophote curvature is large. In these areas, diffusion along the isophote is, to some extent, across the edge as well.

Diffusivity Functions

The diffusion scheme is stated in terms of the usual eigenvectors $\mathbf{v}_1 \parallel \nabla U_\sigma$ and $\mathbf{v}_2 \perp \nabla U_\sigma$ and corresponding eigenvalues λ_1, λ_2 for the diffusion tensor. The diffusivity functions are based on Weickert's diffusivity function from equation 12.10. This diffusivity function is used to decrease the diffusivity across edges in areas with large gradient, as well as to decrease the diffusivity along edges in areas with large corner measure. The corner measure is defined as $|\kappa(U_\sigma)|\ |\nabla U_\sigma|^2$.

$$
\begin{aligned}
w(m, \lambda, s^2) &= \begin{cases} 1 & |\nabla U_\sigma| = 0 \\ 1 - exp\left(\frac{-C_m}{\left(\frac{s^2}{\lambda}\right)^m}\right) & |\nabla U_\sigma| > 0 \end{cases} && (14.4) \\
\lambda_1 &= w(m_G, \lambda_G, |\nabla U_\sigma|^2) && (14.5) \\
cor(U) &= |\kappa(U_\sigma)|\ |\nabla U_\sigma|^2 && (14.6) \\
\lambda_2 &= \lambda_1 + (1 - \lambda_1) w(m_I, \lambda_I, cor(U)^2) && (14.7)
\end{aligned}
$$

The curvature κ is defined by equation 13.4:

$$\kappa(U) = \frac{2U_x U_y U_{xy} - U_x^2 U_{yy} - U_y^2 U_{xx}}{(U_x^2 + U_y^2)^{\frac{3}{2}}} = \frac{-U_{vv}}{U_w}$$

Note that although Weickert's diffusivity function w_m is used in both directions, different values for m are used (m_I in the isophote direction and m_G in the gradient direction). Analogously, different values for λ are used in the two diffusivity functions (λ_G in the gradient direction and λ_I in the isophote direction).

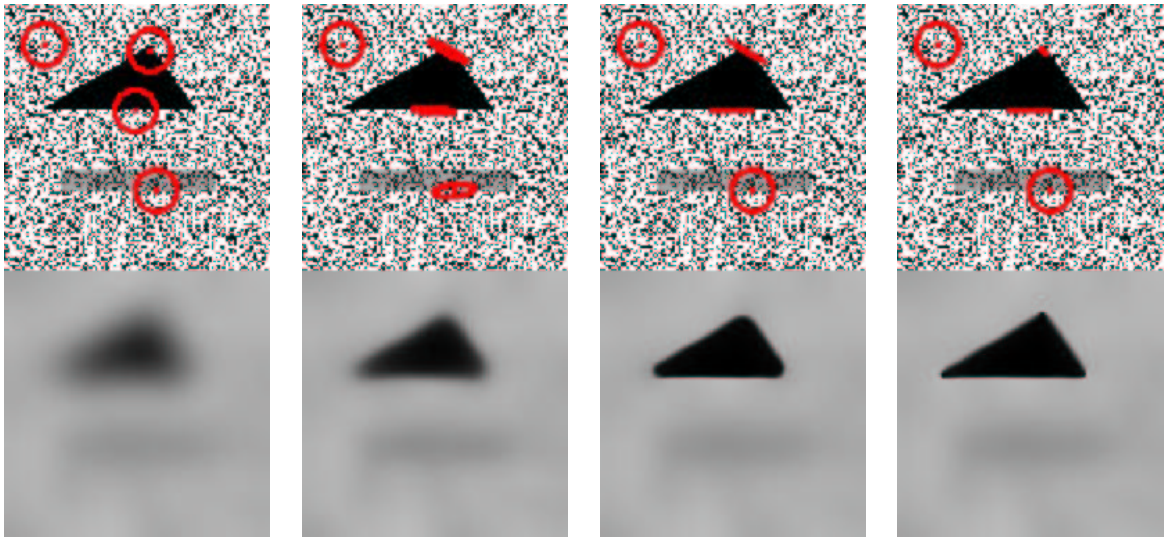
However, the main difference between this diffusion scheme and the ones previously presented is that diffusivity function in the isophote direction is steered by the isophote curvature — not the gradient magnitude.

Expectations for CID

The diffusion scheme CID resembles the GAN scheme. However, it is fundamentally different since the diffusion in the isophote direction is *locally* steered by a corner measure.

The diffusion scheme would be expected to perform quite well in the evaluation. The basic design is sensible — and the large number of parameters allow to scheme to adapt to the specific segmentation task class of images.

However, the diffusion scheme is not evaluated. It is included as an example of a diffusion scheme that is not a special case of the GAN scheme.



Linear Diffusion Perona-Malik EE4 CID

Figure 14.5: *The effects of the diffusion schemes on an articial image. For selected points, the local diffusion is illustrated by a red ellipse that signifies an informal area of support.*
The upper left ellipse is approximately a circle for all diffusion schemes. In a noisy area without significant edges, the diffusion is locally isotropic for all schemes. The bottom ellipse is located at an edge with low contrast. The Perona-Malik scheme protects the edge slightly due to the soft threshold diffusivity function. The parameters for the more aggressive Edge Enhancing Diffusion and CID schemes defines this as a non-edge area. The center ellipse is located at an edge with high contrast. Here the Edge Enhancing Diffusion scheme and the CID scheme become purely anisotropic. The top center ellipse is located at a corner and demonstrates the potential of the CID scheme. The scheme prevents diffusion, and thereby smoothing, at the corner.
The images in the bottom row shows the effects of applying the diffusion schemes on the example image. The images are shown for $t = 40$ *(for the linear scheme,* $\sigma \approx 9$*). The edge preserving properties of the Perona-Malik and Edge Enhancing diffusion schemes are as shown previously in figure 12.3. However, the bottom right images shows that the CID scheme is able to preserve the corners as well.*

Appreciating the CID Diffusion Scheme

Since the CID is not evaluated, it is worth illustrating the fundamental difference between the CID scheme and the previous diffusion schemes. The effects of the local steering of the degree of anisotropic diffusion are illustrated in figure 14.5.

The illustration shows that the CID scheme has promising image enhancing properties. Considering the information available in the image, significantly better corner enhancement is probably impossible without further specialisation.

14.3.4 Properties for GAN and CID

Above, the theoretical properties for the GAN and CID schemes are not treated.

However, the designs of the schemes ensure that the second eigenvalue λ_2 is limited to the interval $[\lambda_1, 1]$. This makes the schemes generally sensible.

Furthermore, the designs ensure that the diffusion tensor is smooth, symmetric and uniform positive definite.

This is sufficient to claim that the diffusion schemes have the same properties regarding smoothness and simplification that is stated for the anisotropic nonlinear diffusion equation below definition 11.

14.4 An Interesting Drop in the Bucket

From the discussions above, it is clear that the presented diffusion schemes by no means cover the entire space of diffusion schemes.

Therefore, the evaluation does not reveal the optimal diffusion scheme — only the best among the evaluated schemes. However the evaluated diffusion schemes have proven their worth in other, closely related, application areas such as preprocessing for segmentation. The scheme with the best performance among the evaluated ones is therefore expected to be competetive with any given alternative diffusion scheme.

A central result of this chapter is the design of three additional diffusion schemes:

- APM — *Anisotropic Perona-Malik* is a simple anisotropic variation of the original regularised Perona-Malik diffusion scheme.
- GAN — *Generalised Anisotropic Nonlinear* diffusion is a single diffusion scheme that encompasses all the other evaluated diffusion schemes as special cases.
- CID — *Curvature Isophote Diffusion* is fundamentally different from the previuosly presented diffusion schemes, however. The CID scheme has a diffusivity function that decreases with increasing isophote curvature.

An interesting aspect of the GAN scheme is that it reveals the type of diffusion scheme best suited for the segmentation method. The parameter set that results from the optimisation method determines whether nonlinear diffusivity functions are preferable to linear and whether anisotropic diffusion is preferable to isotropic.

Thereby, the GAN is not only a generalisation that offers a practical single scheme instead of the previous handful of schemes. It can also be perceived as a characterisation of the deep structure of the geometry of the ground truth data set.

Part IV

Results & Conclusion

Chapter 15: Evaluation Results

Chapter 16: Future Work

Chapter 17: Conclusion

The secret of success is sincerity. Once you can fake that, you've got it made.
Jean Giraudoux

Chapter 15

Evaluation Results

Part One introduces the basic multi-scale watershed segmentation method. This method, developed in [Olsen, 1996], uses the deep structure of watershed catchment basins of a dissimilarity measure to generate building blocks of varying sizes. The original dissimilarity measure, the gradient magnitude, is used exclusively in this thesis. However, the use of other diffusion schemes for the underlying gradient scale-space is explored.

In Part Two, three evaluation methods are designed. The first method measures the processing cost required to reach a perfect segmentation. The second method refines this and measures how good segmentation quality optimal use of the building blocks supports — given a limited number of user actions. The third method is a variant of the first method. The first method measures the processing cost required to reach perfect segmentation by means of optimal use of the building blocks. The third methods measures the processing cost required where a simple heuristic strategy determines the use of the building blocks.

For all evaluation methods, the processing cost is measured in terms of canonical actions — selections and deselections of the building blocks. The perfect segmentation is defined by ground truth data sets, where both simulated and real data sets are used.

In Part Three, the alternative diffusion schemes are presented. Central schemes are the regularised Perona-Malik diffusion, the various anisotropic nonlinear schemes of Joachim Weickert and the mean curvature motion scheme. All these presented schemes are special cases of the anisotropic nonlinear diffusion equation (definition 11). Chapter 14 investigates the design of a unifying diffusion scheme that generalises the diffusion schemes through a canonical one-parameter diffusivity function. It is argued that this search is inherently futile. Instead, a general diffusion scheme that encompasses all the evaluated diffusion schemes is designed. In principle, this diffusion scheme GAN (Generalised Anisotropic Nonlinear diffusion) could replace the remaining schemes in the evaluation. However, the evaluation of the individual diffusion schemes offers much better insight in the nature of the diffusion. Specifically, the comparison of the schemes reveals whether nonlinear diffusion is superior to linear diffusion, and whether anisotropic diffusion is superior to isotropic diffusion — in the context of multi-scale watershed segmentation of MR brain scans.

In this chapter, the results from the evaluation are presented. Due to the computational requirements of the parameter optimisation process, not all diffusion schemes are evaluated on all data sets. Four data sets are selected as the central ones. For the simulated data, the data set with most noise is selected. The simulated data set with almost no noise does not resemble the real data with respect to variation in the images. Thereby, evaluation on these data does not give equally relevant results. The real data sets from the schizophrenic brain are given a lower priority than the sets from the normal brain for similar reasons. Since the physical implications on the geometry of the brain due to schizophrenia are unknown to the author, the evaluation of the data from the schizophrenic brain offer less general results.

Thereby, the central data sets are BW9W, BW9G, IRNW, and IRNG — here BW9x are the simulated data from Brain Web with 9% noise, IRNx are the real data of the Internet brain segmentation Repository from the Normal brain, W and G are for White matter and Grey matter respectively.

The evaluation results are numerous. Therefore the evaluation is presented step by step. Obviously, the overview and the bottom line numbers are saved until the end of the chapter.

The parameter sets used for the different diffusion schemes are generally not displayed in this chapter. However, a complete listing of the optimised parameter sets for each diffusion scheme is enclosed in appendix A.

The reader should be aware that colour prints of the following graphs are essential. It is quite difficult to distinguish the curves from a black and white print.

15.1 The Evaluation Basis

Before the actual evaluation is presented, first a quick reminder of the acronyms used for diffusion schemes and ground truth data sets. The following methods are evaluated.

Acronym	Diffusion Scheme	Section
QT	Quad tree	6.4
LG	Linear Gaussian (fourier domain implementation)	3.2
R2	Linear Gaussian (second order recursive filter)	11.1.2
R4	Linear Gaussian (fourth order recursive filter)	11.1.2
ILD	Isotropic Linear (special case of anisotropic nonlinear diffusion)	12.5.1
RPM	Regularised Perona-Malik	12.5.2
IND	Isotropic Nonlinear (with diffusivity parameter $m = 4$)	12.5.3
EE2	Edge Enhancing (with diffusivity parameter $m = 2$)	12.5.4
EE3	Edge Enhancing (with diffusivity parameter $m = 3$)	12.5.4
EE4	Edge Enhancing (with diffusivity parameter $m = 4$)	12.5.4
MCM	Mean Curvature Motion	13.1
APM	Anisotropic Perona-Malik	14.3.1
GAN	Generalised Anisotropic Nonlinear diffusion	14.3.2

The following ground truth data sets are used. As mentioned above, the data sets BW9W, BW9G, IRNW, and IRNG are considered central. Each data set consists of a number of slices from a 3D data set. The BWxx sets use slices 60, 80, 100, 120, and 140. The IRNx sets use slices 10, 20, 30, 40, and 50. The IRSx sets use slices 20, 36, 52, 68, and 84.

Acronym	Ground truth data set	Section
BW1W	**B**rain **W**eb simulated MR data. Noise level **1**%. Ground truth segmentation of **W**hite matter tissue.	8.3
BW1G	**B**rain **W**eb simulated MR data. Noise level **1**%. Ground truth segmentation of **G**rey matter tissue.	8.3
BW9W	**B**rain **W**eb simulated MR data. Noise level **9**%. Ground truth segmentation of **W**hite matter tissue.	8.3
BW9G	**B**rain **W**eb simulated MR data. Noise level **9**%. Ground truth segmentation of **G**rey matter tissue.	8.3
IRNW	**IBSR** real MR data. **N**ormal brain. Manual ground truth segmentation of **W**hite matter tissue.	8.2
IRNG	**IBSR** real MR data. **N**ormal brain. Manual ground truth segmentation of **G**rey matter tissue.	8.2
IRSW	**IBSR** real MR data. **S**chizophrenic test subject. Manual ground truth segmentation of **W**hite matter tissue.	8.2
IRSG	**IBSR** real MR data. **S**chizophrenic test subject. Manual ground truth segmentation of **G**rey matter tissue.	8.2

15.2 Linear Diffusion Implementations

The schemes LG, ILD, R2, and R4 are different implementations of linear Gaussian scale-space. The purpose of this part of the evaluation is to establish whether the fast recursive filter schemes R2 and R4 performs as well as the original Fourier domain implementation LG and the special case of the anisotropic nonlinear diffusion scheme implementation ILD.

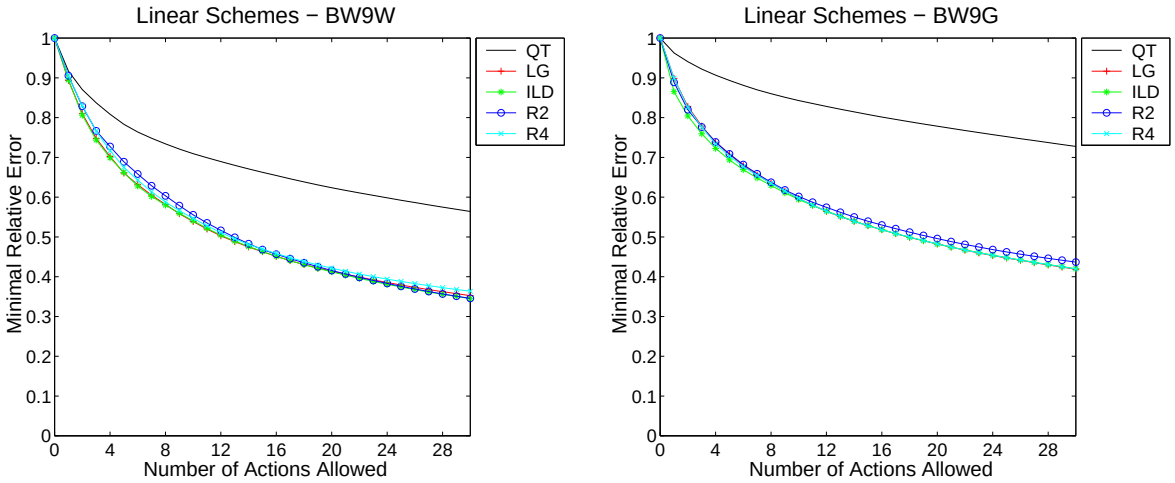


Figure 15.1: *Evaluation of linear isotropic schemes — simulated data.*

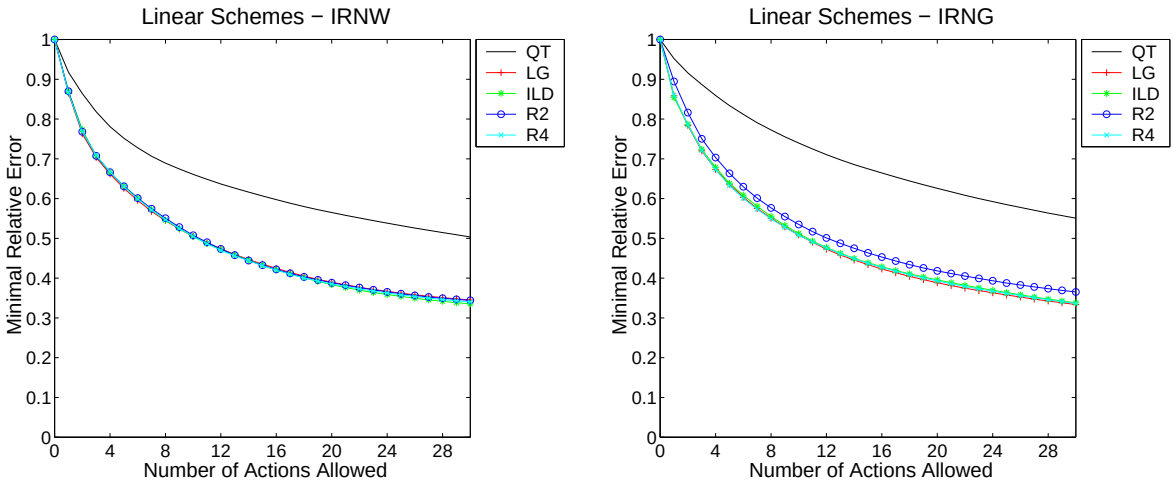


Figure 15.2: *Evaluation of linear isotropic schemes — real data.*

The different implementations perform almost identical. Only for the data set IRNG, the R2 scheme is slightly worse than the other linear Gaussian schemes. The informal notion "slightly worse" is put into perspective by the following evalution results and the overview at the end of the chapter. The original LG scheme is used as representative for the isotropic linear schemes in the following evaluations.

15.3 Anisotropic Nonlinear Schemes

The isotropic regularised Perona-Malik RPM is evaluated with Weickert's isotropic nonlinear scheme ILD and Weickert's anisotropic edge enhancing schemes EE2, EE3, and EE4. Quad tree linking and linear Gaussian diffusion are included for comparison.

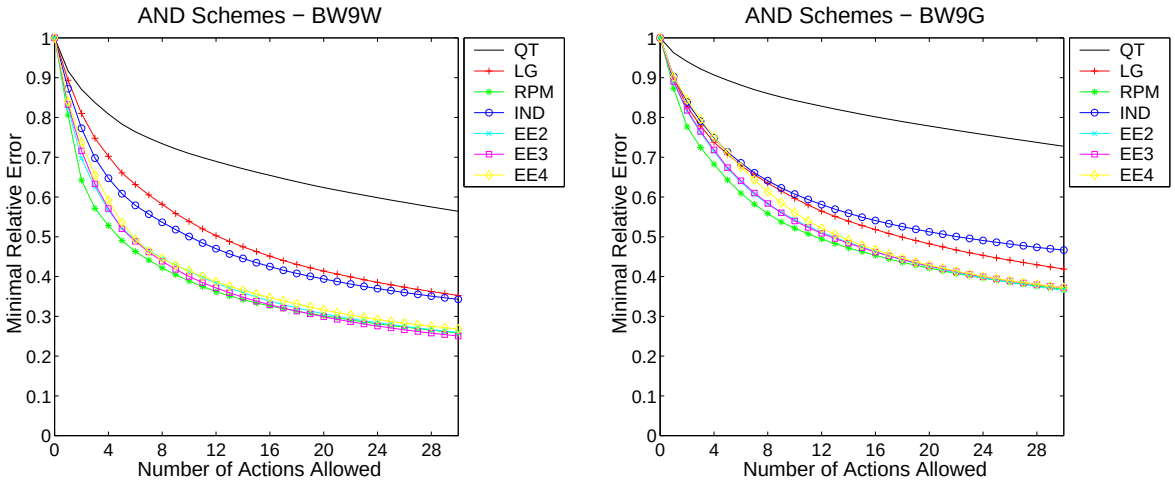


Figure 15.3: *Evaluation of anisotropic nonlinear schemes - simulated data.*

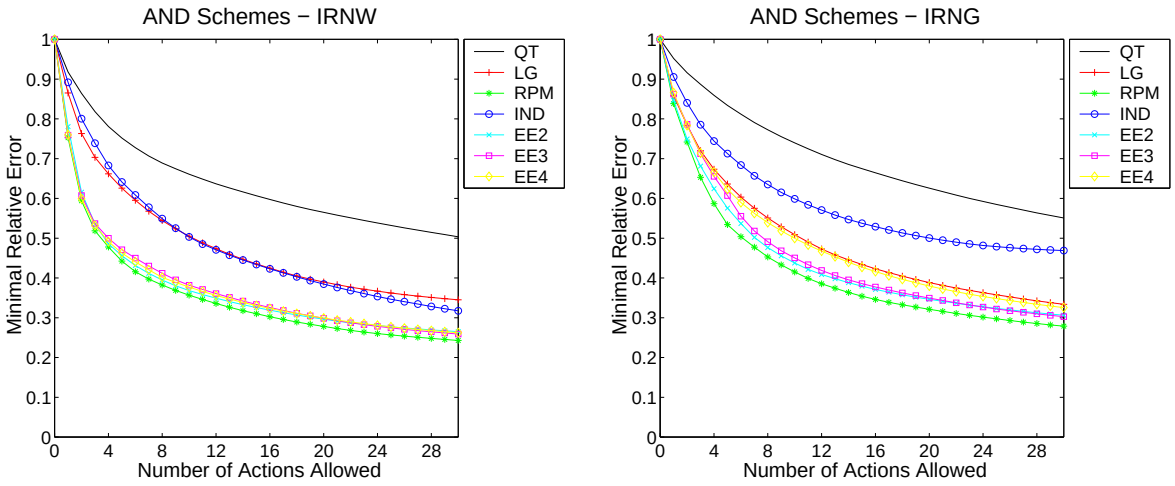


Figure 15.4: *Evaluation of anisotropic nonlinear schemes - real data.*

Apart from IND, the nonlinear schemes are superior to linear diffusion. Somewhat surprising, the isotropic RPM scheme appears to be slightly better than the anisotropic schemes.

15.4 Performance of Mean Curvature Motion

The performance of the mean curvature motion scheme MCM is compared to the characteristic representatives from the previous evaluations: QT, LG, and RPM.

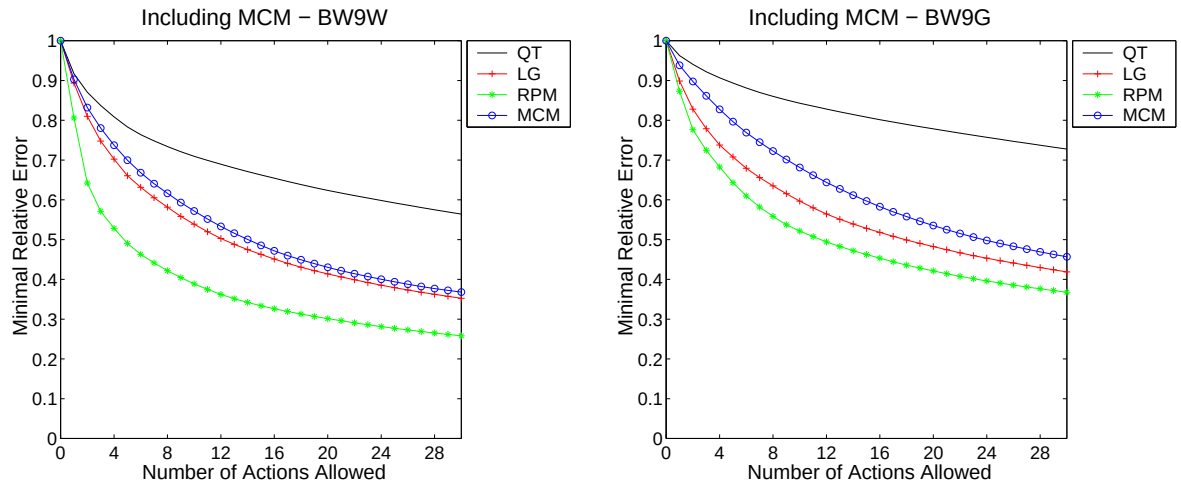


Figure 15.5: *Evaluation of mean curvature motion - simulated data.*

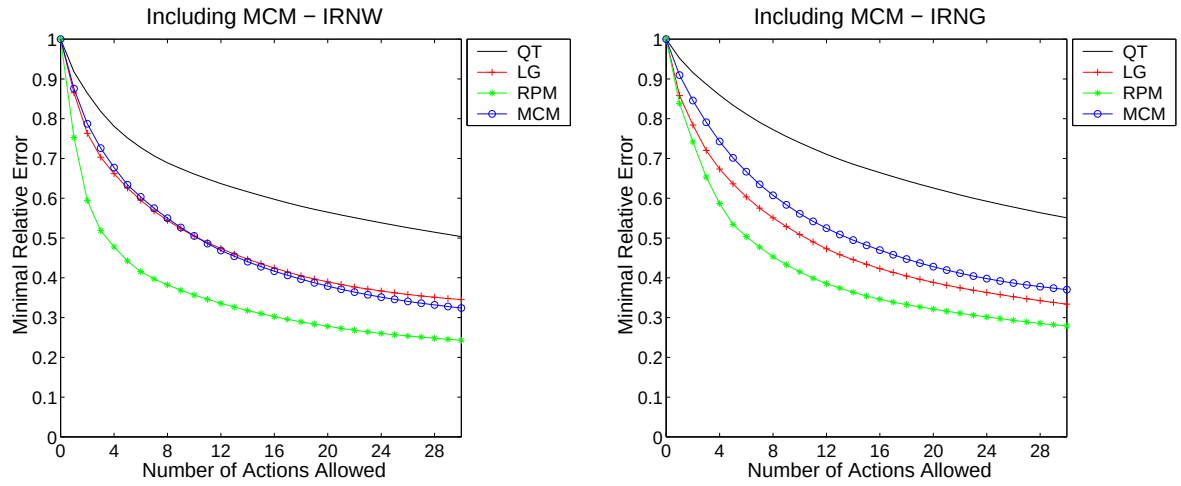


Figure 15.6: *Evaluation of mean curvature motion - real data.*

The pure anisotropic scheme MCM is not expected to be well suited for the segmentation method. However, it is possibly slightly surprising that it performs worse than the original linear Gaussian scheme on average.

15.5 Evaluation of New Schemes: APM and GAN

In chapter 14, three new schemes are designed — two of them are evaluated: The anisotropic extension of the Perona-Malik schemes APM, and the generalised anisotropic diffusion scheme GAN. Quad tree linking and linear Gaussian diffusion are included for comparison.

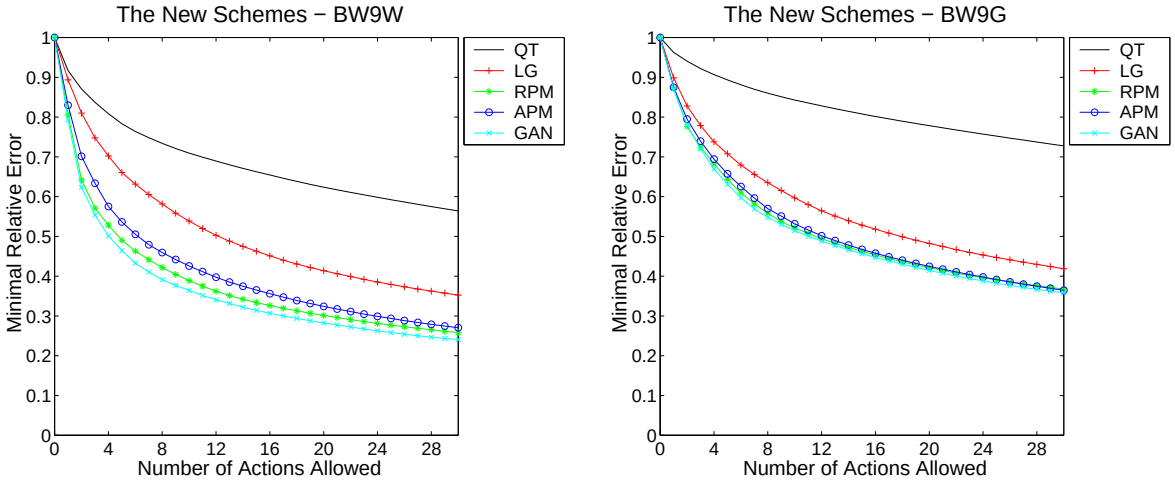


Figure 15.7: *Evaluation of new schemes — APM and GAN on simulated data.*

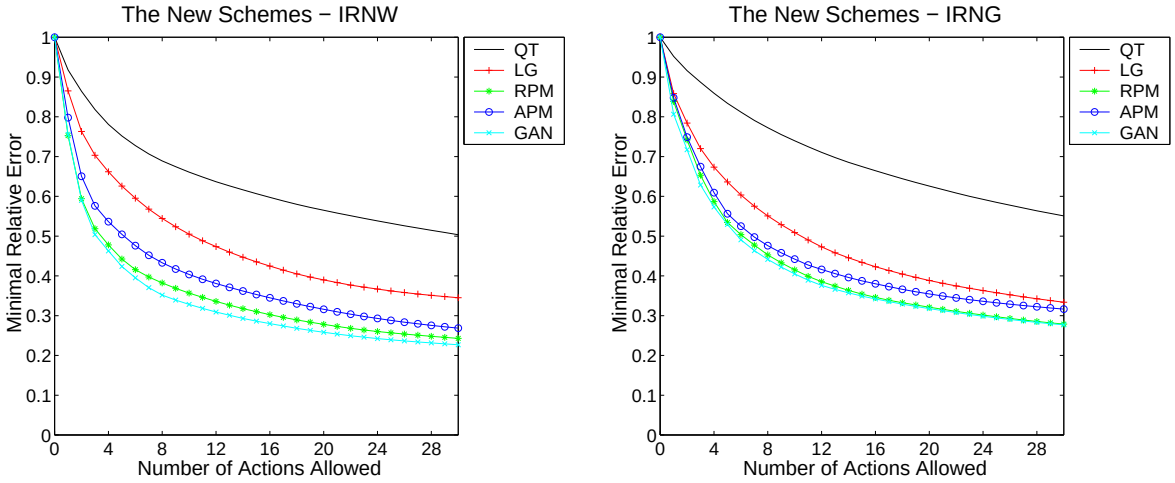


Figure 15.8: *Evaluation of new schemes — APM and GAN on real data.*

The new APM scheme performs worse than the original RPM scheme in all data sets. Again anisotropy appears to have no positive effects on the performance.

The new GAN scheme offers a slight improvement compared to the RPM scheme. Note that the anisotropy parameter θ is close to zero for all data sets (0.06, 0.12, 0.0, and 0.0). The slightly improved performance is due to increased aggressiveness in the diffusivity function (the parameter m is 1.4, 1.0, 1.1, and 1.0 compared to the approximate value of 0.75 for RPM).

15.6 The Remaining Data Sets

For the sake of completeness, the remaining evaluations on the "non-central" ground truth data sets are included. As mentioned above, not all diffusion schemes are evaluated on these data sets.

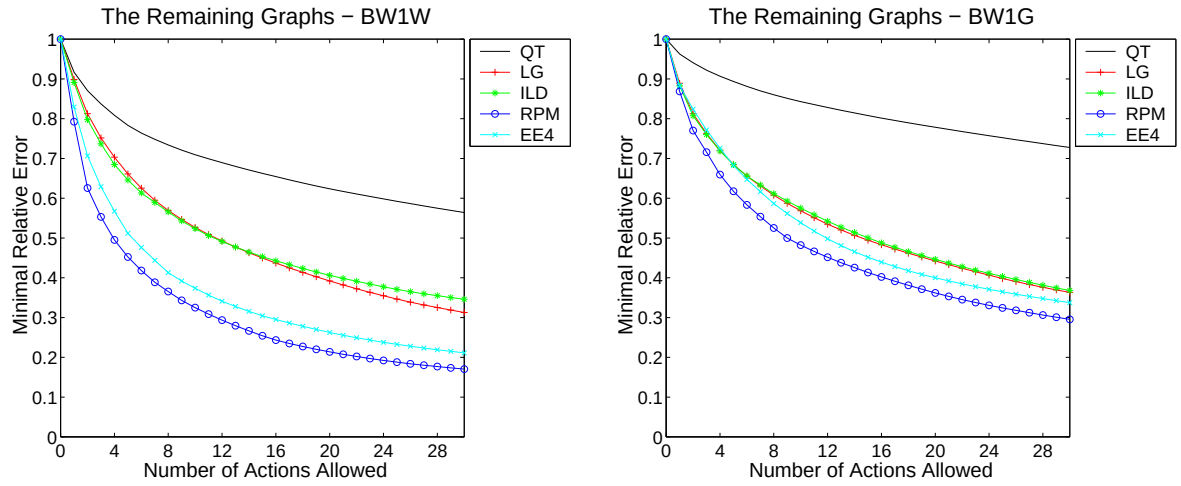


Figure 15.9: *Evaluation of central schemes on low-noise simulated data.*

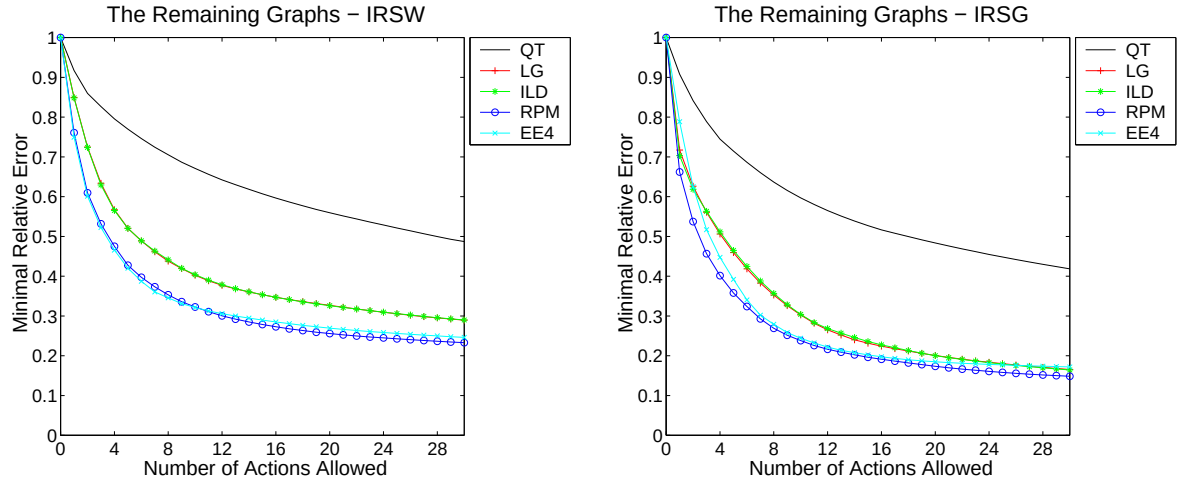


Figure 15.10: *Evaluation of central schemes on schizophrenic brain data set.*

The evaluation results follow the general picture. The isotropic RPM scheme performs best with the edge enhancing scheme EE4 a little worse but still significantly better than linear isotropic diffusion.

15.7 The User-friendly Evaluation

The introduction of the evaluation of the user-friendly selection strategy serves two main purposes. Firstly, the evaluation reveals whether the relative performances of the diffusion schemes measured by the optimal evaluation methods is experienced in the same way by a user. Secondly, the evaluation reveals how close to optimal performance a simple heuristic strategy is.

The table in figure 15.11 compares the optimal evaluation method (method 1, section 6.2.2) with the user-friendly evaluation method (method 3, section 6.3.3) for the linear Gaussian diffusion scheme LG on all data sets. The second column shows the processing costs measured by method 1. These numbers are to be compared to the third column which shows the processing cost measured by method 3. The numbers are relative — the processing costs are around one fourth of the total number of pixels in the ground truth objects. It would appear that the optimal strategy is insignificantly better than the heuristic.

Data	Optimal	User-friendly				Actions		Ratio
	PC	PC	Adds	Rems	Error	Optimal	User	
BW1W	0.212	0.216	0.012	0.006	0.197	0.015	0.019	1.270
BW1G	0.213	0.220	0.010	0.016	0.195	0.018	0.026	1.418
BW9W	0.305	0.309	0.009	0.005	0.295	0.010	0.014	1.344
BW9G	0.287	0.295	0.011	0.016	0.267	0.020	0.028	1.404
IRNW	0.343	0.348	0.011	0.006	0.331	0.013	0.017	1.341
IRNG	0.241	0.247	0.011	0.014	0.222	0.019	0.025	1.320
IRSW	0.291	0.295	0.016	0.004	0.275	0.016	0.020	1.233
IRSG	0.131	0.135	0.005	0.010	0.120	0.011	0.015	1.348
All					0.238	0.015	0.020	1.337
Central					0.279	0.015	0.021	1.355

Figure 15.11: *The processing costs measured by evaluation methods 1 and 3 for the linear Gaussian diffusion scheme. The pixel-level operations are substracted from the total processing costs in the two columns under the heading "Actions". The ratio between the number of actions for the heuristic user strategy and the optimal selection strategy is* 1.337 *on average — and* 1.355 *for the central data sets.*

However, this is not correct. The processing costs for method 3 are split into three numbers: the number of selections (denoted *Adds*), the number of deselections (denoted *Rems* for removals) and the *Error*. The error is equivalent to the number of selections or deselections necessary at the pixel level in order to achieve perfect segmentation. The designs of the two evaluation methods ensure that the number of pixel-level operations is equivalent in the two strategies. Therefore this error count can be subtracted from the total processing costs. This is done in columns 7 and 8, where the number of *Actions* is displayed for both strategies. The numbers in these columns offer a better comparison of the two selection strategies.

The final column contains the ratio between the number of actions for the two strategies. On average for all data sets, the user-friendly selection strategy requires 33% extra actions in order to reach the best possible segmentation without using the building blocks at the pixel level. On average for the central data sets, this number is 35%.

The above comparison of the two selection strategies that form the basis for evaluation methods 1 and 3 are only performed for the linear Gaussian diffusion scheme in figure 15.11. The scheme in figure 15.12 displays the corresponding comparison for the other diffusion schemes. Only the four central data sets are used.

Method	Error	Actions		Ratio
		Optimal	User	
MCM	0.329	0.011	0.014	1.258
R2	0.315	0.012	0.017	1.398
R4	0.280	0.015	0.021	1.344
ILD	0.267	0.016	0.022	1.368
LG	0.279	0.015	0.021	1.355
IND	0.349	0.013	0.014	1.151
EE4	0.254	0.013	0.018	1.339
EE3	0.243	0.014	0.019	1.350
EE2	0.248	0.013	0.017	1.345
APM	0.232	0.016	0.021	1.321
RPM	0.215	0.018	0.024	1.349
GAN	0.212	0.017	0.023	1.347

Figure 15.12: *The processing costs from evaluation methods 1 and 3 where the actions performed at the pixel-level are substracted. The last column shows the ratio between the number of actions required by the heuristic user strategy and the optimal selection strategy.*

The table shows that the ratio between the actions used by the two strategies are quite similar for all diffusion schemes. The exceptions are the IND scheme and to some extent the MCM scheme. Since the overall performances of these schemes are inferior, this is of no concern.

The rule of thumb is that the user needs approximately one third extra actions compared to the optimal strategy.

Since the *Error* is by far the dominating factor in the processing costs, the two evaluation methods automatically agree on the relative performances of the diffusion schemes.

It should be noted that the best diffusion schemes — RPM and GAN — allow the user to capture up to 79% of the ground truth object without using the pixel-sized building blocks.

The evaluation results for evaluation methods 1 and 3 from above is compared to the evaluation results for method 2 in section 15.12.

15.8 The Number of Scale Levels

All diffusion schemes are evaluated with a fixed number of scale levels — 30. However, all methods will, on average, benefit from more scales. More scales give a finer granularity in the available sizes of building blocks.

Three diffusion schemes are evaluated for different numbers of allowed scale levels. Figure 15.13 shows LG, figure 15.14 shows RPM, and figure 15.15 shows the performance for EE3. All schemes are evaluated on the BW9W data set.

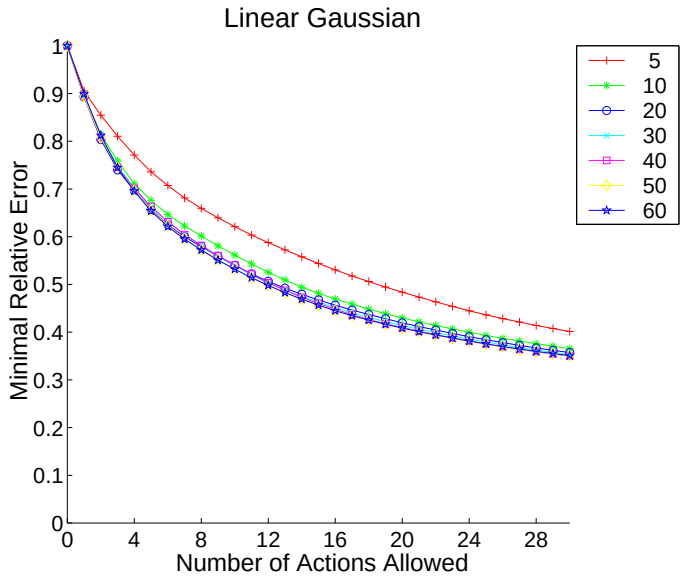


Figure 15.13: *The effect of the number of scale levels — LG evaluated on BW9W for different number of scale levels.*

It appears that the methods have a qualitative "saturation point" where the addition of more scales is insignificant. For LG and RPM, the performance does not increase significantly beyond 20 levels. For EE3, it appears that optimal performance is not reached until 40 scale levels.

This saturation point is relevant for practical reasons. The methods should generate enough scale levels to ensure that the building blocks are near-optimal. However, the generation of an excessive number of scale levels is undesirable. It requires computational time and memory, and in an application more scale levels are confusing for the user.

From a theoretical viewpoint, the differences in saturation points is interesting for two reasons:

- The evaluation is possibly to some extent biased away from the edge enhancing schemes since they are not allowed the necessary number of scale levels to perform optimally.
- The differences in saturation points indicate that the edge enhancing scheme is less stable.

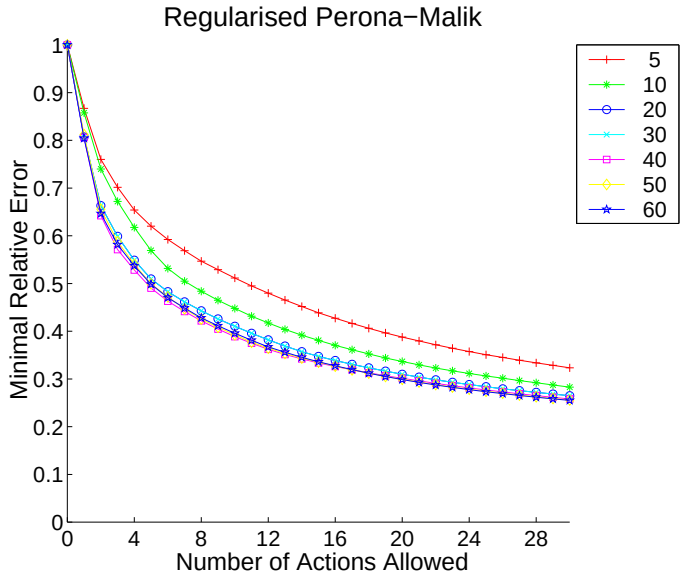


Figure 15.14: *The effect of the number of scale levels — RPM evaluated on BW9W for different number of scale levels.*

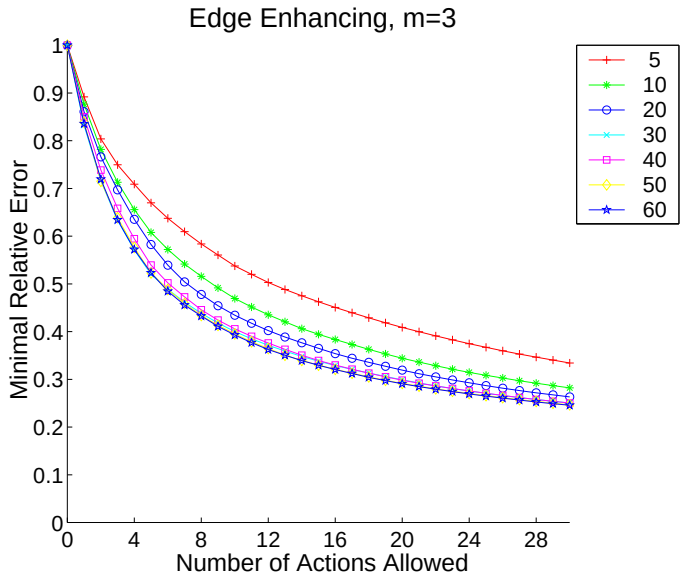


Figure 15.15: *The effect of the number of scale levels — EE3 evaluated on BW9W for different number of scale levels.*

In order to make a proper investigation of the effects of varying the number of scale levels, it is necessary to evaluate the diffusion schemes on several data sets with a number of different scale levels. This requires a tremendous amount of computational time. For this reason it is not explored any further.

15.9 Performance for Increased Action Limit

The optimisation and evaluation is performed with a maximal action limit of 30 actions in the previous evaluations. Obviously, it is interesting to see how the segmentation method performs for higher action limits. This is even more relevant since the best diffusion scheme fails to capture more than 75% of the ground truth objects within the 30 actions.

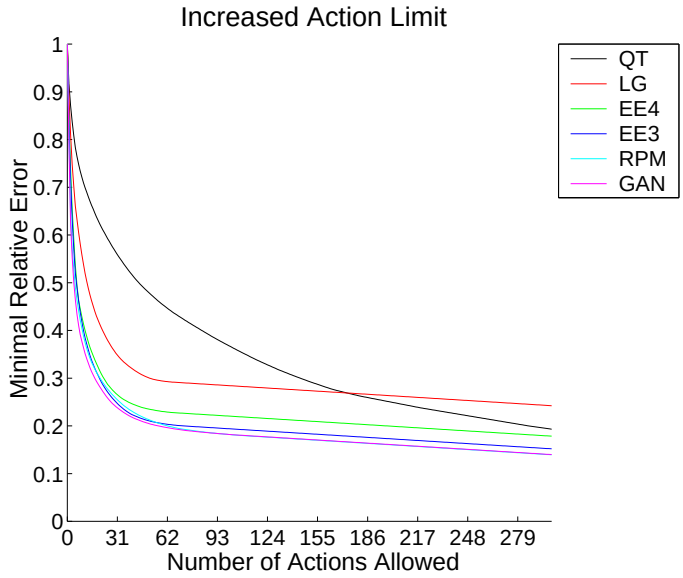


Figure 15.16: *Performance of QT, LG, EE3, and RPM for increased action limit on BW9W.*

The graphs in figure 15.16 reveal that the diffusion schemes LG, EE3, and RPM provide building blocks that allow around 60 profitable user actions. From there, the user is obliged to select and deselect single pixels in order to get further improvement.

The level where the curves flatten is the level where further improvement depends on actions performed on single pixels. It is therefore a measure of how well the diffusion schemes are able to detect the edges that correspond to the ground truth objects. This level corresponds to the error number in table 15.12.

Another striking property is that the QT scheme actually is better than LG from around 170 actions. This implies that the small rectangles are better aligned along the ground truth edges than the building blocks of the linear Gaussian scheme. At first, this seems rather surprising. However, it is a natural consequence of the optimisation process. Since a maximum of 30 actions is used to measure the performance, the methods need not locate the edges perfectly. Consequently, the chosen localisation scales are relatively large. It is of more importance to have large building blocks that recover large parts of the ground truth object fast instead of having small building blocks that allow the finer details to be captured.

This could also explain why the best methods are only able to provide reasonable building blocks for around 80% of the ground truth objects. The evaluation method simply favours parameter sets that allow the schemes to capture 75% of ground truth fast instead of parameter sets that allow the schemes to capture 90% a little less fast.

Thus, in the enlightenment of hindsight, it appears that a maximal action limit around 80 or 90 is more appropriate for the optimisation method. Furthermore, it would be very interesting to compare the results of evaluations based on different action limits. Due to the computational effort required to perform the optimisation and evaluation for all methods on all ground truth data sets this is not feasible.

However, in order to substantiate the observations in this section, the LG and GAN schemes are optimised for the BW9W data set with different action limits. Figure 15.17 displays the performance of these schemes optimised with maximal action limits of 30, 80, and 300.

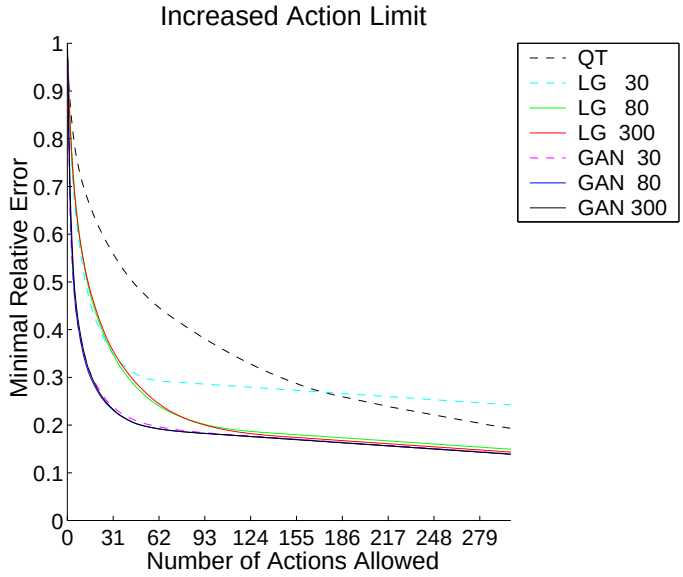


Figure 15.17: *Performance of QT, LG, and RPM for increased action limit on BW9W. The LG and RPM schemes are optimised with maximal action limits of 30, 80, and 300.*

The performances displayed in figure 15.17 are somewhat surprising. As expected, the LG performance curve flattens at a lower level for the parameter set optimised with a maximal action limit of 80 compared to the original maximal action limit of 30. Thereby, the scheme becomes significantly better than the QT scheme for large action counts as well.

However, only minimal improvement is achieved for the LG scheme where the parameter set is optimised with a maximal action limit of 300. The curve flattens at a slightly lower level, but the difference is less than anticipated. This behaviour is even more pronounced for the GAN scheme. The curves for the schemes optimised with the three different maximal action limits are virtually indistinguishable.

The schemes appear to have a qualitative lower bound for the ratio of the ground truth objects than can be captured without use of the pixel-sized building blocks. If the schemes are optimised with a too low maximal action limit, this lower bound level is not achieved. Furthermore, optimisation with a higher maximal action limit allows the schemes to reach this lower bound level. It is worth noticing that this is not achieved at the cost of poorer performance for the first few actions.

In section 9.4 it is stipulated that the optimisation process biases the schemes towards either optimal performance for few actions or optimal performance for many actions. It appears that this is not really the case. Optimisation with a low maximal action limit simply neglects the performance for many actions. Optimisation with a higher maximal action limit improves the performance for many action — but without impairing the performance for few actions significantly.

Therefore, the conclusion is that a large maximal action limit is the proper choice for the optimisation process. For the data sets in this thesis, a maximal action limit at 100 or more would be appropriate.

The consequence is that it is not difficult to "optimise" the maximal action limit parameter for the optimisation process. This obviously makes the optimisation process simpler.

Another observation is that the GAN scheme is apparently more robust that the LG scheme. Optimal performance for few actions automatically ensures near-optimal performance for many actions as well. The LG scheme needs to be challenged beyond 30 actions to be optimised for good performance for many actions as well. However, this is possibly not a quite fair conclusion. Possibly both schemes simply need to be optimised to capture more than two thirds of the ground truth objects. The GAN schemes achieves this from 10-15 actions where the LG needs approximately 40 action to get there.

15.10 The Effects of a Tolerance Area

The previous results show that the best diffusion scheme, GAN, is able to capture approximately 80% of the ground truth objects without the use of the pixel-sized building blocks. Evaluation with a tolerance area (introduced in section 6.3.4) puts this number into perspective.

Figure 15.18 compares the performance of the LG, EE3, EE4, RPM, GAN schemes with and without a tolerance area. For the evaluation with a tolerance area, the width of the tolerance area is one pixel. Without detailed knowledge of the specific segmentation task, this is a fair assumption of the imprecision of the ground truth segmentations.

In practice, this means that a pixel inside a ground truth object in considered inside the tolerance area if some of its four neighbouring pixels are outside the ground truth object. The corresponding pixels outside are then also considered within the tolerance area.

	Tolerance: 0					Tolerance: 1				
	Optimal	User-friendly				Optimal	User-friendly			
	PC	PC	Adds	Rems	Error	PC	PC	Adds	Rems	Error
LG	0.294	0.300	0.011	0.010	0.279	0.081	0.086	0.010	0.009	0.066
EE3	0.257	0.262	0.010	0.009	0.243	0.056	0.060	0.009	0.008	0.043
EE4	0.267	0.272	0.010	0.008	0.254	0.062	0.066	0.009	0.007	0.050
RPM	0.233	0.239	0.012	0.012	0.215	0.046	0.052	0.011	0.011	0.030
GAN	0.229	0.235	0.012	0.012	0.212	0.045	0.050	0.010	0.010	0.030

Figure 15.18: *The effects of a tolerance area on the LG, EE3, EE4, RPM and GAN schemes for the four central data sets. With the tolerance area, the number of actions above the pixel-level is slightly lower. More importantly, the number of selections and deselections of pixel-sized building blocks (the "Error") is significantly lower. With a tolerance area of width one, the GAN scheme captures* 97% *of the ground truth objects without use of pixel-level building blocks — compared to* 79% *without the tolerance area.*

The inclusion of the tolerance area have two effects. With the tolerance area, the optimal number of actions above the pixel-level is slightly lower. However, the main effect is that the number of actions required at the pixel-level is significantly lower with the tolerance area. These effects are identical for the evaluated schemes.

The *Error* numbers for the GAN scheme show that even though only 79% of the ground truth objects are captured without pixel-sized building blocks, 97% of the areas of objects are within the tolerance area.

From a practical point of view, this means that only 3% of the ground truth objects have to be segmented at the pixel level — assuming that a tolerance area of width one corresponds to the desired precision of the segmentations.

However, this figure is also an informal measure of the ability of the diffusion scheme to locate the edges of the ground truth objects. A tolerance area of width one changes the covering of the ground truth objects from "reasonable" to "almost perfect". Therefore this width can be regarded as the imprecision of the edges recovered by the GAN diffusion scheme. Or the imprecision of the ground truth segmentation, for that matter.

15.11 The Source of Improvement

The scheme in figure 15.18 gives valuable information about the nature of the performance for the different schemes. The differences in performances between the different schemes are not primarily determined by the number of selections and deselections on larger regions. The differences are mainly in the ability of the schemes to minimise the number of actions needed at the pixel level.

The need for pixel level actions is equivalent to the ability to align the borders of the building blocks with the boundaries of the ground truth regions. As a contrast, the number of actions performed above the pixel level is determined by the ability of the diffusion schemes to merge the regions inside the ground truth object to a few, large regions.

Since the differences in the schemes in figure 15.18 are mainly in the pixel level error, it is therefore tempting to announce the ability to locate the object borders as the main difference between the diffusion schemes. This is supported by figure 15.16, where the schemes reveal different lower bounds for the ratio of the ground truth areas that they are able to capture without pixel-sized building blocks. However, it is important to realise that this behaviour is beyond the 30 actions that the schemes have as maximal action limit in the optimisation. Therefore, the schemes would be judged by performance in an area where they are not asked to perform well.

Instead the analysis must be based on the curves in figure 15.17 and the table in figure 15.19.

	Tolerance: 0					Tolerance: 1				
	Optimal	User-friendly				Optimal	User-friendly			
	PC	PC	Adds	Rems	Error	PC	PC	Adds	Rems	Error
LG80	0.212	0.219	0.016	0.012	0.190	0.050	0.055	0.016	0.011	0.029
GAN	0.203	0.208	0.010	0.013	0.186	0.040	0.044	0.008	0.011	0.024

Figure 15.19: *The effects of a tolerance area on the LG scheme optimised with a maximal action limit of 80 compared with the GAN scheme on the BW9W data set. The changes in the number of actions are similar to those observed in figure 15.18. However, the LG manages to capture almost as much of the ground truth object without the use of pixel-sized building blocks as the GAN scheme.*

The linear Gaussian scheme optimised with a maximal action limit of 80 reveals a lower bound for the attainable part of the ground truth objects close to the GAN scheme. With the proper optimisation, the linear Gaussian scheme needs almost as few pixel level operations as the GAN scheme.

Figure 15.17 reveals the true difference between the schemes. The lower bounds for the area captured without pixel level operations are similar. However, the GAN scheme is able to reach this lower bound using approximately half the number of actions compared to the LG scheme.

Thus, the main difference between the original Gaussian diffusion scheme and the new GAN scheme is the ability to merge the regions inside the ground truth object into large, profitable building blocks — not the ability to locate the edges. Admittedly, this conclusion is based mainly on the comparison of two schemes on a single data set. Therefore, it is not as solid as the other conclusions in this chapter.

15.12 Overview of Evaluation Results

The presentation of the evaluation results so far gives a general impression of the performance of the schemes. In the following, the qualitative differences between the schemes are replaced by a specific quantitative measure for each scheme.

15.12.1 Consistency of the Evaluation Methods

The evaluations performed by the three evaluation methods offer qualitatively equivalent results. The linear schemes are significantly better than the quad tree linking scheme. The group of anisotropic nonlinear schemes are somewhat better. Close to top performance is the RPM scheme, but the GAN scheme is slightly better. The remaining schemes (MCM, IND, and APM) have the same relative rankings among the other schemes for all three evaluation methods.

However, it is not possibly to determine whether relative performances measured by the three evaluation schemes are quantitatively equivalent.

The evaluation based on method 2 only investigates the performance for up to 30 actions. This is also the performance that the scheme are optimised with respect to. The other two evaluation methods measure the total number of actions required to reach perfect segmentation. Thereby, method 1 and 3 relies heavily on performance that is not measured by method 2, and that is not optimised by the parameter sets for the diffusion schemes.

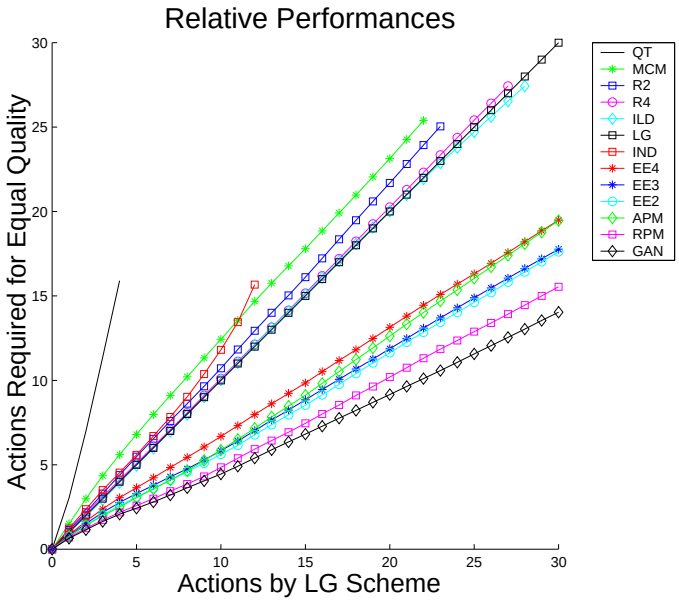
The effect on this is obvious from comparison of the different optimisations of the LG scheme. The performances measured by evaluation method one and three are drastically lower for the schemes optimised with a maximal action limit of 80 and 300 compared to the performance of the scheme optimised with a limit of 30 (see figures 15.18 and 15.19).

In order to compare the results from all three evaluation methods quantitatively, all diffusion schemes need to be optimised with a maximal action limit of 80 or more.

15.12.2 "The Number"

The previous graphs display how far a virtual user can get with a specific number of actions. However, a more appropriate comparison of the schemes is a measure of how much effort is required to obtain a given segmentation quality. Specifically, given the desired quality threshold for a specific segmentation task, the schemes should be compared on the number of actions required to achieve this threshold.

In the following, the performance of the original linear Gaussian diffusion scheme is used as reference frame. In the graphs in figure 15.20, the performances of the schemes are compared. For a given number of actions, the performance of the LG scheme is noted. For each of the other schemes it is then measured how many actions are required to get equally good segmentation quality.



Scheme	Performance
QT	3.83
MCM	1.18
R2	1.08
R4	1.01
ILD	0.99
LG	1.00
IND	1.20
EE4	0.66
EE3	0.59
EE2	0.58
APM	0.63
RPM	0.51
GAN	0.46

Figure 15.20: *The true comparison of the performances of the different diffusion schemes. The linear Gaussian scheme is used as reference frame. For a given number of actions the performance of the LG scheme is used as reference. For each scheme it is then measured how many actions are required to get equally good segmentation quality (on average for the four central data sets). For the other schemes the performances are interpolated linearly between the action limit steps — otherwise the LG scheme would be favoured by the comparison.*
For some schemes the performance of the LG scheme could not be achieved within 30 action for all the four central data sets. These curves for these schemes stop at this point.
The incline of a curve determines the number of actions required to obtain a given segmentation quality — relative to the performance of the linear Gaussian scheme. This performance indicator is displayed in the table for each diffusion scheme. The new GAN scheme requires less than half as many actions than the original linear Gaussian scheme.

The curves in figure 15.20 generally behave beatifully — they are approximately linear. Thereby, the relative performances of the schemes can be stated as one single number: the incline of the approximating line. These numbers are displayed for each method in the table below in figure 15.20.

The significance of these numbers is straightforward. For each action performed through the original linear Gaussian diffusion scheme, the new GAN scheme only requires 0.46 actions on average. In other words, the GAN scheme requires less than half user interaction compared to the original scheme.

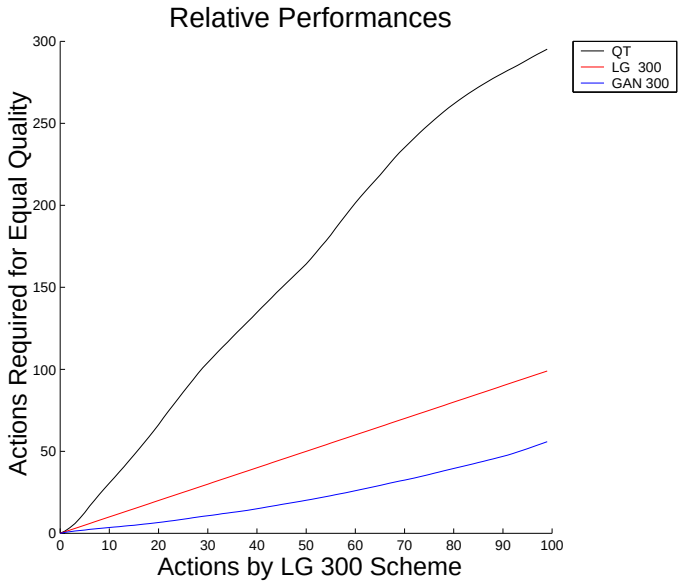
Performance for Increased Action Limit

The numbers above are the effect of optimisation of the parameters of the diffusion with a maximal action limit of 30. Correspondingly, the curves in figure 15.20 are not extended beyond 30 actions.

The behaviour beyond 30 actions should be compared through the versions of the schemes that are optimised with a maximal action limit of 80 or 300. The behaviour beyond 30 actions can then be predicted from figure 15.17.

In the limit, the GAN scheme is not able to obtain double the quality gain achived by the LG scheme. From a certain point, the scheme improves the segmentation by exactly one pixel per action. Thereby the relative advantage of the GAN scheme diminish. Therefore, the linearity of the curves in figure 15.20 disappear in the limit.

However, if the graph is not extended beyond the point where the improvement of the segmentation relies on pixel-sized building blocks, the relative performances of the LG and the GAN schemes are retained. This is illustrated in figure 15.21.



Scheme	Performance
QT	3.23
LG	1.00
GAN	0.48

Figure 15.21: *The performances of the LG and GAN diffusion schemes extended beyond 30 actions.*
The linear Gaussian scheme optimised with a maximal action limit of 300 is used as reference frame. The last part of the curves recede from the linear behaviour as expected. However, the superior performance of the GAN scheme compared to the original linear Gaussian scheme shown in figure 15.20 is retained.

This is beyond the scope of this report.
[Dam et al., 1998]

Chapter 16

Future Work

For most non-trivial fields of research, new questions arise as soon as some of the old questions have been answered. This thesis focuses on two aspects of multi-scale segmentation: evaluation and diffusion. Even within these areas a lot of topics are left unexplored.

Below, some of these open questions are listed. Three of these are emphasised at the end since they are *extremely* interesting. However, this simply implies that the projects mentioned under "Miscellaneous Projects" are only *very* interesting.

16.1 Miscellaneous Projects

16.1.1 The Effects of Noise

The simulated data sets from Brain Web (section 8.2) can be generated with noise levels 0%, 1%, 3%, 5%, 7%, and 9%.

Further evaluation of the diffusion schemes on data sets with different noise levels would allow analysis of whether the specific schemes are more or less sensitive to noise. An indication of the effects of noise is available through inspection of figure 15.9. This figure displays the performance of selected diffusion schemes on the simulated data sets with 1% noise. For comparison, the performance of the same diffusion schemes on the corresponding data sets with 9% are illustrated in figures 15.1 and 15.3.

Not surprisingly, the schemes perform better with less noise. Furthermore, it appears that the differences in performance between the schemes seem to diminish somewhat on the more noisy data sets. This is not surprising either. In the limit with total noise, all diffusion schemes are expected to perform equally bad. However, proper analysis with more data sets is required to clarify whether the diffusion schemes are equally robust with respect to noise.

16.1.2 Evaluation of Dissimilarity Measures

The evaluation reveals that proper choice of diffusion scheme implies a considerably improved performance. Another central aspect of the segmentation method is the dissimilarity measure. It would be quite interesting to explore whether other choices of dissimilarity measures than the gradient squared would offer further improvements. These alternative dissimilarity measures could be inspired by various edge detectors.

The improved performance due to the choice of diffusion scheme is only investigated with respect to segmentation of greyscale images. Another interesting line of work would be to design dissimilarity measures that allow segmentation of color images, textures, multi-modal data sets et cetera.

16.1.3 The Importance of Linking

The segmentation method ensures that dislocation of edges due to diffusion is handled by the linking of the catchment basins. This linking step is definitely crucial for the linear diffusion to work properly. An often stated property of the nonlinear schemes is that the dislocation of the edges is less severe.

It is not trivial to derive a quantitative measure of the dislocation of the watersheds. And it is not necessary for the evaluation of the diffusion schemes. However, a measure of dislocation would provide interesting insight in the nature of the different diffusion schemes.

16.1.4 3D

An obvious extension of the work is a 3D implementation. The segmentation method is only implemented and evaluated in 2D in this thesis.

This would make the work more general from a theoretical point of view. It is tempting to assume that the relative performance of the diffusion schemes are preserved in 3D —- however, proper evaluation is needed to establish this.

From an application point of view, the work would also be more general. Most medical scannings are 3D, and therefore a large application potential is in 3D.

Yet another aspect is that 3D images contain interesting shapes that are not present in 2D — for instance blood vessels. It would be very interesting to design diffusion schemes that allow segmentation of such structures (see for instance [Krissian et al., 1997] and [Lorenz et al., 1997a]).

For more information regarding 3D diffusion schemes see references in [Weickert, 1999] and [Weickert, 1998a].

16.1.5 Efficient Implementations

A limiting factor in the use of the segmentation method is the computational time required to generate the gradient scale-spaces — in particular during the optimisation of the diffusion schemes.

The applicability of the method therefore relies heavily on efficient implementations of the diffusion schemes. The original linear Gaussian is implemented efficiently by a recursive filter algorithm that performs as well as the original implementation.

An efficient implementation of the GAN scheme would obviously be very useful. However, since the RPM performs almost as well, it would possibly be equally relevant to implement this scheme efficiently. Since the scheme is isotropic, the implementation will be simpler and faster.

AOS-schemes and multi-grid algorithms are relevant for these approaches. For more information on efficient implementations see [Weickert, 1998c] and [Weickert, 1998a].

16.1.6 Optimisation of the Optimisation Method

The optimisation method is robust and reliable — but quite time-consuming. For a practical application, the GAN or the RPM scheme needs a few example images with ground truth segmentations for the optimisation. This process might require a few days of computation.

For 3D images, the process would possibly be unfeasible. The computational efforts required by the evaluation methods are insignificant. However, the optimisation method requires the generation of quite a lot of gradient scale-spaces. For 3D images this step is even more time-consuming than for 2D images.

Therefore, it is important to design an optimisation method that requires less steps — without sacrificing the robustness.

16.1.7 Alternative Diffusion Schemes

As mentioned in chapter 14, the evaluated diffusion schemes are only drops in the bucket. The space of diffusion schemes is infinitely dimensional.

It would be interesting to explore the performance of other schemes. Other schemes with edge enhancing properties are mentioned in [Weickert, 1999] and [Weickert, 1998a].

Naturally, it would also be very interesting to evaluate the performance of the CID scheme (section 14.3.3).

16.1.8 Catastrophe Theory Analysis

Analysis of the generic events for the dissimilarity measure for nonlinear schemes is by no means trivial.

However, this analysis would provide insight to the nature of the deep structure of these diffusion schemes beyond the quantitative measures of the evaluation methods.

16.2 Major Projects

As promised, three major directions for future work are emphasised.

16.2.1 Region Merging

In the segmentation method presented in this thesis, the regions from the localisation level are merged through diffusion. Other schemes exist for this region merging process.

Flooding algorithms extend the intuitive background for the watershed algorithm. The watersheds in a landscape are defined as the ridges that separates the catchments basins. The rain analogy allows a *strength* attribute to the watersheds. When it rains, the individual pools are filled gradually. Neighbouring pools with a low watershed boundary are merged after a little while. Regions separated by a high watershed require more rain to merge. This gives an ordering of the merging. This basic flooding technique has possible variations that focus on the volume or the depth of each catchment basin, or the width of the watershed separators.

The main principle is that the regions are merged in a specific order. This is the same basic principle that steers the formation of the building blocks in the multi-scale watershed algorithm. It is therefore possible to make a two-dimensional scale-space where one parameter determines the diffusion scale, and the other parameter determines the flooding level.

A scale-space with two independent scale parameters is quite complicated. Specifically for this thesis, it is not trivial to extend the evaluation methods.

However, the flooding principle could be isolated to be used at a fixed scale level. In an application, the user will then select or deselect not only one building block in each action. The flooding level determines whether the neighbouring building blocks are "sufficiently equal" to be selected/deselected by the same action. In a user interface this could be applied quite elegantly. For instance, the action could select/deselect series of regions while the mouse was pressed. Thereby the duration of the mouse click determines the flooding level. Several other combinations are possible.

In such a framework, the diffusion process is responsible for the creation of good building blocks, while the flooding is responsible for quick selection of building blocks that contribute to the same object. An example clarifies this effect. Figure 16.1 shows an elongated white object. The lines across the object are example watersheds from a diffusion scheme.

It is inherently difficult for any diffusion scheme to capture elongated structure. Linear diffusion tends to capture approximately circular regions — the linking to localisation level allows the regions to get arbitrary shapes but they will still have the approximately same length and width. Anisotropic nonlinear diffusion schemes minimise this problem to some extent but do not remove it. Instead of being split into 10 regions, the white object would possibly only be split into 5 regions. Even though the inside of the object is quite uniform, the dissimilarity measure will have fluctuations that causes several local minima and thereby several catchment basins. However, a simple flooding scheme would allow the regions to be merged quite easy.

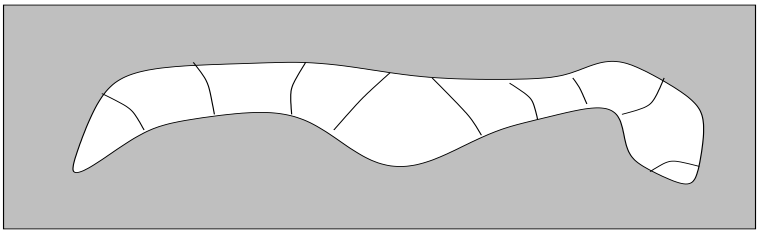


Figure 16.1: *Illustration of combination of diffusion and flooding techniques. The elongated structure is inherently difficult to capture by any diffusion scheme. A simple flooding scheme would allow the 10 regions to be selected in "one" action.*

The example is very simple. However, it illustrates the potential. The nonlinear diffusion schemes are very capable of preserving the edges and thereby create shapes with almost uniform intensity that correspond well to the objects of interest. These shapes are split into a number of regions by the watershed algorithm, but a simple flooding post-processing step can combine them into one single region.

Obviously, this presentation is simplified. The proper flooding level depends on the scale level and the specific image. For some image classes, the optimal flooding technique should be based on a measure determined by average intensity, for others by intensity variation, for yet others by the area or width of the individual regions.

Therefore, the best general flooding technique is possibly based on a simple measure determined by the depth of the regions in the dissimilarity measure. The appropriate flooding level should probably be determined interactively by a user — for instance through the duration of the mouse click.

It is certainly extremely interesting to explore how an added flooding-based selection technique can improve the segmentation method. However, the expectation is that this added step does not change the evaluation of the underlying diffusion schemes. The flooding-based selection will improve the usability of the method, but the quality of the underlying building blocks is still the key to the applicability of the method.

This is also the conclusion in [Lopez et al., 1999]. Here it is concluded that *"An application-dependent region merging postprocessing is also a good solution"* as a post-processing step for a *"suitable filtering scheme to remove local minima"*. Much like the combination outlined above.

Flooding in combination with watersheds is treated in [Meyer & Maragos, 1999].

Other approaches towards region merging are [Sporring & Olsen, 1999], [Pratikakis et al., 1999] and [Liao et al., 1999a] (in combination with [Liao et al., 1999b] and [Li et al., 1999])

16.2.2 The Space of Diffusion Schemes

Chapter 14 explores the possibility for a parameterisation of the space of diffusion schemes. The informal conclusion is that it is not realistic to reach an explicit parameterisation.

However, the seach for a proper characterisation of diffusion schemes seems more promising. The ratio of areas before and after the edge-defining parameter λ in section 14.1 is only intended as a preliminary attempt. No doubt, better and more general characterisations are possibly.

The ambitious goal is to combine the understanding of the space of diffusion schemes with equal understanding of image shapes. An ideal process would be:

- Determine the complexity and type of the shapes of the ground truth object.
- Determine the variation of the shapes and the noise level in the example images.
- Use the knowledge from the two previous steps to choose the proper parameter set for optimisation of a general diffusion scheme.

The design of the GAN scheme (section 14.3.2) can be viewed as a first tiny step in this direction. The parameters allow characterisation of the edges (through the λ parameter), and the diffusion (through the aggresiveness parameter m and the anisotropy parameter θ).

16.2.3 Generalisation of the Evaluation Results

The evaluation results are very satisfying. The GAN scheme implies that less than half as many actions are required compared to the original linear Gaussian diffusion scheme. Furthermore, through a simple heuristic, a user can limit the number of actions to one third above the optimal number.

However, these numbers are specific for segmentation of grey and white matter from MR brain scans. It is extremely interesting to explore whether this behaviour generalises to other classes of ground truth images.

If these properties generalise to other image classes, it implies that the behaviour is in a sense generic for the diffusion schemes. Thereby, the results would offer a qualitative characterisation of the deep structure implied by the diffusion schemes.

This requires ground truth data sets from other non-medical segmentation task. Preferably both data sets with artificial and natural environments.

The conclusions of passion are the only reliable ones.
Søren Kierkegaard

Chapter 17

Conclusion

The chapters 14 and 15 present the central discussions and results of this thesis. This chapter displays the most important conclusions.

17.1 Evaluation of the Diffusion Schemes

In Part One three evaluation methods are designed. The first measures the minimal effort required to reach a perfect segmentation. The second measures the optimal quality of the segmentation given a limited number of actions. The third measures the effort required to reach a perfect segmentation through the use of a simple heuristic.

The evaluation is based on ground truth segmentations of MR brain scans. The objects of interest are the grey and white matter tissues of the brain. Both simulated and real data are used in the evaluation.

Chapter 15 reveals that the evaluation results from the three methods are very similar. The original linear Gaussian scheme is outperformed by a number of nonlinear diffusion schemes. Among the existing schemes, the isotropic nonlinear Regularised Perona-Malik scheme performs best. However, the scheme with the best performance is the new Generalised Anisotropic Nonlinear diffusion scheme.

Compared to the original linear Gaussian scheme, the GAN scheme requires less than half the number of actions to reach a given segmentation quality. Specifically, on average 46% is required compared to the original scheme presented in [Olsen, 1996].

The ability to localise the borders of the ground truth object are similar for the linear Gaussian and the GAN schemes. If the imprecision of the ground truth data is assumed to correspond to a tolerance area of width one, both schemes are able to capture 97% of the area of the desired objects without the use of pixel level building blocks. However, the GAN scheme achieves this is in the less than half the number of actions compared to the linear Gaussian scheme.

Furthermore, the evaluation reveals that the simple heuristic strategy requires approximately one third extra actions compared to the optimal strategy.

17.2 Diffusion

This thesis explores the effects of diffusion on images.

It is not trivial to characterise this effect. A possible way is to study the deep structure of image features. An example of an uncommitted image feature is the contrast. The deep structure of this image feature can be studied through the development of the gradient magnitude watersheds over scale.

The next step is to characterise this development. Since the watersheds correspond to regions boundaries it is interesting to see how these regions correspond to the salient objects in the image. The interesting objects in an image is defined by a specific task. Different tasks perceive different objects as the interesting ones.

In this thesis the interesting objects are defined by ground truth segmentations of the grey and white matter areas in MR scan of the brain. The deep structure of the image features is then characterised by the ability to perform segmentation of the ground truth objects.

The evaluation of the diffusion schemes with respect to segmentation directly reveals the best suited diffusion scheme. However, the design of the GAN scheme allows an inspection of the most important features of the diffusion scheme.

The conclusion is that nonlinear diffusion is significantly superior to linear diffusion. Nonlinearity is essential for the ability to capture the shape of the ground truth objects. Somewhat surprisingly, the anisotropy seems to be insignificant for the diffusion. The GAN scheme achieves the improvement compared to the RPM scheme through optimisation of the aggressiveness of the edge enhancement — not through anisotropy. For all data sets, the optimal anisotropy factor θ is zero or close to zero.

17.3 The Contributions of this Thesis

The main contributions of this thesis are:

Evaluation
: The design of a general evalution method allows a proper evalution of the segmentation method designed in [Olsen, 1996].

Customisation
: The evaluation and the optimisation methods allow the segmentation method to be specialised with respect to a given segmentation task.

Improvement
: The thesis improves the original segmentation method through the selection of the best suited diffusion scheme. Thereby, compared to the original method, less than half the actions are required to perform a segmentation task.

Design of GAN
A new diffusion scheme is designed. The scheme is a simple generalisation of existing schemes. The performance of the scheme is approximately 10% better than the best of the existing schemes. Furthermore, the design of the scheme reveals the important factors of diffusion with respect to watershed segmentation.

Design of CID
The scheme is not evaluated but the local steering of the anisotropy shows promising image enhancement properties.

Diffusion
The thesis provides a little additional insight in the nature of the deep structure of the diffusion schemes. It appears that the significance of anisotropy is limited compared to the evident significance of nonlinearity.

17.4 The Final Example

The presentation of the results and the conclusion are concentrated on graphs, numbers and diffusion schemes. In order to appreciate the consequences for the segmentation method, the thesis is concluded with an example.

The white matter tissue of the center slice from the BW9W data set is segmented using the heuristic selection strategy from evaluation method 2. This is done where the building blocks are the results of diffusion with the linear Gaussian scheme (optimised with a maximal action limit of 80). For comparison it is repeated where the building blocks are the results of diffusion with the GAN scheme.

Figure 17.1 displays the brain scan slice and the corresponding ground truth segmentation of the white matter tissue.

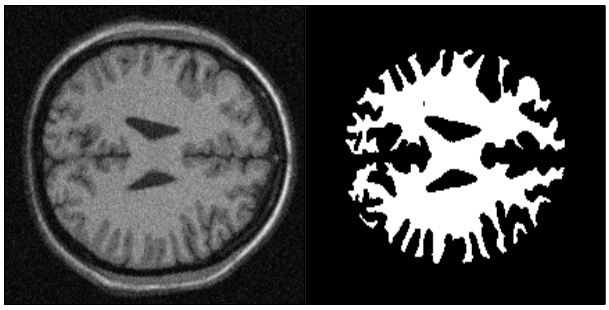


Figure 17.1: *The center slice of data set BW9W with the corresponding white matter ground truth segmentation.*

The segmentation processes are illustrated by an intermediary step, where a large part of the ground truth object is captured, and by the end result. The end result is defined as the best segmentation possible without the use of pixel sized building blocks.

Figure 17.2 shows the intermediary step for the segmentation based on the LG scheme. The selection strategy starts at scale level 30 and works its way down the scale levels. The illustration is from scale level 16 where 70.5% of the ground truth object is recovered through 20 selections and 16 deselections — a total of 36 actions.

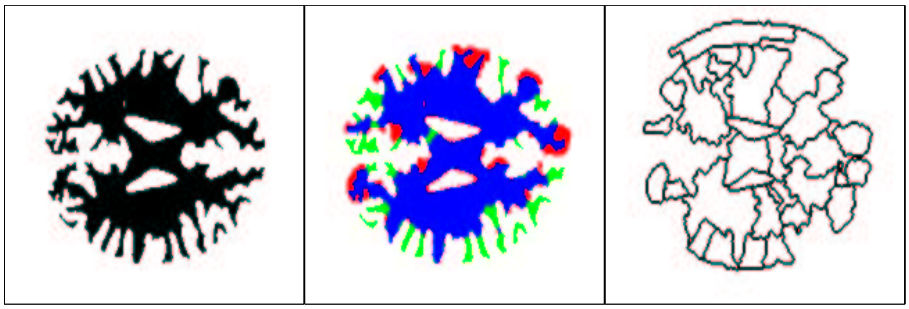


Figure 17.2: *The segmentation based on the LG scheme at scale level 16 where* 70.5% *of the ground truth object is segmented through 36 actions.*
The ground truth object is displayed to the left.
The center image is the current state of the segmentation process. The blue area is the part of the ground truth object that is correctly recovered. The green area is the part of the ground truth object that has not been captured yet. The red area is the part that is selected but does not belong to the ground truth object.
The right image shows the outline of the building blocks that is used in the segmentation process. The large building blocks from the higher scale levels are split in this illustration. For instance, the top region is part of a large region selected at high scale. The region displayed is the building block at a lower level that is deselected in order to remove an undesired part of the larger building block.

Figure 17.3 shows the end result for the segmentation based on the LG scheme. It is possible to capture 89.8% of the ground truth object without use of pixel-sized building blocks. The total number of actions is 174.

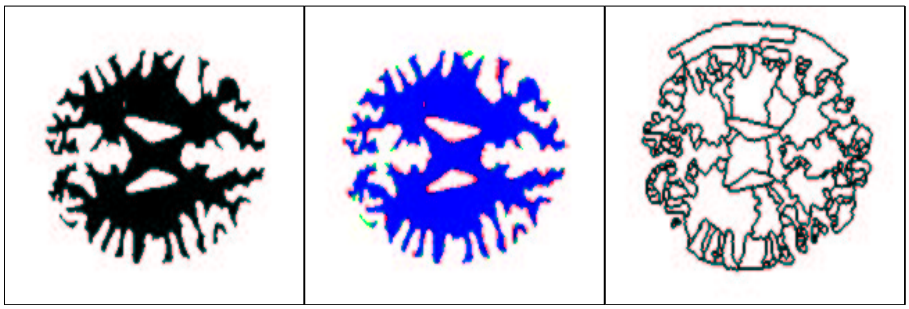


Figure 17.3: *The end result for the segmentation based on the LG scheme. Without the use of pixel level building blocks* 89.8% *is captured through 100 selections and 74 deselections.*

For comparison, the segmentation is performed using the building blocks from the GAN scheme. The intermediary step displayed in figure 17.4 is from level 30 — the highest scale level. Here, 71.2% of the ground truth object is recovered through 11 selections.

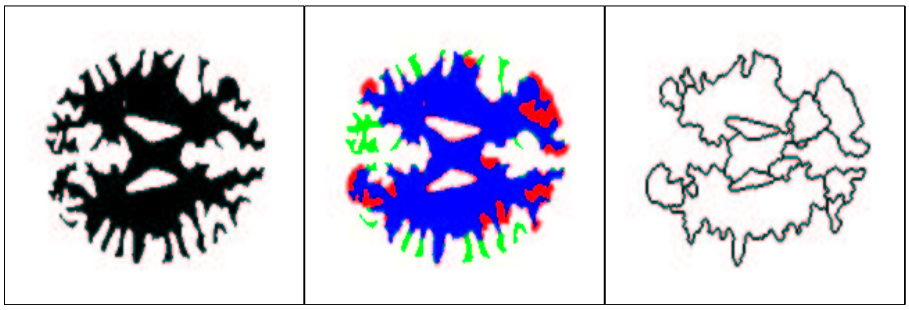


Figure 17.4: *The segmentation based on the GAN scheme at scale level 30 where* 71.2% *of the ground truth object is segmented through 11 selections.*

The end result for the GAN scheme segmentation is displayed in figure 17.5. A total of 89.9% of the ground truth object is recovered through 117 actions.

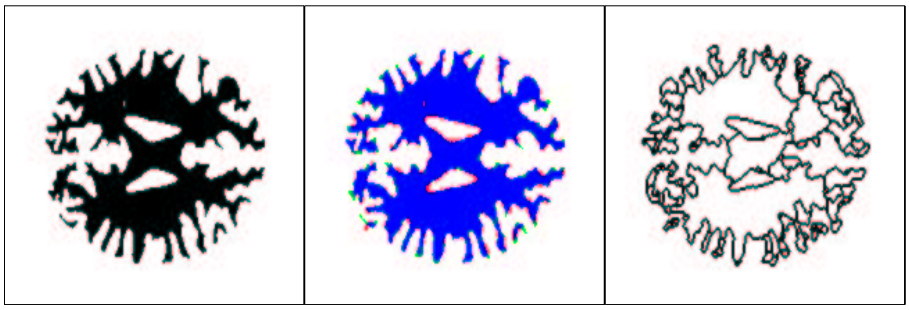


Figure 17.5: *The end result for the segmentation based on the GAN scheme. After a total of 55 selections and 62 deselections* 89.9% *of the ground truth object is recovered without the use of pixel level actions.*

Both schemes allow the user to capture close to 90% of the ground truth object without the use of pixel level building blocks. However, the GAN scheme performs this in 117 actions compared to the 174 actions of the LG scheme.

A striking qualitative difference is the appearance of the building blocks used. The blocks used in the GAN scheme has a more intuitive correspondance with the ground truth object. Large blocks are used for the main part of the object; small blocks are used to refine the details along the tentacles. The building blocks used by the LG scheme leaves a more confusing impression.

Possibly this qualitative difference will cause a user to feel more comfortable with the GAN scheme building blocks. But that is certainly beyond the scope of this thesis!

Appendix A

Optimised Parameter Sets

The parameter sets that result from the optimisation process are displayed for each diffusion scheme for each data set.

The parameters σ, λ, θ, and m are explained in the relevant chapters.

For the Gaussian schemes, the "From" and "To" parameters are the scale for the lowest and highest scale level. The "Steps" parameter is the number of logarithmically distributed scale levels.

For the remaining schemes, the "From" and "To" parameters are the number of iteration for the lowest and highest scale level. The "Step" parameter determines the time step size for each iteration (using an explicit iteration scheme). The "Steps" parameter is again the number of logarithmically distributed scale levels.

LG — Linear Gaussian Diffusion

Data	From	To	Steps
BW1G	0.70	15.00	30.00
BW1W	1.00	15.00	30.00
BW9G	1.00	12.00	30.00
BW9W	1.90	21.00	30.00
IRNG	0.70	25.50	30.00
IRNW	1.90	12.00	30.00
IRSG	0.70	16.50	30.00
IRSW	1.15	12.00	30.00

LG — Increased Maximal Action Limit on BW9W

Maximal Action Limit	From	To	Steps
30	1.90	21.00	30.00
80	0.85	20.00	30.00
300	0.70	18.50	30.00

ILD — Isotropic Linear Diffusion

Data	From	To	Steps	σ	Step
BW1G	1.00	1420.00	30.00	0.40	0.20
BW1W	5.00	1500.00	30.00	1.00	0.20
BW9G	3.00	1580.00	30.00	0.40	0.20
BW9W	3.00	1380.00	30.00	1.00	0.20
IRNG	1.00	1380.00	30.00	0.20	0.20
IRNW	5.00	1500.00	30.00	1.00	0.20
IRSG	1.00	1620.00	30.00	0.40	0.20
IRSW	1.00	1660.00	30.00	1.00	0.20

RF2 — Second Order Recursive Filter Linear Gaussian

Data	From	To	Steps
BW9G	1.45	21.00	30.00
BW9W	1.45	21.00	30.00
IRNG	1.45	15.50	30.00
IRNW	1.75	13.50	30.00

RF4 — Fourth Order Recursive Filter Linear Gaussian

Data	From	To	Steps
BW9G	1.00	12.00	30.00
BW9W	2.20	21.00	30.00
IRNG	0.55	21.00	30.00
IRNW	1.90	12.00	30.00

IND — Isotropic Nonlinear Diffusion

Data	From	To	Steps	σ	λ	Step
BW9G	4.00	1460.00	30.00	2.20	5.30	0.20
BW9W	4.00	1500.00	30.00	1.80	6.50	0.20
IRNG	6.00	1580.00	30.00	3.00	6.20	0.20
IRNW	11.00	1500.00	30.00	0.40	4.40	0.20

EE2 — Edge Enhancing Diffusion

Data	From	To	Steps	σ	λ	Step
BW9G	5.00	1410.00	30.00	0.80	3.70	0.20
BW9W	8.00	1420.00	30.00	0.80	4.70	0.20
IRNG	10.00	1570.00	30.00	0.40	6.10	0.20
IRNW	11.00	1450.00	30.00	0.20	4.30	0.20

EE3 — Edge Enhancing Diffusion

Data	From	To	Steps	σ	λ	Step
BW9G	6.00	1540.00	30.00	1.00	3.50	0.20
BW9W	7.00	1500.00	30.00	0.60	4.40	0.20
IRNG	7.00	1570.00	30.00	0.40	7.00	0.20
IRNW	5.00	1460.00	30.00	0.40	4.10	0.20

EE4 — Edge Enhancing Diffusion

Data	From	To	Steps	σ	λ	Step
BW1G	6.00	1620.00	30.00	0.70	4.00	0.20
BW1W	5.00	1460.00	30.00	0.40	4.40	0.20
BW9G	3.00	1500.00	30.00	1.00	3.50	0.20
BW9W	5.00	1500.00	30.00	1.00	3.50	0.20
IRNG	10.00	1610.00	30.00	0.40	6.10	0.20
IRNW	7.00	1460.00	30.00	0.40	4.70	0.20
IRSG	5.00	1620.00	30.00	1.20	4.40	0.20
IRSW	4.00	1500.00	30.00	0.40	4.40	0.20

MCM — Mean Curvature Motion

Data	From	To	Steps	σ	Step
BW9G	4.00	1540.00	30.00	1.60	0.20
BW9W	4.00	1820.00	30.00	2.20	0.20
IRNG	1.00	1660.00	30.00	1.60	0.20
IRNW	7.00	1500.00	30.00	1.00	0.20

RPM — Regularised Perona-Malik diffusion

Data	From	To	Steps	σ	λ	Step
BW1G	3.00	1460.00	30.00	0.40	3.20	0.20
BW1W	3.00	1460.00	30.00	0.20	2.00	0.20
BW9G	5.00	1740.00	30.00	0.60	2.30	0.20
BW9W	8.00	1460.00	30.00	0.40	1.70	0.20
IRNG	7.00	1500.00	30.00	0.40	3.20	0.20
IRNW	8.00	1700.00	30.00	0.40	2.60	0.20
IRSG	3.00	1400.00	30.00	0.50	2.40	0.20
IRSW	4.00	1660.00	30.00	0.20	1.70	0.20

APM — Anisotropic Perona-Malik diffusion

Data	From	To	Steps	σ	λ	Step
BW9G	7.00	1610.00	30.00	0.30	2.00	0.20
BW9W	9.00	1690.00	30.00	0.50	1.10	0.20
IRNG	4.00	1570.00	30.00	0.70	2.30	0.20
IRNW	4.00	1450.00	30.00	0.30	2.00	0.20

GAN — Generalised Anisotropic Nonlinear diffusion

Data	From	To	Steps	σ	λ	Step	θ	m
BW9G	6.00	1660.00	30.00	0.40	3.30	0.20	0.12	1.00
BW9W	13.00	1480.00	30.00	0.10	3.30	0.20	0.06	1.40
IRNG	8.00	1500.00	30.00	0.40	10.20	0.20	0.00	1.00
IRNW	9.00	1685.00	30.00	0.30	8.00	0.20	0.00	1.10

GAN — Increased Maximal Action Limit on BW9W

Maximal Action Limit	From	To	Steps	σ	λ	Step	θ	m
30	13.00	1480.00	30.00	0.10	3.30	0.20	0.06	1.40
80	20.00	1540.00	30.00	0.50	3.90	0.20	0.06	1.40
300	16.00	1560.00	30.00	0.50	4.00	0.20	0.12	1.50

Bibliography

[Almansa & Lindeberg, 1997] Andrés Almansa & Tony Lindeberg. *Enhancement of Fingerprint Images using Shape-Adapted Scale-Space Operators.* In Jon Sporring, Mads Nielsen, Luc Florack, & Peter Johansen, editors, *Gaussian Scale-Space Theory*, pages 21–30. Kluwer, 1997.

[Alvarez & Morel, 1994] Luis Alvarez & Jean-Michel Morel. *Geometry-Driven Diffusion in Computer Vision*, chapter Morphological Approach to Multiscale Analysis: From Principles to Equations. Kluwer Academic Publishers, 1994.

[Alvarez et al., 1993] L. Alvarez, F. Guichard, P.-L. Lions, & J.-M. Morel. *Axioms and Fundamental Equations in Image Processing.* Arc. Rational Mech. Anal., 123:199—257, 1993.

[Atkins & Mackiewich, 1996] M. Stella Atkins & Blair T. Mackiewich. *Automatic Segmentation of the Brain in MRI.* In Karl Heinz Höhne & Ron Kikinis, editors, *Visualization in Biomedical Computing*, volume 1131 of Lecture Notes in Computer Science, pages 241–246. Springer, 1996.

[Bouma et al., 1996] Carolien J. Bouma, Wiro J. Niessen, Karel J. Zuiderveld, Elma J. Gussenhoven, & Max A. Viergever. *Evaluation of Segmentation Algorithms for Intravascular Ultrasound Images.* In Karl Heinz Höhne & ron Kikinis, editors, *Visualization in Biomedical Computing*, volume 1131 of Lecture Notes in Computer Science. Springer, 1996.

[Bruce & Giblin, 1992] J.W. Bruce & P.J. Giblin. *Curves and Singularities.* Cambridge University Press, second edition, 1992.

[Catté et al., 1992] F. Catté, P.-L. Lions, J.-M. Morel, & T. Coll. *Image Selective Smoothing and Edge Detection by Nonlinear Diffusion.* SIAM Journal of Numerical Analysis, 29:182–193, 1992.

[Cayley, 1859] A. Cayley. *On Contour and Slope Lines.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 18(120):264–268, October 1859.

[Chung & Sapiro, 1999] Do Hyun Chung & Guillermo Sapiro. *A Windows-based User Friendly System for Image Analysis with Partial Differential Equations.* In Mads Nielsen, Peter Johansen, Ole Fogh Olsen, & Joachim Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, pages 453–458. Springer, 1999.

[Cocosco et al., 1997] C.A. Cocosco, V. Kollokian, R.K.-S. Kwan, & A.C. Evans. *BrainWeb: Online Interface to a 3D MRI Simulated Brain Database.* In *Proceedings of 3rd International Conference on Functional Mapping of the Human Brain*, volume 5 of NeuroImage, page 425, May 1997. http://www.bic.mni.mcgill.ca/brainweb/.

[Collins et al., 1998] D.L. Collins, A.P. Zijdenbos, V. Kollokian, J.G. Sled, N.J. Kabani, C.J. Holmes, & A.C. Evans. *Design and Construction of a Realistic Digital Brain Phantom.* IEEE Transactions on Medical Imaging, 17(3):463–468, June 1998. http://www.bic.mni.mcgill.ca/brainweb/.

[Dam & Lillholm, 1999] Erik B. Dam & Martin Lillholm. *Generic Events for the Isophote Curvature.* Student project at the Department of Computer Science, University of Copenhagen. A copy can be obtained through the authors: `erikdam@diku.dk` or `grumse@itu.dk`, 1999.

[Dam et al., 1998] Erik B. Dam, Martin Koch, & Martin Lillholm. *Quaternions, Interpolation and Animation.* Technical Report 98/5, Department of Computer Science, University of Copenhagen, 1998.

[Dam et al., 2000] Erik Dam, Peter Johansen, Ole Fogh Olsen, Andreas Thomsen, Tron Darvann, Andy B. Dobrzeniecki, Nuno V. Hermann, Noriyuki Kitai, Sven Kreiborg, Per Larsen, & Mads Nielsen. *Interactive Multi-Scale Segmentation in Clinical Use.* In *European Congress of Radiology 2000*, March 2000. Abstract and video accepted for CompuRAD.

[Damon, 1995] James Damon. *Local Morse Theory for Solutions to the Heat Equation and Gaussian Blurring.* Journal of Differential Equations, 115(2):368 – 401, January 1995.

[Damon, 1997] James Damon. *Gaussian Scale-Space Theory*, chapter Local Morse Theory for Gaussian Blurred Functions, pages 147 – 163. Kluwer, 1997.

[Deriche, 1990] Rachid Deriche. *Fast Algorithms for Low-Level Vision.* IEEE PAMI, 12(1):78 — 87, 1990.

[Deriche, 1993] Rachid Deriche. *Recursively Implementing the Gaussian and its Derivatives.* Technical Report 1893, INRIA – Sophia Antipolis, 1993.

[Florack, 1993] Luc Florack. *The Syntactical Structure of Scalar Images.* PhD thesis, Universiteit Utrecht, 1993.

[Florack, 1996] L.M.J. Florack. *Data, Models, and Images.* In *IEEE Conference on Image Processing*, pages 469–472, 1996.

[Gauch & Pizer, 1993] John M. Gauch & Stephen M. Pizer. *Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(6):635–646, June 1993.

[Gauch, 1999] John M. Gauch. *Image Segmentation and Analysis via Multiscale Gradient Watershed Hierachies.* IEEE Transactions on Image Processing, 8(1):69 – 79, January 1999.

[Gilmore, 1993] Robert Gilmore. *Catastrophe Theory for Scientists and Engineers.* Dover Publications, New York, 1. edition, 1993.

[Guichard & Morel, 1995] F. Guichard & J.M. Morel. *Partial Differential Equations and Image Iterative Filtering.* Technical Report 9535, Ceremade, Université de Paris IX - Dauphine, 1995.

[IBS, 1999] *The Internet Brain Segmentation Repository*, 1999. `http://neuro-www.mgh.harvard.edu/cma/ibsr`.

[Johansen et al., 1999] Peter Johansen, Mads Nielsen, & Sven Kreiborg. *The Computation of Natural Shape*, 1999. `http://www.diku.dk/research-groups/image/research/NaturalShape/`.

[Kaus et al., 1998] M. Kaus, S. Warfield, F. Jolesz, & R. Kikinis. *Adaptive Template Moderated Tumor Segmentation from Magnetic Resonance Images.* Medical Imaging Analysis, 1998.

[Kimia & Siddiqi, 1996] Benjamin B. Kimia & Kaleem Siddiqi. *Geometric Heat Equation and Nonlinear Diffusion of Shapes and Images.* Computer Vision and Image Understanding, 64(3):305–322, 1996.

[Koenderink, 1984] Jan J. Koenderink. *The Structure of Images.* Biological Cybernetics, 50:363–370, 1984.

[Koster, 1995] André Koster. *Linking Models for Multi-scale Image Sequences.* PhD thesis, University of Utrecht, 1995.

[Krissian et al., 1997] Karl Krissian, Gregoire Malandain, & Nicholas Ayache. *Directional Anisotropic Diffusion Applied to Segmentation of Vessels in 3D Images.* In Bart ter Haar Romeny, Luc Florack, Jan Koenderink, & Max Viergever, editors, *Scale-Space Theory in Computer Vision, First International Conference, Proceedings*, number 1252 in Lecture Notes in Computer Science, pages 345–348. Springer, 1997.

[Kwan et al., 1996] R.K.-S. Kwan, A.C. Evans, & G.B. Pike. *An Extensible MRI Simulator for Post-Processing Evaluation.* Lecture Notes in Computer Science, 1131:135–140, 1996. http://www.bic.mni.mcgill.ca/brainweb/.

[Leclerc, 1989] Yvan G. Leclerc. *Constructing Simple Stable Descriptions for Image Partitioning.* International Journal of Computer Vision, 3:73–102, 1989.

[Leclerc et al., 1999] Yvan G. Leclerc, Q.-Tuan Luong, & Pascal Fua. *Characterizing the performance of Multiple-image Point-correspondance Algorithms using Self-Consistency.* In Bill Triggs, Richard Szeliski, & Andrew Zisserman, editors, *Vision Algorithms — Theory & Practice*, pages 51–58, 1999. Final proceedings are to appear in Springer LNCS in 2000.

[Li et al., 1999] Yi Li, Ming Liao, Songde Ma, & Hanqing Lu. *Robust Estimation of Global Motion Using Gradient Watershed.* E-mail: `liyi@nlpr.ia.ac.cn`, 1999.

[Liao et al., 1999a] Ming Liao, Yi Li, Songde Ma, & Hanqing Lu. *Robust Tracking of Video Objects through Topological Constraints on Homogeneous Motion.* E-mail: `mliao@nlpr.ia.ac.cn`, 1999.

[Liao et al., 1999b] Ming Liao, Yi Li, Songde Ma, & Hanqing Lu. *Scale Space of Gradient Watershed.* E-mail: `mliao@nlpr.ia.ac.cn`, 1999.

[Lifshitz & Pizer, 1990] Lawrence Lifshitz & Stephen Pizer. *A Multiresolution Hierarchical Approach to Image Segmentation Based on Intensity Extrema.* IEEE PAMI, 12(6):529 – 540, June 1990.

[Lindeberg, 1990] Tony Lindeberg. *Scale-Space for Discrete Signals.* IEEE PAMI, 12:234–254, 1990.

[Lindeberg, 1994] Tony Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[Lindeberg, 1996a] Tony Lindeberg. *Edge Detection and Ridge Detection with Automatic Scale Selection.* Technical Report ISRN KTH/NA/P-96/06-SE, Royal Institute of Technology, Sweden, 1996.

[Lindeberg, 1996b] Tony Lindeberg. *Scale-space: A framework for handling image structures at multiple scales.* In *Proceedings for CERN School of Computing*, Egmond aan Zee, The Netherlands, September 1996.
`http://www.nada.kth.se/~tony/cern-review/cern-html/cern-html.html`.

[Lopez et al., 1999] Antonio M. Lopez, Felipe Lumbreras, John Serrat, & Juan J. Villanueva. *Evaluation of Methods for Ridge and Valley Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4):327–335, April 1999.

[Lorenz et al., 1997a] C. Lorenz, I. C. Carlsen, T. M. Buzug, C. Fassnacht, & J. Weese. *A Multi-scale Line Filter With Automatic Scale Selection Based on the Hessian Matrix for Medical Image Segmentation.* In Bart ter Haar Romeny, Luc Florack, Jan Koenderink, & Max Viergever, editors, *Scale-Space Theory in Computer Vision, First International Conference, Proceedings*, number 1252 in Lecture Notes in Computer Science, pages 152–163. Springer, 1997.

[Lorenz et al., 1997b] C. Lorenz, I. C. Carlsen, T. M. Buzug, C. Fassnacht, & J. Weese. *A Multi-scale Line Filter With Automatic Scale Selection Based on the Hessian Matrix for Medical Image Segmentation.* In *Computer Vision, Virtual Reality and Robotics in Medicine, CVRMed, Proceedings*, number 1205 in Lecture Notes in Computer Science, pages 233–242. Springer-Verlag, Berlin Heidelberg, April 1997.

[Malladi et al., 1995] R. Malladi, J. A. Sethian, & B. C. Vemuri. *Shape Modeling with Front Propagation: A Level Set Approach.* IEEE PAMI, 17(2):158–175, 1995.

[Maxwell, 1870] J.C. Maxwell. *On Hills and Dales.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 4th Series, 40(269):421–425, December 1870.

[Meyer & Maragos, 1999] Fernand Meyer & Petros Maragos. *Multiscale Morphological Segmentations Based on Watershed, Flooding, and Eikonal PDE.* In Mads Nielsen, Peter Johansen, Ole Fogh Olsen, & Joachim Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, pages 351–362. Springer, 1999.

[Moravec, 1999] Hans Moravec. *Rise of the Robots.* Scientific American, pages 86 – 93, December 1999.

[Morse, 1931] Marston Morse. *The Critical Points of a Function of n variables.* Transactions of American Mathematical Society, 33:72 – 91, 1931.

[Nielsen et al., 1997] M. Nielsen, L. Florack, & R. Deriche. *Regularization, Scale-Space and Edge Detection Filters.* Mathematical Imaging and Vision, 7:291–307, 1997.

[Niessen et al., 1997] Wiro J. Niessen, Koen L. Vincken, Joachim A. Weickert, & Max A. Viergever. *Nonlinear Multiscale Representations for Image Segmentation.* Computer Vision and Image Understanding, 66(2):233–245, May 1997.

[Nitzberg & Shiota, 1992] Mark Nitzberg & Takahiro Shiota. *Nonlinear Image Filtering with Edge and Corner Enhancement.* IEEE Tr. Pattern Analysis and Machine Intelligence, 14(8):826 – 833, 1992.

[Olsen & Nielsen, 1997] Ole Fogh Olsen & Mads Nielsen. *Generic events for the gradient squared with application to multi-scale segmentation.* In *Scale-Space Theory in Computer Vision, Proc. 1st International Conference*, volume 1252 of Lecture Notes in Computer Science, pages 101–112, Utrecht, The Netherlands, July 1997.

[Olsen, 1996] Ole Fogh Olsen. *Multi-Scale Segmentation of Grey-Scale Images.* Technical Report 96/30, Department of Computer Science, University of Copenhagen, 1996.

[Olsen, 1997] Ole Fogh Olsen. *Multi-Scale Watershed Segmentation.* In Jon Sporring, Mads Nielsen, Luc Florack, & Peter Johansen, editors, *Gaussian Scale-Space Theory*, pages 191–200. Kluwer, 1997.

[Oomes & Snoeren, 1996] A.H.J. Oomes & P.R. Snoeren. *Structural Information in Scale-Space.* In *Classical Scale-Space Theory, PhD-summerschool, University of Copenhagen.* 1996.

[Ott, 1993] Edward Ott. *Chaos in Dynamical Systems.* Cambridge University Press, 1993.

[Pedersen & Nielsen, 1999] Kim S. Pedersen & Mads Nielsen. *The Hausdorff Dimension and Scale-Space Normalization of Natural Images.* In Mads Nielsen, Peter Johansen, Ole Fogh Olsen, & Joachim Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, pages 271–282. Springer, 1999.

[Perona & Malik, 1990] Pietro Perona & Jitendra Malik. *Scale-Space and Edge Detection Using Anisotropic Diffusion.* IEEE PAMI, 12(7):629 – 639, July 1990.

[Pratikakis et al., 1999] I. Pratikakis, H. Sahli, & J. Cornelis. *Hierarchical segmentation using dynamics of multiscale gradient watersheds.* In Bjarne Kjær Ersbøll & Peter Johansen, editors, *The 11th Scandinavian Conference on Image Analysis*, pages 577–584, June 1999.

[Samet, 1984] H. Samet. *The Quadtree and Related Hierarchical Data Structures.* Surveys, 16(2):187–260, June 1984.

[Sethian, 1996] J. A. Sethian. *Level Set Methods.* Cambridge University Press, 1996.

[SPL, 1999] *The SPL & NSG Brain Tumor Database*, 1999. Contact Marianna Jakab (`marianna@bwh.harvard.edu`) at Brigham and Women's Hospital, Boston (`http://splweb.bwh.harvard.edu`).

[Sporring & Olsen, 1999] Jon Sporring & Ole Fogh Olsen. *Segmenting by Compression Using Linear Scale-Space and Watersheds.* In Mads Nielsen, Peter Johansen, Ole Fogh Olsen, & Joachim Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, pages 513–518. Springer, 1999.

[Sporring, 1996] Jon Sporring. *The Entropy of Scale-Space.* In *Classical Scale-Space Theory, PhD-summerschool, University of Copenhagen.* 1996.

[Sporring et al., 1997] Jon Sporring, Mads Nielsen, Luc Florack, & Peter Johansen. *Gaussian Scale-Space Theory.* Kluwer Academic Publishers, Dordrecht, 1997.

[Sporring et al., 1998] Jon Sporring, Mads Nielsen, Joachim Weickert, & Ole Fogh Olsen. *A Note on Differential Corner Measures.* Technical Report DIKU-98/1, Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2200 Copenhagen East, Denmark, 1998.

[Szeliski & Zabih, 1999] Richard Szeliski & Ramin Zabih. *An Experimental Comparison of Stereo Algorithms.* In Bill Triggs, Richard Szeliski, & Andrew Zisserman, editors, *Vision Algorithms — Theory & Practice*, pages 59–66, 1999. Final proceedings are to appear in Springer LNCS in 2000.

[ter Haar Romeny, 1994] Bart M. ter Haar Romeny. *Geometry Driven Diffusion.* Kluwer Academic Publishers, Dordrecht, 1994.

[VHP, 1995] *The Visible Human Project*, 1995. `http://www.nlm.nih.gov/research/visible/visible_human.html`.

[Vincken, 1995] Koen Vincken. *Probabilistic Multi-scale Image Segmentation by the Hyperstack.* PhD thesis, University of Utrecht, 1995.

[Weickert & Benhamouda, 1997] Joachim Weickert & Brahim Benhamouda. *Why the Perona-Malik Filter Works.* Technical Report 97/22, Department of Computer Science, University of Copenhagen, 1997.

[Weickert, 1998a] Joachim Weickert. *Anisotropic Diffusion in Image Processing.* B. G. Teubner, Stuttgart, 1998.

[Weickert, 1998b] Joachim Weickert. *Efficient Image Segmentation Using Partial Differential Equations and Morphology.* Technical Report 98-10, Department of Computer Science, University of Copenhagen, 1998.

[Weickert, 1998c] Joachim Weickert. *Fast Segmentation Methods Based on Partial Differential Equations and the Watershed Transformation.* In P. Levi, R.-J. Ahlers, F. May, & M. Schanz, editors, *Mustererkennung 1998*, pages 93–100. Springer, 1998.

[Weickert, 1999] Joachim Weickert. *Coherence-Enhancing Diffusion Filtering.* International Journal of Computer Vision, 31:111–127, 1999.

[Weickert et al., 1997] J. Weickert, S. Ishikawa, & A. Imiya. *Scale-space has been Discovered in Japan.* Technical Report 18, Department of Computer Science, University of Copenhagen, 1997.

[Witkin, 1983] Andrew P. Witkin. *Scale-Space Filtering.* In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.