Segmentation of elongated structures in medical images

Joes Staal

This book was typeset by the author using LaTeX 2ε.

Front cover: *De Denker* by Bas Sebus

ISBN 90–393–3842–6

The author can be reached at joes@staalkemade.nl

Printed by PrintPartners Ipskamp, Enschede, the Netherlands.

Segmentation of elongated structures in medical images

Segmentatie van langgerekte structuren in medische beelden

(met een samenvatting in het Nederlands)

PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR AAN DE UNIVERSITEIT UTRECHT OP GEZAG VAN DE RECTOR MAGNIFICUS, PROF. DR. W. H. GISPEN, INGEVOLGE HET BESLUIT VAN HET COLLEGE VOOR PROMOTIES IN HET OPENBAAR TE VERDEDIGEN OP WOENSDAG 13 OKTOBER 2004 DES MIDDAGS TE 2:30 UUR

DOOR

Jozef Johannes Staal

geboren op 2 maart 1971 te Brielle.

Promotor:	prof. dr. ir. M. A. Viergever University Medical Center Utrecht
Copromotores:	dr. B. van Ginneken University Medical Center Utrecht
	dr. S. N. Kalitzin Dutch Epilepsy Clinics Foundation

The research described in this thesis was carried out at the Image Sciences Institute, University Medical Center Utrecht, Utrecht, the Netherlands, under the auspices of ImagO, the Utrecht graduate school for Biomedical Image Sciences. The project was financially supported by the Netherlands Organization for Scientific Research (NWO).

Financial support for the publication of this thesis was kindly provided by Philips Medical Systems Nederland B.V. (Medical IT – Advanced Development), ImagO and the Röntgen Stichting Utrecht.

To Pippi and Scooter

Contents

Chapter 1

Introduction

1.1 Motivation

HUMAN beings are excellent signal processing units in terms of qualitative measurements. A brief glance at a family picture suffices to interpret the scene, to tell how many people are visible or to identify acquaintances. When listening to music we recognize melodies, feel rhythms and can tell if the music is contemporary or not. Effortlessly, we understand people speaking to us and can respond to them.

However, for making quantitative measurements we have to rely on specialized tools, like a rod, a pair of scales or a spectrometer. Standardizing quantitative measurements is difficult: the accuracy and precision of the measurements are influenced by the skills and condition of the observer. We are also not well adapted for more complex tasks, like searching for a specific pattern in a large amount of images. The time needed might be prohibitive and there is a high chance of missing or falsely signalling patterns, especially if the pattern is difficult to identify.

Machines that could take over such tasks have great advantages. They do not get tired (although they need maintenance), are not disturbed by other tasks or their mental state and, if constructed correctly, deliver reproducible results. Often machines can perform tasks much faster than a human being.

In the field of medical image analysis and processing ample use is made of such machines: computers. Nowadays, there is a large variety of medical image data consisting of X-ray images, ultrasound images, MRI (magnetic resonance imaging) scans, and CT (computed tomography) scans, to mention a few. Computers are used for aiding in diagnosis (although not too often, yet), e.g. to determine the presence of a tumor, for planning operations, e.g. to establish the size of a stent that is needed, and for treatment, e.g. to let a radiotherapist pinpoint the exact location and shape of a tumor before radiotherapy is applied.

In this thesis methods are developed that can facilitate certain medical image processing applications. Of special interest is the detection of elongated structures, such as vessels and bones.

1.2 Methodology

Computers work with discretized data and images are commonly represented on a rectangular grid. For two-dimensional images a point on the grid is called a pixel, which is an abbreviation of **pict**ure **el**ement. In higher dimensional images a grid-point is referred to as a voxel[1]. At every grid-point a color or a gray-value is present.

When we look at an image on our monitor, we are normally not aware of the grid on which the image is represented. In a sense, the pixel grid is artificial: it is a *representation* to store an image in the computer's memory. The pixel values themselves give no "information" about the picture, it is their constellation that gives a picture its "meaning". With meaning we denote the exact location and shape or the recognition of an object. Compare the grid representation with written text: we use letters to write down words, but the letters in their own right have no meaning, it is the words they constitute which we understand.

The example of the letters introduces another important concept: bottom-up and top-down approaches. In a bottom-up approach we start with letters, which form words, which form sentences, which form paragraphs, which form sections, which form chapters, which form a book. In a top-down approach we have a book, which breaks up in chapters, which break up in sections, etc. Reading a book is a bottom-up procedure, while writing a book is more a top-down activity.

In this thesis the emphasis is on bottom-up approaches. We start with the pixel grid representation and continue from there to construct other representations. As in the book example, we will obtain a chain of representations which derive from each other. The notion of small basic entities forming entities on a higher level is called *grouping*. The entities on the next higher level are more *global* representations than the entities on the previous level. But they form the *local* entities for the next higher level. Examples of this concept are abundant and the most important is probably the following: quarks forming electrons, neutrons and protons, which together form atoms, which form molecules, which form cells, which form organs, which form organisms, which form societies.

The focus in this thesis is on the detection of elongated structures in medical images. More precise, we want to detect elongated structures of a certain nature, e.g. vessels. So we do not only want to detect elongated structures, we also want to recognize if they belong to a certain class. To distinguish between vessel and non-vessel structure use can be made of classifiers. Classifiers are tools that use properties or features of an object to assign that object to a specific category. Human beings act like classifiers when they recognize an apple from a pear based on features like color, shape and taste. The reason that we can distinguish between such objects, is that we have learned to use the features to draw a conclusion. Learning and the availability of example data is an important factor in the construction of classifiers.

In the case of detecting vessels, we could start with computing a lot of features

[1] Voxel is an incorrect abbreviation of volume element.

for every pixel in the image (supposing examples are available and we have an idea how to compute features) and then classify every pixel in the image. The question is whether the pixels are the best representation to use for this purpose. It is like trying to find out whether we are dealing with an apple or a pear by looking at properties of the constituting atoms. A more global representation might be more advantageous. First of all, such a representation may naturally suggest sensible features. Secondly, the amount of elements that must be classified will be smaller.

In this thesis methods are presented to extract the locations of the centerlines, or *ridges*, of elongated structures. With the knowledge of these locations, entities on a higher level can be constructed: line elements. It is this representation that will be used for classification, e.g. between vessel and non-vessel structure. The next stage is the grouping of the requested line elements into curves. At this level it is possible to recognize anatomical structures, e.g. specific arteries.

Note that the approach of extracting and grouping centerlines does not yield a full segmentation of the objects we are looking for. If a full segmentation is required, the centerlines can serve as a starting point. In this thesis a few examples are given of methods that can be used for this purpose. One could say that the centerlines are the "skeleton" on which the elongated objects are mounted.

1.3 Motivation for the proposed methodology

1.3.1 The visual system

The approach that is advocated in this thesis is partly inspired by the functioning of the human visual system.

The part of the brain that is dedicated to our vision is located at the back of the head and known as the visual cortex. Light that enters our eyes causes nerve-cells in the retina to fire. These signals are transfered to the lateral geniculate nucleus, a sort of relays station. From there on, the signals are projected to the visual cortex. A schematic diagram is shown in figure 1.1.

The visual cortex consists of many interconnected areas, of which the primary visual cortex, or V1, is the largest. The lateral geniculate nucleus projects primarily to this area. In the early 1960's researchers in neurophysiology started to measure the output of single cells in the areas of the visual cortex of cats and macaque monkeys [46, 47]. This was done with a small electrode that was stuck in the visual cortex. On a screen a visual input was generated at different locations of the visual field, see figure 1.1. It turned out that the cells in V1 that were measured, responded to a specific visual pattern at a specific location in the visual field. The patterns to which the cells responded were blobs, bars and boundaries.

Later research revealed that many connections go from V1 to other areas of the visual cortex and vice versa. It seems that the visual system is able to group the basic building blocks to higher level structure.

This short overview of the human visual system supports the strategy that is

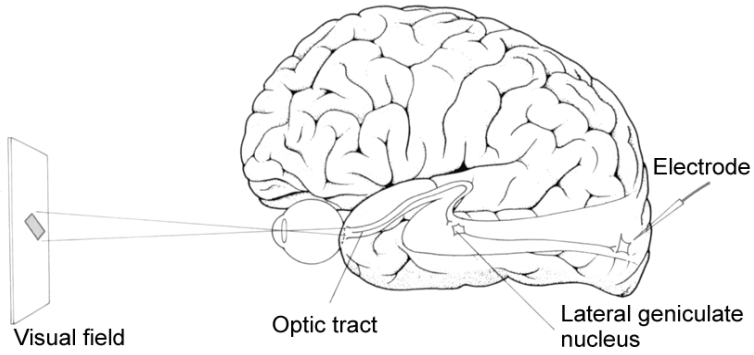


Figure 1.1 A schematic overview of the visual pathway. Picture is taken from S. Zeki, *A vision of the brain* [133] (permission for reproduction granted by Blackwell Scientific Publications).

followed in this thesis. The retina can be seen as a representation of the outside world similar to the pixel grid. The primitive representation is constructed in V1, while grouping, recognition and segmentation seem to be happening in other visual cortical areas.

Note, that we have given a far too simplistic view of how the human visual system operates. However, a more extensive description is beyond the scope of this thesis. In the books of Hubel [45], Zeki [133] and Rodieck [107] comprehensive descriptions of the visual system can be found.

1.3.2 Marr's primal sketch

The importance of using appropriate representations at different stages of an image processing system was recognized by D. Marr in his important book *Vision* [74]. On the first page of his book, Marr defines vision as follows:

> Vision is the *process* of discovering from images what is present in the world, and where it is.
>
> Vision is therefore, first and foremost, an information-processing task, but we cannot think of it just as a process. For if we are capable of knowing what is where in the world, our brains must somehow be capable of *representing* this information.

In his book, Marr sets out a strategy to the understanding of images by computers. He observes that there are different structures present in images, such as blobs, edges and bars. With this representation he derives a description in which images are decomposed in those primitives. He calls this the *primal sketch*. The primal sketch serves as a starting point for building more complex representations, which Marr refers to as grouping. His ultimate goal is to detect and

recognize shapes in two-dimensional images and to build a three-dimensional representation out of these scenes.

The approach we follow is related to the work of Marr. We too stress the importance of using appropriate representations and construction of primitives. But, we are not aiming for a three-dimensional description of the world in terms of two-dimensional representations. If we have a task that involves two-dimensional data, then we treat the data as being two-dimensional. If we have three-dimensional data at our disposal, we can obtain the three-dimensional representation directly from the data.

1.4 Prerequisites

The rest of this chapter is devoted to the introduction of some technical concepts that are used throughout this thesis. We start with notational issues and continue with a discussion on computing derivatives of image intensities on a discrete grid. Since supervised classification plays an important role in our methodology, next, an overview is given on three classifiers that are used in this work. We end with a discussion on how to evaluate the performance of the systems that we develop in this thesis.

1.4.1 Notation

$\mathbb{N}$ is the set consisting of zero and all positive integers. $\mathbb{N}^+$ is the set of all positive integers (zero excluded). $\mathbb{Z}$ is the set of all integers (positive, zero and negative). $\mathbb{R}$ is the set consisting of all real numbers.

A vector will be denoted by a symbol in bold lowercase italic font, like $\boldsymbol{v}$. The elements of $\boldsymbol{v}$ are in italic normal font and have a subscript to index their position in the vector: v_i is the i-th element of $\boldsymbol{v}$.

Matrices are written as symbols in bold uppercase italic font, like $\boldsymbol{M}$. The elements of a matrix are in italic normal font and doubly indexed: M_{ij} is an element of $\boldsymbol{M}$, where i refers to the row and j to the column.

Einstein summation convention

In this chapter and in chapter 2 we will use the convention introduced by Einstein [26] that repeated indices are implicitly summed over[2]. This can simplify and shorten equations involving tensors. For example, using Einstein summation, we get

$$\sum_i a_i b_i = a_i b_i$$

[2]The story goes that Einstein later joked to a friend [87, p. 216]: "I have made a great discovery in mathematics; I have suppressed the summation sign every time that the summation must be made over an index which occurs twice…"

and

$$\sum_j a_{ij} b_{jk} = a_{ij} b_{jk}$$

Two important tensors need to be introduced: the Kronecker delta tensor δ_{ij} and the n-dimensional Levi-Civita or permutation tensor $\varepsilon_{i_1 \cdots i_n}$ [113].

The Kronecker delta is defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} ,$$

and can be thought of as the identity matrix.

The Levi-Civita tensor is defined as

$$\varepsilon_{i_1 \cdots i_n} = \begin{cases} +1 & \text{if } i_1 \cdots i_n \text{ is an even permutation of } 1 \cdots n \\ -1 & \text{if } i_1 \cdots i_n \text{ is an odd permutation of } 1 \cdots n \\ 0 & \text{if two or more indices are equal} \end{cases} .$$

As an example $\varepsilon_{123} = 1$, $\varepsilon_{213} = -1$ and $\varepsilon_{ijj} = 0$.

1.4.2 Regularization of differential operators

For the development of our methods we need to take derivatives of the image intensity with respect to the image coordinates. Taking derivatives of functions on continuous domains is defined as a limiting process. For example, the first partial derivative of a function $f(\boldsymbol{x})$ on $\mathbb{R}^d$ with respect to x_i is computed as

$$\frac{\partial f}{\partial x_i}(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon} , \tag{1.1}$$

with $\partial / \partial x_i$ the derivative operator and $\boldsymbol{e}_i$ a vector that is one at index i and zero elsewhere.

There are two fundamental problems with the definition in eq. (1.1). First of all, this operation is only defined if the limit exists. If it does, the function is said to be differentiable. Secondly, with our images discretized on a pixel grid, we cannot take these limits.

In the 1950's a theory was developed by Schwartz [109, 110] that describes a measurement (as our image is) as a "filtered" observation of the "reality". According to Schwartz, the "reality" is a *distribution* $r(\boldsymbol{x})$, that is measured by a test function $\phi(\boldsymbol{x})$ which produces an observation f as follows

$$f = \int r(\boldsymbol{x}) \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} ,$$

where the integral runs over the whole space. Observations at different points in space are obtained by translating the measurement device

$$f(\boldsymbol{y}) = \int r(\boldsymbol{x}) \phi(\boldsymbol{x} - \boldsymbol{y}) \mathrm{d}\boldsymbol{x} .$$

Schwartz built a mathematical theory of making measurements and defined a special class of test functions, the so-called Schwartz class $\mathcal{S}$. A function ϕ belongs to $\mathcal{S}$ if it is an infinite number times differentiable and if it goes to zero at its tails as

$$|\phi(\boldsymbol{x})| \leq M_n \|\boldsymbol{x}\|^{-n}, \qquad \|\boldsymbol{x}\| \to \infty, \quad \forall n \in \mathbb{N}^+ \ ,$$

where M_n is a constant that only depends on n [117, 134]. The distributions that can be measured with these test functions must not grow faster than any polynomial if $\|\boldsymbol{x}\| \to \infty$. Distributions that adhere to this demand are called tempered distributions.

One of the reasons that Schwartz wanted the test functions to be infinitely differentiable, is that it enabled him to transfer differentiation from the distribution to the test function

$$\int \frac{\partial r}{\partial x_i}(\boldsymbol{x})\phi(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{x} = \int r(\boldsymbol{x})\frac{\partial \phi}{\partial y_i}(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{x} \ .$$

In the above equation we used integration by parts. The stock terms are zero, because the test function goes faster to zero at infinity than the tempered distribution is allowed to grow. As a result, the observations can be differentiated by mimicking the measurement process with a differentiated test function

$$\frac{\partial f}{\partial y_i}(\boldsymbol{y}) = \int r(\boldsymbol{x})\frac{\partial \phi}{\partial y_i}(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{x} \ .$$

In general, the distribution r and the test function ϕ are unknown. The "reality" that is available are the observations f. To obtain differentiated versions of the observations, the observations themselves are taken as tempered distributions and the test function is chosen. With these substitutions we get

$$\frac{\partial f}{\partial y_i}(\boldsymbol{y}) = \int f(\boldsymbol{x})\frac{\partial \phi}{\partial y_i}(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{x} \ . \tag{1.2}$$

Equation (1.2) is called a weak or regularized derivative of f. It enables us to compute well behaving derivatives by integration. Note that higher order derivatives are computed by taking higher order derivatives of the test function.

A well-known member of $\mathcal{S}$ is the d-dimensional Gaussian kernel

$$G(\boldsymbol{x};\sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{x_i x_i}{2\sigma^2}\right) \ ,$$

where σ is a parameter that determines the width of the kernel. This parameter can be regarded as the scale at which the measurement is made. It is this kernel that will be used throughout this thesis to compute derivatives of images. Note that the scale is a free parameter. Taking no derivative at all (zeroth order derivative) one gets a blurred version of the image

$$f(\boldsymbol{y},\sigma) = \int f(\boldsymbol{x})G(\boldsymbol{x}-\boldsymbol{y},\sigma)\mathrm{d}\boldsymbol{x} \ . \tag{1.3}$$

where the amount of blur depends on the value of the scale parameter. Since $G(\boldsymbol{x}, \sigma)$ converges to the Dirac-delta distribution $\delta(\boldsymbol{x})$ for $\sigma \downarrow 0$, $f(\boldsymbol{x}, \sigma)$ goes to $f(\boldsymbol{x})$ if $\sigma \downarrow 0$, provided that f is continuous. In this respect, we can interpret the weak derivatives at a scale σ as blurred derivatives. If f is differentiable, i.e. the limit in eq. (1.1) exists, then the weak derivative converges to the "real" derivative if $\sigma \downarrow 0$.

In the 1980's, Koenderink [64] investigated the differential structure of images. He was looking for a process to get "simpler" versions of an image and it turned out that the following diffusion process produced what he was aiming for

$$\frac{\partial^2 f(\boldsymbol{x}, \sigma)}{\partial x_i \partial x_i} - \frac{1}{\sigma} \frac{\partial f(\boldsymbol{x}, \sigma)}{\partial \sigma} = 0 \quad \text{with } f(\boldsymbol{x}, 0) = f(\boldsymbol{x}) \ .$$

The solution to this equation is

$$f(\boldsymbol{x}, \sigma) = \int f(\boldsymbol{y}) G(\boldsymbol{x} - \boldsymbol{y}, \sigma) \mathrm{d}\boldsymbol{y} \ . \tag{1.4}$$

Note the close connection between eq. (1.4) and eq. (1.3). In fact, these equations are equivalent, because $G(-\boldsymbol{x}) = G(\boldsymbol{x})$.

The work of Koenderink started a new area of research in image processing: Gaussian scale space analysis [72, 122, 123], which is primarily interested in the topological deformation of images as a function of the scale. Florack [28, 29] recognized the coupling between Gaussian scale space analysis and regularization theory. One of the main properties of Gaussian scale space is that with increasing scale image structure is suppressed. A proper choice of the scale of observation can be used to tune the detection of objects of a specific scale. This is called scale selection [72].

1.4.3 Supervised classification

In this section we introduce some terminology and theory regarding classification that is needed for this thesis. An excellent review on statistical pattern recognition can be found in [48].

Classifiers

A classifier C is a mapping that takes as input a feature vector $\boldsymbol{\phi} \in \mathbb{R}^n$. Depending on the type of classifier, the output can be either an integer, that denotes the class number of the object for which $\boldsymbol{\phi}$ is representative, or it can be an m-dimensional vector of real numbers that gives the chances that $\boldsymbol{\phi}$ belongs to any of the m possible classes ω_i. The first type of classification is called hard classification, as opposed to the second type which is called soft classification.

Many classifiers exist, like neural networks, linear discriminant classifiers, quadratic discriminant classifiers, support vector machines, Parzen classifiers, k-nearest neighbor classifiers to mention the most well-known [9, 21, 31, 41, 48,

105]. All these classifiers bear in common that they need example data to "learn" the mapping and are therefore called supervised classifiers. The example data consist of a set of feature vectors (the training set) for which the output class is known.

In the experiments in this thesis, three classifiers have been applied: the linear discriminant classifier, the quadratic discriminant classifier and the *k*-nearest neighbor or *k*NN-classifier.

Bayes' formula and the decision rule

In order to classify a feature vector to a specific class, Bayesian decision theory can be used. Before the construction of a classifier, the only knowledge that is available is the fraction of examples that belong to class ω_i. This fraction is called the prior probability and is denoted by $P(\omega_i)$. The idea of using classifiers is to compute a posterior probability $P(\omega_i|\boldsymbol{\phi})$, that describes the chance of finding a sample belonging to class ω_i given its measured features $\boldsymbol{\phi}$. The prior and posterior probabilities are related by Bayes' formula [21]

$$P(\omega_i|\boldsymbol{\phi}) = \frac{p(\boldsymbol{\phi}|\omega_i)P(\omega_i)}{p(\boldsymbol{\phi})} \ , \tag{1.5}$$

where $p(\boldsymbol{\phi}|\omega_i)$ is called the likelihood and $p(\boldsymbol{\phi}) = \sum_{j=1}^{m} p(\boldsymbol{\phi}|\omega_j)P(\omega_j)$, so that the sum of the posterior probabilities over all classes equals 1.

With the posterior probabilities the class $\omega^\star$ to which a sample is assigned can be determined. Define $\lambda(\omega_i|\omega_j)$ as the loss of assigning a sample to class ω_i while the true class is ω_j. Now, we can define the expected loss $R(\omega_i|\boldsymbol{\phi})$ as

$$R(\omega_i|\boldsymbol{\phi}) = \sum_{j=1}^{m} \lambda(\omega_i|\omega_j)P(\omega_j|\boldsymbol{\phi}) \ .$$

In Bayesian decision theory, a sample is assigned to the class that minimizes the expected loss, i.e.

$$\omega^\star = \arg\min_i R(\omega_i|\boldsymbol{\phi}) \ . \tag{1.6}$$

If errors are to be avoided, the following loss function is the natural choice

$$\lambda(\omega_i|\omega_j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases} \ . \tag{1.7}$$

Substitution in eq. (1.6) yields

$$\omega^\star = \arg\min_i \sum_{j\neq i} P(\omega_j|\boldsymbol{\phi}) = \arg\min_i \left[1 - P(\omega_i|\boldsymbol{\phi})\right] \ ,$$

or, equivalently

$$\omega^\star = \arg\max_i P(\omega_i|\boldsymbol{\phi}) \ , \tag{1.8}$$

showing that for this loss function the expected loss is minimal if $\boldsymbol{\phi}$ is assigned to the class with the highest posterior probability.

Linear and quadratic discriminant classifiers

The linear and quadratic classifiers assume that the classes are normally distributed, where the linear classifier adds as extra assumption that the covariance matrices of the features of all classes are identical. These assumptions imply the following likelihood for class ω_i [21]

$$p(\boldsymbol{\phi}|\omega_i) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma_i}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{\phi}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\phi}-\boldsymbol{\mu}_i)\right) \ ,$$

where $\boldsymbol{\mu}_i$ is the mean and $\boldsymbol{\Sigma}_i$ the covariance matrix of the features of class ω_i. For the linear discriminant the covariance matrix is independent of the classes: $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$. Maximizing the posterior probability is equivalent to maximizing the logarithm of the numerator of eq. (1.5), giving

$$\omega^\star = \arg\max_i \left[-\frac{1}{2}(\boldsymbol{\phi}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\phi}-\boldsymbol{\mu}_i) - \frac{1}{2}\log_e|\boldsymbol{\Sigma}_i| + \log_e P(\omega_i)\right] \ ,$$

where terms independent of i have been dropped. The form between the square brackets is called the discriminant function. Because this form is quadratic in $\boldsymbol{\phi}$, the classifier is called a quadratic discriminant classifier.

If the covariance matrix between the classes is independent of i, the above equation can be rewritten, after dropping terms that do not depend on i, as

$$\omega^\star = \arg\max_i \left[\boldsymbol{\mu}_i^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi} - \frac{1}{2}\boldsymbol{\mu}_i^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \log_e P(\omega_i)\right] \ ,$$

which is linear in $\boldsymbol{\phi}$, explaining the name linear discriminant classifier.

For the training of these discriminant classifiers the mean vectors and covariance matrices of the features are estimated from the training data for every class.

kNN-classifier

For the linear and quadratic classifiers assumptions are made on the likelihood $p(\boldsymbol{\phi}|\omega_i)$, so that an analytical expression for eq. (1.5) can be derived. In kNN-classification, no assumptions with respect to the underlying probability density function are made and the posterior probability is directly estimated from the training data, without explicitly using eq. (1.5).

Training a kNN-classifier is extremely simple, just store all examples. For classification of a sample, its k closest neighbors must be found. Euclidean distance is often used for defining what closest means, but other distance measures can be used as well. The posterior chance that the unclassified sample $\boldsymbol{\phi}$ belongs to class ω_i is given by [21]

$$P(\omega_i|\boldsymbol{\phi}) = \frac{n_i}{k} \ ,$$

where n_i is the number of neighbors with class number ω_i. Hard classification can be obtained by choosing the class that maximizes the posterior probability (cf. eq. (1.8)).

In a straightforward implementation of a kNN-classifier the number of distances that have to be computed for every sample is linearly proportional to the total number of samples in the training set. In the work of Arya et al. [5] an algorithm is derived that makes use of a hierarchical decomposition of the feature space, which is called a balanced box-decomposition (BBD) tree. The construction of this tree decreases the look-up time for finding the neighbors of a query point. The algorithm proposed by Arya et al. can reduce the look-up time even further by using an approximate nearest neighbor algorithm. In approximate nearest neighbor searching the distance from the input sample to the k-th nearest neighbor may exceed the true distance by a factor $(1 + \epsilon)$, with $\epsilon > 0$. Setting a value of $\epsilon > 0$ can speed up computations considerably without a significant loss of accuracy. An implementation of the algorithm by Arya et al. [5] is freely available on the internet.

The classifiers that are discussed in this section have been tested on all the data sets that are used in this thesis and it turns out that the kNN-classifier gives the best performance. For that reason, the kNN-classifier is mostly used this in thesis.

Transforming the input data

Because a kNN-classifier uses distances between features, the output of the classifier will be different if, e.g., the values of a specific feature are multiplied with a scalar. Features with a large range dominate over features with a small range. Removing the dominance can be accomplished by subtracting from each feature set its mean value and subsequently dividing by its standard deviation. The mean and standard deviation can be estimated from the training set. On a new query point this transformation must also be applied. In figure 1.2 an example is given.

Feature selection

An important question in the construction of a classifier is: How many features are needed for good performance? Intuitively one could think that more information leads to better separability of the classes, so: the more, the better. In practice this is not true and the performance of a classifier can degrade when adding more and more features. This phenomenon is known as the "curse of dimensionality". Adding more and more features, or equivalently increasing the dimensionality of the feature space, yields a feature space which becomes almost empty. Such a sparse covering by the data is often not representative for the mapping to be learned. If the density of training points in the feature space has to remain constant, the number of samples has to increase exponentially with the dimensionality [9, 48].

Another reason why the performance of a classifier may degrade, is that some features add no meaningful information. Take for example the kNN-classifier, which relies on comparing the distances in feature space. Suppose we have the situation in figure 1.3(a) where a point is classified in a one-dimensional feature

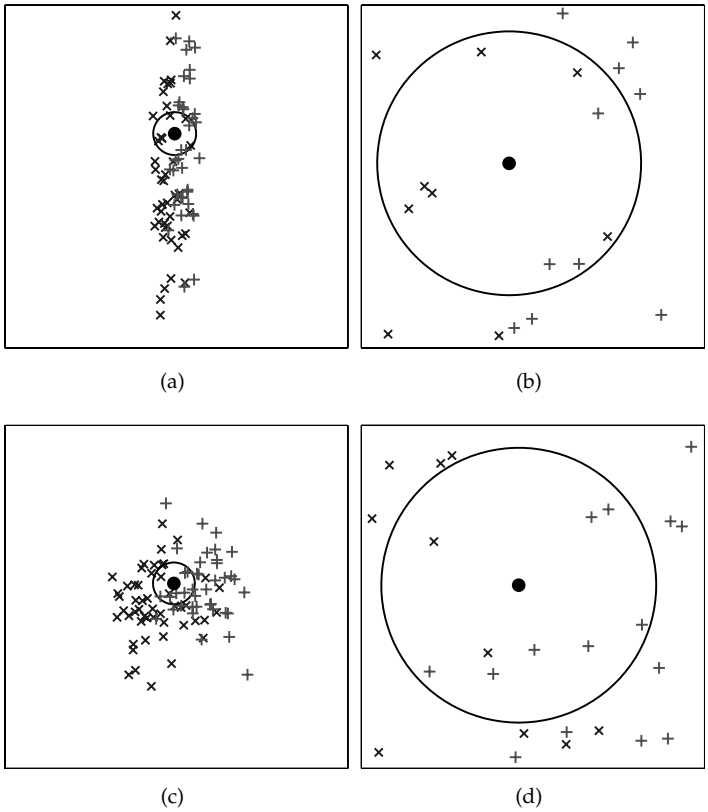
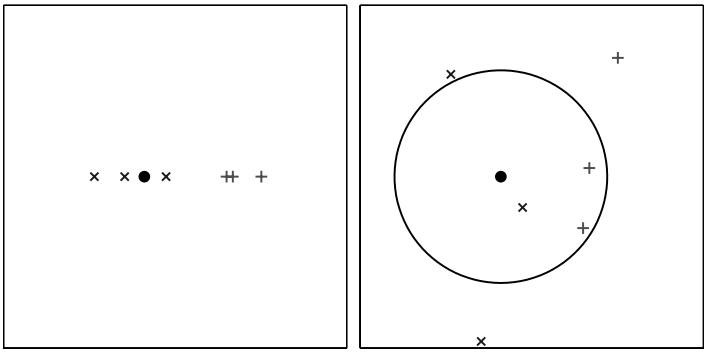


Figure 1.2 (a) A scatterplot of two classes in a two-dimensional feature space (the classes are marked with '+' and '×'). The black dot denotes a new query point, the circle encloses the 9 nearest neighbors. (b) Zooming in on (a) shows that 6 neighbors are of class '×' and 3 of class '+'. (c) Rescaling the features to unit variance makes the distribution of the features more isotropic. (d) Zooming in on (c) shows that now 2 neighbors belong to class '×' and 7 to class '+'.

space. Now, we add a second feature, in this case a random number. This situation is depicted in figure 1.3(b). If the three closest neighbors to the query point are taken, the classification result changes from class '×' in figure 1.3(a) to class '+' in figure 1.3(b).

To discover the right features in a data set, feature selection can be employed. Feature selection is a method that tries to find the features that give good performance. There exist no general methods to find the optimal subset of features, except exhaustive search which for a moderate number of features already becomes infeasible.

Several techniques exist, see [48] for an overview. Here sequential forward selection (SFS) is discussed [127]. This algorithm starts with a null feature set and, for each step, the best feature that satisfies some criterion function is included



(a) (b)

Figure 1.3 (a) In a one dimensional feature space, the black dot will be classified as '×'. (b) Adding a "bad" feature can change the distances between the samples in such a way that the black dot is incorrectly classified as '+'.

with the current feature set. In our experiments the area under the ROC-curve is taken as the criterion function (ROC-curves are discussed in section 1.4.4). After all features have been included, the subset that gives the best performance is chosen.

A feature selection method that generally gives better results is sequential forward floating search (SFFS) [102]. The difference with SFS is that after a feature has been added, earlier added features are unselected as long as that improves the performance. SFFS finds "nested" sets of features that remain hidden in SFS, at the cost of an increase in the number of feature set evaluations. In our experiments, SFFS is for computational reasons often not feasible.

1.4.4 Evaluation of performance

Independent test set and cross-validation

For the evaluation of the performance of a classifier an independent test set is indispensable [21], i.e. a set of samples for which the outputs are known and for which the intersection with the training set is empty. However, sometimes there is not enough data available and in that case an alternative is cross validation [21]. In cross validation the training set is divided in m non-overlapping sets (folds) of approximately equal size. One performs m experiments, where in each experiment i the i-th fold is removed, the classifier is trained on the remaining data, and the trained classifier is tested on the held-out fold. After performing the m experiments, every example has been used in a test set exactly once. As a consequence, all the test sets are independent (assuming that for all i the samples in fold i are independent from those in the other folds). The training sets, however, are not independent, since they overlap each other substantially.

We will mainly be concerned with two-class problems, or foreground vs. back-

ground classification. In a two-class setting, $P(\omega_1|\boldsymbol{\phi}) + P(\omega_2|\boldsymbol{\phi}) = 1$ and we will from hereon take ω_1 as the foreground class.

We will consider two methods to evaluate classification results. The first deals with hard classification and confusion matrices. The second one uses ROC-analysis to measure the performance of soft classification.

Confusion matrix based analysis

If a test set is fed into a classifier, the output of the classifier can be compared to the known output, which acts as a reference set. In the ideal situation the classifier makes no errors and all foreground samples are classified as foreground and all background samples as background. In reality a classifier will make errors. The results of a classifier are conveniently summarized with a confusion matrix (or contingency table) [65, 104]. A confusion matrix is a table showing how many samples are classified in a specific class vs. the class in which they should be classified according to the reference set, see table 1.1.

		ref. set	
		foreground (ω_1)	background (ω_2)
obs.	foreground (ω_1)	TP	FP
	background (ω_2)	FN	TN

Table 1.1 Confusion matrix for a two class problem.

Samples that are correctly classified as foreground are called true positives (TP). Similarly, correctly classified background elements are called true negatives (TN). Samples misclassified as foreground are called false positives (FP) and samples misclassified as background are false negatives (FN). Note that the total number of samples is the sum of all the elements in the confusion matrix. From these numbers some useful fractions can be computed. We give the three that are mostly used. Accuracy is given by [65]

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \ ,$$

and measures the rate of correct predictions made by the classifier over the complete test set. A classifier which classifies all samples correctly has an accuracy of 1. A second measure is the sensitivity [65] or true positive fraction (TPF), which computes the fraction of correctly classified foreground samples with respect to the number of samples that should be classified as foreground

$$\text{sensitivity} = \text{TPF} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ .$$

A classifier that makes no errors in classifying foreground samples has a sensitivity of 1. A last measure is the specificity [65] or true negative fraction. It gives the

fraction of correctly classified background samples with respect to the number of samples that should be classified as background

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \ .$$

The specifity of a classifier is 1 if the classifier labels all background elements correctly.

ROC analysis

The measures above make only sense in case of hard classification. In eq. (1.8) Bayes' decision rule was introduced for transforming a soft classification into a hard classification. In a two-class system there is an alternative. A threshold t_{P} can be set on the posterior probability $P(\omega_1|\boldsymbol{\phi})$. If $P(\omega_1|\boldsymbol{\phi})$ exceeds the threshold the sample is counted as foreground, otherwise as background. As a result the confusion matrix becomes dependent on t_{P}. A convenient way to visualize the behavior of the sensitivity and the specificity is to make an ROC-curve (receiver operating characteristic[3] curve) [81]. An ROC-curve is a graphical plot of the true positive fraction vs. the false positive fraction as the discrimination threshold t_{P} is varied. The false positive fraction (FPF) is given by

$$\text{FPF} = 1 - \text{specificity} = \frac{\text{FP}}{\text{TN} + \text{FP}} \ .$$

In figure 1.4 an example of an ROC-curve is given. A perfect classifier will yield a graph that is a point in the top left corner, i.e. 100% sensitivity (no false negatives are found) and 100% specificity (no false positives are found). A classifier that assigns random probabilities to the samples will give a straight line at an angle of 45 degrees from the horizontal, from bottom left to top right: this is because, as the threshold is raised, equal fractions of true and false positives would be let in. Results below this no-discrimination line suggest a detector that gives wrong results consistently, and can therefore be simply used to make a detector that gives useful results by inverting its decisions.

The performance of a classifier is often summarized by A_z, the area under the ROC-curve, which has the following statistical meaning. If a true positive and a true negative are selected at random from the test set, A_z is the chance that they are ordered correctly by the classifier, i.e. the true positive is given a higher probability than the true negative [39]. A_z equals 1 for a perfect classifier and 0.5 for a classifier that ranks samples randomly. A higher value of A_z denotes better performance.

[3] The name "receiver operating characteristic" comes from signal detection theory, developed during World War II for the analysis of radar images. Radar operators had to decide whether a blip on the screen represented an enemy target, a friendly ship, or just noise. Signal detection theory measures the ability of radar receiver operators to make these important distinctions. Their ability to do so was called the receiver operating characteristic.

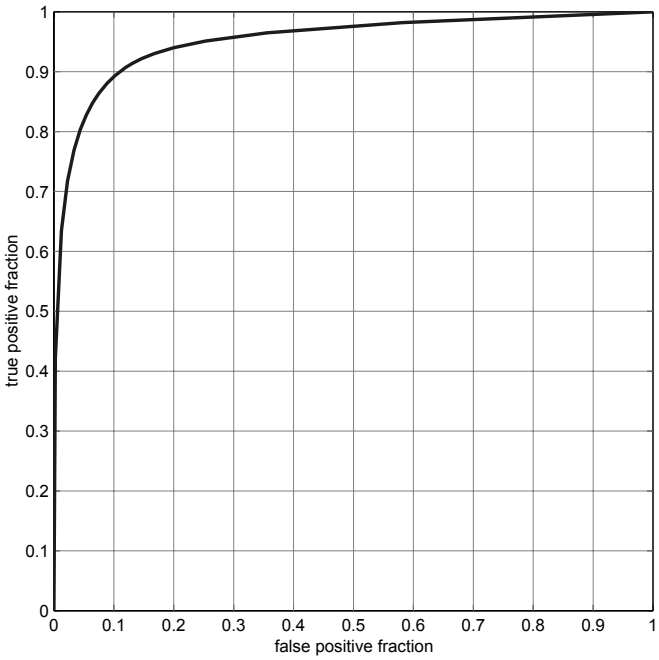


Figure 1.4 Example of an ROC-curve.

Comparing systems

When different classifiers are used one wants to know which system gives the best result. For this purpose, two mutual excluding hypotheses are stated. The null hypothesis, H_0, assumes that the systems are not different. If this hypothesis can be rejected, the alternative hypothesis, H_1 is accepted, which states that the systems are different. The idea of hypothesis testing is to construct a test characteristic which gives the chance P that the data observed obey the null hypothesis. If P is low, H_0 is rejected and the difference between the classifiers are called statistically significant. A value of 0.05 is normally taken as the upper boundary for rejecting the null hypothesis.

If two systems are tested on the same datasets, a paired t-test is the appropriate method to compute the P-value [104]. Suppose we have N datasets, for which the A_z values of both classifiers are available. For every dataset the difference d_i between the two A_z values is computed. These differences are distributed according to a Student t-distribution [3, 118][4] with $N-1$ degrees of freedom. To

[4]Student was the pseudonym of William Gosset, who worked for the Guinness Breweries. His employer did not allow him to publish under his own name. Gosset was hired by the brewery because Guinness wanted to have a method to quantify the quality of beer.

compute the t-statistic, the mean

$$\mu_d = \frac{1}{N}\sum_{i=1}^{N} d_i \ ,$$

and standard deviation

$$s_d = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d_i - \mu_d)^2} \ ,$$

of the differences need to be known.

With the mean and standard deviation the t-statistic is computed as

$$t = \sqrt{N}\,\frac{\mu_d}{s_d} \ .$$

The Student t-distribution with $N-1$ degrees of freedom gives the chance P to observe t given the null hypothesis. If t is close to zero, this chance will be higher than if t is very different from zero. In the first case the A_z values do not differ much, while in the second case they do.

The measure of comparison, d_i, does not necessarily have to be the difference between the A_z values. The difference between accuracies or between any other appropriate measure of performance can be used. For more information on hypothesis testing, we refer to [3, 20].

1.5 Outline of the thesis

This thesis consists of 7 chapters. In chapter 2 a general method is developed for the detection of image structure in the form of (sub-dimensional) critical point sets, i.e. local maxima, minima and saddle points. It is shown that critical points are the singularities of a vector field. As a result, the detection of ridges and edges can be formulated as finding the singularities of a vector field with respect to local sub-dimensional coordinate frames. Because ridges are located on the centerlines of elongated structures special attention is being paid to their detection.

Chapter 3 is a short chapter on the construction of image primitives from sub-dimensional critical point sets. When the sub-dimensional point sets are one-dimensional (curves) approximations to line elements are found. From two-dimensional point sets (surfaces) patches can be constructed.

In chapter 4 the techniques developed in chapters 2 and 3 are applied for the segmentation of the vasculature in two-dimensional color images of the human retina. The segmentation is based on a two-step classification. Firstly, for each primitive the chance is computed that it is part of the centerline of a vessel. Secondly, the pixels surrounding the primitives are classified.

Next, in chapter 5 we investigate the use of a binary spin-glass model as an alternative classifier. The idea is that image primitives can have interaction with

their neighbors and in this way improve the detection of foreground elements. We consider the training of this classifier based on local and bilocal features defined on the primitives.

The classifier presented in chapter 5 is employed in chapter 6 to detect the primitives of rib centerlines in multi-detector row CT (MDRCT) images of the thorax. The detected primitives are grouped to individual centerlines and labeled to rib number. To obtain the full segmentation of the rib cage, the labeled centerlines initialize a seeded region growing algorithm.

Finally, in chapter 7 a general discussion is given and an overview of other possible applications is presented.

Chapter 2

Detection of image structure: sub-dimensional topological point sets

2.1 Introduction

IN this chapter we introduce a topological quantity to characterize the neighborhood of every pixel in scalar images of arbitrary spatial dimensions. This quantity is an integer number that can single out critical points such as extrema and saddle points. It can also be generalized to define the membership of the point to extended sub-dimensional structures such as ridges and edges. The topological homotopy class number was introduced in [54, 55] in relation to its importance for deep-structure image analysis and application to multi-scale segmentation. In its essence this number reflects the behavior of the gradient image vector in a close neighborhood around a given test point. In the simplest one-dimensional case, the topological class of a point is defined as the difference of the sign of the signal derivative taken from both sides of the point. Clearly this number is zero everywhere except in the local extrema.

The proper generalization to higher dimensions is provided by the homotopy class $\pi_{D-1}(S^{D-1})$ that parameterizes the space of non-equivalent (non-deformable smoothly into each other) mappings between two $D-1$ dimensional spheres (D is the number of dimensions of the image). The mapping is defined by the normalized gradient vector taken on a closed surface (homotopic to a sphere) surrounding a test point. One can show that the set of non-equivalent mappings for $D > 1$ can be labeled with an integer number. This is particularly evident in the $D = 2$ example where the topological number reduces to the well-known winding number [57] indicating the number of times the gradient vector rotates around its origin when a point is circumventing a test point. Extrema points have winding number $+1$, saddle points have winding number -1 and for regular points the number vanishes. A detailed discussion on the 2D case is presented in [54] and in the next section we give a concise summary of the construction as well as of its generalization to arbitrary image dimensions.

The main advantage of using homotopy classes for the localization of singular points lies in their explicit and constructive nature. In finding the zero crossings of a real function, for example, the only sensible task would be to find the intervals where the function changes sign. The size of these intervals is the precision with which we are searching for the zero crossings. Our topological construction is in many aspects analogous to this generic example. The size of the neighborhood (the closed surface) around the test image point is the spatial precision with which we localize the singular point. Therefore our method is a natural generalization of interval calculus to higher dimensional signals.

Another advantage of the proposed quantity is its non-perturbative nature. To compute the homotopy class of a point, we do not need to know the values of any set of derivatives *in* the point. We only sum certain quantities (differential forms) *around* the given image location. The quantity is *additive* so the total topological number surrounded by a given surface is the sum of the topological number of all image points inside. This feature enables the selection of the neighborhood size as the singularity resolution parameter and makes the whole construction well posed.

In addition to segmentation of critical points, the method is extended in this chapter for localization of points lying on *relative critical sets*. To this end we introduce a *relative homotopy class* of a given test image point defined as the homotopy class calculated on a linear sub-space in the neighborhood of the test point. This sub-space can be defined as a linear envelope spawned over the vectors of a given vector frame field. A particularly important case is that of the frame formed by a subset of the eigenvectors of the Hessian (the matrices of the second image derivatives). In this case we obtain a constructive definition for *topological ridges*. We can classify such a point set by the number of the Hessian eigenvectors forming the sub-space and by the signs of the corresponding eigenvalues. We show in this work that our definition is equivalent in some particular cases to the definitions given in [19, 22, 24, 58, 82]. In the last works the point sets are defined implicitly whereas we propose a direct computational technique that segments the point sets explicitly. In addition to ridges, we can define other particular relative critical sets by using any globally biased frame field where a number of vectors is chosen independently of the image structures.

Another interesting case is the hyper-dimensional critical set defined relative to the gradient vector itself and segmented by the zero-crossings of the second derivative in the direction of the gradient. This point set can be interpreted as a *topological edge*.

Critical points and ridges play an essential role in uncommitted image analysis as revealed in [64, 71]. Similar topological structures were studied in the context of multi-scale image analysis [50, 72, 83, 86]. They form a sort of topological back-bone on which image structures are mounted. One specific feature of our construction is that it relies on quasi-local image properties. This distinguishes it from the watershed model [86].

The rest of the chapter is organized as follows.

In section 2.2 we remind the basic definitions and properties of the topological

feature introduced in [54] that enables localizing singular points in gray-scale images. Section 2.3 extends the latter definitions to the concept of critical point sets of higher sub-dimension. We introduce a classification of these critical sets that can be relevant for image analysis. In particular, we define relative critical point sets defined by the eigenvectors of the local Hessian. These point sets can be interpreted as topological ridges of various signature. Further we consider another class of critical point sets that can be interpreted as topological edges. In section 2.4 we present some examples in various image dimensions. The technique is applied either to the original image or to an expanded data space built over the image. We consider the case of a scale-space image representation and the case of an orientation space representation. Section 2.5 discusses the applicability of the method, possibly in conjunction with other techniques, to segmentation tasks in image processing. The main conclusions of this chapter are summarized in section 2.6. Finally, a discrete implementation of the method is outlined in an appendix.

2.2 Homotopy numbers

We start this section with two simple and intuitive examples—the set of critical points in one- and in two-dimensional signals (images). The purpose of these examples is to provide an inductive basis for generalizations in images of arbitrary dimension.

A critical point of a D-dimensional image $L(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_D)^T$ is a point where the gradient vector $g_i = \partial_i L(\boldsymbol{x})$ vanishes (we write often ∂_i as short-hand for ∂_{x_i}). In the one-dimensional case this is a point where the first derivative has a zero crossing. To localize such a point, it suffices to consider in the one-dimensional case the difference $\nu_P = \frac{1}{2}(\mathrm{sign}(g_x(x_B)) - \mathrm{sign}(g_x(x_A)))$ for any x_A, x_B, with $x_A < x_P < x_B$ in the close vicinity of a test point x_P. Obviously, the signal's derivative changes sign at the zero crossings which therefore can be characterized with the condition $\nu_P \neq 0$. Moreover, because in 1D signals critical points are either minima or maxima, $\nu_P = +1$ indicates that there is a minimum in the neighborhood of x_P and $\nu_P = -1$ a maximum. The size of the interval $[x_A, x_B]$ defines the *precision* with which the critical point is localized. If we consider the above quantity $\nu(x)$ for all signal points x, we obtain, roughly speaking, the distribution of the critical points for the whole signal.

The same procedure is far less trivial in the case of a two-dimensional image $L(x, y)$. To localize the spatial locations where the image gradient $\boldsymbol{g} = (g_x, g_y)^T$ vanishes we can consider the *winding number* [57]

$$\nu_P = \frac{1}{2\pi} \oint_C (g_x \mathrm{d}g_y - g_y \mathrm{d}g_x) \ , \tag{2.1}$$

where the integral is taken along a closed curve surrounding the test point $\boldsymbol{x}_P$. The expression in the integral in eq. (2.1) is the infinitesimal angle of rotation of the gradient vector as the integration goes along the curve C. It is not difficult to

see that if the contour C encompasses an extrema point (minimum or maximum) the quantity (2.4) takes value $+1$. If C encloses a saddle point, then $\nu_P = -1$. These two cases are illustrated in figure 2.1.

In degenerate saddle points, named also as "monkey saddles" the topological number is $-(n-1)$ where n is the number of ridges or valleys converging to the saddle point.

In the same way as in the one-dimensional case, by computing the quantity (2.1) for all image points we can obtain another image containing information about the positions and the type of the critical points in the original gray-scale image. The spatial precision in this case is determined by the size of the area surrounded by the contour C.

Formula (2.1) is the starting point for the higher dimensional extension of the concept of the winding number. In [54] we give self-contained definitions of our quantities as well as a detailed proof of their essential properties. Here we present the highlights of the construction and introduce some brief notations from the theory of homotopy groups [84] that provide the natural basis for the introduction of a topological number associated to any singular image point.

Suppose $\boldsymbol{x}_P$ is a point in the image (singular or regular) and V_P is a region around $\boldsymbol{x}_P$ which does not contain any singularities except possibly $\boldsymbol{x}_P$. Now we will define a quantity characterizing the image in the surrounding of the point $\boldsymbol{x}_P$. Let S_P be a closed hyper-surface, topologically equivalent to a $(D-1)$-dimensional sphere, such that it is entirely in V_P and our test point $\boldsymbol{x}_P$ is inside the region W_P bounded by S_P. In other words,

$$\boldsymbol{x}_P \in W_P \ , \quad S_P = \partial W_P \ . \tag{2.2}$$

Because by assumption there are no singularities in $S_P \in V_P$, the normalized gradient vector field

$$\chi_i = \frac{g_i}{\sqrt{(g_j g_j)}} \ ,$$

is well defined on the surface S_P. In this chapter, unless stated differently, a sum over all repeated indices is assumed (cf. section 1.4.1).

The space of all unit-length D-dimensional Euclidean vectors is isomorphic to the $(D-1)$-dimensional sphere of unit radius $S_1^{(D-1)}$. Therefore the vector field $\boldsymbol{\chi}$ defines a mapping

$$S_P \to S_1^{(D-1)} \ .$$

But recalling that S_P is a manifold homotopic to a $(D-1)$-dimensional sphere, we see that the above mapping can be classified by an element of the homotopy group $\pi_{(D-1)}(S^{(D-1)})$. This group comprises all homotopically non-equivalent mappings between two $(D-1)$-dimensional spheres. Because $\pi_{(D-1)}(S_1^{(D-1)}) = \mathbb{Z}$, which is the Abelian group of all integer numbers (where addition is the group operation), we can characterize the vector field $\boldsymbol{\chi}$ on the surface S_P taken around the chosen point $\boldsymbol{x}_P$ by the homotopy number (the element of $\pi_{(D-1)}(S^{(D-1)})$) of

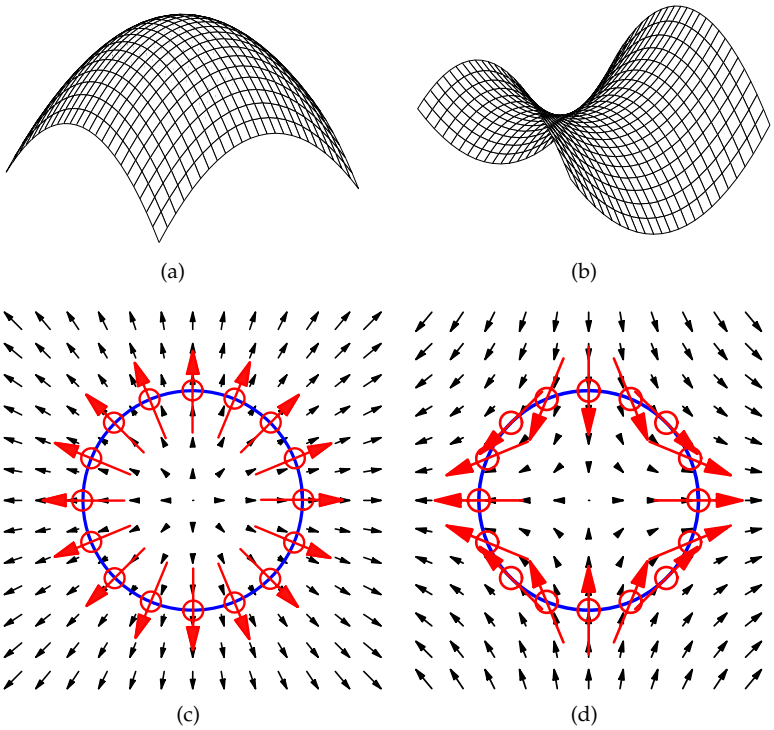


Figure 2.1 (a) Graph of a synthetic 2D image with a maximum in $(0,0)$ described analytically as $L(x,y) = -(x^2+y^2)$. (b) Graph of a synthetic image with a saddle point in $(0,0)$ described analytically as $L(x,y) = x^2 - y^2$. (c) The gradient vector field is represented as a lattice of small black arrows, the integration contour C is the blue circle and at the 16 running integration positions (red circles) the gradient vector is represented as a red arrow. It makes a 2π rotation around the contour. (d) As in (c) but for the saddle point image. The gradient vector returns in its original position after a -2π turn.

the mapping it defines. This number ν is independent of the surface S_P as long as $S_P \in V_P$, since then the surface does not surround singularities other than possibly $\boldsymbol{x}_P$. Therefore the so defined local topological number ν characterizes only the image neighborhood of the point $\boldsymbol{x}_P$ and not the hyper-surface on which it is measured.

To make the above ideas computationally explicit, we give the operational definition of the quantity ν.

Definition 2.1 *Let* $L(\boldsymbol{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$ *be a differentiable* D*-dimensional scalar image represented by its gray-scale function with at most isolated singularity points. At a* non-singular *point* $\boldsymbol{x}_A$ *we define a* $(D-1)$*-form*

$$\Phi(\boldsymbol{x}_A) = \chi_{i_1} \mathrm{d}\chi_{i_2} \wedge \cdots \wedge \mathrm{d}\chi_{i_D} \varepsilon^{i_1 i_2 \ldots i_D} \quad , \tag{2.3}$$

where $\varepsilon^{i_1 i_2 \ldots i_D}$ is the Levi-Civita tensor (see section 1.4.1).

Definition 2.2 *With the same conditions as in definition 2.1, let S be a closed ($\partial S = 0$), oriented hyper-surface. If there are no singularities on S then we define the quantity*

$$\nu_S = \frac{1}{\Gamma_D} \oint_{\boldsymbol{x}_A \in S} \Phi(\boldsymbol{x}_A) \ , \tag{2.4}$$

where the factor Γ_D is the area of a D-dimensional hypersphere of radius 1. The integral above is the natural integral of a $(D-1)$-form over a $(D-1)$-dimensional manifold without border.

An important property of the $(D-1)$-form Φ is that it is a closed form, or

Proposition 2.1

$$\mathrm{d}\Phi(\boldsymbol{x}_A) = 0 \ . \tag{2.5}$$

The above property of the form (2.3) is essential for the applications of the topological quantity (2.4). If W is a region where the image has no singularities, then the form Φ is defined for the entire region W and we can apply the generalized Stokes theorem [10, 25]

$$\oint_{\partial W} \Phi^{(D-1)} = \int_W \mathrm{d}\Phi^{(D-1)} = 0 \ ,$$

because of eq. (2.5). Therefore we obtain

Corollary 2.1 *If the gray-scale function $L(\boldsymbol{x})$ has no singularities in a given region W, then the topological quantity* (2.4) *is identically zero.*

Consider now a smooth local deformation of the surface S. If no singularities are crossed by S in the process of this deformation, then the region swept by the surface will be free of singularities and therefore the topological number on its border is zero. But the border of this region is composed exactly of the initial surface, with its orientation inverted, and the deformed surface. It is easy to see that the number (2.4) is *additive* or, in other words $\nu_{S_1 \cup S_2} = \nu_{S_1} + \nu_{S_2}$ where S_1 and S_2 are two hyper-surfaces. The integral defining ν_S obviously changes its sign when changing the orientation of the surface of integration. Therefore the quantity (2.4) on the initial and the deformed surfaces are equal as their difference is zero. This leads to the property that the topological number (2.4) is invariant under smooth deformations of the hyper-surface S, as long as no singularities are crossed by the boundary. The last property justifies the term "topological" that we assign to the quantity ν_S. It depends on the properties of the image in the region where S is placed, but in general not on the surface S itself. More precisely, the topological number depends only on the number and type of singularities surrounded by the surface S. From an implementation point of view, the topological nature of the

quantity (2.4) allows the computation to take place on any surface around the test point.

Topological numbers ν_P can be associated with every point of the image if in eq. (2.4) S is *any* closed oriented hyper-surface taken closely around a test point $\boldsymbol{x}_P$. The surface S must be close to $\boldsymbol{x}_P$ in order to ensure that no other singularities are surrounded. It is clear from proposition 2.1 and the integral Stokes' theorem however, that the topological number of a non-singular point is zero. If we plot the value of ν_P in every point of an image, we will obtain a map of the singularities of the image representing their topological "charge". We can go one step further and define a *scalar density field*, $\nu(\boldsymbol{x})$, that gives the *distribution* of the topological singularities in a given image. The point set defined as the set of locations (pixels) where $\nu \neq 0$ will give the set of all singular points in the image. Examples for this construction in different spatial dimensions were presented in the beginning of this section and can be found in more detail in [54].

2.3 Critical sets relative to vector sub-frames

2.3.1 General construction

In what follows we extend the definition of a local homotopy number by introducing suitable linear sub-neighborhoods around every image point. The idea is to select at every image point $\boldsymbol{x}_P$ a *local linear sub-space* K_P and to project the gradient vector $\boldsymbol{g}$ onto it

$$g_\alpha(\boldsymbol{x}) = v^i_\alpha(\boldsymbol{x}) g_i(\boldsymbol{x}) \ . \tag{2.6}$$

Here $\boldsymbol{v}_\alpha$, $\alpha = 1, \ldots, D_K$ are the local frame vectors defining the linear sub-space K with dimension $D_K < D$.

The next step is to compute the *relative homotopy number* in point $\boldsymbol{x}_P \in \mathbb{R}^D$ in analogy with eq. (2.4).

Definition 2.3 *The topological charge in point $\boldsymbol{x}_P$ relative to the local sub-frame $\boldsymbol{v}_\alpha$ is*

$$\nu_P(K_P) = \oint_{\boldsymbol{x}_A \in S_P, S_P \in K_P} \Phi(\boldsymbol{x}_A) \ , \tag{2.7}$$

where now the closed surface S_P around $\boldsymbol{x}_P$ is of dimension $D_K - 1$ and lies in the sub-space K_P, which is the linear envelope of $\boldsymbol{v}_\alpha(\boldsymbol{x}_P)$. The $(D_K - 1)$-form Φ is computed as in eq. (2.3) *but from the vector field* (2.6)*.*

We assume that the set of vectors $\boldsymbol{v}_\alpha(\boldsymbol{x}_P)$, $\alpha = 1, \ldots, D_K$ is of maximal rank D_K and therefore the local linear space K_P is of dimension D_K.

Now we define our central construct, the topological critical set, relative to the sub-frame $\boldsymbol{v}_\alpha$.

Definition 2.4 *Let $\boldsymbol{v}_\alpha(\boldsymbol{x})$ be a local non-degenerate sub-frame of dimension (rank) D_K and let the local linear envelope of this frame be $K_{\boldsymbol{x}}$. The point set*

$$\mathrm{PS}(\boldsymbol{v}_\alpha) = \{\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^D, \nu_{\boldsymbol{x}}(K_{\boldsymbol{x}}) \neq 0\}$$

is the relative critical point set (RCPS) *associated to the local frame* $\boldsymbol{v}_\alpha$.

From the property of the topological number given in corollary 2.1, definition 2.4 implies $g_\alpha(\boldsymbol{x}_P) = 0$ when $\boldsymbol{x}_P \in \text{PS}(\boldsymbol{v}_\alpha)$. Therefore our RCPS are equivalent to those defined in [19, 22, 24] for the same frame fields.

It is clear that if $D_K = D$, i.e. if the local sub-frame is complete, the relative homotopy class (2.7) is just the full homotopy class. The manifold defined in definition 2.4 will then be of dimension 0. In fact this set is the set of all critical points in the image.

In addition to the feature that the topological number (2.7) is non-zero, the RCPS can be characterized by the value of this number and eventually by some characteristic properties of the sub-frame $\boldsymbol{v}_\alpha$.

Now we address the question of the *local topological structure* of the RCPS associated with an arbitrary sub-frame $\boldsymbol{v}_\alpha$. We show in what follows that

Proposition 2.2 *If point* $\boldsymbol{x}_P$ *belongs to the relative critical point set from definition 2.4 and if the Hessian matrix* $\boldsymbol{H}$ *with elements* $H_{ij} = \partial_i \partial_j L$ *is non-singular in* $\boldsymbol{x}_P$*, then the point set is locally isomorphic to a linear space of dimension* $D - D_K$.

The following proof uses some techniques from differential geometry that provide the adequate covariant formalism.

Proof *Let the sub-frame* $\boldsymbol{v}_\beta$, $\beta = D - D_K + 1, \ldots, D$ *be chosen in such a way that the system of vectors* $\{\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta\}$ *represents a complete frame in the D-dimensional space of the image and* $v_\alpha^i v_\beta^i = 0$ *for all* α, β *from the corresponding ranges. If the point* $\boldsymbol{x}_P$ *belongs to the* RCPS *defined by the sub-frame* $\boldsymbol{v}_\alpha$ *then obviously* $g_\alpha = v_\alpha^i g_i(\boldsymbol{x}_P) = 0$. *If* $\boldsymbol{x}_P$ *has coordinates* x^i *then the infinitesimally close point with coordinates* $x^i + \delta x^i$ *will belong to the same* RCPS *if* $\delta x^A \nabla_A g_\alpha = 0$ *where the index A takes all values of* $1, \ldots, D$, $\delta x^A = (v^{-1})_i^A \delta x^i$ *and* ∇_A *is the* covariant derivative *induced by the frame* $\boldsymbol{v}_A$ *(such that* $\nabla_A \boldsymbol{v}_B = 0$*). Applying these notations we obtain the following equation for the variation* δx^i

$$\delta x^i v_\alpha^j H_{ij}(\boldsymbol{x}_P) = 0 \ . \tag{2.8}$$

Using the completeness of the system $\boldsymbol{v}_A$ *and the orthogonality between* $\boldsymbol{v}_\beta$ *and* $\boldsymbol{v}_\alpha$ *we get*

$$\delta x^i H_{ij}(\boldsymbol{x}_P) = C^\beta v_\beta^j \ ,$$

where the C^β *are* $D - D_K$ *arbitrary constants. If* $\boldsymbol{H}$ *is non-singular it can be inverted giving the following form of the* RCPS *coordinate variations*

$$\delta x^i = (H^{-1})^{ij} v_\beta^j C^\beta \ . \tag{2.9}$$

Clearly the system $w_\beta^i = (H^{-1})^{ij} v_\beta^j$ *has rank* $D - D_K$ *as it is obtained by a non-singular linear transformation from the sub-frame* $\boldsymbol{v}_\beta$. *Therefore, the variations in eq.* (2.9) *belong to the* $(D - D_K)$*-dimensional linear envelope over the sub-frame* $\boldsymbol{w}_\beta$. ■

The above proposition motivates the introduction of RCPS as topological manifolds. It is clear from the proof that these manifolds are continuous for all points where both the image Hessian and the sub-frame are defined and non-singular.

In the following subsection we give some important examples of RCPS associated with a subset of the principal eigenvectors of the local Hessian.

2.3.2 Definition and detection of topological ridges

Let the local Hessian field $\boldsymbol{H}(\boldsymbol{x})$ have eigenvalues $\lambda_1, \ldots, \lambda_D$ with corresponding eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D$. We can assume that the eigenvalues are labeled in decreasing order of their absolute values: $|\lambda_1| > \cdots > |\lambda_D|$. There are different ways to select a subset of eigenvectors to form the sub-frame $\boldsymbol{v}$ in definition 2.4. In what follows the following definition is most suitable for the interpretation of the RCPS as height ridges and their generalizations.

Definition 2.5 *A topological ridge set $R^{(m_+,m_-)}(L)$ of co-dimension $m_+ + m_- = D_K$ is defined as an* RCPS *associated with the Hessian eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{D_K}$ corresponding to the D_K largest eigenvalues in magnitude λ_α, $\alpha = 1, \ldots, D_K$. Excluded are points where the Hessian is degenerate so that there is no unique set of λ_α. In the so defined point set only those points are included where there are exactly m_+ positive and m_- negative Hessian eigenvalues.*

This definition puts a natural label on the topological ridge. For example if $m_+ = 0$, $D_K = m_-$ we obtain a classical height ridge, if $m_- = 0$, $D_K = m_+$ we have a valley. In the general cases where $m_+ \neq 0$, $m_- \neq 0$ we can talk of "saddle ridges". The definition extends the $D_K = D$ case where the ridge is of dimension zero. Then we have the possibilities of maxima, minima and saddle points of different signature. Note that the classification implied by definition 2.5 is richer than the one induced from the value of the topological number alone. For non-degenerate critical points the homotopy class is equal to the sign of the determinant of the Hessian (see detailed proof in [54]). Obviously a lot of different signatures discriminated by definition 2.5 will have the same homotopy number.

As a consequence from proposition 2.2, the topological ridges are manifolds of dimension $D - D_K$. In this case the vectors $\boldsymbol{v}_\alpha$ are eigenvectors of the Hessian and we can choose the frame $\{\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta\}$ from the proof 2.2 to be the entire system of eigenvectors of $\boldsymbol{H}$ (and therefore also of $\boldsymbol{H}^{-1}$).

In the case of topological ridges we have for all $\alpha = 1, \ldots, D_K$ that $H_{ij} v_\alpha^j = \lambda_\alpha v_\alpha^i$ (no summation over α) and therefore eq. (2.8) takes the form

$$\lambda_\alpha \delta x^i v_\alpha^i = 0 \quad \text{(no summation over } \alpha\text{)} \ .$$

It is easy to see that if $\lambda_\alpha \neq 0$ for all α then eq. (2.9) now takes the simpler form

$$\delta x^i = v_\beta^i C_\beta \ .$$

Therefore, in addition to general RCPS properties, the topological ridges are locally *orthogonal* to the vector system $\boldsymbol{v}_\alpha$.

A different definition may include the eigenvectors corresponding to the first D_K eigenvalues of the Hessian ordered by their *signed value*. Such a scheme is extensively studied in [19]. The corresponding classification in that case will partially overlap with ours, namely for the cases of $R^{m,0}$ and $R^{0,m}$ RCPS (strict positive or strict negative ridges). For the mixed signatures the two schemes will segment out different topological manifolds. We note that our general constructive approach from definition 2.4 can be used for both definitions.

2.3.3 Inclusion hierarchy of ridges

A direct consequence of the definition of topological ridges as relative critical sets is the relation

$$m^+ \geq {m'}^+; m^- \geq {m'}^- \rightarrow R^{(m_+,m_-)}(L) \in R^{(m'_+,m'_-)}(L) \ . \tag{2.10}$$

The importance of this inclusion is discussed in the next section where it provides a way for detection of both critical lines and their annihilation points in one-dimensional signals. Another practical application includes detection of the optimal scale for elongated image structures and simultaneously establishing a link to the finest scale.

2.3.4 Topological edges

In order to illustrate our geometrical construction we modify definition 2.5 to localize edges, or borders between gray-scale objects in images. A trivial approach to this issue would be to define the system of edges as the point set $R^{0,1}$ taken on the gradient magnitude of the image $L(\boldsymbol{x})$. This construction can be useful but it involves higher order differentials (the Hessian of the gradient will contain third order derivatives of the original image) so we propose a *second order alternative*.

Definition 2.6 *A topological edge is the point set $E(L)$ where the quantity*

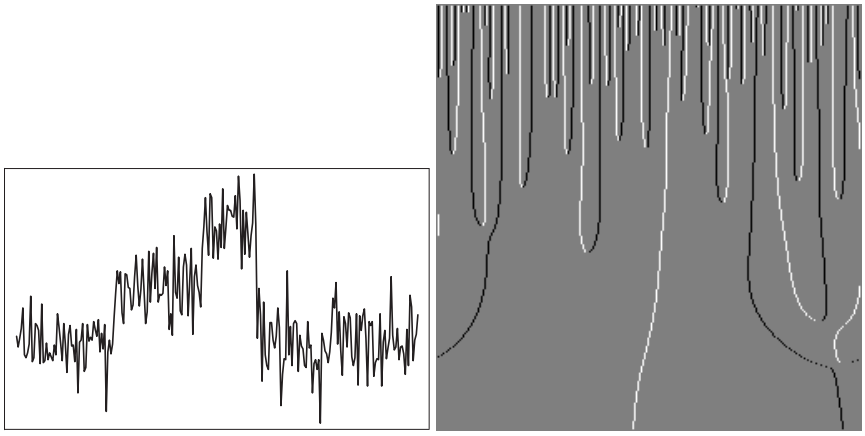
$$\eta(\boldsymbol{x}) = \frac{1}{2}\left(\operatorname{sign}\left(\chi_i\chi_j H_{ij}(\boldsymbol{x}+\rho\boldsymbol{\chi})\right) - \operatorname{sign}\left(\chi_i\chi_j H_{ij}(\boldsymbol{x}-\rho\boldsymbol{\chi})\right)\right) \tag{2.11}$$

is equal to -1. Here H_{ij} is the local Hessian tensor, χ_i is the normalized gradient vector as in eq. (2.3) *and ρ is a small constant defining the precision with which the edge is localized.*

In fact eq. (2.11) is just the difference in sign of the second derivative in the direction of the gradient. In our notation we interpret the topological edge as a RCPS defined by the one-dimensional vector frame constituted by the gradient field. Note, that we can regard $\boldsymbol{\chi}$ as the sub-dimensional frame field $\boldsymbol{v}$ in which we want to detect the singularities of the vector field $H_{ij}\chi_i$.

m_- \ m_+	0	1
0	regular	minimum
1	maximum	×

Table 2.1 The possible point sets for one-dimensional signals. An '×' stands for an impossible configuration.



(a) (b)

Figure 2.2 (a) One-dimensional signal of 256 data points. (b) Scale-space evolution of the one-dimensional topological number. Vertical axis represents the scale parameter. Light and dark lines represent evolution of maxima and minima respectively. Both types of singular points annihilate at catastrophe points.

2.4 Examples of RCPS in images

Here we present different instances of the point sets $R^{m_+,m_-}(L)$ and $E(L)$.

The first example presented in figure 2.2 illustrates both the concept of the one-dimensional homotopy number and the scale-space evolution properties of the corresponding point sets (the sets of critical points in this case). All possible cases are summarized in table 2.1. The RCPS in this case are discrete point sets that consist of all maxima $R^{0,1}$ and all minima $R^{1,0}$.

The second example uses the same input as in figure 2.2 but instead of tracing the scale-space evolution of the critical points we regarded the whole scale-space of the original signal as a two-dimensional signal (the second dimension is the scale). We can then localize the ridges $R^{1,0}$, $R^{0,1}$ as well as the 2D critical point sets $R^{2,0} \cup R^{0,2}$ (this is the set of all extrema) and $R^{1,1}$ (the set of all saddle points). For a listing of all cases refer to table 2.2. We see that the inclusion property (2.10) assigns the set of saddle points as the set of annihilation points for the 1D critical trajectories in figure 2.2 where they appear as 2D positive and negative ridges in figure 2.3. Noticeably, there are no extrema points ($R^{2,0}$ or $R^{0,2}$) in this figure. The

m_- \ m_+	0	1	2
0	regular	negative ridge	minimum
1	positive ridge	saddle	×
2	maximum	×	×

Table 2.2 The possible point sets for two-dimensional signals. An '×' stands for an impossible configuration.

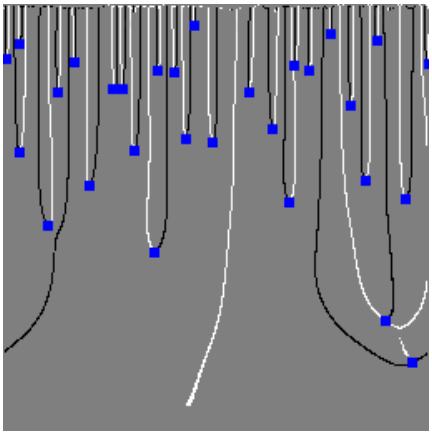


Figure 2.3 For the same 1D signal as in figure 2.2(a), we first build the scale-space and then localize the ridges (white) and valleys (black) and the singular saddle points (blue).

reason is that the linear diffusion that generates the scale-space of the signal does not create new minima or maxima beyond the initial scale. This also implies that linear diffusion does not allow for scale selection.

An example where all relative point sets are present can be obtained by taking the natural derivatives of the 1D signals and their scale evolution. This means that new maxima and minima can be created beyond the initial scale, allowing for scale selection. We illustrate this with a one-dimensional signal from which we take the natural second derivative across scale, and look as in the previous case at the two-dimensional critical point sets of this scaling derivative. Natural derivatives are dimensionless derivatives, which allows for comparing derivatives of different orders. They are defined by multiplying the normal derivative by the scale

$$\overline{\partial}_i = \sigma \partial_i \ ,$$

with $\overline{\partial}_i$ the natural derivative and $\sigma = \sqrt{t}$ the scale.

Figure 2.4 shows the point sets for the second natural derivative of the one-dimensional signal of figure 2.2(a). We see that the local maxima and minima are subsets of the ridges and valleys, as was discussed in section 2.3.3. Most of

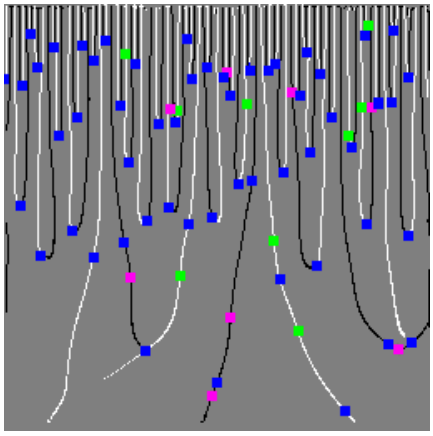
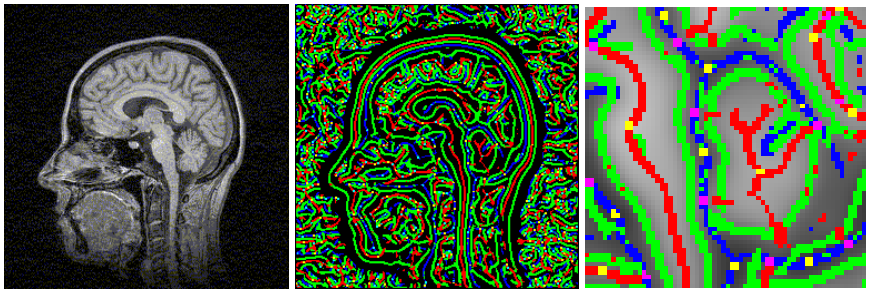


Figure 2.4 The topological point sets for the second natural derivative of the 1D signal of figure 2.2(a). White corresponds to positive ridges, black to negative ridges, blue to saddle points, green to maxima and purple to minima.



(a) (b) (c)

Figure 2.5 (a) An MRI 256×256 sagittal image of a human head. (b) Complete set of 2D positive ridges (red) and negative ridges (blue), edges (green) and singular points (extrema: yellow, saddle points: purple). The calculations are done for a spatial scale of 2 pixels. (c) A 64×64 fragment of the original image is superimposed with the corresponding RCPS.

the saddle points are annihilation points of the ridges and valleys. Saddle points on a ridge are local minima in the direction along the ridge, i.e. the direction of the eigenvector of the smallest absolute eigenvalue. On a valley they are local maxima in that direction.

We see in figure 2.4 that all possible point sets as classified in table 2.2 occur. The local maxima and minima points, $R^{0,2}$ and $R^{2,0}$, serve as scale selectors, the points at which maximum response—in absolute sense—of the filter is found. These points are an indication for the scale at which structure in the original signal is observed. Via the ridge or valley to which an extremum belongs, we can trace back to the finest scale, which restores the location of the local structure. Since we have taken the second derivative, this will be a needle like structure.

m_- \ m_+	0	1	2	3
0	regular	negative surface	negative string	minimum
1	positive surface	saddle string	saddle point (1)	×
2	positive string	saddle point (2)	×	×
3	maximum	×	×	×

Table 2.3 The possible point sets for three-dimensional signals. An '×' stands for an impossible configuration.

For the large scales the positive and negative ridges annihilate each other in saddle points. These points select the scale up to which structure exists, i.e. up to which scale it can be observed.

In the previous two examples the 2D technique is used in scale-spaces of a one-dimensional signal. The complete relative critical set system of a 2D image is given in figure 2.5. Note the inclusion relations between the different RCPS.

Now we turn to the 3D case. The possible RCPS are listed in table 2.3. Note that in this case we have ridges of two different dimensionalities representing surfaces and strings. Furthermore, the saddle points are of two possible types: $R^{2,1}$ and $R^{1,2}$.

As an example of the topological method in 3D we analyze the 3D scale-space built over a two-dimensional image of an anisotropic Gaussian blob with $\sigma_x = 5.0$ and $\sigma_y = 35.0$ pixels, see figure 2.6(a). We computed the scale-space of this image, i.e. we blurred it with a two-dimensional Gaussian filter, for 64 scales, running exponentially from 1.0 to 20.0 pixels. The only critical sets we can find from the scale-space are surfaces and strings. Local extrema ($R^{0,3}$ and $R^{3,0}$) can not be found, as was the case in the two-dimensional scale-space example (figure 2.3), because of the attenuation of the diffusion equation. In figure 2.6(b) we show in blue the positive surface of the scale-space and in green the string of maxima. These are the only critical subsets that can be found for this image. Note that on a fixed scale the positive surface is nothing more than the two-dimensional ridge and that the string of maxima is the local maximum of the blob at that scale.

In figure 2.6(d) we have depicted the sets $R^{0,1}$, positive surface, $R^{1,0}$, negative surface, $R^{0,2}$, string of maxima and $R^{2,0}$, string of minima of the natural Laplacian of the blob (figure 2.6(c)). The natural Laplacian is the operator $\sigma^2\partial_{ii}$. The middle surface has become a negative surface in comparison to figure 2.6(b) because the main lobe of the second derivative of the Gaussian is negative. The positive surfaces are due to the second derivative. The strings of maxima are subsets of the positive surface, cf. section 2.3.3.

For clarity we have shown in figure 2.6(e) again the negative surface and the strings of minima, and added the set $R^{3,0}$, a local minimum, which was invisible in the middle frame because of the positive surfaces. As with the positive strings, the strings of minima are subsets of the negative surface whereas the local minimum is a subset of the negative strings. Going from small scales to large scales,

we see two strings start at the endpoints of the Laplacian of the blob. For increasing scales they start to converge to the middle. In the same scale range we see a string at the middle, which is oriented in the scale direction. Then we have a scale range in which no strings are found, because the scale-space structures interfere with each other and it is no longer possible to define a local coordinate frame based on the Hessian. For still larger scales we find a primarily horizontal string, marking the optimal scale at which the negative part of the Laplacian of the blob can be detected. On the horizontal string, we find in red a local minimum. For the larger scales three strings start to disperse for increasing scale, the middle one being vertical. If we extrapolate the small scale vertical string upwards, i.e. to larger scales, and the large scale vertical string downwards, they connect and cross the horizontal string in the local minimum, which is the optimal minimum with respect to scale.

As a final set of examples we show how we can introduce model information by choosing a particular local vector frame instead of the local Hessian frame field from the general definition 2.4. We will concentrate on the detection of elongated structures in two-dimensional images.

First we show an example of a synthetic two-dimensional image of horizontal bars, see figure 2.7. This image has clearly two optimal scales at which elongated structures can be detected. At small scales we perceive the horizontal bars separately, but at larger scales they will be grouped into a vertical line. If we would use the natural Laplacian for building a scaling representation of the image we could compute the one-dimensional RCPS for detecting these two types of structures, but in such a construction we would also detect vertical strings representing the scale evolution of the local extrema and saddle points of the two-dimensional image. To suppress the vertical strings we have selected the two-dimensional frame field $\boldsymbol{v}(\boldsymbol{x})$ from the general definition 2.4 with one of the vectors fixed in the direction of increasing scale. The other frame vector is determined by the largest absolute eigenvalue of the Hessian in the xy-plane. We refer to these strings as *scale-space strings*. Clamping one of the frame vectors in the direction of the scale parameter is a natural choice for detecting elongated structures of two-dimensional images in scale-space. It is dictated by the interpretation of the final result as elongated structures in the original image *lifted* to a certain scale. This scale we can interpret as the *optimal scale* of response for the corresponding structure.

Figure 2.7(b) shows the scale-space strings of figure 2.7(a). For small scales (bottom of the box) we find the horizontal bars whereas for larger scales they are grouped on a vertical line.

The following example is a two-dimensional fundus reflection image from a scanning laser ophthalmoscope, see figure 2.8, in which we want to detect the vessels of the retina at their optimal scale, i.e. we want to detect two-dimensional elongated structures at their optimal scale. To this end we have detected the scale-space strings of the natural Laplacian, as discussed in the previous example. Figure 2.8(b) shows in red the scale-space strings, which detect the elongated structures at their optimal scale in scale-space.

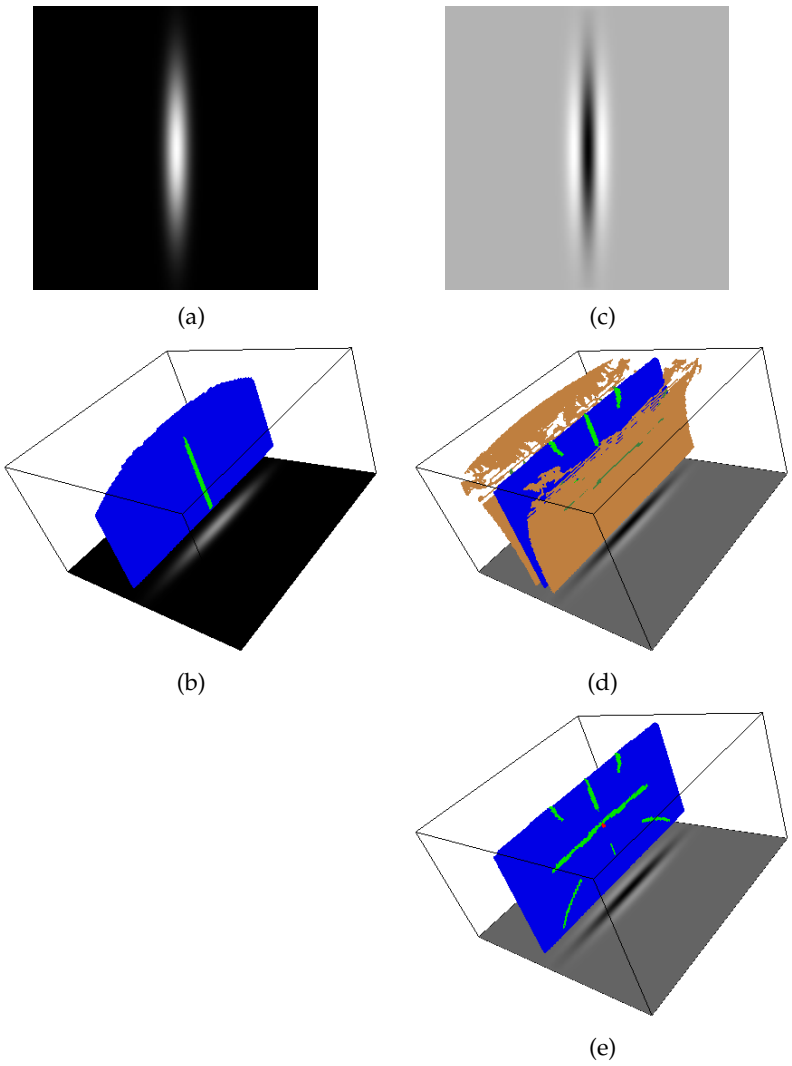
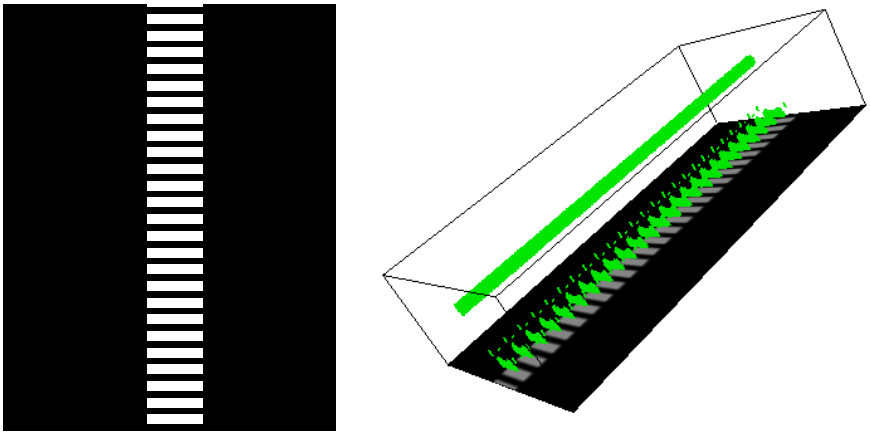
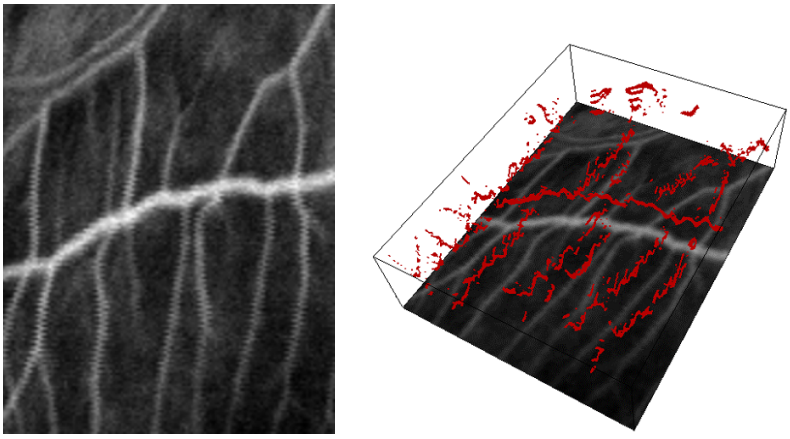


Figure 2.6 (a) A two-dimensional anisotropic Gaussian blob with $\sigma_x = 5.0$ and $\sigma_y = 35.0$ pixels. (b) The positive surface (blue) and the string of maxima (green) of the blob in scale-space. At the bottom of the box we have shown the blob again. The scale runs exponentially in 64 steps from 1.0 to 20.0 pixels upwards in the figure box. The image is 128 pixels square. (c) The natural Laplacian of the blob at a scale of 1.0 pixel. (d) Positive surfaces (brown) of the natural Laplacian of the blob in scale, negative surface (blue) of the natural Laplacian of the blob in scale, strings of maxima (dark green, on the positive surface), and strings of minima (light green, on the negative surface). (e) Negative surface (blue) of the natural Laplacian of the blob in scale, strings of minima (green) and a local minimum (red).



(a) (b)

Figure 2.7 (a) An image of 64×128 pixels with bars of 11×3 pixels. The bars are separated by 2 pixels. (b) The scale-space strings of the natural Laplacian of the left image. The scale runs exponentially from 1.0 to 10.0 pixels in 32 steps.



(a) (b)

Figure 2.8 (a) Vessel structure of the retina. (b) Scale-space strings (red) of the natural Laplacian of the left image. We fixed one vector of the local frame field in the direction of increasing scale.

Another way for detecting elongated structures in two-dimensional images can be provided by local orientation analysis. With any 2D gray-scale image we can associate a 3D orientation bundle given as

$$F(x,y,\theta) = \int \mathrm{d}x'\mathrm{d}y'\Psi(x-x',y-y',\theta)L(x',y') \ ,$$

where $\Psi(x,y,\theta)$ is any orientation filter. In such a representation the bundle can be regarded as a 3D signal in the space (x,y,θ).

A natural choice for the frame field in this case is a 2D vector sub-frame in which one of the vectors is always in the direction of angular changes, i.e. in the vertical direction, and the second vector points in the direction perpendicular to the angle at the point (x,y,θ), i.e. $(-\sin\theta,\cos\theta,0)$, since maximal response for an elongated structure is to be expected *along* the direction of the orientation filter.

In the following examples we have chosen as the orientation filter an anisotropic Gaussian kernel with a scale of 1.0 pixel along the short axis and 8.0 pixels along the long axis. In figure 2.9 we depict the positive strings relative to the above defined two-dimensional vector frame for an image of two crossing lines. The angle runs from 0 to π in 32 steps.

We clearly see that the two lines "unsplit" in the orientation bundle. They cross at the same spatial position but at different angles. We have only computed the bundle for $\theta \in [0,\pi)$ because of the $\theta \to \theta + \pi$ symmetry of the filter.

Finally, in figure 2.10 we present the same analysis as in figure 2.9 for the fundus reflection image. In figure 2.10(a) we show all the positive strings that are detected. In figure 2.10(b) the 22 longest strings are shown. Both images show clearly the unsplitting of crossing structures in orientation space.

2.5 Applicability of the method

The introduced method is based on the *constructive* definition 2.4 of the RCPS and therefore there is no need for intrinsic validation of the scheme. Any data point that obeys the definitive condition belongs to the set and vice versa. Different RCPS definitions can lead of course to different point sets, but each one will be consistent with its own definition.

An important validation question arises however in relation to the *interpretation* of the RCPS of various signature as specific image properties with perceptual significance. Such a correspondence will justify (or deny) the application of these mathematical constructions in real image analysis.

RCPS represent in general sub-dimensional structures in images such as one-dimensional string like structures (catheters, blood vessels etc.) or two-dimensional surface types of structures (edges, fault surfaces in seismic data etc.). If a given image feature can be *modeled or approximated* with such structures, then the RCPS method can applied successfully.

We show an example in figure 2.11(a) from a real clinical application where a dense network of blood-vessels in an optical image of the human retina has to

be segmented before laser treatment. As a starting point, we model the blood-vessels as one-dimensional structures and approximate them with their center luminance line. It can be seen from the result in figure 2.11(b) that the sensitivity of the system of ridges when used to represent the center lines of the blood vessels is nearly 100%. At the same time, the specificity of the method is low because many ridges do not represent actual blood vessels. The system of ridges serves as a superset from which one has to recognize and select those ridges corresponding to blood vessels. A method for the detection and full segmentation of the vasculature in *color* images of the retina is described in chapter 4.

2.6 Conclusions

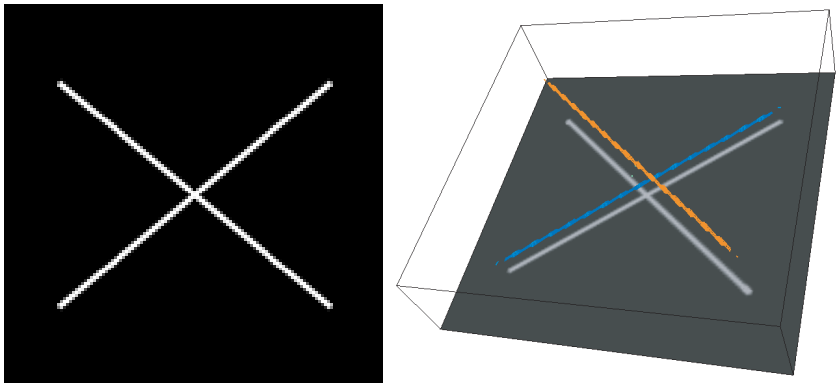
In this chapter we propose a constructive definition of relative critical sets in images of any number of spatial dimensions. The definition is very flexible because it associates critical sets to an arbitrary chosen local vector frame field. Depending on the visual task, different model structures can be identified with the relative critical sets. As a consequence our construction can be purely intrinsic (defined only by the image structures), or it can involve externally specified frames. The last situation may be useful for exploiting additional model information. In this chapter we demonstrate the examples of one of the most popular intrinsic cases: the set of topological ridges.

The relative critical sets are in general connected sub-manifolds. Therefore, our technique provides indeed a method for *perceptual grouping* achieved with only *local* measurements. In a sense such a technique can be viewed as a particular extension of the threshold techniques where the connected entities are the level surfaces (or lines in 2D).

The grouping properties of the system of ridges are demonstrated in optimal scale selection of a multi-scale connected object (see figure 2.6). The method also provides a *linkage* from the scale-space structure down to the original image space. We refer to these applications as *topological deep-structure analysis*.

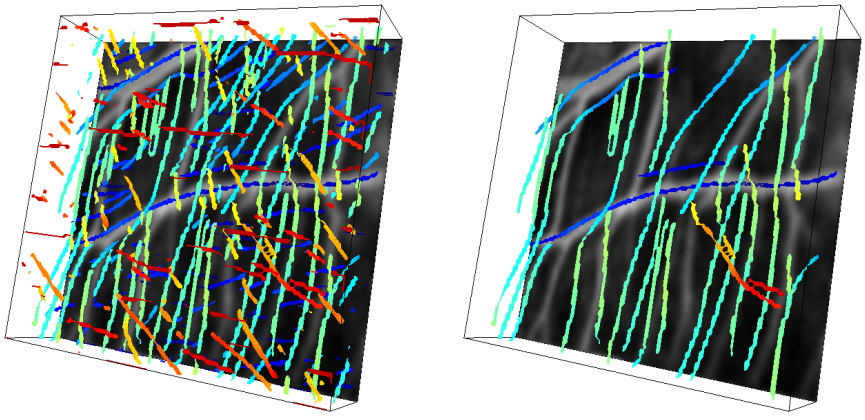
In the above examples the externally specified vector frames incorporate model information. One example concerns scale selection (figures 2.7 and 2.8). Another example shows detection of elongated structures in an orientation bundle representation of 2D images (figures 2.9 and 2.10).

As already discussed in section 2.5, the major criteria for applying this technique is the correspondence of the RCPS to *essential image features*. The important features of the image are those that determine the contextual content of the image. As can be observed from e.g. figure 2.11, further processing is required to select only those subsets of the RCPS that are relevant for the larger scale context. In chapters 4 and 6 the techniques for RCPS detection are used as an initial step for obtaining full segmentations of elongated structures in medical images.



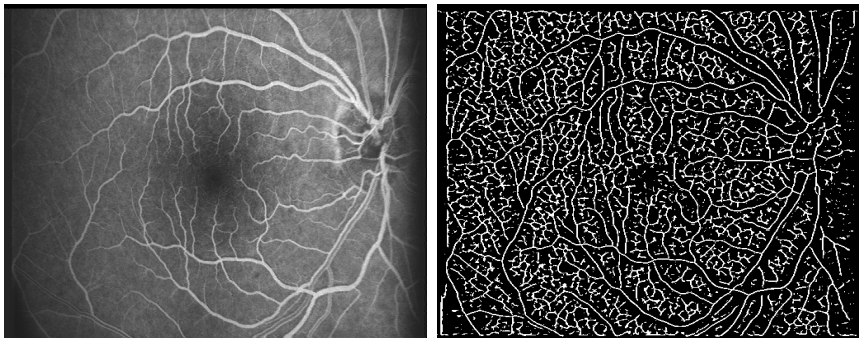
(a) (b)

Figure 2.9 (a) Simulated 128x128 pixels image representing two crossing lines. (b) Orientation space strings obtained as RCPS derived from the 2D frame field where one vector points always vertically in the θ direction. The other vector is $(-\sin\theta, \cos\theta, 0)$, i.e. it is perpendicular to the local orientation of the filter. The local angle (vertical axis) is color-coded (0 is dark blue, π is red) for better rendering effect.



(a) (b)

Figure 2.10 Orientation space strings for a retinal image. (a) All computed positive strings. (b) The 22 longest strings. The vector frame is chosen as in the computation presented in fig. 2.9.



(a) (b)

Figure 2.11 (a) Optical fluorescence image of a human retina. (b) Positive ridges computed at regularization scale of 1 pixel.

Appendix: numerical implementation

In this section we discuss the computational algorithm of formulas (2.7) and (2.3). According to the previous sections, the integral has to be calculated on a surface lying in a linear subspace and taken close around a given test point. The computation of the vectors defining this subspace (when necessary) depends on the application and will not be presented here. In the case of ridges, for example, the principal directions of the Hessian can be obtained by any diagonalization procedure. We assume in what follows that the linear space is defined in every image point by a given orthonormal system of vectors v_α^i, $i = 1, \ldots, D$, $\alpha = 1, \ldots, D_K$.

To compute the number (2.7) we will use its topological property that it can be defined by integration of eq. (2.3) on *any* surface surrounding the test point $\boldsymbol{x}_P$. Therefore we can select for a border of a D_K-dimensional *cube* those ribs which are parallel to v_α^i. The surface is formed by $2D_K$, $(D_K - 1)$-dimensional cubes, parallel in pairs, which we denote as S_A^+, S_A^-, $A = 1, \ldots, D_K$. If $D_K = 2$ for example, $S_A^\pm$, $A = 1, 2$ are just line segments; if $D_K = 3$ then $S_A^\pm$, $A = 1, 2, 3$ are the six walls of the cube, etc. Next we *parameterize* the integration points on each of the cubes $S_A^\pm$ as

$$\boldsymbol{x}_A^\pm(s_1, \ldots, s_{D_K-1}) = \boldsymbol{x}_P \pm \frac{\rho}{2}\boldsymbol{v}_A + \sum_a^{D_K-1} (s_a \frac{\rho}{N} - \frac{\rho}{2})\boldsymbol{v}_{t(A,a)} \ . \qquad (2.12)$$

The real constant ρ measures the size of the neighborhood around the point P and represents the spatial precision. The integer constant N represents the number of integration points along each direction of the cube border. Finally, the function $t(A, a)$ gives the a-th element from the set $[1, \ldots, D_K] - [A]$, or in other words $t(A, a) = a$ when $a < A$ and $t(A, a) = a + 1$ when $a \geq A$. Surface coordinates s_a, $a = 1, \ldots, D_K - 1$ in eq. (2.12) take integer values $1, \ldots, N$.

We can compute the projected gradient vector in these points according to eq. (2.6)

$$g^{\pm}_{\alpha,A}(s_1,\ldots,s_{D_K-1}) = \sum_i v^i_\alpha(\boldsymbol{x}) g_i(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}^{\pm}_A(s_1,\ldots,s_{D_K-1})} \ . \tag{2.13}$$

For the examples in this work we have used a simple multidimensional interpolation algorithm of first order to calculate the gradient values with sub-pixel precision in eq. (2.13). Because of the topological nature of the quantity (2.7) any interpolation algorithm will give the same result.

Substituting the differentials in eq. (2.3) with finite differences on the borders $S^{\pm}_A$ and replacing the integration in eq. (2.7) with summation over the indices $s_1,\ldots,s_{D_K-1}$ we obtain the following expression for the discrete computation of the topological number

$$\begin{aligned}\nu_K(\boldsymbol{x}_P) = \frac{1}{N^{D_K-1}} \sum_A^{D_K} \sum_{\sigma=-,+} \sigma \sum_{\alpha_1,\ldots,\alpha_{D_K}} \varepsilon^{\alpha_1,\ldots,\alpha_{D_K}} \\ \sum_{s_1,\ldots,s_{D_K-1}} g^{\sigma}_{\alpha_1,A}(s_1,\ldots,s_{D_K-1}) g^{\sigma}_{\alpha_2,A}(s_1+1,\ldots,s_{D_K-1}) \\ g^{\sigma}_{\alpha_3,A}(s_1,s_2+1,\ldots,s_{D_K-1})\cdots g^{\sigma}_{\alpha_{D_K},A}(s_1,\ldots,s_{D_K-1}+1) \ .\end{aligned}$$

In the above formula the anti-symmetry of the tensor ε has been used to exclude components symmetric in the indices α.

Chapter 3

Primitive construction

3.1 Introduction

IN the previous chapter we have seen how to detect interesting image structure in the form of sub-dimensional topological point sets, such as ridges and edges. In this chapter we investigate the construction of useful primitives based on these point sets. We first consider 1D point sets in 2D and 3D images and continue with 2D point sets in 3D images.

3.2 Line elements

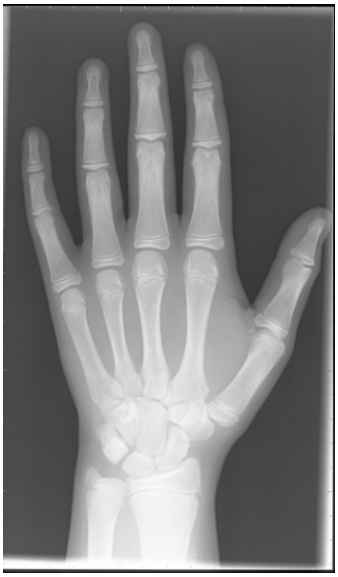
Point sets that are intrinsically one-dimensional form curves, see figure 3.1. Since line elements are a natural representation for curves of arbitrary shape, we put forward an algorithm to construct line elements. An important ingredient in our algorithm is the concept of a convex point set.

A convex point set is a set of points such that with every couple of points that belong to the set, all points that lie in-between these two points on the line connecting them also belong to the set. In more formal terms

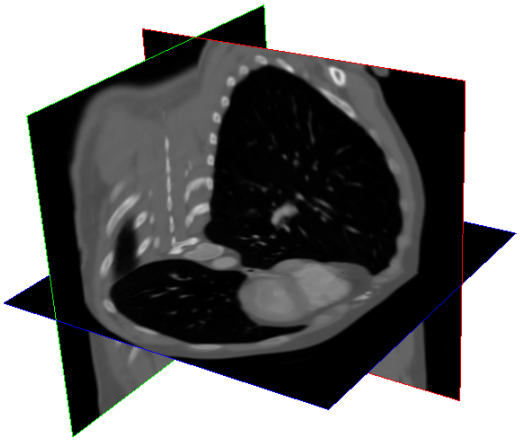
Definition 3.1 *A point set P is convex if for all points $\boldsymbol{x} \in P$, $\boldsymbol{y} \in P$, and for a scalar $s \in [0,1]$ the point $\boldsymbol{z}(s) = \boldsymbol{x}s + \boldsymbol{y}(1-s) \in P$.*

The only convex sets of dimension equal to one are straight line elements. So, constructing convex sets from 1D point sets gives the line elements we require. However, since the point sets are defined on the discrete lattice, some regularization is needed. Therefore, we propose an adapted form of the basic definition for convex sets, in which the directional information of the point sets is exploited. Note that the point sets are not only characterized by their position, but also by a vector $\boldsymbol{v}$, e.g. the gradient vector in the case of edges and one of the principal eigenvectors of the Hessian matrix in the case of ridges (cf. section 2.3). Using the directional information replaces Euclidean convexity by geodesic convexity. The resulting sets are called *affine* convex sets.

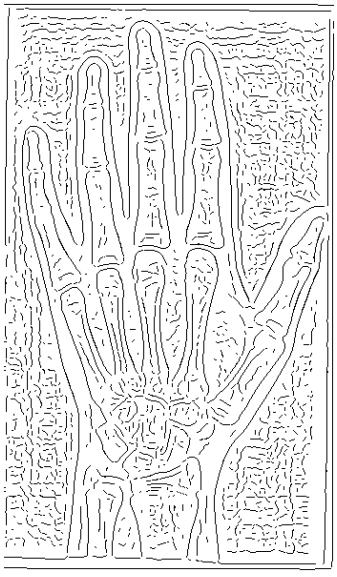
The mechanism to obtain affine convex sets is a region growing algorithm which compares the direction of $\boldsymbol{v}$ at a seed point $\boldsymbol{x}_s$ with the directions of $\boldsymbol{v}$ at



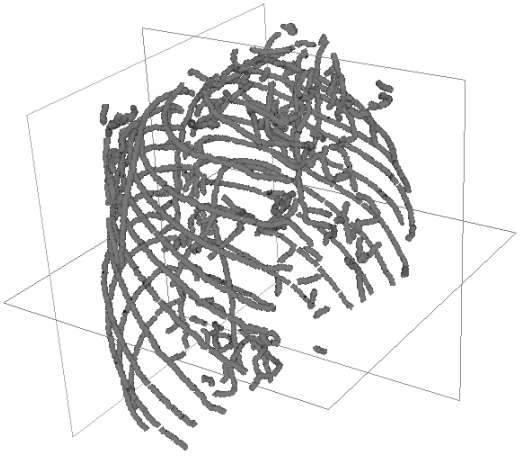
(a)



(b)



(c)



(d)

Figure 3.1 (a) X-ray image of a left hand. Image size is 348×598 pixels. (b) Three orthogonal slices of a CT scan. Image size is $256 \times 256 \times 235$ voxels. (c) The edges of (a) at $\sigma = 2.0$ pixel. (d) Ridge strings of (b), after thresholding the CT scan at 100 Hounsfield units and blurring the binary image with $\sigma = 2.0$ voxel (for further details, see chapter 6).

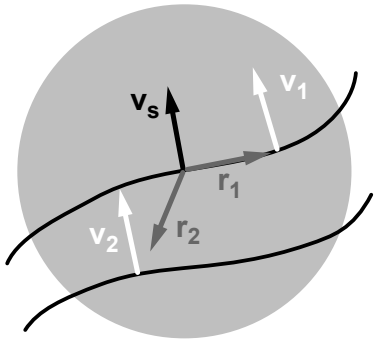


Figure 3.2 Construction of convex sets. The dark curved lines are two topological point sets. The diameter of the disk is ϵ_c. The vector v_s characterizes direction of the seed point, v_1 and v_2 are the vectors at unconnected points. The vectors r_1 and r_2 are unit vectors pointing from the seed point to the unconnected points. The point that belongs to the same point set will be added to the convex set, because it satisfies the conditions in (3.1)–(3.3). The pixel on the parallel point set does not satisfy condition (3.3) and will not be added. Note that we have shown the situation in 2D; in 3D the directional vectors are tangential to the point sets.

unconnected points $\boldsymbol{x}_u$ in a neighborhood of radius ϵ_c. Here, the subscript 's' stands for seed, the subscript 'u' for unconnected and the subscript 'c' for connectivity. The condition for adding an unconnected point to the set of connected points within the neighborhood is based on two comparisons

1. Is the direction of $\boldsymbol{v}(\boldsymbol{x}_s)$ at the seed point and the direction of $\boldsymbol{v}(\boldsymbol{x}_u)$ at the unconnected point similar?

2. If so, are the points on the same point set or are they on parallel point sets?

The first question can be checked by taking the scalar product of $\boldsymbol{v}(\boldsymbol{x}_s)$ and $\boldsymbol{v}(\boldsymbol{x}_u)$. If the orientations are similar, the scalar product will be close to 1. However, points in parallel point sets may possess similar orientations, see figure 3.2. That is why the second question is posed. For point sets in 2D, parallellity can be checked by taking the unit-length normalized vector $\boldsymbol{r}$ between the locations of the two points under consideration and computing the vector product between $\boldsymbol{r}$ and $\boldsymbol{v}(\boldsymbol{x}_s)$. If the points are on the same segment, the magnitude of the vector product will be close to 1. For ridge strings in 3D images, the situation is a little bit different. Here we use the tangential direction to characterize the orientation of the point set, which is in the same direction as $\boldsymbol{r}$ for ridge points on the same segment. To test on parallellity in that case, the inner product of $\boldsymbol{v}(\boldsymbol{x}_s)$ and $\boldsymbol{r}$ is computed, which is close to 1 for points on the same segment.

To summarize, the following inequalities are checked

$$\text{connectivity:} \quad \|\boldsymbol{x}_\mathrm{s} - \boldsymbol{x}_\mathrm{u}\| \leq \epsilon_\mathrm{c} \ , \tag{3.1}$$

$$\text{orientation:} \quad |\boldsymbol{v}(\boldsymbol{x}_\mathrm{s}) \cdot \boldsymbol{v}(\boldsymbol{x}_\mathrm{u})| \geq \epsilon_\mathrm{o} \ , \tag{3.2}$$

$$\text{parallellity:} \quad \|\boldsymbol{v}(\boldsymbol{x}_\mathrm{s}) \diamond \boldsymbol{r}\| \geq \epsilon_\mathrm{p} \ , \tag{3.3}$$

where $\diamond$ represents the vector product in 2D and the scalar product in 3D. The subscript 'o' stands for orientation and 'p' for parallelism. The conditions posed in eqs. (3.1)–(3.3) are visualized in figure 3.2.

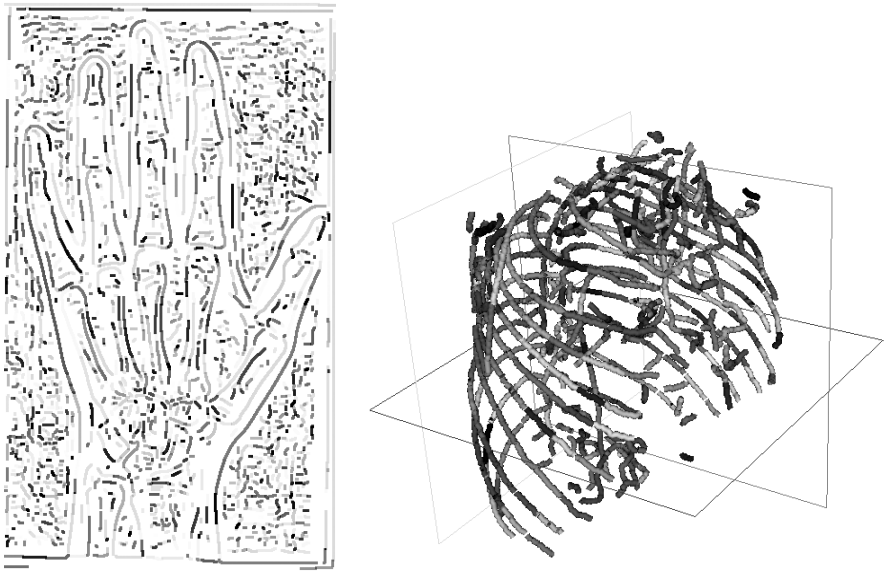
If the detection of the point sets is not perfect, e.g. due to interfering image structure or noise, ϵ_c can overcome the discontinuity caused by gaps in the point sets. The parameter ϵ_o controls the amount of curvature that is allowed in the convex set. If it is set close to zero, highly curved convex sets can be formed. If it is set close to one, straighter convex sets are obtained. The parameter ϵ_p prevents the growing process to jump to parallel point sets. A value close to one is recommended.

The algorithm for the construction of convex sets is as follows. Pick a point in the point set at random, remove it from the point set and mark it as seed. Find the neighbors around the seed. If the neighbors adhere to the conditions in eqs. (3.1)–(3.3) remove the one that is closest to the seed from the point set and add it to the convex set. Mark the newly added point as seed and continue until no more neighbors can be found. In figure 3.3 the convex sets for the images in figure 3.1 are shown.

Applying some pre- and postprocessing in the construction of the convex sets can be beneficial. In later chapters the convex sets will be used to extract image features. In 2D images some of these features are based on the intensity profiles perpendicular to pixels in the convex set, i.e. in the direction of $\boldsymbol{v}$. And in 2D and 3D a measure for the curvature of the convex set is computed based on $\boldsymbol{v}$. Because both $\boldsymbol{v}$ and $-\boldsymbol{v}$ are valid eigenvectors of the Hessian tensor, there is a sign ambiguity when convex sets are constructed for ridge point sets. To estimate the profiles and curvature correctly, this sign ambiguity must be removed. This can be done if there is an ordering in the convex sets in such a way that if the points in the convex set are indexed from 1 to N, the convex set is traversed from one end-point to the other end-point. In that case we can, by traversing the convex set from one end-point to the other end-point, adjust the sign of the $\boldsymbol{v}$'s in such a way that they are e.g. left oriented, see figure 3.4.

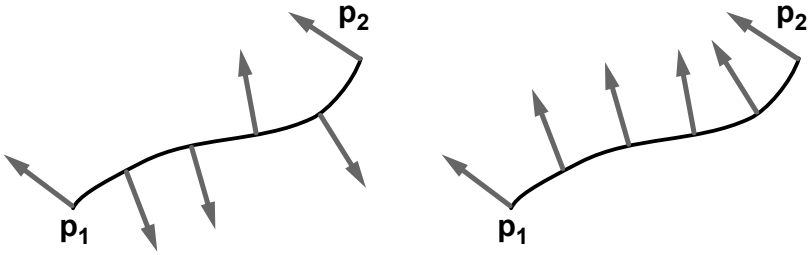
Traversing a convex set is trivial if the set is thinned, that is, each pixel is connected to at most two other pixels. Unfortunately, this is not the case in our detection scheme. We can achieve this, however, by applying skeletonization. Of particular interest are topology conserving skeletonization techniques, since they do not introduce spurious branches or remove holes. In this thesis, skeletonization is performed before the construction of the convex sets. In 2D, a method described in [23] is used. In 3D, the algorithm of [89] is used. Both methods preserve the topology. An overview of skeletonization techniques is given in [68].

During the construction of the convex sets, we allow unconnected pixels to



(a) (b)

Figure 3.3 (a) The convex sets of the edges in figure 3.1(c). Settings used are $\epsilon_c = 3.0$ pixel, $\epsilon_o = 0.95$, $\epsilon_p = 0.95$. (b) The convex sets of the ridge strings in figure 3.1(d). Settings used are $\epsilon_c = 5.0$ voxel, $\epsilon_o = 0.9$, $\epsilon_p = 0.8$. As an extra constraint, the maximum size a convex set can attain is 20 voxels (see chapter 6).



(a) (b)

Figure 3.4 (a) Because both v and $-v$ are eigenvectors of the Hessian matrix, a convex set can have vectors pointing to both its sides. (b) Traversing the convex set from p_1 to p_2, the sign ambiguity can be resolved by demanding all vectors to be left oriented.

be added, so that gaps can be closed. If a gap is encountered, the "missing" pixels can be inserted, for example by using Bresenham's line algorithm, see [11]. Closing a gap means that the vectors v must also be added. In 2D we accomplish this by inserting a vector that is perpendicular to the line through the inserted points. In 3D we take the vector that is tangential to this line.

3.3 Surface patches

For sheet like topological point sets, such as edge and ridge surfaces in 3D, we can construct primitives in the form of surface patches in a similar way as is done for the line elements. Here too, preprocessing in the form of skeletonization can be applied [88]. Because the connectivity of 2D structures in 3D is much better than those of 1D structures in 2D or 3D we only investigate neighbors in a 26-neighborhood on the image lattice (no larger connectivity neighborhood is used, although for poorly connected point sets, introduction of ϵ_c can be considered). In the algorithm, we start again with a randomly chosen seed point. Next we locate all its neighbors in a 26-neighborhood and add them to the patch if they satisfy the conditions in eq. (3.2) and (3.3). Then, the neighbors of the just added points are tested with respect to the seed point and added if they pass the test. This scheme is repeated until no more suitable points can be found. The seed point remains the same throughout the construction of the patch.

Note that in the case of surfaces in eq. (3.3) the vector product is used, because the vector v is always perpendicular to the point set.

In figure 3.5 an example of surface patches is shown.

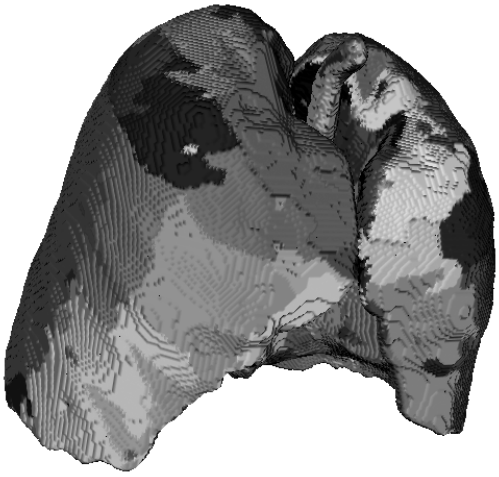


Figure 3.5 Surface patches of the edges of the lungs in a CT scan. Settings used are $\epsilon_o = 0.9$, $\epsilon_p = 0.9$.

Chapter 4

Vessel segmentation in color images of the retina

4.1 Introduction

ASSESSMENT of the characteristics of vessels plays an important role in a variety of medical diagnoses. For these tasks measurements are needed of e.g. vessel width, color, reflectivity, tortuosity, abnormal branching, or the occurrence of vessels of a certain width. When the number of vessels in an image is large, or when a large number of images is acquired, manual delineation of the vessels becomes tedious or even impossible.

The focus of this chapter is on the automated segmentation of vessels in color images of the retina. These images, also known as fundus images, are acquired by making photographs of the back of the eye. We are interested in vessel segmentation for screening of diabetic retinopathy. Diabetes is a disease that affects about 5.5% of the population worldwide, a number that can be expected to increase significantly in the coming years. About 10% of all diabetic patients have diabetic retinopathy, which is the primary cause of blindness in the Western World. Since this type of blindness can be prevented with treatment at an early stage, the WHO advises yearly ocular screening of patients. Automation will facilitate this screening [63].

Knowledge about the location of the vessels can aid in screening of diabetic retinopathy, e.g., to reduce the number of false positives in the detection of microaneurysms [30, 69, 116], to serve as a means for registration of images taken at different time instants or at different locations of the retina [131], or to find the location of the optic disc and the fovea.

Previous methods for vessel segmentation in images of the retina can be divided into two groups. The first group consists of rule-based methods and comprises vessel tracking [13, 16, 32, 124], matched filter responses [15, 33, 43], grouping of edge pixels [98], model based locally adaptive thresholding [49], topology adaptive snakes [79] and morphology based techniques [77, 125, 132]. The second group consists of supervised methods, which require manually labeled images for training. To the best of our knowledge, the only published method in this

category is the neural network scheme for pixel classification by Sinthanayothin et al. [114, 115]. Our method belongs to the last category.

In our opinion, a pixel representation is not optimal to describe the vessel structure. Therefore our approach is based on the intrinsic property that vessels are elongated structures. This observation leads to a primitive based method, which we refer to as PBM. Our algorithm uses image primitives formed from image ridges that are grouped into sets that approximate straight line elements. The sets are used for two purposes. First, features are computed which together with a classifier give a probability that the line element is part of a vessel. Second, the sets divide the image into patches by assigning every pixel of the image to the nearest primitive. Within each patch, the line element defines a local coordinate frame in which local features can be extracted for every pixel. The probability that the line element is part of a vessel is one of these features. The features are used to classify the pixels in the patch into vessel and non-vessel.

Many of the published methods have not been evaluated on large datasets or fail to give good results for large numbers of images as encountered in a screening process. In [114] and [124] evaluation is done on vessel segments and bifurcations. Only in [43] and [49] an evaluation on complete manually labeled images is presented.

We have constructed a database of manually labeled images for training and evaluation of our method. The database consists of 40 images taken from a screening programme for diabetic retinopathy in the Netherlands. We compare our method with two rule-based methods. The first one is the method of Hoover et al. [43], the second one the method of Jiang and Mojon [49]. Hoover et al. have collected a database of manually labeled images, which is publicly available together with the results of their method. For comparison, our system is evaluated on their database too.

The chapter is organized as follows. In section 4.2 a method for extracting ridges and constructing line elements in 2D images is reviewed (cf. chapters 2 and 3). The line elements are used to subdivide the image into patches. Section 4.3 describes the classifier used and the features that are extracted. A description of the material used is given in section 4.4. Section 4.5 presents the results, which are summarized and discussed in section 4.6.

4.2 Representation of vessels

4.2.1 Ridge detection

Since image ridges are natural indicators of vessels, we start our analysis with a short overview of ridge detection for two-dimensional gray value images. For a more extensive discussion on this subject, see [22, 53]. The ridge detection method used in this chapter is described in full detail in [53] and chapter 2. Because the green channel of color fundus images, when formatted as RGB, gives the highest contrast between vessel and background [43], this channel is used for

extraction of the image ridges.

Ridges are defined as points where the image has an extremum in the direction of the largest surface curvature. In chapter 2 a detailed discussion is given on the detection of ridge sets. Here we give a short overview of a simplified algorithm for 2D images.

The direction of largest surface curvature is the eigenvector $\boldsymbol{v}$ of the matrix of second order derivatives of the image corresponding to that eigenvalue λ which has the largest magnitude. This matrix is referred to as the Hessian matrix $\boldsymbol{H}$. The sign of λ determines whether a local minimum ($\lambda > 0$) or a local maximum ($\lambda < 0$) is found.

Because taking derivatives of discrete images is an ill-posed operation, they are taken at a scale σ using the Gaussian scale-space technique, see section 1.4.2. The main idea is that the image derivatives can be taken by convolving the image with derivatives of a Gaussian

$$L_{x_j} = \frac{\partial L(\boldsymbol{x},\sigma)}{\partial x_j} = \frac{1}{2\pi\sigma^2} \int_{\boldsymbol{x}' \in \mathbb{R}^2} \frac{\partial \mathrm{e}^{-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/2\sigma^2}}{\partial x_j} L(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}',$$

where x_j is the image coordinate with respect to which the derivative is taken. Mixed and higher order derivatives are computed by taking mixed and higher order derivatives of the Gaussian kernel.

It is now possible to define a scalar field $\rho(\boldsymbol{x},\sigma)$ over the image that takes value -1 for ridges of local minima, $+1$ for ridges of local maxima and 0 elsewhere as follows:

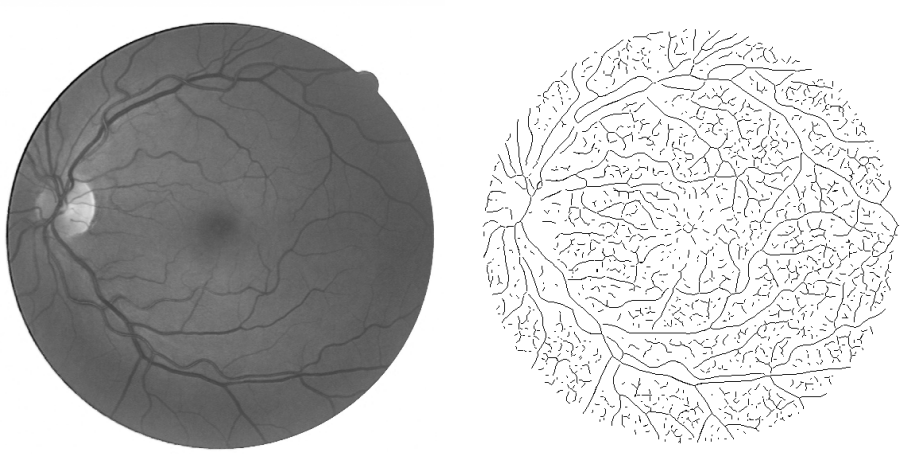
$$\rho(\boldsymbol{x},\sigma) = \begin{cases} -1 & \text{if } L(\boldsymbol{x},\sigma) < L(\boldsymbol{x}-\epsilon\boldsymbol{v},\sigma) \text{ and } L(\boldsymbol{x},\sigma) < L(\boldsymbol{x}+\epsilon\boldsymbol{v},\sigma) \\ +1 & \text{if } L(\boldsymbol{x},\sigma) > L(\boldsymbol{x}-\epsilon\boldsymbol{v},\sigma) \text{ and } L(\boldsymbol{x},\sigma) > L(\boldsymbol{x}+\epsilon\boldsymbol{v},\sigma) \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

where the parameter ϵ denotes the spatial accuracy with which the point sets are detected. In the continuous case the limit $\epsilon \to 0$ is taken, but in the discrete pixel case $\epsilon = 1.0$ pixel is a natural choice. In eq. (4.1) v is evaluated at $(\boldsymbol{x},\sigma)$. The locations at which $L(\boldsymbol{x} \pm \epsilon\boldsymbol{v},\sigma)$ is evaluated are in general not on a grid point and linear interpolation is used to obtain these values.

Figure 4.1 shows an example of ridge detection in a fundus image. Since the vessels are dark structures, only local minima ridges are shown. In fluorescein angiography, where the vessels are the brighter structures, the local maxima ridges should be used.

4.2.2 Primitive construction: affine convex sets

The next step is the formation of primitives from the ridge pixels. The aim is to obtain primitives which represent approximately straight line elements. For this purpose we use the algorithm to construct affine convex sets, as developed in section 3.2. In this chapter $\epsilon_{\mathrm{c}} = 3.0$ pixels, $\epsilon_{\mathrm{o}} = 0.95$ and $\epsilon_{\mathrm{p}} = 0.95$ is used. In



(a) (b)

Figure 4.1 (a) Green channel of a fundus image obtained from a digital fundus camera. The diameter of the FOV is 540 pixels. (b) The local minima ridges of (a), σ = 2.0 pixel. A subset of the ridges coincide with the vessels.

figure 4.2 the convex sets of the ridges of figure 4.1(b) are shown.

The kth element in convex set number i, consisting of K_i points, will be denoted by $\boldsymbol{c}_i(k), k = 1, \ldots, K_i$. The vector $\boldsymbol{c}(k)$ is the location $\boldsymbol{x}$ of the kth element in the set. The subscript i is omitted when no particular convex set is referred to. For every point in a convex set i there is a corresponding direction $\boldsymbol{v}_i(k)$, the direction in which the ridge is detected.

4.2.3 Convex set regions

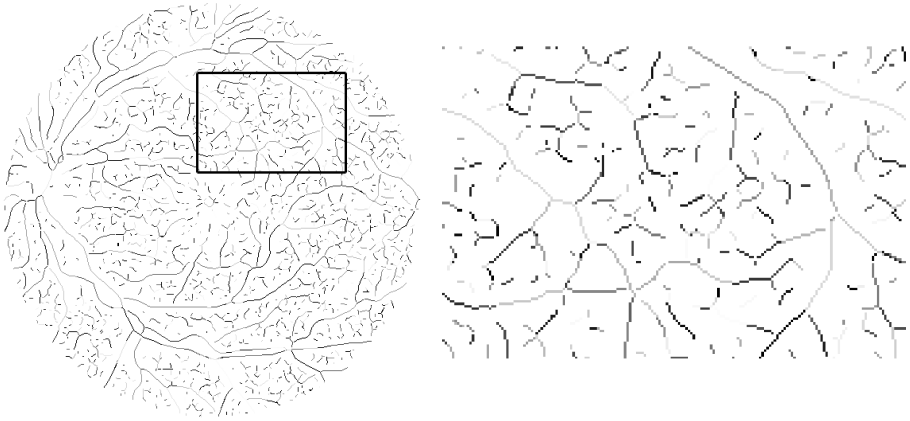
The image can now be partitioned into patches based on the convex sets. Every pixel is assigned to the convex set to which it is closest. Figure 4.3 shows the result of this operation on the convex sets of figure 4.2(a). The patches are referred to as convex set regions (CSR).

A fitted straight line through a convex set can be used as the main axis of a local coordinate frame for the pixels in the corresponding CSR.

4.3 Features and classifiers

The goal of this work is to classify every pixel in an image as vessel or non-vessel. For this purpose labeled examples or training sets, features and a classifier are needed. Here we present the material that is needed in this chapter, for a more detailed discussion see section 1.4.3.

From the training sets feature vectors are constructed that can be labeled as vessel or non-vessel, so every feature vector belongs to one of two classes. The



(a) (b)

Figure 4.2 (a) The convex sets of the ridges of figure 4.1(b). Every convex set has its own color. (b) Blow up of (a). Note that the number of ridge pixels in (a) is equal to the number of ridge pixels in figure 4.1.

idea is that feature vectors from a particular class cluster together in the feature space and that a classifier can be designed that determines a decision boundary between the different classes. After the training, a non-labeled feature vector can be classified by determining on which side of the decision boundary it is situated. With some classifiers it is possible to approximate the chance, given the features, that a pixel is vessel or not. This is called soft classification.

4.3.1 Classification and performance

In initial experiments three classifiers have been compared, a kNN-classifier, a linear discriminant classifier and a quadratic discriminant classifier [21]. Performance of the kNN-classifier was superior in all experiments, so this classifier has been selected. We use the optimized implementation for kNN-classifiers that has been made available by Arya et al. [5].

Using k neighbors of which n are labeled as vessel the *a posteriori* probability for being part of a vessel is approximated as

$$P(\text{vessel}) = \frac{n}{k} \quad . \tag{4.2}$$

This converges to the true probability in the limit of an infinite number of examples [21].

The performance of the system is measured with ROC-curves [81]. An ROC-curve plots the fraction of pixels that is falsely classified as vessel against the fraction that is correctly classified as vessel. The fractions are determined by setting a

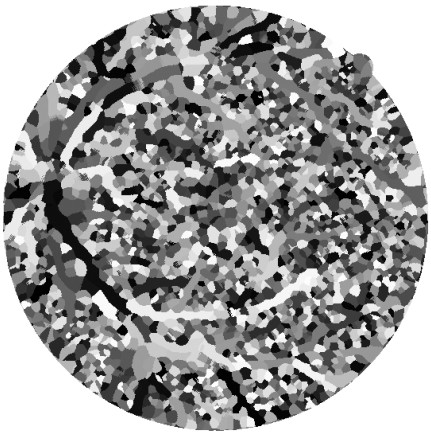


Figure 4.3 Convex set regions of the convex sets of figure 4.2(a).

threshold on the posterior probability. The closer a curve approaches the top left corner, the better the performance of the system. A single measure to quantify this behavior is the area under the curve, A_z, which is 1 for a perfect system. A system that makes random classifications has an ROC-curve that is a straight line through the origin with slope 1 and $A_z = 0.5$. For more details, see section 1.4.4.

4.3.2 Feature selection

Below, many features will be defined and it is not known beforehand which ones will give good classification results and which ones not. Therefore, feature selection is applied. The scheme used in this chapter is the Sequential Forward Selection method [127]. This algorithm starts with a null feature set and, for each step, the best feature that satisfies some criterion function is included with the current feature set. In this work, A_z is taken as the criterion function. After all features have been included, the set that gives the best performance is chosen.

4.3.3 Convex sets features

In figure 4.2 the convex sets that are part of a vessel can be distinguished from the non-vessels by their local appearance. For that reason, features based on the profiles perpendicular to the convex sets seem to be a good choice. To extract profile information, a profile is sampled for every point k in a convex set from the image's green plane in the direction of the vector $\boldsymbol{v}(k)$ with $\boldsymbol{c}(k)$ as midpoint. The obtained profiles are averaged and the mean profile $\psi(n)$, with $n = -N, \ldots, N$, is taken as the profile of the convex set. The averaging operation performs some smoothing, so blurring of the profile is not considered. Features that are extracted from the profile are:

1. The height of the profile: $h = \psi(0)$.
2. The width of the profile defined as the distance between the strongest right and left edge of the profile: $w = n_{\text{re}} - n_{\text{le}}$ ($n_{\text{re}} = \arg\max_{n>0} \psi'(n)$, with ψ' the first derivative of the profile. n_{le} is defined similarly for $n < 0$).
3. The height divided by the width: h/w.
4. The edge strength, defined as: $s_{\text{e}} = \psi'(n_{\text{le}}) + \psi'(n_{\text{re}})$.
5. The edge strength divided by the width: s_{e}/w.
6. The edge height: $h_{\text{e}} = \frac{1}{2}(\psi(n_{\text{le}}) + \psi(n_{\text{re}}))$.
7. The height minus the edge height: $h - h_{\text{e}}$.
8. The height divided by the edge height: h/h_{e}.

There are also features extracted that are not profile related, but might give useful information. The following are computed:

9. The distance between the first and last point of a convex set: $d = \|\boldsymbol{c}(1) - \boldsymbol{c}(K)\|$.
10. The length of a convex set: $l = \sum_{k=2}^{K} \|\boldsymbol{c}(k) - \boldsymbol{c}(k-1)\|$.
11. The curvature of a convex set, approximated by: $\kappa = \sum_{k=2}^{K} \boldsymbol{v}(k) \cdot \boldsymbol{v}(k-1)$.
12. A rectangular image patch of size $1.5 \times w$ by K is sampled in the green plane around a convex set. The mean μ_{g} for this patch is computed.
13. The standard deviation σ_{g} for the green patch.
14. The mean value of the green plane at the locations of the convex set divided by the the mean value of the red plane.
15. At different scales σ the mean value of $\lambda(\sigma)$ at the locations of the convex set: $\bar{\lambda}(\sigma) = \frac{1}{K} \sum_{k=1}^{K} \lambda(\boldsymbol{c}(k), \sigma)$, this is a measure of ridge strength (see section 4.2.1).

4.3.4 CSR pixel features

Fitting a straight line through a convex set, a local coordinate system can be established for every CSR. For the origin of the coordinate the center of mass of the convex set is chosen. The first axis is along the direction of the fitted line, the second axis is perpendicular to the first axis and its direction is chosen so that a right hand oriented coordinate system is established. The features computed for the pixels in the CSR can be subdivided into features that take no information of the convex set into account (number 1–3 in the list below), features that take convex set information into account (features 4–9 below) and features that exploit the use of the local coordinate system (features 10–12).

1. The value of the red plane of the image at the pixel location: $r(\boldsymbol{x})$.
2. The value of the green plane of the image at the pixel location: $g(\boldsymbol{x})$.

3. The ratio of the green values and red values of the pixel: $g(\boldsymbol{x})/r(\boldsymbol{x})$.
4. The chance that the corresponding convex set belongs to a vessel: $P(\boldsymbol{c} = \text{vessel})$.
5. The distance between the pixel and the closest point on the convex set: $d_{\text{closest}} = \|\boldsymbol{x} - \boldsymbol{c}_{\text{closest}}\|$.
6. The difference in the red values of the pixel and the closest point on the convex set: $r(\boldsymbol{x}) - r(\boldsymbol{c}_{\text{closest}})$.
7. The ratio of the red values of the pixel and the closest point on the convex set: $r(\boldsymbol{x})/r(\boldsymbol{c}_{\text{closest}})$.
8. The difference in the green values of the pixel and the closest point on the convex set: $g(\boldsymbol{x}) - g(\boldsymbol{c}_{\text{closest}})$.
9. The ratio of the green values of the pixel and the closest point on the convex set: $g(\boldsymbol{x})/g(\boldsymbol{c}_{\text{closest}})$.
10. The coordinate of the pixel with respect to the first axis: x'_1.
11. The coordinate of the pixel with respect to the second axis: x'_2.
12. $L_{x'_1}$, $L_{x'_2}$, $L_{x'_1x'_1}$ and $L_{x'_2x'_2}$ at different scales σ. (Note that these are the number of scales $\times$ 4 features).

Not all of the features defined in this and the previous subsection are independent, but this is not a concern for the kNN-classifier used here.

In the databases we use, the blue channel is often empty or contains a lot of noise. Therefore, no features are extracted from that channel.

4.4 Material

Two databases with images are used. The first one is obtained from a screening programme in the Netherlands and will be referred to as the DRIVE database[1]. From this database 40 images are taken, containing 7 images with pathology (exudates, hemors, pigment epithelium changes). The images are captured in digital form from a Canon CR5 non-mydriatic 3CCD camera at 45º field of view. The images are of size 768×584 pixels, 8 bits per color channel and have a field of view (FOV) of approximately 540 pixels in diameter. The images are in compressed JPEG-format, which is unfortunate for image processing but is commonly used in screening practice. Figure 4.1(a) shows an example of the green channel of such a fundus image.

The images were manually segmented by three observers, A. Scheenstra (a computer science student), N. Niemeijer (a fellow PhD-student) and the author. They were asked to mark all pixels for which they were for at least 70% certain that they were vessel. The observers were trained by an ophthalmologist, M. Abràmoff.

[1]Digital Retinal Images for Vessel Extraction. The database, consisting of the images and manual segmentations, is publicly available at `http://www.isi.uu.nl/Research/Databases/DRIVE/`.

The 40 images have been divided into a train and test set, each containing 20 images. The images in the train set were segmented by the first (14 images) and second observer (6 images). The images in the test set were segmented twice, resulting in a set A and a set B. The images in set A were labeled by the first (13 images) and second observer (7 images). Set B was segmented by the last observer.

The train set contains 3 images with pathology. Performance is computed with respect to the test set (the segmentations of set A are used as ground truth). The observers of set A marked 577,649 pixels as vessel and 3,960,494 as background (12.7% vessel), for set B these numbers are 556,532 and 3,981,611 respectively (12.3% vessel).

The second database has been collected by Hoover et al. [43] and consists of 19 images. This database will be referred to as the Hoover database. The Hoover images are digitized slides captured by a TopCon TRV-50 fundus camera at 35^o field of view. The slides were digitized to 700$\times$605 pixels, 8 bits per color channel. The FOV in the images are approximately 650$\times$550 pixels in diameter. Two observers manually segmented all images. The first observer segmented 615,726 pixels as vessel and 5,293,034 as background (10.4% vessel), the second observer marked 879,695 pixels as vessel and 5,029,065 as background (14.9% vessel). Nine images contain pathology. The first observer segmented far less (small) vessels than the second observer and there is a large variability between the observers. Performance is computed with the segmentations of the first observer as ground truth.

Hoover et al. used the complete image for measuring the performance. Since the dark background outside the FOV is easily extracted, in this article all experiments are done on the FOV only.

We compare the performance of our algorithm to that of Hoover et al. [43] using their publicly available results. To compare our system with the method of Jiang and Mojon [49] we implemented that method.

4.5 Results

All experiments for the PBM (primitive based method) are carried out with the following settings, which were found after a pilot study.

The ridges in the images are extracted from the green channel at scale $\sigma = 1.5$ pixel. To obtain approximately straight lines, the maximum size of the convex sets is set to 25 pixels.

Training sets for the convex sets are constructed by counting how many of their pixels intersect with the vessel pixels in the manually labeled ground truth images. If more than 50% of the pixels in a convex set intersect they are labeled as vessel, else as non-vessel.

The profile features for the convex sets are computed with a half profile width $N = 15$, i.e. the profile consists of 31 pixels. To compensate for the lighting variations and to enhance local contrast, the pixels of every color channel C_i of the

images are locally normalized to zero mean and unit variance

$$N_i(\boldsymbol{x},\sigma) = \frac{C_i(\boldsymbol{x}) - \mathcal{E}_\sigma\{C_i\}(\boldsymbol{x})}{\sqrt{\mathcal{E}_\sigma\{C_i^2\}(\boldsymbol{x}) - \mathcal{E}_\sigma^2\{C_i\}(\boldsymbol{x})}} \quad ,$$

with

$$\mathcal{E}_\sigma\{C\}(\boldsymbol{x}) = \frac{1}{2\pi\sigma^2}\int_{\boldsymbol{x}'\in\mathbb{R}^2} C(\boldsymbol{x}')\mathrm{e}^{-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/2\sigma^2}\mathrm{d}\boldsymbol{x}' \quad ,$$

acting as a local averaging operator. A value of σ = 8.0 pixels is used. In [114] a similar filtering is performed with a square filter.

4.5.1 Settings

The ridge measures are extracted from the green channel and scales σ = 0.5, 1.0, 2.0 and 4.0 pixels are used.

With these settings a total 18 features per convex set is extracted. For the training of the classifier, only every fourth convex set is taken. This reduces computation time and memory resources.

For the computation of the features for the CSR the following settings are used. In the training phase, the *a posteriori* probabilities for the convex sets are computed using eq. (4.2) in a leave-one-out fashion, i.e. the convex sets of one image are classified with a classifier trained on the convex sets of the other images in the training set.

The derivatives with respect to the local coordinate systems are taken at scales σ = 0.5, 1.0, 2.0 and 4.0, resulting in 27 features. For the training, only every fourth pixel in the x_1 and x_2-directions is used.

The feature selection is also done on a leave-one-out basis. This is done for every image and the A_z-value of all images is averaged to obtain a criterion upon which it is decided to include a feature or not.

The kNN-classifiers for the classification of the convex sets and the CSR use k = 101.

Because kNN-classifiers are sensitive to scaling between different features, in all experiments each feature is normalized independently to zero mean and unit variance (cf. section 1.4.3).

4.5.2 The DRIVE database

The convex sets detection on the train set gives 12,648 vessel segments and 48,838 non-vessel segments (20.6% vessel).

The results of feature selection on the train set are presented in table 4.1 for the convex sets and in table 4.2 for the CSR. Every row in the table gives the performance of the best feature that is added to the feature set (so including the previously selected features). In the last line the performance is shown when all features are included.

DRIVE database		Hoover database	
	A_z		A_z
$\lambda, \sigma = 2.0$	0.843	$\lambda, \sigma = 2.0$	0.934
h	0.864	d	0.945
κ	0.870	μ_g	0.948
σ_g	0.871	$\lambda, \sigma = 4.0$	0.947
$\lambda, \sigma = 1.0$	0.872	$\lambda, \sigma = 1.0$	0.948
μ_g	0.873	h_e	0.950
$\lambda, \sigma = 4.0$	0.874	all features	0.940
all features	0.864		

Table 4.1 Selected features for the convex sets. For every added feature the obtained area under the ROC-curve is given.

DRIVE database		Hoover database	
	A_z		A_z
$P(\boldsymbol{c} = \text{vessel})$	0.8755	$P(\boldsymbol{c} = \text{vessel})$	0.8955
$g(\boldsymbol{x}) - g(\boldsymbol{c}_{\text{closest}})$	0.9314	$g(\boldsymbol{x})/g(\boldsymbol{c}_{\text{closest}})$	0.9653
d_{closest}	0.9372	d_{closest}	0.9677
$L_{x'_1}, \sigma = 1.0$	0.9437	$L_{x'_1x'_1}, \sigma = 2.0$	0.9683
$L_{x'_2}, \sigma = 2.0$	0.9463	$L_{x'_2x'_2}, \sigma = 2.0$	0.9680
$L_{x'_2x'_2}, \sigma = 1.0$	0.9472	$r(\boldsymbol{x})/r(\boldsymbol{c}_{\text{closest}})$	0.9682
$L_{x'_1x'_1}, \sigma = 2.0$	0.9485	$L_{x'_1}, \sigma = 1.0$	0.9684
$r(\boldsymbol{x})/g(\boldsymbol{x})$	0.9490	all features	0.9589
$L_{x'_2x'_2}, \sigma = 2.0$	0.9497		
$L_{x'_2}, \sigma = 1.0$	0.9498		
$g(\boldsymbol{x})$	0.9500		
all features	0.9493		

Table 4.2 Selected features for the convex set regions. For every added feature the obtained area under the ROC-curve is given.

Figure 4.4 shows the ROC-curves for the segmentations obtained by the PBM and the method of Jiang and Mojon on the images of the test set. The performance of the observers of set A versus B is also plotted, together with the result of thresholding the classified images at $P = 0.5$. This is equivalent to a hard classification in two classes.

Figure 4.5 shows for the PBM the images with the highest and lowest A_z, the results of hard classification of these images and the corresponding manually segmented images for the observers of set A and set B.

In table 4.3 an overview is given of the results of the different methods. The upper part of the table shows the A_z-values. The A_z-value for the method of Jiang

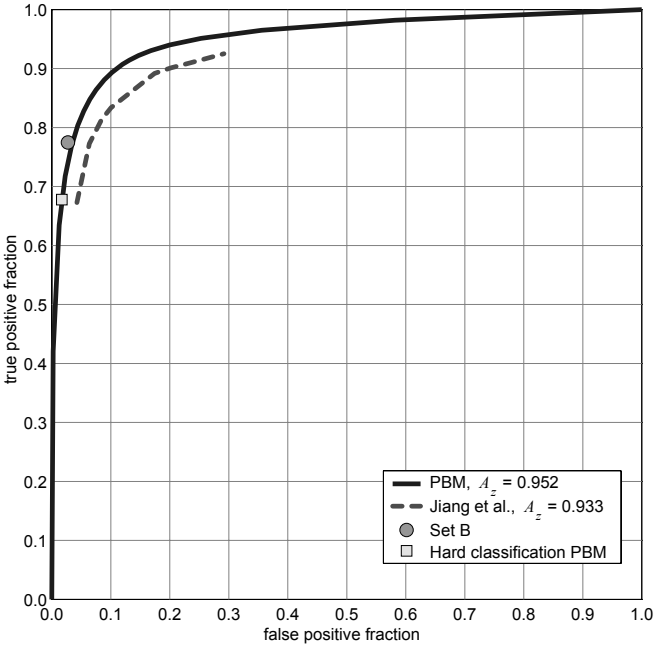


Figure 4.4 Results for the DRIVE database. The PBM gives $A_z = 0.952$ and the method of Jiang and Mojon [49] $A_z = 0.933$. Comparing set B to set A, false and true positive fractions of (0.0275, 0.775) are found. Performing a hard classification on the results of the PBM gives (0.017, 0.678) for the false and true positive fractions.

and Mojon is approximated by adding the points (0, 0) and (1, 1) to the curve. The lower part of the table lists the accuracies for the different methods.

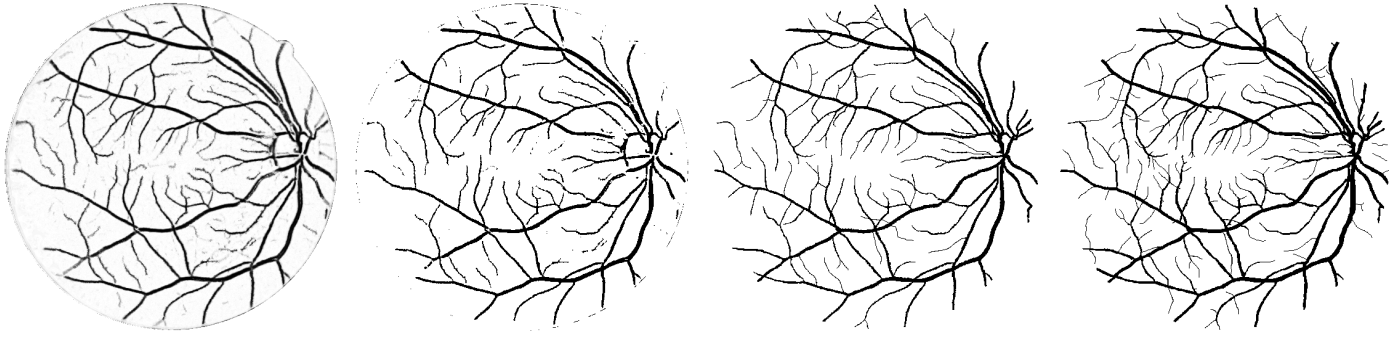
For comparison, the last row shows the results when all pixels are segmented as the most likely class, in this case the background.

Paired t-tests on the A_z-values for the individual images of the database, show that the PBM performs significantly better than Jiang and Mojon with $P < 0.01$. Using the accuracies of each image, all values in table 4.3 are significantly different from each other with $P < 0.01$, except for Jiang and Mojon versus the most likely class, where $P < 0.02$.

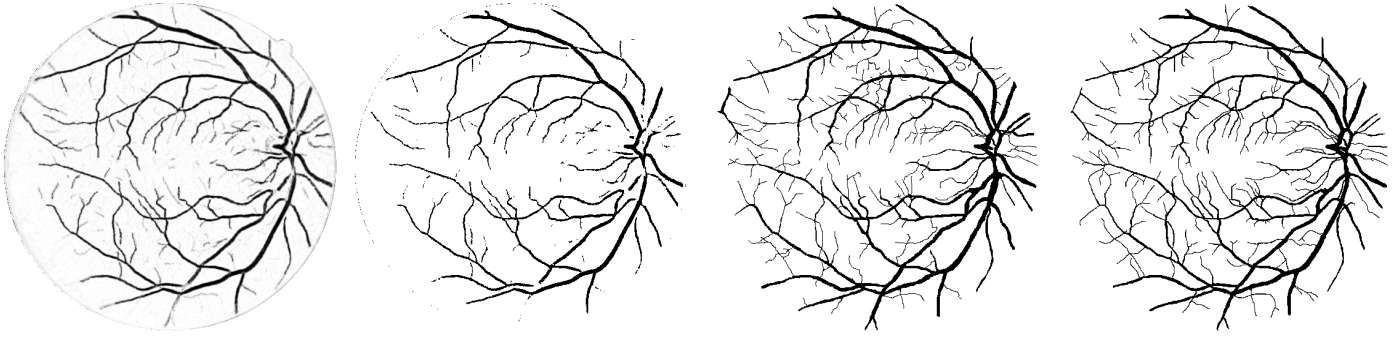
A t-test for the PBM shows no significant differences when the results of the images with pathology are compared to those without pathology.

4.5.3 The Hoover database

Because no independent test set is available for the Hoover database, leave-one-out experiments are performed, i.e. every image is classified using the other 18 images as training set. However, the computation time required for feature selection is very large and is therefore only done once for both the convex sets and the



(a) PBM, $A_z = 0.972$ (b) Hard classification, acc. = 0.962 (c) Observer set A (d) Observer set B



(e) PBM, $A_z = 0.937$ (f) Hard classification, acc. = 0.934 (g) Observer set A (h) Observer set B

Figure 4.5 First column: Best and worst result of the PBM. The grey value denotes the prior probability of the pixel being vessel (bright for the lower values and dark for the higher values). Second column: hard classification. Third column: observer from set A. Fourth column: observer from set B.

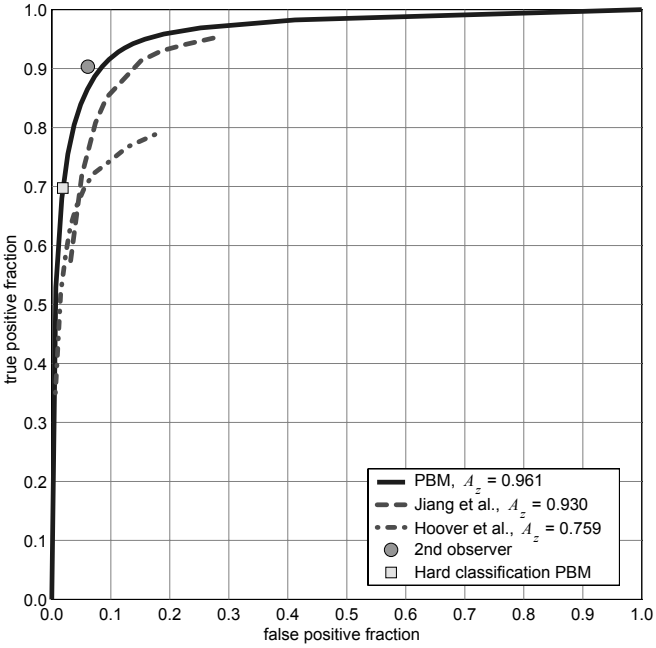


Figure 4.6 Results for the Hoover database. $A_z = 0.961$ for the PBM, $A_z = 0.930$ for the algorithm by Jiang and Mojon [49] and $A_z = 0.759$ for the algorithm by Hoover et al. [43]. Comparing the second observer to the first observer, false and true positive fractions of (0.061, 0.903) are found. Performing a hard classification on the results of the PBM gives (0.019, 0.697) for the false and true positive fractions.

CSR, using 9 of the 19 images (4 with and 5 without pathology).

Tables 4.1 and 4.2 list the features that have been selected. In the train set 8,615 convex sets are vessel, and 57,996 non-vessel (12.9% vessel).

After the feature selection, the images are segmented in leave-one-out experiments.

The ROC-curves for the PBM and for the methods of Jiang and Mojon and Hoover et al. are shown in figure 4.6. The results of observer 1 versus observer 2 and of hard classification are plotted too.

Table 4.3 shows the A_z-values and the accuracies for the different methods. Paired t-tests on the A_z-values show that the PBM performs significantly better than the two other methods and that the method of Jiang and Mojon is significantly better than the method of Hoover et al., all with $P < 0.01$. For the accuracies, the PBM is significantly better than the other methods (including the second observer) with $P < 0.01$. The method of Hoover et al. is significantly better than Jiang and Mojon and than classifying every pixel to its most likely class with $P < 0.01$, but differs only significantly with the second observer with $P < 0.1$. The method by Jiang and Mojon is not significantly better than classifying the

Criterion	Method	Database	
		DRIVE	Hoover
A_z	Hoover		0.7590
	Jiang	0.9327	0.9298
	PBM	0.9520	0.9614
Accuracy	2nd obs.	0.9473	0.9351
	Hoover		0.9275
	Jiang	0.8911	0.9009
	PBM	0.9441	0.9516
	Most likely class	0.8727	0.8958

Table 4.3 Results for the different databases and methods. Rows 1–3 give the area under the ROC-curves, rows 4–8 the accuracy (the sum of the number of correctly classified foreground and background pixels, divided by the total number of pixels).

pixels to the most likely class.

The PBM performs significantly better on the images without pathology than on those with pathology ($A_z = 0.969$ vs. $A_z = 0.950$).

4.6 Discussion and conclusions

The results show that the method proposed in this chapter (the PBM) outperforms the previously published rule-based methods by Hoover et al. [43] and by Jiang and Mojon [49]. A possible explanation is that it is very difficult to come up with good rules for problems where many features interact in a complex manner. Supervised methods with feature selection like the PBM seem to be better equipped for these tasks. In the Hoover database this is illustrated by the fact that the accuracy of the PBM is higher than that of an independent second observer. The PBM is able to adapt to the first observer, who disagrees especially on the number of small vessels with the second observer.

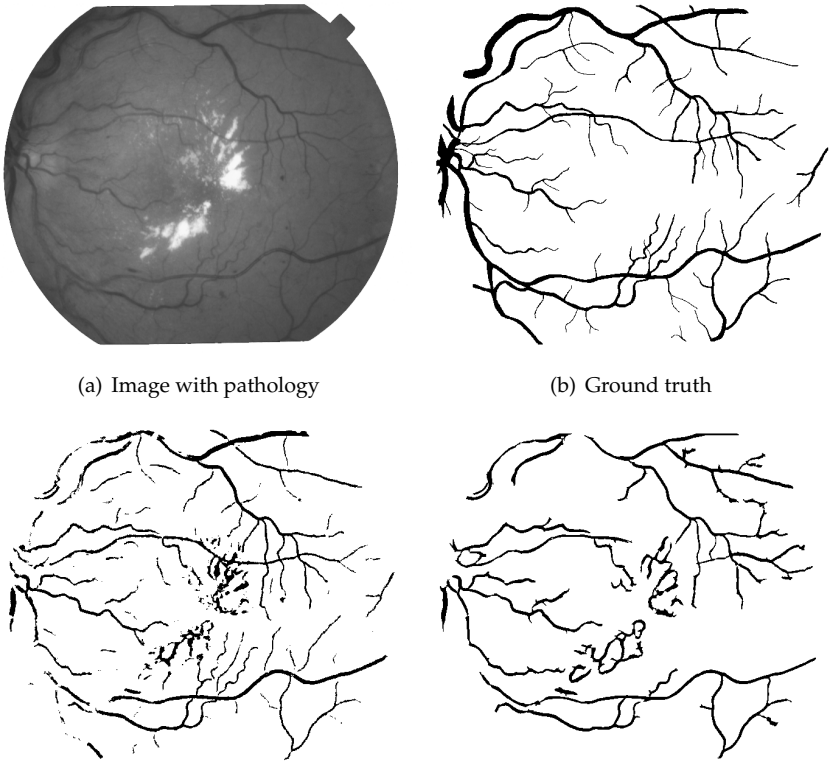
A disadvantage of supervised methods is the need for (manually) labeled training data. For the DRIVE database it took an observer 2 hours on average to label a single image.

Figures 4.4, 4.6 and table 4.3 demonstrate that it is possible to design a system that approaches the performance of human observers. It can be noticed from these figures that the second observer does not achieve an accuracy of 1, because the first and second observer disagree in their manual segmentations. One reason is that due to JPEG-artifacts it is hard to discern vessels of pixel or subpixel width. Another reason is the subjective decision that the human observers must make regarding pixels at the border of the vessels. Do they belong to the vessel or not? Asking the observers to rate the probability for such pixels being vessel would increase the time required for the segmentation prohibitively.

Two types of errors can be distinguished. The first type is over- and undersegmentation of the vessels. This is important in applications where determination of the vessel width is needed. The second type of error is the missing or erroneous detection of vessel branches. Suppose for example that two observers both label a small vessel, but in one of the segmentations the vessel is slightly shifted so that both vessels do not overlap. This will degrade accuracy more than if one of the observers had labeled no vessel at all. Type 2 errors are expected to be encountered with small vessels mostly. Removing the small vessels by a morphological opening, the type 1 error for large vessels can be approximated. The accuracy between first and second observer increased from 0.947 to 0.964 in the DRIVE database and from 0.935 to 0.948 in the Hoover database.

A perfect system would yield $A_z = 1$, which is not reached by our method. There are a few reasons. First, in all images high probability for being vessel is found around the boundaries of the FOV, while there are no vessels. See figure 4.5(a) and 4.5(e) for an example. This is an artifact of the method, introduced by the blurring that is needed for detecting the ridges of the image. Second, in some images part of the boundary of the optic disc is marked as vessel, cf. figure 4.5(a), while a human observer can clearly discern between the two. This is an indication that the amount of training data might be insufficient. Third, related to the last observation, is the appearance of (severe) pathology of which figure 4.7 gives an example. The method selects with high probability some vessels between bright areas where there are clearly no vessels. More training data that includes the various types of pathology encountered in screening practice might overcome this problem. No experiments were done to investigate the influence of the amount of training data. However, certain types of pathology might be detected and labeled in a preprocessing step. A fourth reason for imperfect performance of the method can be the incorrect labeling of the convex sets in the training sets. Or even worse, no ridge is detected at the location of a vessel, which can happen for very small vessels. This can occur because the locations of the ridges are perturbed by the blurring that is needed for their detection. For large vessels, the detected ridge pixels are still within the vessel, but for small vessels they can be a little off. A possible solution might be to do ridge detection with scale selection, like in [72]. Another interesting reason for scale selection is the use of scale as a feature.

For the PBM it takes about 15 minutes to segment an image in the DRIVE database on a Pentium-III PC, running at 1.0GHz with 1GB memory. Most time is spent in the classification. However, our implementation is experimental and could be optimized. For example, table 4.2 shows that for both databases, the most important feature is the probability that a convex set is part of a vessel. It can be expected that for a set with low probability of being vessel the pixels in its CSR will have low probability. This measure can be used to speed up the segmentation by not processing CSR belonging to convex sets with a low chance of being vessel. In the DRIVE database, we observed that only about 10% of the convex sets have a probability higher than 0.5. In [49], processing times of about half a minute are reported. Computation times for [43] are not available.



(a) Image with pathology (b) Ground truth

(c) Hard classification of the PBM (d) Result of Hoover et al.

Figure 4.7 Results on an image with pathology from the Hoover database.

In this chapter, evaluation has been done using accuracy of hard classifications and A_z-values of soft classifications. Other evaluation measures might be more appropriate, depending on the application at hand. For example, if one is interested in examining the tortuosity of the vessels, the width of the vessels might not be important, only the centerlines. The measures used do not take into account the number of branches, the connectedness of the vessels or the number of branching points, which all might be relevant in specific applications. Another good evaluation measure might be the performance of a computer-aided diagnosis system for retinal images that uses the results of a vessel segmentation algorithm in its analysis.

Chapter 5

A spin-glass model for primitive interaction

5.1 Introduction

As long as the field of digital image analysis exists, segmentation has been the bottleneck to achieve object extraction, object specific measurements, and fast object rendering from multi-dimensional image data. Simple segmentation techniques based on local pixel-neighborhood classification fail to apprehend the globality of objects and often require intensive operator assistance to produce acceptable results. The reason is that the notion of a object does not necessarily follow the characteristics of its local image representation; only in idealized cases local operations directly yield a definition of an object. Local properties, such as textures, edgeness, ridgeness etc. generally do not represent connected features of an object. Therefore, a method is needed that *groups* pieces of image primitives into objects. In such a method the solution of the *segmentation* problem will involve the use of domain knowledge that derives from the *recognition* task. Similar arguments have motivated earlier work on model-driven grouping and segmentation applied to real-world images [14, 37, 42, 52, 59, 76, 91, 92, 97, 103, 106, 112, 128].

In our view the segmentation problem can only be tackled successfully in conjunction *with* the recognition problem. The recognition task provides a notion of the objects to be defined using the segmentation method; this allows us to incorporate model knowledge of the objects in the grouping process, either by predefining properties that are characteristic of an object, or by deriving such properties by statistical means from example data.

The grouping process, described in this study, relies on *local* and *bilocal* prior object probabilities that have been based on the predefined recognition task. In the grouping process, image primitives interact with each other and through these interactions posterior probabilities for being part of the object are computed. In this sense, the method is based on Bayesian statistics.

The proposed method can be regarded as finding the mean state of a spin-glass system subject to the Gibbs-Boltzmann-distribution. The energy functionals that are needed for such a spin-glass system are based on the local and interac-

tion prior probability densities of the image primitives. If these densities are not known beforehand, they can be estimated from example data. In this chapter it is shown how this can be accomplished using classifiers from statistical pattern recognition [21].

The idea of grouping image primitives using a spin-glass model was investigated in [42] for the segmentation of edgels. The main differences between their approach and ours is firstly that we use a *local* part in the energy functional, that does not rely on the interaction probabilities. Secondly, we use example data to learn the energy functionals and do not define them in analytical form. Finally, instead of searching the configuration that maximizes the posterior probabilities, the mean values of all possible configurations of the image primitives determine the posterior probabilities. This is also an important difference of the proposed method with respect to Boltzmann-machine like approaches [1]. And unlike relaxation labeling methods, the method does not have to re-evaluate the probabilities of an objective function [62].

Other methods for grouping define affinity matrices between primitives and try to define a splitting of the primitives based on eigenanalysis [92, 126] or normalized cuts [112]. The problem with such methods is that their foundation is not statistical in nature, so that one has to incorporate user defined rules in constructing the affinity matrices, although in [106] an attempt has been made to overcome this shortcoming.

The proposed method is illustrated in two examples: grouping of line elements in synthetic and real-world data. In [76] a Markovian approach to this problem is given. Although the approach is interesting, the connection field and interaction matrices are manually constructed and it is not clear how they could be learned. Also, the use of local (non-interacting) knowledge is not incorporated in this scheme.

The purpose of the present study is to come up with a general statistical method that improves the *local* classification of image primitives by introducing (bilocal) interaction between them.

The setup of the chapter is as follows. In section 5.2 the grouping process is considered as a spin-glass system. The solution of the grouping problem is formulated as finding the means of the state variables. Section 5.3 gives an overview of implementational issues, followed in section 5.4 by illustrations of the approach, both on synthetic and on real-world images. Concluding remarks and a discussion of the results are presented in section 5.5.

5.2 Formal problem statement

In this section the grouping process will be regarded as a spin-glass system. The system is governed by an energy functional, consisting of a local and a bilocal part. In the next subsection, the spin-glass formulation will be derived. Section 5.2.2 discusses the definitions for the local and bilocal energies.

5.2.1 Probabilistic formulation

The task is to group K elements ξ_i out of a set $\Xi = \{\xi_1, \ldots, \xi_N\}$ of N elements. The number K is unknown beforehand. The elements can be image pixels, line elements, image patches, etc. Every element is regarded as having a spin s_i, that can be in one of two states: down or 0 and up or 1. If s_i is up, ξ_i belongs to the group (or object), if it is down, ξ_i belongs to the background. It is useful to introduce the probability of a certain grouping, i.e. a configuration $\{s_i\}$ of the spins, as

$$P(\{s_i\}) = \frac{1}{Z} \mathrm{e}^{-\beta E(\{s_i\})} \ . \tag{5.1}$$

Equation (5.1) is known as the Gibbs-Boltzmann-distribution. The constant Z is the partition function, which is the sum over all configurations of the spins of $\mathrm{e}^{-\beta E(\{s_i\})}$. It takes care that the sum over all configurations of $P(\{s_i\})$ equals 1. The functional $E(\{s_i\})$ plays the role of the energy belonging to a state $\{s_i\}$ of the system, whereas β is a control parameter, which is equivalent to the inverse temperature of a physical spin system.

We want to model the spins upto pairwise interaction in such a way that elements belonging both to the foreground lower the energy. The energy function that is used to accomplish this is

$$E(\{s_i\}) = \sum_i L_i s_i + \frac{1}{2} \sum_i \sum_j B_{ij} s_i s_j \ , \tag{5.2}$$

where L_i is the value of a local potential function induced by s_i and B_{ij} is the value of a bilocal potential function induced by the pair s_i and s_j. In [35, 42] similar energy functions are being used.

The bilocal part of the energy can be viewed upon as a discrete Hopfield network [44] with connections B_{ij} between the neurons. For the grouping process it is important that the elements influence each other, which is accomplished by Hopfield networks since they exhibit strong feedback-coupling.

Many Gibbs-based methods try to minimize the energy functional in order to obtain a maximum a posterior (MAP) estimate from eq. (5.1). A fast deterministic solution for energy functionals with binary variables and constant B_{ij} is given in [35]. A recently published paper [67] investigates what energy functions can be maximized using graph cuts. The constraints that the energy functionals must satisfy are not met in our case. For that reason, we estimate the mean state of the system governed by eq. (5.1). Following the terminology of [75], another loss function is adopted from a Bayesian theoretic point of view.

The mean $\langle s_i \rangle$ of a spin s_i is given by

$$\langle s_i \rangle = \sum_{\{s_j\}} s_i P(\{s_j\}) \ ,$$

where the sum runs over all configurations. Elements with mean spins close to one are very probable in the group, whereas those with values close to zero

belong to the background. Once the mean values of the spins are determined the grouped elements can be extracted by setting a threshold. For each element ξ_i, its mean spin plays the role of the *a posteriori* probability of being part of the object.

The computation of the values of the mean spins can efficiently be computed using the Metropolis algorithm [80], which will be discussed in section 5.3.1. But first, definitions for the potential functions will be given.

5.2.2 Definition of the potentials

For computation of the mean spins the potentials L_i and B_{ij} need to be known. We want to base the potentials on properties of the elements ξ_i. Therefore, it is assumed that *a priori* knowledge of every element ξ_i is available in the form of a local probability $P_i = P(s_i = 1)$. Information of the interaction between a pair of elements ξ_i and ξ_j should be available in the form of a bilocal probability $P_{ij} = P(s_i = 1|s_j = 1)$. A method for the determination of these *a priori* probabilities is given in section 5.3.2. Since every spin can only be in one of two states, we have that $P(s_i = 0) = 1 - P_i$, and likewise, $P(s_i = 0|s_j = 1) = 1 - P_{ij}$. Note that local probabilities have a single index, whereas bilocal probabilities are doubly indexed.

If there is no interaction between the elements, eq. (5.2) reduces to

$$E(\{s_i\}) = \sum_i L_i s_i \ .$$

With eq. (5.1) the probability that the system is in state $\{s_i\}$ is

$$P(\{s_i\}) = \frac{1}{Z} \prod_i \mathrm{e}^{-\beta L_i s_i} \ ,$$

showing that without interactions the classification of the elements is independent of each other. Because of the independence we can consider the whole system as a set of N systems each consisting of one spin. The Gibbs-Boltzmann-distribution for the system concerning s_i is then given by

$$P(s_i) = \frac{1}{Z_i} \mathrm{e}^{-\beta L_i s_i} \ ,$$

and we find for the *a posteriori* probability

$$\langle s_i \rangle = \frac{0 \cdot \mathrm{e}^{-\beta L_i \cdot 0} + 1 \cdot \mathrm{e}^{-\beta L_i \cdot 1}}{Z_i} = \frac{1}{Z_i} \mathrm{e}^{-\beta L_i} \ , \tag{5.3}$$

with $Z_i = 1 + \mathrm{e}^{-\beta L_i}$ the partition function for the system corresponding to s_i.

Without interaction between the elements, the energy should be defined in such a way that the *a posteriori* probability equals the *a priori* probability, i.e.

$$\langle s_i \rangle = P_i \ . \tag{5.4}$$

Substitution of eq. (5.4) in eq. (5.3) and solving for L_i yields

$$L_i = -\frac{1}{\beta} \log_e \frac{P_i}{1 - P_i} \ . \tag{5.5}$$

With the above equation we have expressed the local potential function in terms of the *a priori* local probabilities. In the absence of bilocal interaction the system is calibrated in such a way that it classifies the elements according to their *a priori* probabilities.

In analogy with the demand of eq. (5.4), we would like the system to be calibrated in such a way that if only ξ_i and ξ_j have interaction and if there is no contribution of the local potential, that the *a posteriori* conditional probability for $s_i = 1$ given $s_j = 1$ equals the *a priori* conditional probability $P_{ij} = P(s_i = 1|s_j = 1)$

$$\langle s_i | s_j = 1 \rangle = P_{ij} \ , \tag{5.6}$$

where $\langle s_i|s_j = 1\rangle$ is the *a posteriori* conditional probability. Under these conditions we find that

$$\langle s_i | s_j = 1 \rangle = \frac{1}{Z_{ij}} e^{-\beta B_{ij}} \ ,$$

with $Z_{ij} = 1 + e^{-\beta B_{ij}}$. Note that there are only two states, viz. $s_i = 0 \wedge s_j = 1$ and $s_i = 1 \wedge s_j = 1$. Solving for B_{ij} in the above equations gives

$$B_{ij} = -\frac{1}{\beta} \log_e \frac{P_{ij}}{1 - P_{ij}} \ . \tag{5.7}$$

In order to derive expressions for L_i and B_{ij} in terms of *a priori* knowledge their contributions to the energy functional have been investigated in isolation. Eq. (5.4) holds only when the bilocal potential is excluded and eq. (5.6) is only valid in a two-spin system without a local potential. When the potentials as defined in eqs. (5.5) and (5.7) are combined in eq. (5.2), the expressions in eqs. (5.4) and (5.6) are perturbed. In particular, the *a posteriori* probability in eq. (5.4) becomes

$$\langle s_i \rangle = P_i + \Delta_i \ , \tag{5.8}$$

where the sign and magnitude of Δ_i resemble the outcome of the competition between the local and bilocal contributions to the energy.

To get a feeling how the interactions between the spins influence the system, we consider the following cases. If $P_{ij} > \frac{1}{2}$, then $B_{ij} < 0$ and selection of both $s_i = 1$ and $s_j = 1$ is favored by the system since that will lower the energy. For $P_{ij} < \frac{1}{2}$ putting at least one of the spins to 0 will be preferred, since in that case the energy is not increased. An element that has $P_i < \frac{1}{2}$, and thus $L_i > 0$, would like to have $s_i = 0$. However, if one of ξ_i's neighbors, say ξ_j, is a foreground element and it has strong interaction with ξ_i, then, given that $s_j = 1$, $B_{ij} < 0$, which favors $s_i = 1$. Clearly, there will be competition between the local and bilocal contributions to the energy. If the interaction is strong enough, the neighbor will

cause the locally weak element to become part of the foreground object, i.e. it increases the mean spin of s_i. If ξ_i is a locally strong element, i.e. $P_i > \frac{1}{2}$ and it has weak interactions with its neighbors, then this will encourage the spins of the neighbors to be set to zero and the mean spin of ξ_i will not change with respect to the local *a priori* probability P_i.

Note that with the definitions for the potentials in eqs. (5.5) and (5.7) the parameter β drops out in eq. (5.1).

5.3 Implementation

In this section we will start with discussing the Metropolis algorithm, a method for finding the expected values of the state variables of a system that is characterized by the Gibbs-Boltzmann-distribution. After this discussion, determination of the *a priori* probabilities P_i and P_{ij} will be dealt with.

5.3.1 The Metropolis algorithm

The Metropolis algorithm [80] is a Monte-Carlo method for calculating the expected values of the state variables of a system that is subject to the the Gibbs-Boltzmann-distribution.

At the start of the algorithm the system is in a certain state, quite probably not the equilibrium state. The algorithm begins by choosing an element at random and reverses its spin. The reversal changes the energy of the system. If the change of energy $\Delta E < 0$ the reversal of the spin is accepted, if $\Delta E > 0$ the reversal is accepted with probability $\exp(-\beta \Delta E)$. The remaining $N - 1$ elements are checked in random order and the system changes its state with the same rules as before. This procedure is referred to as a Metropolis step. The Metropolis step is repeated M times, where M has to be large enough in order to represent the system's (thermal) equilibrium.

The mean of the spins is found by

$$\langle s_i \rangle = \frac{1}{M} \sum_{k=1}^{M} s_i(k) \ , \tag{5.9}$$

where $s_i(k)$ is the value of the spin after the k-th Metropolis step. In the limit $M \to \infty$, eq. (5.9) converges to the true expected value.

With the choices for the potentials in this chapter, knowledge of the energy itself is not necessary, only $\Delta E(s_i \mapsto \bar{s}_i) = (2\bar{s}_i - 1)(L_i + \sum_j B_{ij} s_j)$ is needed, where $\Delta E(s_i \mapsto \bar{s}_i)$ is the change of the energy due to the spin reversal of element ξ_i and $\bar{s}_i = 1 - s_i$.

5.3.2 Determination of the a priori probabilities

The main issue in the implementation of the proposed method is the determination of the *a priori* probabilities P_i and P_{ij}. Once those are known, the potentials

from eq. (5.5) and eq. (5.7) can be computed. For the determination P_i and P_{ij} two approaches are possible

1. Prior information suggests an analytical functional based on properties of the elements.
2. The probability densities are estimated from example data based on properties of the elements.

In this chapter the second option is taken, and the probabilities P_i and P_{ij} have been learned from manually labeled example data. In the example data every primitive ξ_i is given a label "true" (1) or "false" (0) which serves as a reference. To estimate the local density, a feature vector $\boldsymbol{\phi}_i$ is introduced for every element ξ_i in the example data, so that P_i can be estimated as function of the features. The set of $\boldsymbol{\phi}_i$'s combined with the reference labeling is the local training set.

For the bilocal probabilities a training set is built following a similar rationale. Recall that P_{ij} is the conditional probability that $s_i = 1$ given that $s_j = 1$. This means that we train with those ξ_i for which the neighbor element ξ_j in the local reference set is labeled as "true". The target in the bilocal reference set must be set to 1 if ξ_i is labeled as "true" in the local reference set and to 0 if it is labeled as "false". For every pair ξ_i and ξ_j which appears in the bilocal reference set, a vector $\boldsymbol{\psi}_{ij}$ is computed which stores the interaction features. This enables the estimation of P_{ij} as function of the features. These vectors together with the bilocal reference set form the bilocal training set.

The training sets are used to train a local and a bilocal classifier to estimate the probability densities for P_i and P_{ij}. An example of a classifier that is capable to accomplish this task is a feed-forward neural network [9, chapter 6]. Because training feed-forward neural networks can be difficult and many parameters need to be adjusted, we have chosen to use the k-Nearest-Neighbor (kNN) classifier for approximating the probability densities. There exist optimized and fast implementations for kNN-classifiers, see [5]. The training of a kNN-classifier is extremely simple: all feature vectors with their corresponding labeling are stored. The probability P that a feature vector with unknown label (a query point) has a label equal to 1 is estimated by inspecting the k closest neighbors of this vector in feature space. Suppose that n of those neighbors have a label equal to 1, then [21]

$$P = \frac{n}{k} \ . \tag{5.10}$$

For determining which feature vectors are closest to the query point the Euclidean distance is used in this work. Because kNN-classifiers are sensitive for scaling between the different features, each feature is normalized independently to zero mean and unit variance. The parameters for this linear transformation are obtained from the training data.

To summarize, by constructing feature sets for the local and bilocal reference sets, a local and a bilocal kNN-classifier can be trained. These classifiers enable the

estimation of P_i and P_{ij} for unlabeled feature sets using eq. (5.10). The classifiers can be regarded as "look-up" tables for the local and bilocal probabilities.

To reduce the number of bilocal probabilities that have to be learned, and to avoid long-range interaction, a neighborhood or "clique" can be used, in which element ξ_i only interacts with a limited number of other elements. Such a neighborhood also reduces the number of computations in the Metropolis algorithm, since less neighbors have to be taken into account.

5.4 Examples

In this section we will illustrate the proposed method in two examples. The first example deals with synthetic data and shows grouping of line elements into a cord. In the second example real-world data is used in the detection of the vasculature in retinal fundus images.

5.4.1 Grouping line elements into a cord

As a first example we experiment with cords existing of line elements. Ten training images of size 400×400 pixels are generated, which contain $2,020$ line elements of which 20 form a cord (an example is shown in figure 5.1). All line elements have a length of 10.0 ± 1.0 pixels (all distributions to generate the training and test data in this section are uniform). The orientation of the line elements that form the background vary between 0^o and 360^o. The cords have a random orientation θ and the orientation of their constituting line elements varies between $\theta - 1.8^o$ and $\theta + 1.8^o$.

Only one local feature is taken into account, the mean μ_i of the gray values of the line elements, which is 5.8 ± 4.2 for background elements and 10.0 ± 0.2 for foreground elements. These values cause some overlap between the distributions of the local features. With these settings a fair amount of foreground elements will be selected as background in local classification.

For the bilocal *a priori* probabilities five bilocal features are computed, three based on the geometry of the line elements, and two on the local features. The latter two are the sum and the absolute difference of the μ's of every considered pair. The geometrical features are a measure for distance (distance between the closest end-points), a measure for mutual orientation (inner product between the unit vectors aligned with the line elements) and an alignment measure, see figure 5.2. Note that a parallel displacement of one line element with respect to another, does not change their mutual orientation.

To decrease computational costs and to avoid long range interaction, only the 10 closest neighbors are taken into account.

The method is tested with ten test images that are constructed in the same way as the training data. The kNN-classifiers needed to approximate P_i and P_{ij} are both used with $k = 11$. After classification a line element is classified as foreground if the mean of its spin is larger than 0.5 and to background otherwise.

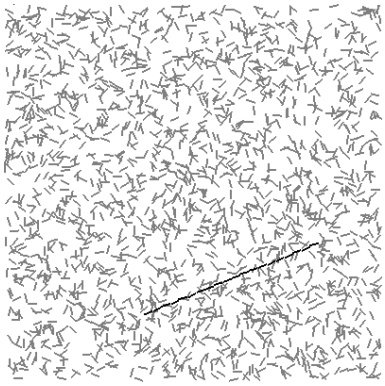


Figure 5.1 Example of a training image. In black the elements that constitute the cord, in gray the background elements.

A local classification for one of the images, shown in figure 5.3(a), is presented in figure 5.3(b). Notice that about 50% of the cord is classified as background, as is to be expected from the distributions for the means.

Figure 5.3(c) shows the results after bilocal classification. The Metropolis algorithm is run $1,000$ times. All but one of the elements of the cord are classified as foreground.

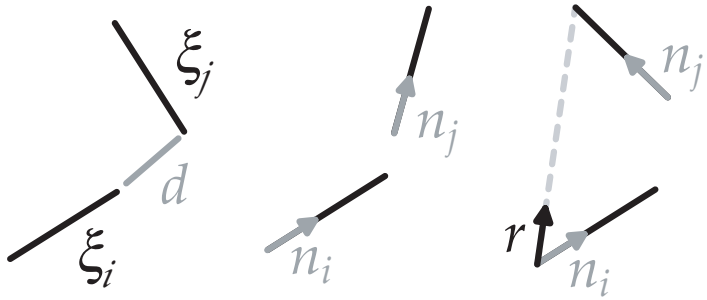
To evaluate the result of the grouping, several measures have been computed: the number of true positives TP (elements correctly classified as foreground), the number of true negatives TN (elements correctly classified as background), the number of false positives FP (elements incorrectly classified as foreground) and the number of false negatives (elements incorrectly classified as background). Their values are listed in table 5.1. The table shows clearly that the classification result after grouping increases: instead of an error of 55.5% in foreground classification, an error of 12.5% is obtained.

Finally, we investigated how much the means of the spins changed on average after the bilocal classification, cf. eq. (5.8). The changes for the elements classified as foreground show a increase of 0.245 on average for the bilocal *a posteriori* probabilities with respect to the local *a priori* probabilities. For the background elements no changes are found.

5.4.2 Segmenting ridges in retinal fundus images

In this subsection the method is tested on two-dimensional medical images of the retina of the human eye, for an example see figure 5.4(a). These images, also known as fundus images, are acquired by making photographs of the back of the eye. The image processing task is to delineate the vessel structure.

Since image ridges are natural indicators of vessels, we start our analysis with ridge detection. In chapter 2 a detailed discussion is presented on this subject. The ridges of figure 5.4(a) are shown in figure 5.4(b).



(a) (b) (c)

Figure 5.2 (a) The shortest distance between the end-points of two line elements is taken as the distance d between the elements. Note that there are four distances between the end-points of two elements. (b) The angle between two line elements is characterized by the absolute value of the inner product of the unit vectors $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, that are aligned with the line elements. (c) A (symmetric) measure for alignment is found by looking for the end-points which are closest to each other and forming a vector $\boldsymbol{r}$ of unit length along the line between the two other end-points. Note that those end-points are not necessarily the end-points with the longest distance between the two line elements. The alignment measure is now defined as the mean of the absolute values of the inner product of $\boldsymbol{r}$ with $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$: $\frac{1}{2}(|\boldsymbol{r} \cdot \boldsymbol{n}_i| + |\boldsymbol{r} \cdot \boldsymbol{n}_j|)$.

The problem of detecting the vessels in figure 5.4(a) is thus reduced to detecting which ridge pixels in figure 5.4(b) delineate vessels. It is obvious from the abundance of ridges in figure 5.4(b) that this representation is still suboptimal.

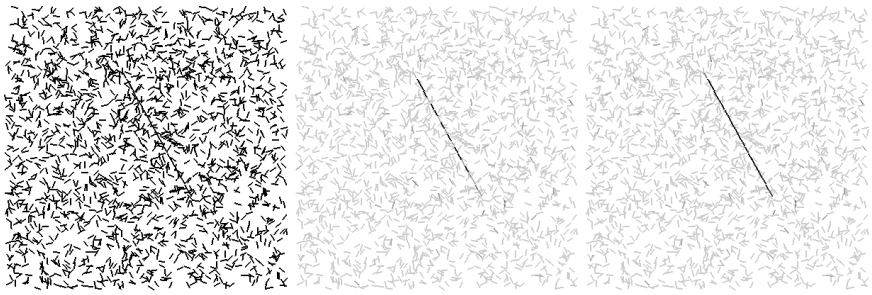
To improve the representation, the ridge point sets are fragmented into convex subsets. Each of these convex subsets represents a line segment. The so obtained set of line segments is the basic "grouping set" of geometrical image primitives. For the construction of the convex sets we refer to chapter 3.

The convex sets of the ridges in figure 5.4(b) are displayed in figure 5.6(a). These convex sets have been used for local and bilocal classification. For this purpose, 30 fundus images have been taken for which the convex sets were computed. The ridges are detected at a scale $\sigma = 0.8$ pixel. For the convex sets the following settings are used $\epsilon_{\mathrm{c}} = 3.0$ pixel, $\epsilon_{\mathrm{o}} = 0.98$ and $\epsilon_{\mathrm{p}} = 0.98$. This resulted in $106,206$ sets, of which, after manual labeling, $28,501$ turned out to be marked as vessel.

The 30 images are divided in a training set of 15 images and a test set of the remaining 15 images. To approximate P_i and P_{ij}, local and bilocal features are computed and kNN-classifiers using the corresponding training sets are built. To avoid long range interactions and to reduce computational costs, for P_{ij} only the 10 closest neighbors are taken into account.

The following local features are computed for every convex set i:

1. The mean μ_i of the image gray values at the M_i pixel locations of the convex



(a) (b) (c)

Figure 5.3 (a) Input image. (b) Locally classified image. The gray value of the elements measure the probability on spin up (darker denotes higher probability, lighter denotes lower probability). (c) As in (b), but now with bilocal interaction added. Note that the spins of the elements in the cord have become stronger.

	Local	Bilocal
TP	91	175
TN	20,000	20,000
FP	0	0
FN	109	25
av. Δ_i foreground		0.245
av. Δ_i background		0.000

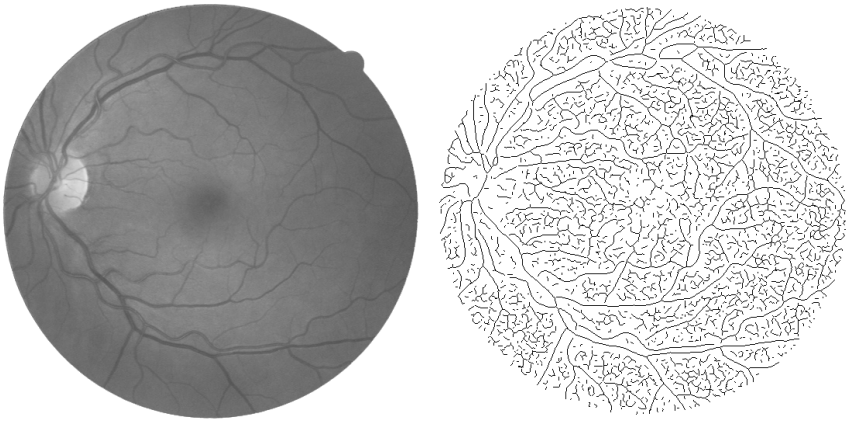
Table 5.1 Results from the experiments of section 5.4.1. The total number of background elements considered is 20,000 and the total number of foreground elements 200. The first row shows the true positives. The second row shows the true negatives. The false positives and false negatives are given in row three and four respectively. The fifth row shows how much the means of the spins of the foreground elements increase on average because of the grouping (see eq. (5.8)). The last row shows the same for the background elements.

set

$$\mu_i = \frac{1}{M_i} \sum_m L(x_{m,i}, y_{m,i}) \ ,$$

where L denotes the gray value image and $(x_{m,i}, y_{m,i})$ the pixel locations of the i-th convex set.

2. A measure for the width of vessels is computed in the following way. For every pixel $(x_{m,i}, y_{m,i})$ in the convex set the principal direction $\boldsymbol{v}_{m,i}$ is known (see appendix and discussion above). The principal directions are perpendicular to the ridges, i.e. perpendicular to the vessels. One-dimensional gray value profiles centered at $(x_{m,i}, y_{m,i})$ and in the direction of $\boldsymbol{v}_{m,i}$ are extracted from the image. In every profile, the edges on the left and right hand side of $(x_{m,i}, y_{m,i})$ are detected. The distance between the locations in the profile with strongest edge response on the left and right side is taken



(a) (b)

Figure 5.4 (a) An image of the fundus of the human retina. The field of view is approximately 540 pixels. (b) The ridges (black) obtained at scale $\sigma = 0.8$ pixel. Note the large response of the ridge detector with respect to the noise in the background.

as the width $\delta_{m,i}$ for profile m. The measure w_i for the width is the mean of the widths of all profiles

$$w_i = \frac{1}{M_i} \sum_m \delta_{m,i} \ .$$

3. A measure σ_i for the edge strength in the convex set is computed as follows. The response of the strongest edges on the left and right side of the profiles (see previous item), $\lambda_{m,i}$ and $\rho_{m,i}$ respectively, are averaged, yielding

$$\sigma_i = \frac{1}{M_i} \sum_m \lambda_{m,i} + \rho_{m,i} \ .$$

4. The curvature κ_i of the convex set, defined as

$$\kappa_i = \frac{1}{M_i - 1} \sum_{m=1}^{M_i - 1} \boldsymbol{v}_{m,i} \cdot \boldsymbol{v}_{m-1,i} \ ,$$

where $\boldsymbol{v}_{m,i}$ is the principal direction corresponding to pixel m of the convex set.

And for the bilocal features between convex sets i and j the following measures are taken:

1. The Euclidean distance between the closest endpoints of the sets (see also figure 5.2).

2. The sum of μ_i and μ_j (see item 1 of the local features).
3. The absolute difference of μ_i and μ_j (see item 1 of the local features).
4. The sum of σ_i and σ_j (see item 3 of the local features).
5. The absolute difference of σ_i and σ_j (see item 3 of the local features).
6. The sum of κ_i and κ_j (see item 4 of the local features).
7. The absolute difference of κ_i and κ_j (see item 4 of the local features).
8. The mutual orientation (see also figure 5.2).
9. The mutual alignment (see also figure 5.2).

The method is evaluated using the test set. All 15 data sets are classified, both locally and bilocally. The local and bilocal kNN-classifiers are used with $k = 21$ and 2,500 Metropolis steps are taken.

The performance of the system is measured with ROC-curves, see section 1.4.4. For the test set $A_z = 0.851$ is found for local classification and $A_z = 0.881$ for bilocal classification. The curves are plotted in figure 5.5(a).

Another measure for evaluation is the accuracy of the system

$$\text{accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FP} + \text{FN}} \ ,$$

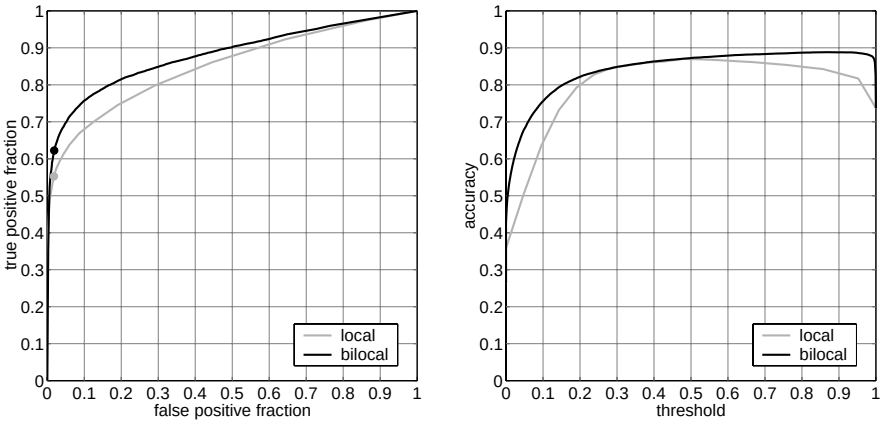
which is also dependent on the value of the threshold value. The threshold for optimal accuracy can be estimated from the training set with leave-one-out experiments[1]. Every image in the training set is classified using the 14 other images in the training set. For various threshold values the accuracy is computed and the threshold at which maximum accuracy is found is used for computing the accuracy of the test set. In figure 5.5(b) the accuracy of the training set as function of the threshold is given. Maximum accuracy for the local classification is found if $\langle s_i \rangle$ is thresholded at 0.5. For the bilocal classification this value is 0.95. The accuracies of the test set at these threshold values are 0.870 for local classification and 0.886 for bilocal classification. The points of maximum accuracy on the ROC-curves are plotted in figure 5.5(b). In table 5.2 an overview of all computed measures is given.

An example of the classification results is shown in figures 5.6(c) and 5.6(d).

From table 5.2 it can be concluded that the foreground classification has benefited from the grouping method. The total number of correctly classified vessel sets has increased. It also causes an increase in the number of correctly classified background elements, whereas the number of wrongly classified elements is reduced.

Not only the number of true positives increased, their *a posteriori* probabilities increased on average by 0.138, which is the purpose of the method. The *a posteriori* probabilities decreased on average with 0.016.

[1] With respect to the discussion on Bayes' formula and the decision rule, in chapter 1, we have changed the cost function that is presented in eq. (1.7).



(a) (b)

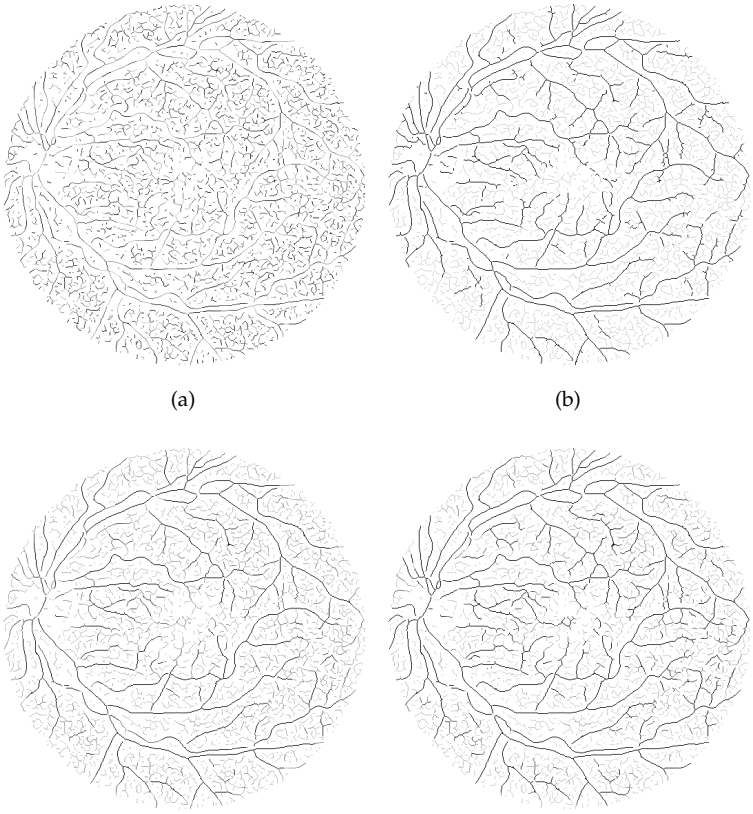
Figure 5.5 (a) ROC curves for the local and bilocal classification of retinal fundus images. The area under the curve is 0.851 for the local curve and 0.881 for the bilocal curve. The dots on the ROC-curve are at the location of maximum accuracy. (b) Accuracy of the training set as function of the threshold value. Maximum accuracy is found at $\langle s_i \rangle = 0.5$ for local classification and at 0.95 for bilocal classification.

In figure 5.7(a) the distribution of Δ_i, see eq. (5.8), is shown in a histogram for the foreground elements (width of the bins is 0.1). Figure 5.7(b) shows the same for the background elements. The distributions are centered around zero, but skewed to the right for the foreground elements and to the left for the background elements. As figure 5.7(a) shows, 46.5% of the foreground elements have a change of the spins between -0.05 and 0.05. The area of the bins above $\Delta_i = 0$ shows that 42.1% of the elements had an increase of their mean spin value, while the area of the bins below zero show that 11.4% had a decrease of the mean spin value. For the background elements, 41.5% of the mean values changed only between -0.05 and 0.05. An increase of the mean values was found for 17.0%, vs. a decrease for 41.5% of the elements.

5.5 Discussion

In this chapter a method is presented for grouping image primitives based on local and bilocal features. The method performs well on synthetic data. Compared to local classification the number of classification errors is reduced and the confidence with which the elements are classified is increased.

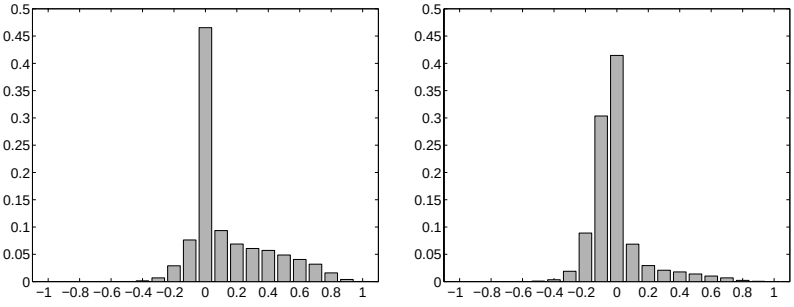
In the retinal fundus images, ROC-analysis shows that bilocal classification is better than local classification. The area under the curve increases from 0.851 to 0.881. For a threshold of 0.5 on $\langle s_i \rangle$ in the local case and of 0.95 in the bilocal case,



(a) (b)

(c) (d)

Figure 5.6 (a) The convex sets of the ridges of figure 5.4(b). Every grouped set has its own color. Note that sets which consist of 1 pixel have been removed. (b) Manually labeled convex sets for the vessels. (c) Locally classified convex sets. Sets with higher mean spin are shown darker. (d) Bilocally classified convex sets. Again darker elements mean higher mean spin.



(a) (b)

Figure 5.7 Distribution of Δ_i for (a) the foreground and (b) the background elements.

	Local	Bilocal
A_z	0.851	0.881
TP	8117	8783
TN	38760	38980
FP	949	729
FN	6062	5396
Accuracy	0.870	0.886
Sensitivity	0.572	0.619
Specificity	0.976	0.982
av. Δ_i foreground		0.138
av. Δ_i background		-0.016

Table 5.2 Results from the experiments of section 5.4.2. The thresholds used are 0.5 in the local case and 0.95 in the bilocal case. The first row shows A_z. The second row shows the number of true positives, the third row the true negatives, the fourth row the false positives and the fifth row the false negatives. The accuracy, sensitivity and specificity are displayed in rows six to eight. The ninth row shows how much the means of the spins of the foreground elements increase on average because of the grouping (see eq. (5.8)). The last row shows the same for the background elements.

the average increase of the posterior probabilities for correctly classified convex sets is 0.138. The number of true positives and negatives increases, whereas the number of false positives and negatives decreases as well.

It must be noted that the test on the fundus images is meant to serve as an illustration. For a genuine evaluation of the approach on real world images, the characteristic features of the image at hand must be determined by performing feature selection on a larger variety of features.

The method itself can be applied to a variety of grouping and classification problems. In this study, we consider grouping of line elements and convex sets, but grouping of individual pixels, pixel sets or other structures can be studied as well. Of course, in those cases other type of features will be needed, but the basis of the algorithm remains the same. Extension to higher dimensional images is straightforward. The complexity will remain the same, $O(N^2)$ for fully connected bilocal interactions, with N the number of elements. It is also possible to include higher order interactions (trinary, n-nary), although for this extension the complexity will increase as $O(N^n)$.

As an alternative for the definition of B_{ij} based on the conditional probabilities, the joint probability can be used, which we will denote by Q_{ij}. In that case, by demanding that $\langle s_i s_j \rangle = Q_{ij}$, the following formula similar to eq. (5.7) can be derived

$$B_{ij} = -\frac{1}{\beta} \log_{\mathrm{e}} \frac{3Q_{ij}}{1 - Q_{ij}} \ .$$

The factor 3 in the nominator appears because now the partition function has to

take 4 possible configurations into account: $s_i = 0 \wedge s_j = 0$, $s_i = 1 \wedge s_j = 0$, $s_i = 0 \wedge s_j = 1$ and $s_i = 1 \wedge s_j = 1$.

For the estimation of Q_{ij} with a classifier, the training set has to be built by setting the label to "true" if both elements belong the foreground and to "false" otherwise. Note that the training set for the joint probabilities will consist of more feature vectors than the training set for the conditional probabilities. This can increase the time for training and classification.

Experiments we did with the joint probabilities show similar behavior as with the conditional probabilities. The performance is better than with local classification only. However, in our examples, the conditional probabilities give better results than the joint probabilities. The time to classify one image increased from about 20 seconds to about 30 seconds (unoptimized code).

With the definitions we have given for B_{ij}, only pairs of spins which are both in the "on" state contribute to the energy in eq. (5.2). This is in accordance with the goal that only grouped elements should contribute to the energy. Extra terms like $C_{ij}s_i(1 - s_j)$ could be added to discriminate between foreground and background, or $D_{ij}(1 - s_i)(1 - s_j)$ for grouping background elements. Here, C_{ij} and D_{ij} are the bilocal potentials for these cases respectively. It depends on the application whether or not such terms make sense.

Finally, it is possible to obtain a MAP-estimation by using simulated annealing methods [1, 34, 61]. In that case, the factor β^{-1} should be removed from eqs. (5.5) and (5.7), so that the Metropolis algorithm becomes dependent on β. The Metropolis algorithm is started with a value of β close to zero. After the system has come to equilibrium the value of β is increased and the Metropolis algorithm is executed again. This scheme is repeated until β is so large that the system is forced into a "frozen" configuration ($\beta \to \infty$). However, the rate in which β is allowed to increase makes this algorithm very slow. Increasing β faster than allowed can yield unstable and non-unique results, because the energy may have multiple minima.

Chapter 6

Detection and segmentation of elongated structures in 3D images

6.1 Introduction

THE automatic detection, recognition and segmentation of objects in three-dimensional medical images is considered. This is an extremely challenging task, for several reasons: objects often have no clear boundaries; there may be many similar objects in close vicinity; the constellation of objects and their neighborhoods may vary widely from individual to individual; some objects or parts of objects may be missing, and scans from clinical practice often contain pathology.

To overcome these challenges, prior knowledge should be included in the analysis. This can be done by learning characteristics about the objects from examples. Such supervised image analysis methods are becoming increasingly popular in the medical image analysis community. One can distinguish local and global approaches. Local approaches, for example voxel classification, can be thought of as bottom-up processes. Due to their nature, they usually do not capture the overall structure and shape of the objects of interest. Global approaches, on the other hand, are usually top-down. They impose a model of a complete object that is fitted to the data. Examples are active appearance models [17] or the m-reps framework [99]. However, the constraints set by global models may impede the incorporation of local knowledge to adapt to local structure. Furthermore, global object models often require initialization close to their actual position, which requires an initial detection and recognition step. They may also have difficulties in dealing with (partly) missing data; local abnormalities in the data (e.g. pathology or artifacts) may lead to a global failure of the segmentation method.

In many cases, it can be advantageous to detect a constellation of objects. In this way, anatomical knowledge about the expected location of structures relative to other structures can be exploited. Human observers use such strategies, sometimes perhaps without being aware of it. In this process, local and global analysis

are intermingled.

In this work a general framework for detection, recognition and segmentation is proposed that we believe to be closer to human image interpretation capabilities than purely local or global algorithms. It starts with a local description and works up to a more global description at every next stage. The frameworks consists of five stages: voxel detection, primitive construction, primitive labeling, primitive grouping and recognition, and finally, segmentation.

Such a framework could be applied to a wide range of problems. In this chapter we focus on the detection, labeling and segmentation of the complete rib cage in chest CT-scans. The latest generation of CT-scanners produces data with slice thickness and slice gaps of size similar to the in plane resolution. As a consequence, radiologists are at risk of being overwhelmed by the amount of data they have to examine. To overcome this problem computerized analysis methods may be of great practical interest. In [119] numerous examples are given, of which automated rib segmentation is one. The ribs are always depicted in chest CT, so they should be reported on. In practice, rib anomalies and fractures are frequently missed [101]. Hence our interest in a completely automatic extraction of the rib cage. This can be used for effective visualizations of the rib cage and for computerized detection of bone abnormalities. The segmented ribs can also act as reference objects to segment other structures.

Not much attention has been paid to rib cage segmentation. Some methods have been implemented that can be used to segment the rib cage, or, more generally, segment elongated structures in CT-data. None of these methods is completely automatic. In [6, 7] centerlines are extracted from which the widths of the objects are estimated. In [60] a tracking algorithm is described, that proceeds from one 2D slice to the next. The method uses seeded region growing, for which the seeds must be supplied manually. The work in [56] is based on 3D region growing using locally adaptive thresholds.

The remainder of this chapter is organized as follows. Section 6.2 outlines each step in the general framework for detection, recognition and segmentation. In section 6.3 the application to rib cage detection and segmentation is discussed in detail. The material used is presented in section 6.4. Section 6.5 presents the experimental results of the method. The chapter ends with discussion and conclusions in section 6.6.

6.2 Overview of the general framework

The framework consists of five stages: (1) detection of voxels with relevant image structure; (2) construction of primitives from these voxels; (3) classifying which primitives are part of the objects of interest; (4) grouping and recognition of the object primitives, and finally, (5) a full segmentation based on the extracted groups.

The actual implementation of each stage will depend on the application. In principle each stage can be implemented as a supervised system, trained by ex-

amples. For the first two stages, however, it is typically more appropriate to use generic operations to detect blobs (spherical objects), ridge-like elongated structures (vessels, bones), edges (object boundaries), sheet-like patches (sternum, scapula, pelvis, skull) and heuristic grouping algorithms to construct primitives from these voxels. In this way, the first two stages are related to Marr's primal sketch [74]. After the primitives have been extracted, we are now no longer dealing with voxels, but have replaced them with a more natural representation for the problem at hand.

An important additional effect of the first two stages is that the complexity of the classification problem, in the third stage, is vastly reduced. Typically the number of primitives is several orders of magnitudes smaller than the number of voxels in a scan. To achieve this reduction, primitives should preferably contain many voxels. On the other hand, the primitives should not be too big, as each primitive should belong to a single object of interest to be classified unambiguously.

The third stage performs this classification. For most tasks, it will be difficult to design simple rules for such a classification and a supervised approach will be beneficial. A set of features should be computed, and each primitive is represented as a point in this feature space. The problem is now in the realm of statistical pattern theory and different classifiers and feature selection and/or extraction techniques can be explored see e.g. [21, 31, 40].

Typically each object of interest will contain several primitives, otherwise the second stage of the framework would have solved the recognition problem already. The fourth stage groups the foreground primitives into objects and thus performs recognition. This can be a very challenging stage, especially if the results of previous stages are far from perfect or when there are many primitives and objects to group. Here the coupling between local and global models takes place: interrelationships between primitives must be modelled. If a satisfactory solution cannot be obtained in this stage, the system should reconsider the results of the earlier stages. We conjecture that the human visual system is very adept at exploiting such feedback control, in order to solve complex recognition tasks almost instantly. The incorporation of such flexibility in computer vision algorithms is still a very open research area. For the application in this work, feedback between the stages is not considered, and a fairly simple heuristic algorithm suffices for grouping and recognition.

Finally, in the fifth stage, the grouped primitives are used to obtain the full segmentation. These groups provide a "skeleton" or an approximate outline of the objects, and can be used as a precise initialization. A wide range of segmentation methods can be used, both heuristic such as region growing [2], deformable models [57], level sets [111], or supervised, e.g. active shape or appearance models [17, 18], m-reps [99], deformable organisms [78], etc.

We distinguish three categories of tasks to which this framework could be applied: (1) segmentation of a single object (no recognition needed), (2) segmentation of multiple similar objects and (3) segmentation of multiple different objects. As examples we mention segmentation of the aorta (cat. (1)), segmentation of the

rib cage (cat. (2)), segmentation of the aorta and iliac and renal arteries (cat. (2)) or the full skeleton (cat. (3)). For category (1) and (2) segmentation tasks only a single type of primitives is needed. For category (3) probably different types of primitives are needed.

In some category (2) tasks it is possible to construct all primitives directly from the output of the first stage. However, in cases where the objects to be segmented are of different sizes, like aorta and iliac and renal arteries, a multi-scale construction is probably preferable.

For category (3) tasks different primitives must be constructed at multiple scales, and the classification stage should be performed on any of these different types of primitives in turn.

The demands increase with the categories (1)–(3). For category (1) tasks, each primitive must be assigned to one of the objects. For category (2) tasks, putting primitives in the right order is not sufficient for recognition, and interrelationships between the objects need to be established. The same is true for category (3) tasks, although recognition of certain parts might be easier due to the different nature of the primitives.

In the remainder of this chapter we focus on an instance of the framework on a category (2) task: segmentation of the rib cage in chest CT-scans.

6.3 Method

6.3.1 Stage 1: 1D ridge detection in 3D

Because ribs are 1D elongated structures, we detect in the first stage 1D ridge voxels. To detect the ridge voxels of the bone structure in CT-data, the images are first thresholded at a Hounsfield $t_{\mathrm{H}} = 100$ resulting in a binary image, see figure 6.1.

Next the thresholded image is blurred with a Gaussian kernel. As a result, there will be a local maximum at the centerline in the plane perpendicular to the rib, see figure 6.2. In the blurred image the ridge voxels are detected with the algorithm that is given below. This algorithm is a simplified version of the method presented in chapter 2 to detect the point sets $R^{0,2}$. Other schemes for ridge detection are described in [22, 73].

An elongated structure in a 3D image is an 1D curve. The tangential vectors to this curve are determined by the eigenvector of the Hessian matrix $\boldsymbol{H}$ with smallest eigenvalue in absolute value. The matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \begin{pmatrix} L_{xx} & L_{xy} & L_{xz} \\ L_{yx} & L_{yy} & L_{yz} \\ L_{zx} & L_{zy} & L_{zz} \end{pmatrix} \ ,$$

where L_{ij}, $i, j \in \{x, y, z\}$ represents the second order derivative of the luminance with respect to the coordinates. Because taking derivatives of discrete images is an ill-posed operation, they are taken at a scale σ using the Gaussian scale-space

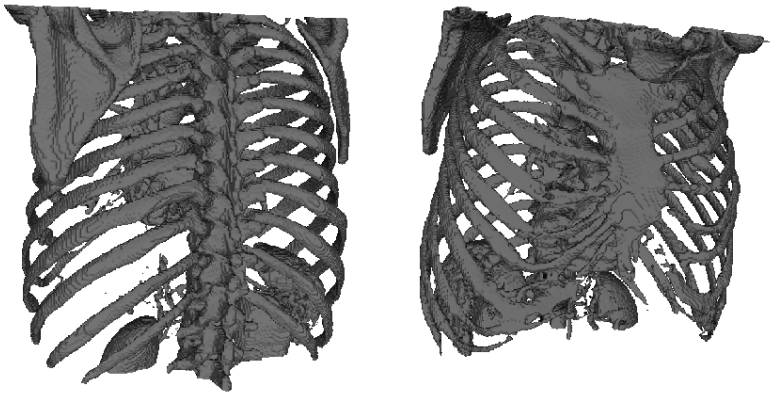


Figure 6.1 Two surface renderings, one from the back and one from the front, of a thresholded CT-scan to obtain the bone structure. Note that because of the presence of contrast agent also the heart, some vessels and other structure is found.

technique, see section 1.4.2. The main idea is that the image derivatives can be taken by convolving the image with the derivatives of a Gaussian

$$L_{ij} = L_{ij}(\boldsymbol{x};\sigma) = \int_{\mathbb{R}^3} L(\boldsymbol{x}) G_{ij}(\boldsymbol{x}-\boldsymbol{x}';\sigma)\mathrm{d}\boldsymbol{x}' \ ,$$

with $\boldsymbol{x} = (x,y,z)^T$ and

$$G(\boldsymbol{x};\sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} \exp\left(-\frac{\boldsymbol{x}\cdot\boldsymbol{x}}{2\sigma^2}\right)$$

the Gaussian kernel in 3D.

By ordering the eigenvectors $\boldsymbol{v}_i$, $i \in \{1,2,3\}$, in decreasing magnitude of their corresponding eigenvalues λ_i, i.e. $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$, the vector tangential to the elongated structure is given by $\boldsymbol{v}_3$. The other two eigenvectors span the plane perpendicular to $\boldsymbol{v}_3$. A bright elongated structure has a maximum in this perpendicular plane (and a dark structure a minimum), see figure 6.2. All eigenvectors are of unit length.

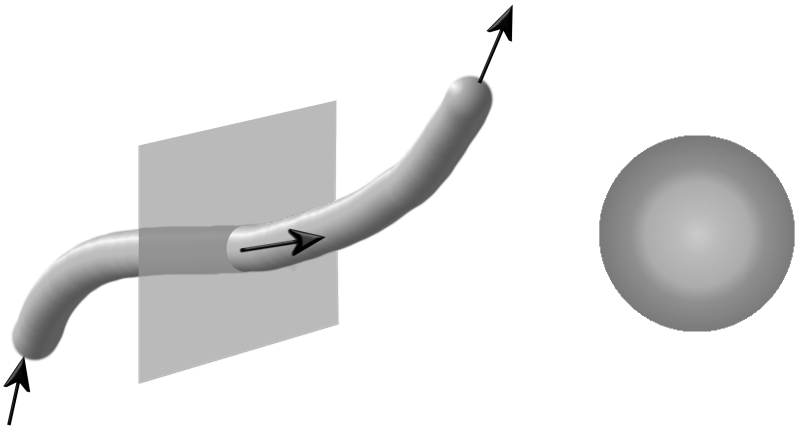
As a result, detection of the 1D ridges is reduced to determining at every voxel whether there is a local maximum or minimum of the intensity in the plane normal to $\boldsymbol{v}_3$.

The detection of the maximum in the 2D plane is done as follows. Around the voxel $\boldsymbol{x}$, the circle

$$\boldsymbol{c}(\theta) = \boldsymbol{x} + \rho(\boldsymbol{v}_1\cos\theta + \boldsymbol{v}_2\sin\theta) \ , \quad \theta \in [0,2\pi) \ ,$$

is defined, with ρ the radius of the circle. The image has a maximum on a 1D ridge if

$$L(\boldsymbol{x}) - L(\boldsymbol{c}(\theta)) > 0 \ , \text{ for all } \theta \ .$$



(a) (b)

Figure 6.2 (a) An elongated structure in 3D. Along the planes perpendicular to its centerline (subdimensional) maxima (bright structures) or minima (dark structures) are found. (b) Image intensities in a plane perpendicular to the centerline.

For a minimum $L(\boldsymbol{x}) - L(\boldsymbol{c}(\theta)) < 0$.

In the mathematical continuous world, the limit of ρ towards zero would be taken. On the discrete image lattice, a choice of $\rho = 1.0$ voxel seems natural. Furthermore, the polar angle θ must be discretized. In this chapter we have chosen to use 8 different angles, corresponding to an eight-connected neighborhood in 2D images. Finally, note that in general $\boldsymbol{c}(\theta)$ will fall in between grid-points. To evaluate L at those points, linear interpolation is used.

6.3.2 Stage 2: from ridge voxels to line elements

The next step is the construction of primitives in the form of line elements from the set of 1D ridge voxels. For this purpose we use the algorithm to construct affine convex sets, as described in section 3.2. To prevent leaking of the region growing process, so that one elongated structure connects to another, a threshold ϵ_{m} can be set on the maximum size a convex set may attain.

6.3.3 Stage 3: classification of the primitives

After the formation of primitives, some of them will belong to the objects we want to segment (ribs) and some will not (background). We use a classifier to distinguish ribs from other primitives. A classifier maps a feature vector extracted from the objects under investigation to class numbers or to a vector of probabilities, where each element in the vector is the probability of belonging to that class. The mapping must be learned from example data, where the examples have been labeled to the appropriate class numbers.

The features used in this work can be divided in two parts. Features that encode information from a single primitive and features that encode information between two primitives. If only the first type of features is used, any classifier from pattern recognition theory can be used, e.g. linear and quadratic discriminant classifiers, neural networks, kNN-classifiers [9, 21]. However, if we want to include features of the second type as well, there is no straightforward method to use these classical classifiers. For that reason, we use the spin-glass based classifier that is discussed in chapter 5.

For the local features geometrical information of the primitives and intensity based information is extracted. The geometrical information used is the Euclidean length of the primitive, its curvature and the x, y and z coordinate of the main axis of the primitive (5 features). For the intensity based features (combinations) of Gaussian derivatives are computed on 4 scales ($\sigma = 1.0, 2.0, 4.0$ and 8.0 voxels). The values of the derivatives are averaged over the primitive, i.e. a feature $\phi_i = \frac{1}{N}\sum_j D_j$ where j runs over the locations of the primitive, N is the number of voxels in the primitive and D_j is the derivative (combination) of interest. All zeroth, first and second order partial derivatives are computed (1 + 3 + 6 features times 4 scales), resulting in 40 features. The gradient magnitude, the determinant of $\boldsymbol{H}$ and the trace of $\boldsymbol{H}$ are also included, adding another 12 features (3 features times 4 scales). So, in total 57 local features are extracted.

For the features between two primitives ξ_i and ξ_j we compute a measure for distance (distance between the closest end-points), a measure for mutual orientation (inner product between the unit vectors aligned with the primitives) and an alignment measure (3 features), see figure 5.2. Furthermore we use the mean and the absolute difference of the curvatures (2 features), the mean and absolute difference of the zeroth order derivatives (4 scales times 2 features) and the mean and absolute difference of the x, y and z coordinate of the main axes of the primitives (6 features). As a result, 19 features are extracted.

For the evaluation of the classifier an independent test set is needed, i.e. a set with known labeling which is not used in the training stage of the classifier. Because the classifier gives as output a posterior probability, we quantify the results of the classification process with ROC-curves, see section 1.4.4.

Beforehand, it is unknown which features produce good results and which features deteriorate the classification process. A means to determine the appropriate features is to conduct feature selection [21]. The feature selection method that has been applied is sequential forward selection, see section 1.4.3.

After training and feature selection, new data can be classified. However, since the output of the classifier is a probability, we must set a threshold t_{P} to get a classification in foreground and background. This threshold can be estimated from the training set, by taking that value $t_{\mathrm{P,max}}$ which maximizes the accuracy of the system[1]. The accuracy measures the rate of correct predictions made by the classifier over the complete set.

[1]With respect to the discussion on Bayes' formula and the decision rule, in chapter 1, we have changed the cost function that is presented in eq. (1.7).

6.3.4 Stage 4: grouping of primitives

The goal of this stage is two-fold: to group the rib primitives into centerlines of the specific ribs and to label the centerlines to rib side and number (e.g. 10th left rib).

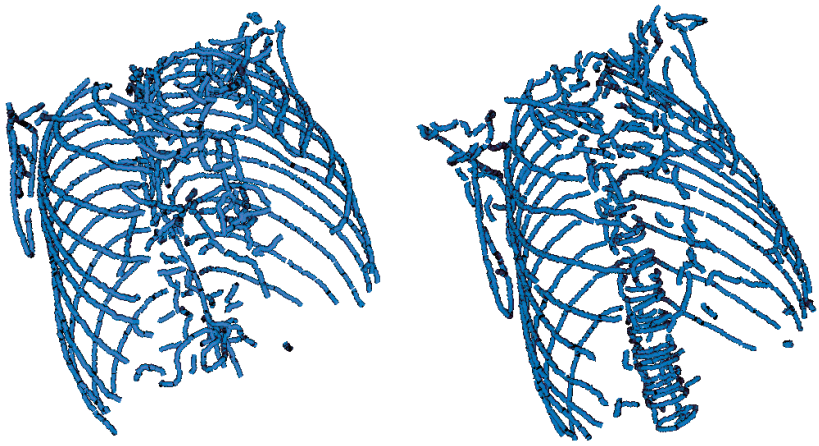
For the centerline construction, the mechanism presented in section 6.3.2 is ran again, but this time on the ridge voxels that constitute the primitives classified as foreground only. Due to the absence of the background primitives, the constraints on orientation and parallelism can be relaxed and the maximum size requirement can be removed.

For the labeling of the centerlines two simple heuristic algorithms prove to be sufficient. First, the centerlines are divided in left and right parts by computing the center of gravity of all centerlines. Comparing the center of gravity of a specific centerline with the total center of gravity decides to which side the centerline belongs.

Then, to obtain the rib numbers for one side, the longest centerline from that side is picked. Next, the centerlines that are closest above and below are detected. The centerline found above is used to find the next centerline above. In a similar fashion, the next centerline below is detected. This scheme is repeated until no more centerlines can be detected. To determine which centerline is closest, we define a distance between two centerlines as follows. Every ridge voxel in the first centerline is mapped to the closest (in Euclidean distance) ridge voxel in the second centerline and vice versa. Note that multiple-to-one mapping is possible. Now, the distance between two centerlines is taken as the average of the distances between the mapped ridge voxels. At the start of the number assigning, it has to be decided which of the two closest centerlines is above and which is below the longest centerline. The average z value of the constituting ridge voxels is used for that purpose. The algorithm stops when 11 ribs are found. Then, the final rib, which is not present in all scans, is searched for. A final rib is only included if it is not too small (similar size as closest rib) and if its distance to the closest rib is comparable with the distances between the other ribs.

6.3.5 Stage 5: full segmentation

After the classification stage the centerlines of the objects that must be segmented have been estimated. For the full segmentation, we have chosen to use a seeded region growing algorithm as described in [2]. At the start of the algorithm, the mean gray values of the image per centerline are computed. Then a list is initialized to which all the neighboring voxels of all centerlines are added. The voxel in this list that has the smallest difference in gray value as compared to the mean of its neighboring centerline is removed from the list and added to the centerline. Next, the mean of the centerline is updated and the neighbors of the added voxel, if not already present, are added to the list of boundary voxels. The algorithm continues until the list of boundary voxels is empty. To circumvent leaking in non bone structures, only voxels above a gray level threshold t_b and and closer



(a) (b)

Figure 6.3 Ridges of two chest CT-scans. (a) Input image size is $256 \times 256 \times 235$. (b) Input image size is $256 \times 256 \times 288$.

than a distance $d_{\max}$ to the centerline are added to the boundary list.

To prevent leaking into structures that are not foreground, the region growing is done on the centerlines and the non-grouped background primitives.

6.4 Material

In this study, 40 CT-scans of the thorax of different patients are used. The scans have been randomly selected from clinical practice and have been acquired at the University Medical Center Utrecht on a 16-detector CT scanner (Mx8000IDT, Philips, Best, the Netherlands). Different protocols have been used, but in all scans the slice thickness is 1.0 mm and the slice spacing 0.7 mm. All scans have been reconstructed to 512×512 matrices, and the in-plane resolution varies between 0.57 mm and 0.91 mm, depending on patient size. The number of slices varies from 407 to 706. In 36 of the scans intravenous contrast material has been administered.

The high resolution of the scans is not needed for the first four stages of our method. To reduce computation time, the scans are subsampled by a factor 2, resulting in sizes of $256 \times 256 \times 203$ to $256 \times 256 \times 353$ voxels. The final segmentation, however, is done on the full resolution images.

In order to test the classifier with a independent test set, the scans are divided randomly in 20 scans for training and 20 for testing.

		reference		
		rib	non-rib	total
computer	rib	3105 (31.0%)	147 (1.5%)	3252 (32.5%)
	non-rib	103 (1.0%)	6661 (66.5%)	6764 (67.5%)
	total	3208 (32.0%)	6808 (68.0%)	10016

Table 6.1 Confusion matrix for the primitives in the test set consisting of 20 scans for $t_{\mathrm{P,max}} = 0.85$.

	rib 1	rib 2	rib 3 – rib 11	rib 12
left	2	19	20	17
right	0	20	20	17

Table 6.2 Results of rib detection divided in left and right ribs. Every entry gives the number of ribs found for the 20 test images. For rib 1 no labels were set in the test set. For rib 2 upto and including rib 12, 20 ribs should have been found.

6.5 Results

This section presents the results of each processing stage and lists particular parameter values.

6.5.1 Stage 1: ridge detection

After thresholding the data at a Hounsfield $t_{\mathrm{H}} = 100$, the resulting binary image is blurred with a Gaussian for which $\sigma = 2.0$ voxel. The blurred image is used to detect the ridges. Output of the ridge detection is shown in figures 6.3(a) and 6.3(b) for two of the data sets.

6.5.2 Stage 2: primitive construction

To extract the primitives, the convex sets algorithm is ran with $\epsilon_{\mathrm{c}} = 5.0$, $\epsilon_{\mathrm{o}} = 0.9$, $\epsilon_{\mathrm{p}} = 0.8$ and $\epsilon_{\mathrm{m}} = 20$.

In figure 6.4 the obtained primitives of the datasets in figure 6.3 are shown.

6.5.3 Stage 3: classification

For the training of the classifier a labeled example set (training set) is needed and for the evaluation of the system an independent test set. To enable manual labeling a 3D interface has been written that enables an observer to select primitives by clicking. The observer has the opportunity to view slices of the original CT-scan together with the primitives. It is also possible to perform rotations, zooming and window leveling. Labeling of one image took about 5 minutes. The observer also added information to which rib a primitive belonged (e.g., left side, second rib).

Because the first left and right ribs are not always visible in the scans, these are not labeled.

In the training set 3,290 primitives out of 9,455 primitives are labeled as rib (35%), in the test set this number is 3,208 out of 10,016 (32%).

For the classification, the Metropolis algorithm is run with $M = 1,000$ steps and the *a priori* probabilities P_i and P_{ij} have both been estimated using kNN-classifiers with $k = 21$.

The feature selection is performed by splitting the training set randomly in 10 scans for training and 10 scans for evaluation. After the feature selection, 17 local features and 7 interaction features are retained. The optimal threshold on the posterior probabilities $t_{\mathrm{P,max}}$ is found to be 0.85.

Running the classifier on the test set results in the ROC-curve of figure 6.5. The value for A_z is 0.992. The corresponding confusion matrix after thresholding the posterior probabilities with $t_{\mathrm{P,max}}$ is shown in table 6.1.

From table 6.1 we find that the accuracy is 0.975, the sensitivity 0.968 and the specificity 0.978.

An example of the classification results on the primitives of figure 6.4 is shown in figure 6.6. Note that in figure 6.6(b) some primitives of the fifth right rib are misclassified (the upper rib is the second rib of the subject).

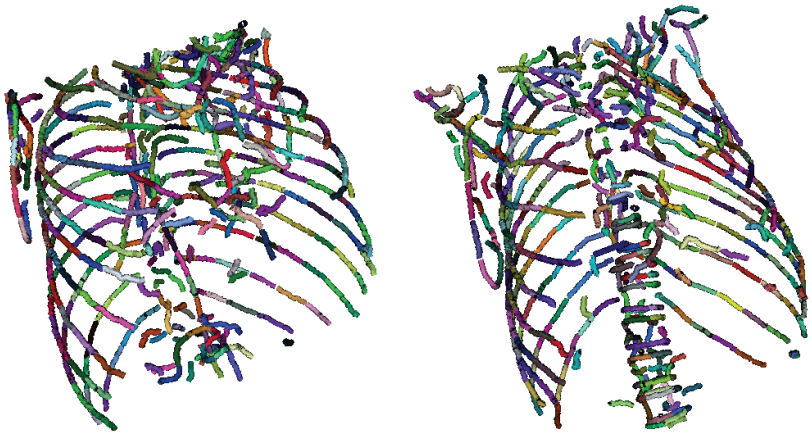
6.5.4 Stage 4: grouping of foreground sets

Grouping of the primitives is done by running the convex sets algorithm on the primitives classified as rib. There is no restriction on the size of the output sets and the orientation and parallellity constraint are relaxed to $\epsilon_{\mathrm{o}} = 0.5$ and $\epsilon_{\mathrm{p}} = 0.7$. To be able to close some gaps, $\epsilon_{\mathrm{c}} = 11$ voxels is taken. Of the thus constructed centerlines, the 11 centerlines on the left and right side detected and labeled to rib number as described in section 6.3.4. If after the labeling a twelfth rib can be found, it is added to the set of foreground centerlines. The size of the final rib must be at least 80% of the closest centerline. Its distance to the closest centerline must be in between 0.9 and 1.1 of the mean distance of the already recognized centerlines. These values are estimated from the training data. In figure 6.7 the results are depicted for the data sets of figure 6.6.

An evaluation on rib detection shows that in 3 images 20 of the 22 selected ribs are found. In 2 of these images the left and right 12th ribs are missing, in 1 image a 2nd and a 12th rib are missing. In 1 image 21 ribs are found (12th rib missing), and in 2 images 23 ribs are detected (2 of the non labeled 1st ribs were found). All other ribs in all images are correctly found and labeled. The detection results are further specified in table 6.2. Taking only rib 2 upto and including rib 12 into account, $433/440 \times 100\% = 98.4\%$ of the ribs is recognized.

6.5.5 Stage 5: full segmentation

The previous stages have been performed at half the resolution of the original scans. For the full segmentation the full resolution images are used and there-



(a) (b)

Figure 6.4 The primitives of figures 6.3(a) and 6.3(b). Every set has it own color.

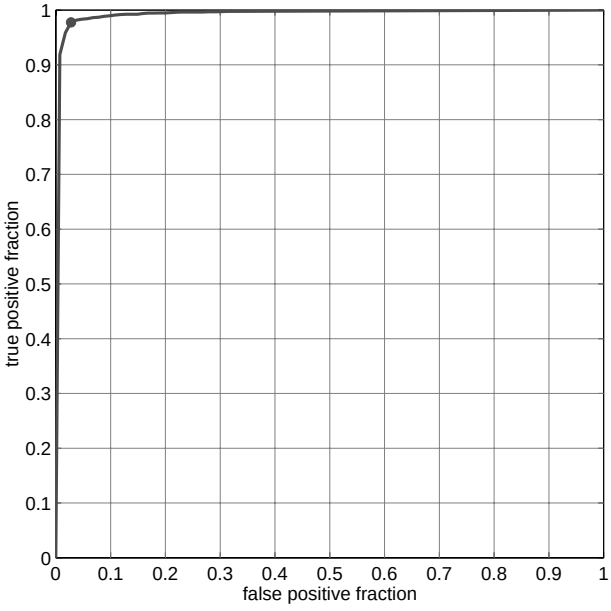
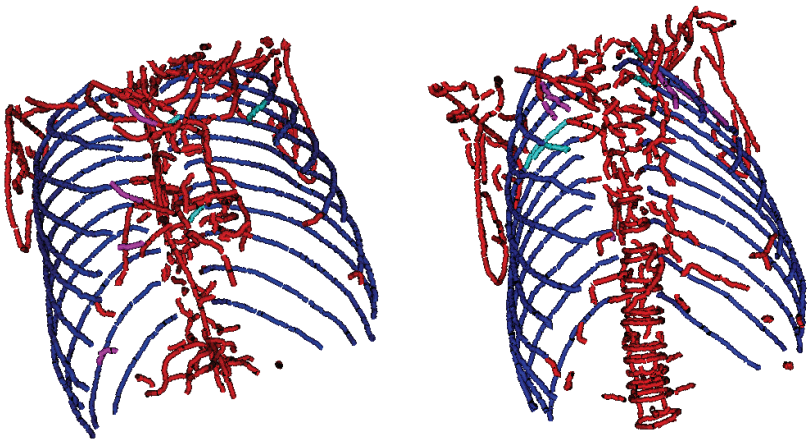
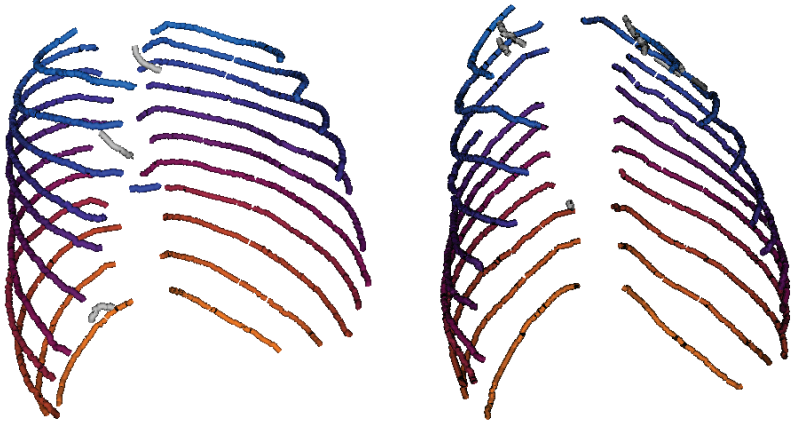


Figure 6.5 The ROC-curve for the test set. The area under the curve is 0.992. The point denotes the performance of optimal accuracy.



(a) (b)

Figure 6.6 Results of the classification of the primitives of figure 6.4. True positives are shown in dark blue, true negatives in red, false positives in magenta and false negatives in cyan.



(a) (b)

Figure 6.7 Result of grouping the foreground primitives of figure 6.6. Centerlines not considered ribs are in light gray. The numbering of the ribs is color coded: from the upper centerline in blue via purple to the lower centerline in orange. Note that because some of the primitives of the fifth rib in figure 6.6(b) are misclassified, in (b) these parts are missing.

	into spine	too short	too long
total	115	26	18
average per scan	5.75	1.35	0.9
percentage of detected ribs	26.4%	6.2%	4.1%

Table 6.3 Results of a qualitative evaluation of the rib segmentations. For each rib it has been determined if it has grown into the bony structures of the spine, if it is too long or if it is too short. In the first row the total number of ribs, in the second row the average per scan (20 scans are evaluated) and in the last row the results as percentage of the number of detected ribs (in total 435 ribs are detected).

fore the coordinates of the centerlines are multiplied by a factor 2. In the seeded region growing algorithm, only voxels in a neighborhood of 40 voxels around every centerline are taken into account. Voxels with a Hounsfield unit below 130 are considered to belong to background. In figure 6.8 the results for the data sets of figure 6.7 are shown. The 6th and 7th rib on the left side in figure 6.8(a) are dented, an observation that is not easily made when inspecting the scan slice by slice. A rendering from a different perspective highlighting this detail is shown in figure 6.9. Note that due to the missing primitives of the 5th right rib in figure 6.7(b), also the full segmentation in figure 6.8(b) is incomplete.

Since we have no manual reference for the full segmentation, a qualitative evaluation has been conducted. First, the observer inspected surface renderings, similar to the ones in figure 6.7, to check whether the ribs had grown into the bony structures of the spine. After checking the surface rendering, the observer inspected the scans slice by slice with the segmentation overlayed on it. The observer had the possibility to turn on and off the overlay during inspection. In this stage, the observer was asked to count the ribs that are too long (growing into the cartilage) or too short (typically because a primitive has been misclassified). The results of this evaluation are shown in table 6.3. With respect to the length of the ribs, we conclude that the majority of the ribs is segmented correctly. The table shows that leaking of the region growing process into the spine occurs often. However, the results shown figure 6.8 are typical examples, where for many ribs only a small region has connected to the spine.

6.6 Discussion

In this work a five stage method has been presented for automatic detection, recognition and segmentation of elongated structures in 3D data. The approach can easily be adapted to other types of structures, such as ridge or edge surfaces, for which primitives in the form of patches can be constructed. The method can also be used for other data dimensionalities.

Local segmentation methods like voxel classification are not the most suitable approaches when large data sets have to be processed. There are a few rea-

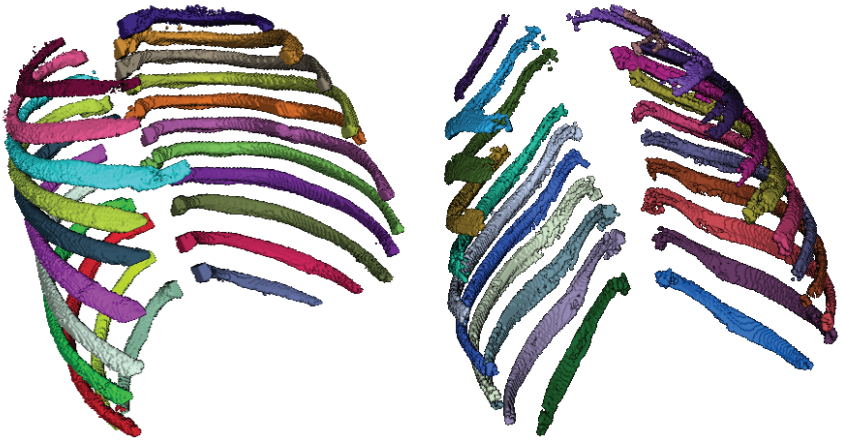
sons. First of all, obtaining (manually) labeled reference sets, which are needed for training, is almost impossible. Second, when a large amount of features is needed, the time required for computing the features, training the classifier and segmenting the set sets can be prohibitively large. And finally, local methods yield no recognition of the segmented objects.

Global schemes, like active shape models and active appearance models, return a segmentation in which recognition is granted. However, these methods rely on a careful initialization. Using these models e.g. for rib cage segmentation can easily lead to results where the ribs in the detected rib cage are shifted with respect to the ground truth. This is due to the fact that the model will locally be well fitted to the data, and therefore cannot escape from this local minimum. Another important issue with these schemes is that example data is needed with reliably annotated corresponding landmarks, which is extremely difficult in 3D data sets. This hampers use of these models, especially because they need many example sets to capture the anatomical variability that is inherently present in 3D medical data.

In our proposed approach, we try to surmount some of the limits discussed above. The construction of primitives allows for a quick classification scheme, because the number of primitives is order of magnitudes lower than the number of pixels. For example, the CT-scans in this chapter contain about 10^7 voxels, as opposed to only 500 primitives on average. Not only is the computation of a lot of features now possible, but manual labeling is feasible too. In the CT-scans on average 160 primitives had to be clicked per image (the other 340 being background) and manually labeling the primitives of one image took about 5 minutes.

With respect to the method's computation time for the rib application, the whole scheme can be ran in the order of a few minutes. Computing the features for an image takes about 1 minute, the detection and classification of the primitives about 2.5 minutes and the region growing about 3 minutes. All computations were done on a PC with an AMD Athlon XP 1800+ processor and 1 GB memory. The code used for this algorithm has not been optimized.

For the classification and recognition of the primitives, primitives that are as large as possible are favorable. The size of the primitives is governed by the four parameters that are introduced in chapter 3 and section 6.3.2. The connectivity parameter ϵ_c is used to bridge gaps which might have been caused by interference of other image structure (crossing of ridges, noise). A too large value of this parameter will give erroneous results if no good candidates are found in the close neighborhood. A value that is too small will give many fragmented primitives. However, the setting is not critical, for a larger setting (between 5 and 10 voxels) we found similar results. The choice for the parameters for orientation and parallellity, ϵ_o and ϵ_p, have been chosen strict in the construction for the primitives in section 6.5.2. If the conditions are weakened (say we choose 0.7 for both parameters) primitives will grow into different objects. For the same reason the threshold ϵ_m on the maximum size is included. If it is made smaller the number of primitives that has to be classified increases. If it is made larger, primitives which



(a) (b)

Figure 6.8 Full segmentation of the data sets of figure 6.7. (a) The 6th and 7th rib on the left side are dented, an observation that is not easily made when inspecting the scan slice by slice. (b) The fifth right rib on the right is not complete, because the primitive that should have initialized the region growing algorithm is not classified as rib.

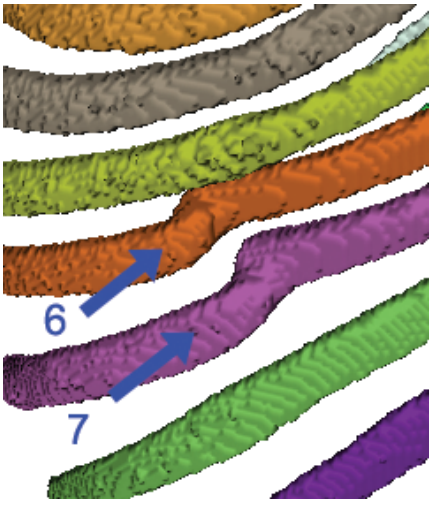


Figure 6.9 Detail of the dented ribs 6 and 7 on the left side of figure 6.8(a). Surface rendering is taken from a different perspective.

belong to rib and other structures will merge. So, clearly there is a tradeoff. After selection of the foreground primitives, however, relaxation of these conditions gives good grouping power, as is shown in the rib detection application.

In the majority of the scans, the ridge voxels of some of the vessels are detected because the patients have often been administered an intravenous contrast agent. However, the use of a classifier enables the discrimination between vessel and rib primitives.

Of course, there are disadvantages with our approach as well. Misclassification of a primitive means missing or adding a substantive structure. An example is shown in figures 6.7(b) and 6.8(b). A human being would probably not make such a mistake, because he or she would infer from the global model that a part is missing. We conjecture, that in our proposed model, similar behavior can be introduced by a feed-back coupling between the fourth and the third stage. However, the details of such a feed-back coupling scheme remains an open question at this moment and is subject of future research. In the construction of the rib cage centerlines, we are able to employ simple heuristics to group the primitives and feed-back seems not to be necessary for most of the detected ribs (only 1.35 ribs per scan turned out to be too short).

In the grouping stage, prior knowledge can be introduced in the form of global information: it is known that human beings have 24 ribs. We exploit this knowledge when the 11 most plausible centerlines on each side are taken as ribs and a possible 12th rib on both sides is looked for. As a result of using global information, the number of false positives is reduced. Again, feed-back coupling between the classification and grouping stage might also be advantageous to reduce the number of false negatives. In general, we argue that modelling of multiple parts to detect one object increases the robustness of the detection and recognition stage.

The classification task in this chapter has been examined with other classifiers as well, such as kNN-classifiers and quadratic discriminant classifiers [21]. Note that there is no straightforward way to incorporate information between primitives with these classifiers, so we tested them with local features only. Their performance is significantly worse than with the spin-glass based classifier presented in chapter 5.

In the final stage of our algorithm, we have used a seeded region growing algorithm [2], because of its simplicity and the nature of the data. However, to obtain better results or in case of different data, the found centerlines can serve as initialization for other, more powerful segmentation schemes, such as deformable models [57], active shape models [18], active appearance models [17], level sets [111] and m-reps [99]. The segmentation as such, however, is not the main contribution of this work. The key contributions that we identify are (1) the presentation of a general applicable framework; (2) a spin-glass model that can be trained for classification; and (3) a completely automatic scheme that groups and recognizes image primitives.

Chapter 7

Discussion and summary

7.1 Introduction

THIS final chapter consists of three parts. In the first part, we illustrate that the methods developed in this thesis can be applied to a wide range of clinical applications. First, we consider finger detection in hand radiographs, then we continue with spinal cord detection in CT-scans of the thorax and we end with showing an example of bone extraction in a full-body CT-scan.

The second part of this chapter discusses some issues that are not raised elsewhere in this thesis and that may be very important for extensions of the proposed framework to other applications.

In the last part of the chapter, a summary of the thesis is given.

7.2 Other areas of application

7.2.1 Finger detection in hand radiographs

The assessment of the skeletal or bone age is a common procedure in pediatric radiology. Comparing skeletal age to chronical age can reveal atypical skeletal development. For patients diagnosed with endocrine disorders skeletal age assessment is of importance for the therapeutic effect of treatment. In general, estimation of the skeletal age serves as an indication whether the growth of patients is accelerating or delaying. In many cases the decision to treat patients with growth hormones is based on the outcome of their skeletal age estimation.

Skeletal age is usually determined from a radiograph of the left hand and wrist, see figure 7.1. An established clinical method to determine the skeletal maturity is the Tanner-Whitehouse 3 (TW3) method [121], which divides the skeletal development into nine stages. In the TW3 method twenty regions of interest (ROIs) are examined, which are independently labeled into one of nine stages. With each ROI and stage a numerical score is associated and an overall maturity score is computed by adding all scores.

The modular structure of the TW3 method makes it suitable for automation. If one can compute features for every ROI, then a classifier can be trained and

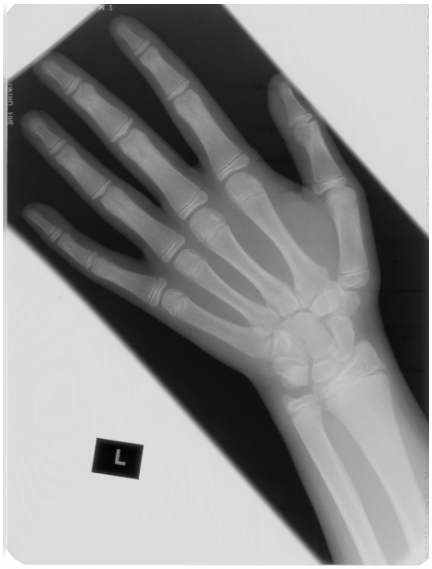


Figure 7.1 Radiograph of the left hand. Image size is 461 × 615 pixels.

used to assign skeletal age [36, 85, 94–96, 120]. However, a key point, when using the TW3 system is the availability of the ROIs, for which no automatic extraction method is currently available.

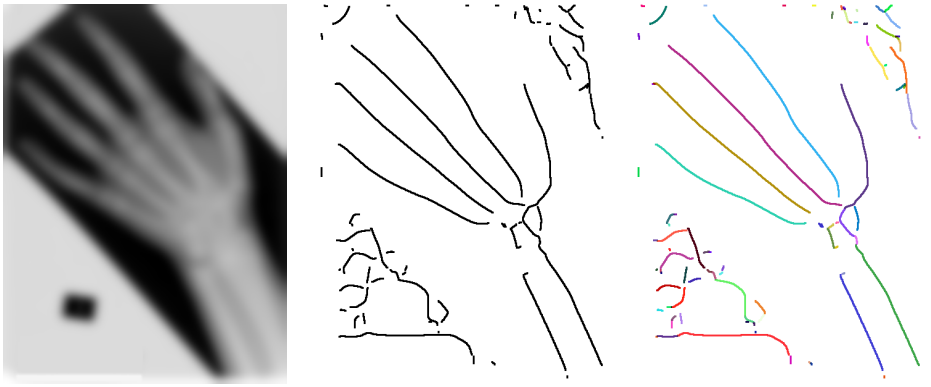
With the techniques developed in this thesis, we give a first start for such a method by automatic detection of the fingers in hand radiographs.

Method

If the hand radiographs are blurred with a large scale, the fingers become elongated white structures, see figure 7.2(a). The locations of the centerlines can be detected using ridge detection (chapter 2), for which the result is shown in figure 7.2(b). Next, primitives can be constructed, for which we use the convex sets algorithm (chapter 3), see figure 7.2(c).

As can be observed from figure 7.2(c), the fingers are a subset of the convex sets. Extraction of the fingers can be done with a classifier. The features are based on the mean profile $\psi(n)$, $n \in -N, \ldots, N$, of the convex sets (cf. section 4.3.3). The extracted features are:

1. The height of the profile: $h = \psi(0)$.
2. The width of the profile defined as the distance between the strongest right and left edge of the profile: $w = n_{\mathrm{re}} - n_{\mathrm{le}}$ ($n_{\mathrm{re}} = \arg\max_{n>0} \psi'(n)$, with ψ' the first derivative of the profile. n_{le} is defined similarly for $n < 0$).
3. The height divided by the width: h/w.
4. The edge strength, defined as: $s_{\mathrm{e}} = \psi'(n_{\mathrm{le}}) + \psi'(n_{\mathrm{re}})$.



(a) (b) (c)

Figure 7.2 (a) The image of figure 7.1 blurred with a Gaussian of scale $\sigma = 8.0$ pixel. (b) The ridges of the image in (a). (c) The convex sets of (b). Settings used here are $\epsilon_c = 5.0$ pixel, $\epsilon_o = 0.95$ and $\epsilon_p = 0.95$

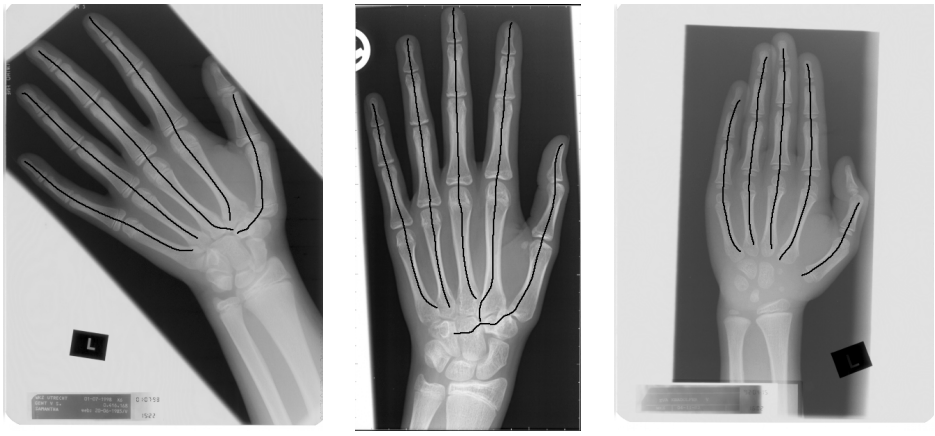
5. The edge strength divided by the width: s_e/w.
6. The edge height: $h_e = \frac{1}{2}(\psi(n_{le}) + \psi(n_{re}))$.
7. The height minus the edge height: $h - h_e$.
8. The height divided by the edge height: h/h_e.
9. The length l, which is computed as the Euclidean length between the endpoints of a convex set.

Material and results

For these experiments a database with 62 radiographs of the left hand of girls is used. The age of the girls varies between 6 and 17 year. All images have been scanned from film and afterwards been down-sampled by a factor 2. The final image sizes vary between 290–536 $\times$ 573–719 pixels. Before ridge detection, all images have been blurred with a Gaussian kernel of scale $\sigma = 8.0$ pixels. For the convex set construction, we use $\epsilon_c = 5.0$ pixel, $\epsilon_o = 0.95$ and $\epsilon_p = 0.95$. In order to have an independent test and training set, the database has been split in two at random. The manual labeling of the convex sets has been done using a tool where a set can be labeled by clicking on it.

The classifier that has been used is a kNN-classifier, with $k = 21$. Sequential floating forward selection has been applied using the training set, which has been split in two for this purpose. The criterion function used is the area under the ROC-curve. The following four features have been selected: (1) h, (2) s_e/w, (3) $h - h_e$, (4) l.

Final classification on the test set is done in the following way. The five convex sets that have the highest posterior probability are taken as finger, the rest as background. With this system, all fingers in the test set are classified correctly



(a) (b) (c)

Figure 7.3 Some results of the finger detection system. The detected fingers are overlayed on the original hand radiograph. (a) Skeletal age is 12 years. (b) Skeletal age is 14 years, 6 months. (c) Skeletal age is 6 years.

and no errors in the background are being made. A few examples are given in figure 7.3.

Discussion

We have shown that classification of elongated structures can be used for detection of fingers in hand radiographs. Note, however, that in some fingers we have detected extraneous parts (see the thumb in figures 7.3(a) and 7.3(b)) or that parts are missing (e.g. the top of the thumb in figures 7.3(a) and 7.3(c)). Also, we have not labeled the individual fingers, which is clearly necessary to obtain the ROIs that are needed in the TW3 method. Using the constellation of the detected convex sets it should be possible to identify the individual fingers, cf. the methodology of chapter 6. After that has been accomplished, detection of the locations of the ROIs can be considered.

7.2.2 Spinal cord detection in CT

Introduction

In radiotherapy treatment patients are exposed to radio-active radiation in order to destroy malignant tissue, while normal tissue should be spared as much as possible. The spinal cord is an extremely radiosensitive organ and overexposure to radiation may cause complications such as paralysis or other neuronal disorders [38, 93]. Recently, two papers have been published to delineate the spinal cord in CT-scans. The paper by Burnett et al. [12] is semi-automatic and needs manual initialization. The paper by Archip et al. [4] is far more ambitious and

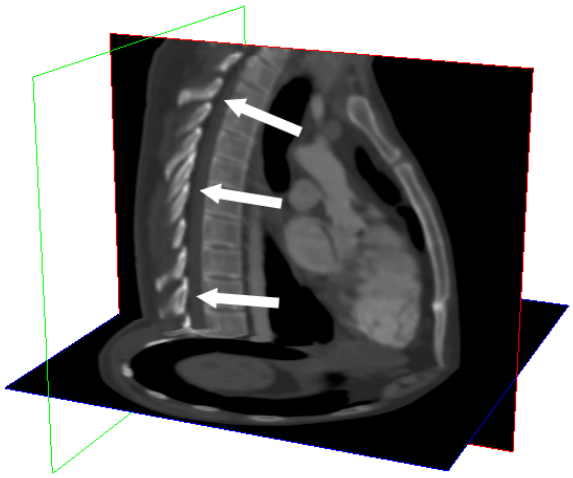


Figure 7.4 Two orthogonal slices in a CT-scan of the thorax. The spinal cord is a dark elongated structure (three locations are marked by the white arrows).

tries to detect and segment the spinal cord using a "macro-anatomy" model of the human thorax, which lays down spatial relationships between different anatomical structures. A disadvantage of both methods is that they do not fully take the 3D nature of the data into account.

The purpose of this section is to show how a slight modification of the ridge detection scheme as presented in chapter 6 can be used to detect the centerline of the spinal cord.

Method

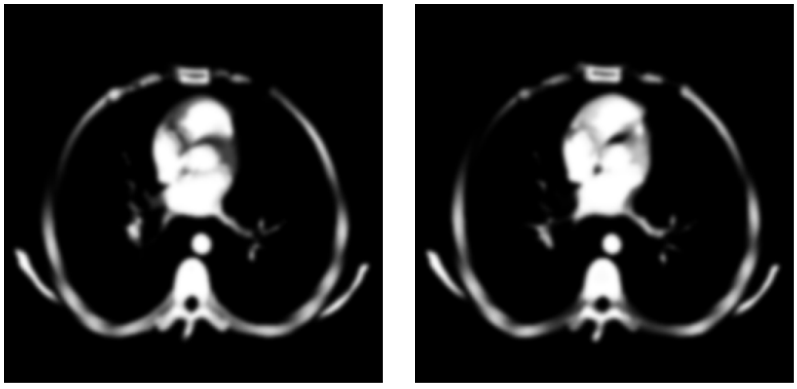
The spinal cord is an elongated and dark structure in a CT-scan, see figure 7.4. In chapter 6 we detect bone structure, which is bright, by thresholding the image at an appropriate Hounsfield unit. The centerlines of the ribs can then be found after blurring the binary image and constructing primitives, which are subsequently classified and grouped. Because the spinal cord is largely surrounded by bone tissue, we again threshold to obtain the bone structure. After blurring we expect to detect the location of the centerline of the spinal cord as local minima in planes perpendicular to the spinal cord, cf. section 6.3.1. However, there is a caveat, because the spinal cord is not completely surrounded by bone tissue, see figure 7.5.

To make sure that we can extract the spinal cord as a ridge of minima, we adjust the blurring in the z-direction by using a larger scale than in the x- and y-direction. Some results are shown in figure 7.6.

The rest of the algorithm is similar to the method presented in chapter 6.

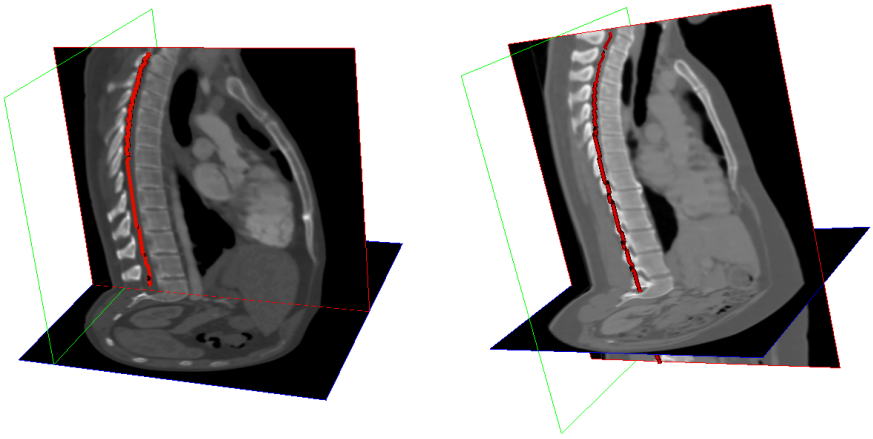
(a) (b)

Figure 7.5 Two axial slices of a CT-scan. To extract the bone tissue, the scan is thresholded at 130 Hounsfield units. (a) The spinal cord is located in the black disc, which is surrounded by the bone tissue to which the arrow points. (b) At some slices, the spinal cord is not completely surrounded by bone tissue.



(a) (b)

Figure 7.6 Results of anisotropic blurring on the two slices of figure 7.5. Both in (a) and (b) the spinal cord is now surrounded by brighter gray values. The image are blurred with $\sigma = 1.5$ pixel in the x- and y-directions and with $\sigma = 4.0$ pixel in the z-direction.



(a) (b)

Figure 7.7 Result of the spine detection algorithm in two CT-scans (full resolution).

The ridges (this time minima) are detected, followed by convex sets construction (chapter 3). An easy grouping stage follows in which convex sets are connected for which the end-points are close. Finally, in every scan the set with the largest distance between the end-points in the z-direction is taken as the centerline of the spinal cord.

Results

We tested the algorithm on all 40 CT-scans that are used in chapter 6. Since the full-resolution images are not needed, the images are subsampled with a factor 2 in all directions. For more details on the scans, see section 6.4.

All image are thresholded at 130 Hounsfield units and blurred with a Gaussian of $\sigma = 1.5$ pixel in the x- and y-direction and with $\sigma = 4.0$ pixel in the z-direction. After the ridge detection, the convex sets are constructed with $\epsilon_{\mathrm{p}} = 7.0$ voxel, $\epsilon_{\mathrm{o}} = 0.8$ and $\epsilon_{\mathrm{p}} = 0.8$. No restriction is set on the maximum size a convex set can attain. In the grouping stage, all sets are connected which have a distance between their closest end-points which is smaller than 10 voxels. Some results of the detected spine centerlines are shown in figure 7.7. To obtain results at the full resolution, the coordinates of the centerlines are multiplied by 2. All spine centerlines in the 40 scans are correctly identified.

Discussion

We have shown that it is possible to detect reliably the centerline of the spinal cord. A full segmentation of the spinal cord might now be obtained using e.g. deformable models, as is proposed in [12], with the centerline as automatic initialization. But other segmentation schemes that exploit the detection of the cen-

terline can be used as well. Note that the detection of the centerline of the spinal cord also opens possibilities for the detection and segmentation of the vertebrae. Finally, we would like to mention that the rib detection scheme that is described in chapter 6 might benefit as well from knowledge of the location of the spinal cord.

7.2.3 Detection of elongated bones in full-body CT

With the latest generation of CT-scanners, it has become possible to obtain full-body scans at nearly isotropic resolution, see figure 7.8 for an example. The classical procedure of inspection of individual 2D slices becomes infeasible with these large datasets and new visualization methods are needed [119]. In this section we investigate the detection of elongated bones in such datasets.

The procedure that we follow is analogous to the one in chapter 6. The CT-scan is, after down-sampling, thresholded at 130 Hounsfield units and the binary image is blurred. Because the diameters of the bones in the body vary, we have detected the ridges at different scales. From the detected ridge sets again primitives can be constructed. We show two examples in figure 7.9. After the full-body scan has been isotropically be resampled, it is down-sampled with a factor 4. In figure 7.9(a) the convex sets are displayed for a scan that is blurred after thresholding with a scale $\sigma = 2.0$ voxel. In figure 7.9(b) the convex sets of the same scan are shown, but now the thresholded scan is blurred with $\sigma = 4.0$ voxel.

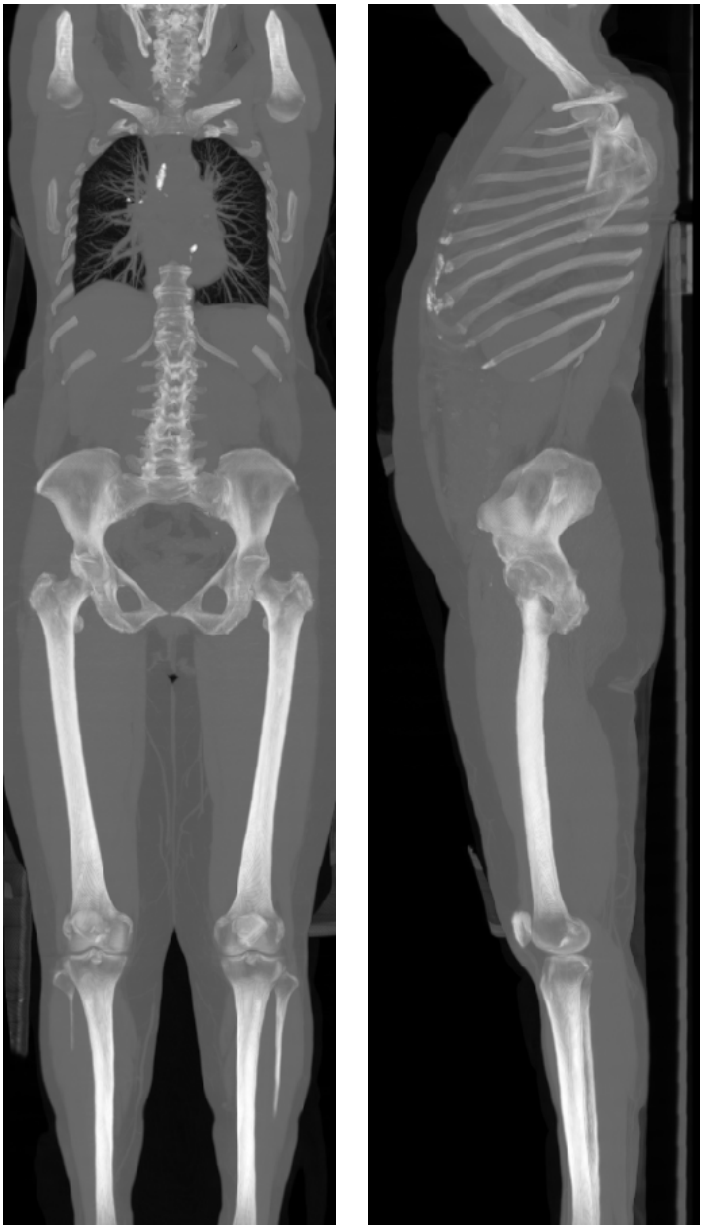
At the finer scale, the rib structure is captured together with the tibia and fibula. At the larger scale, the vertebrae are merged into one elongated structure and the fibula has merged with the tibia. The femur is better detected at the larger scale. The elongated parts of the pelvis are also detected.

If a system can be built that recognizes the individual parts that are detected by this scheme, specialized algorithms can be developed to obtain automatically a full segmentation of the skeleton.

7.3 The road ahead

In chapter 6 all elements that have been developed in this thesis come together to form a coherent and powerful framework for detection, recognition and segmentation. The first three stages of this framework—detection of important image structure, construction of primitives and classicisation of primitives—can be considered to be finished. The fifth stage, full segmentation initialized by the primitives is an interesting area of research, but probably less of an issue than the fourth stage: grouping of the primitives. In our opinion, the challenge is the development of algorithms with respect to this topic.

As is already argued in chapter 6, global information is in general indispensable in the grouping stage. How to include global constraints, however, is still an open question. Issues that need to be solved include: (1) How to incorporate feedback so that missing primitives can be added or erroneous primitives can be



(a) (b)

Figure 7.8 Thin slab maximum intensity projection (MIP) slices of a full-body scan. (a) Coronal slice. (b) Sagittal slice. A thin slab MIP shows the maximum intensity projection of a few slices around a reference slice [129]. The original data set consists of $512 \times 512 \times 1437$ voxels. For the analysis conducted in this section, the volume has been isotropically resampled to $128 \times 128 \times 468$ voxels.

(a) (b)

Figure 7.9 Convex sets of the scan in figure 7.8 at two scales. (a) Convex sets at scale $\sigma = 2.0$ voxel. (b) Convex sets at scale $\sigma = 4.0$ voxel. Before the thresholding and blurring, the scan has been down-sampled to $128 \times 128 \times 486$ voxels.

removed; (2) How to handle variability in the number of primitives that make up one object (e.g. the number of primitives that belong to the same rib); (3) How to handle missing data (e.g. in some CT-scans the primitives of all 24 ribs are detected, while in other less are found).

In the sections below we will introduce some methods and ideas that are related to this subject and that might be an inspiration for continuation of the line of research that is advocated in this thesis.

7.3.1 Matching

Suppose we have an example set of primitives, where each primitive is labeled to a particular class, e.g. left rib 7, left shoulder blade, background, etc. If correspondences can be found between an unlabeled set of primitives and the example set, than grouping and recognition of the primitives is accomplished.

An interesting approach for solving such a matching problem has been put forward by Belongie et al. [8]. In the first stage of their algorithm, the edges in an image are detected. The set of edge locations is subsampled, so that a sparse representation is obtained. Next, for every edge pixel, a 2D shape histogram is computed. The bins of this histogram store the number of neighboring edge pixels at a specific distance in a specific direction.

If two sets of points, say P and Q, are to be matched, a cost matrix $\boldsymbol{C}$ is computed, where the entries $C_{ij} = C(p_i, q_j)$ denote the cost between a point $p_i \in P$ and a point $q_j \in Q$. In [8] the χ^2 test statistic of the shape histograms of p_i and q_j is taken

$$C_{ij} = \frac{1}{2} \sum_{k,l} \frac{\big(h_i(k,l) - h_j(k,l)\big)^2}{h_i(k,l) + h_j(k,l)} \ ,$$

with h_i the shape histogram of p_i and h_j the shape histogram of q_j.

Given $\boldsymbol{C}$, a correspondence can be found by finding the permutation π that minimizes

$$\sum_i C(p_i, q_{\pi(i)}) \ .$$

This is an instance of the weighted bipartite graph matching problem [90], which can be solved efficiently [51]. Note that the matrix $\boldsymbol{C}$ must be square (one-to-one matching), which can be accomplished by adding "dummy" nodes to each point set with constant weighting cost ϵ_{dummy}. In fact, in [8] it is argued that adding dummies to both sets can improve the robustness of the matching procedure. A point which can only be matched to another point if the cost is larger than ϵ_{dummy}, is now matched to a dummy.

The approach sketched above might be extended to match the primitives in two sets. The most straightforward extension is defining shape histograms for every primitive and computing a cost matrix. Note, that the dimension of the shape histograms are equal to the dimensionality of the image. In images with a small amount of primitives, it is questionable whether the cost matrix is well represented by the histograms, because they become sparse. However, instead of

shape histograms, one can also base the entries of the cost matrix on an affinity between two primitives, which can be computed with a classifier, cf. chapter 5 and chapter 6. For this purpose, multiple example sets are needed, for which the correspondence must be known. The input to the classifier would be the features of two primitives (one for each set), and its output a measure for correspondence. Although this alternative is tempting, there is a caveat: for the sets to be matched there must be a one-to-one correspondence. If in one set a specific object is represented by two primitives and in the other by one, then this algorithm will fail to match at least one of the primitives correctly.

7.3.2 Graphs and hierarchy

In the previous section we have shown that the one-to-one correspondence is a constraint that might limit the approach of Belongie et al. [8]. In this section we would like to philosophize on this subject in the context of graphs.

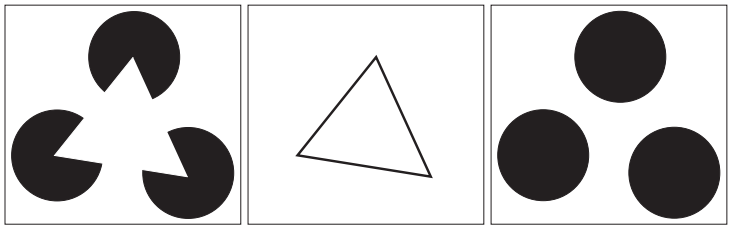
A graph consists of nodes which can be connected with vertices. So, it can be used to model the connectivity (vertices) between primitives (nodes). A special type of graphs are hierarchical graphs. One node in such graphs is denoted as "root" and its vertices connect to children, which may have children again, etc. An example of a root is the skeleton. One of its children is the rib cage, whose children are the individual ribs. Note that hierarchy implies recognition on different levels.

In our opinion, the grouping stage, that is proposed in chapter 6, should produce a hierarchical graph. What is needed to obtain such a graph, is an algorithm that is able to match primitives to the nodes at the "lowest" level, e.g. the rib primitives to the ribs nodes. Such an algorithm must be able to apply many-to-one assignments [108]. Another important ingredient is the definition of a good measure of fit between model and primitives. How to solve this problem in the context of clinical applications is still an open question.

7.3.3 Biologically motivated computer vision

In chapter 1 we mentioned that the functioning of the visual system inspired part of our research. In this last section of the discussion, we would like to relate our work with some computational models of the primary visual cortex that recently have been proposed [66, 70]. These models describe the time-dependent neural activity between cells as a system of coupled differential equations, where every cell sends output to and receives input from its neighboring cells. At this moment, these models take no feedback coupling from higher visual areas into account.

Li [70] shows that her model produces a saliency map of the input image, that highlights edges and ridges. Kokkinos et al. [66] establish a link between their model and variational approaches that are used in the computer vision community [130]. The models in [66, 70] explicitly encode orientation, as the visual system does [45, 133]. The cells are organized in a 3D grid, where two dimensions are for the spatial coordinates and one dimension is for the orientation coordinate. In



(a) (b) (c)

Figure 7.10 The Kanizsa triangle illusion. When we see (a), we fill the triangle as in (b) and complete circles as in (c).

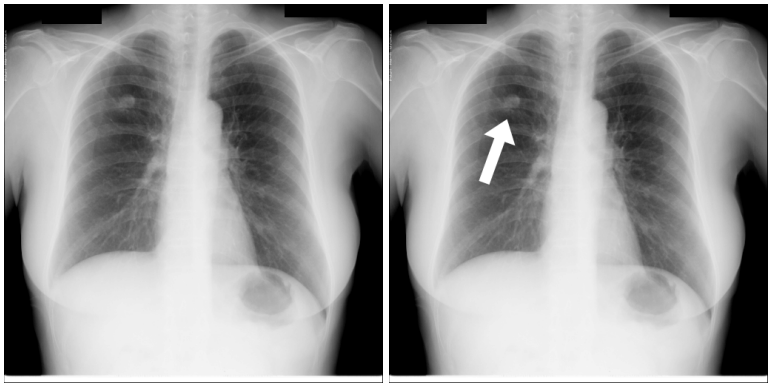
the differential geometric approach for edge and ridge detection (cf. chapter 2) the orientation is implicitly encoded by computing eigenvectors.

Since the cells are coupled, grouping is automatically accounted for, even if "parts" are missing, as in the Kanizsa triangle [27], that is depicted in figure 7.10. The primitive based approach in this thesis might infer relations as in figure 7.10 if in the grouping stage missing data can be handled. However, the models of Kokkinos et al. [66] and Li [70] can fill in and complete missing parts automatically. A grouping algorithm handling such cases can be of great value, e.g. to close gaps that are created when a primitive is incorrectly classified as background.

A big difference between the approach in this thesis and the biologically motivated approach is the use of example data. Training classifiers based on this example data is the key to recognition. At this moment, object recognition is not a part of the models in [66, 70]. Li [70] states that model information might be introduced via feedback coupling on a higher visual processing level that can take care of the recognition task. However, it remains unclear how example data can be exploited to learn the recognition of specific objects.

A human being that for the first sees objects in image, without knowing what they are, will probably be able to sketch a rough outlines. Take e.g. a medical student who observes for the first time an X-ray image of the chest, see figure 7.11(a). Due to some prior knowledge, gained at the university, the student will be able to identify the lung fields and the ribs. If he or she has an understanding of the human anatomy, the clavicles and heart should also not be much of a problem. But even if the observer has no prior knowledge, he or she will be able to draw the outlines of these structures, without recognition. This is essentially the output that can be obtained from the models of Kokkinos and Li.

After the observer has seen a few examples of chest X-rays, he or she will make an internal representation of the objects in the images. With this representation it will be possible for the observer to make a sketch of the structures and tell how they are located with respect to each other. The observer is now able to tell whether an image is upside down or left-right flipped. However, it takes an expert to teach the student that the X-ray in figure 7.11(a) contains a physiological abnormality (there is a nodule in the right lung, see the arrow in figure 7.11(b)). This kind of information is not governed by the biologically motivated models



(a) (b)

Figure 7.11 (a) A chest radiograph. Note that chest X-rays are always displayed as facing the patient: the right lung is displayed on the left and the left lung on the right. (b) The arrow points to a nodule in the right lung.

(yet?), and it would be interesting to see if prior information can be incorporated to solve the recognition problem.

7.4 Summary of the thesis

In this thesis we developed methods to detect, segment and recognize elongated structures in medical images.

Chapter 1 starts with a general introduction on techniques that are employed in this thesis, such as computing derivatives in discrete images, some topics from statistical pattern recognition, and measuring performance of image processing systems.

In chapter 2 a computational method is developed for the segmentation of topological sub-dimensional point sets in scalar images of arbitrary spatial dimensions. The technique is based on calculating the homotopy class defined by the gradient vector in a sub-dimensional neighborhood around every image point. This neighborhood is defined as the linear envelope spawned over a given sub-dimensional vector frame. In the simplest case where the rank of this frame is maximal, a technique is obtained for localizing the critical points, i.e. extrema and saddle points. An important case that is considered is where the frames are formed by the principal directions of the Hessian matrix. The method then segments positive and and negative ridges as well as other types of critical surfaces of different dimensionalities. The signs of the eigenvalues associated to the principal directions provide a natural labeling of the critical sub-sets.

The result in general is a constructive definition of a hierarchy of point sets of different dimensionalities linked by inclusion relations.

Because of its explicit computational nature, the method allows segmentation

of height ridges and edges in different applications. We demonstrate the properties of our construction by presenting two different cases where an extra image coordinate is introduced. In one of the examples we consider image analysis in the framework linear scale-space, where topological properties are gradually simplified through the scale parameter. In the second example, a local orientation parameter is used for segmenting elongated structures.

The output of chapter 2 are point and vector sets (the locations of the topological points and the directions of the sub-dimensional vector frame). In chapter 3 these sets are taken as input for the construction of primitives in the form of affine convex sets. In 2D images, edges and ridges form one-dimensional curves. In 3D images, edges form two-dimensional surfaces. Ridges can be one-dimensional (curves or strings) or two-dimensional (surfaces) depending on the dimensionality of the sub-dimensional vector frame. For the curves in 2D and 3D images, the primitives approximate straight line elements. The surfaces in 3D images are represented by surface patches. The construction of the primitives depends on a few parameters that control the curvature of the primitive and prevent inclusion of points that belong to parallel point sets.

The material that is covered in the first three chapters is applied in chapter 4 for the automatic detection and segmentation of the vasculature in 2D color images of the retina. This method can be used in computer analyses of retinal images, e.g. in automated screening for diabetic retinopathy.

The system is based on extraction of image ridges, which coincide approximately with vessel centerlines. The ridges are used to compose primitives in the form of convex sets. On the basis of the convex sets, an image is partitioned into patches by assigning each image pixel to the closest convex set. Every convex set constitutes a local coordinate frame for its corresponding patch. For every pixel, feature vectors are computed that make use of properties of the patches and the convex sets. A set of features is designed and the feature vectors are classified using a kNN-classifier and sequential forward feature selection. The algorithm is tested on a database consisting of 40 manually labeled images.

The method achieves an area under the ROC-curve of 0.952. The method is compared with two recently published rule-based methods of Hoover et al. [43] and Jiang and Mojon [49]. The results show that our method is significantly better than the two rule-based methods ($P < 0.01$). The accuracy of our method is 0.944 versus 0.947 for a second observer.

With the standard classifiers from statistical pattern recognition it is not obvious how to use features that are defined between primitives. In chapter 5, therefore, an alternative classifier is presented that is able to incorporate such features. The classifier is based upon a spin-glass system, where the image primitives are treated as possessing a spin.

The system is subject to an energy functional consisting of a local and a bilocal part, allowing interaction between the image primitives. Instead of defining the state of lowest energy as the grouping result, the mean state of the system is taken. In this way, instabilities caused by multiple minima in the energy are being avoided. The means of the spins are taken as the *a posteriori* probabilities for the

grouping result.

It is shown how the energy functional can be learned from example data.

The energy functional is defined in such a way, that in case of no interactions between the elements, the means of the spins equal the *a priori* local probabilities. The classifier enables the fusion of the *a priori* local and bilocal probabilities into the *a posteriori* probabilities.

The method is illustrated both on detection of line elements in synthetic images and on vessel detection in images of the human retina.

The topic of chapter 6 is a general framework for automatic detection, recognition and segmentation of objects in three-dimensional medical images. The framework consists of five stages: (1) detection of important image structure, (2) construction of image primitives, (3) classification of the primitives, (4) grouping and recognition of the classified primitives and (5) full segmentation based on the groups. In this chapter, techniques from chapters 1–3 and 5 are employed.

As an instance of this framework, automatic segmentation and labeling of the complete rib cage in chest CT-scans is considered. For this application 1D ridges are extracted from the 3D data. Then, primitives in the form of line elements are constructed from the ridge voxels. Next a spin-glass based classifier is trained to classify the primitives in foreground and background. In the grouping stage centerlines are formed from the foreground primitives and rib numbers are assigned to the centerlines. In the final segmentation stage, the centerlines act as initialization for a seeded region growing algorithm.

The method is tested on 20 CT-scans. Of the primitives, 97.5% is classified correctly (sensitivity is 96.8%, specificity is 97.8%). After grouping, 98.4% of the ribs is recognized.

This thesis ends in chapter 7 with some pilot studies on finger detection in hand radiographs, spine detection in CT-scans and elongated bone detection in full body CT-scans. A discussion follows on the possible extensions of the approach that is taken in this thesis. We conclude from this discussion that the greatest challenge that lies ahead is the development of algorithms that are capable to group primitives into meaningful objects.

Bibliography

[1] E. Aarts and J. Korst. *Simulated annealing and Boltzmann machines*. John Wiley and Sons, 1989.

[2] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(16):641–647, 1994.

[3] D. G. Altman. *Practical statistics for medical research*. Chapman and Hall, London, 1993.

[4] N. Archip, P. J. Erard, M. Egmont-Petersen, J. M. Haefliger, and J. F. Germond. A knowledge-based approach to automatic detection of the spinal cord in CT images. *IEEE Trans. Med. Imag.*, 21(12):1504–1516, 2002.

[5] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[6] S. R. Aylward and E. Bullit. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Trans. Med. Imag.*, 21(2): 61–75, 2002.

[7] S. R. Aylward, E. Bullitt, S. Pizer, and D. Eberly. Intensity ridge and widths for tubular object segmentation and description. In *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 131–138, 1996.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(4):509–522, 2002.

[9] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[10] W. M. Boothby. *An introduction to differential geometry and Riemannian geometry*. Academic Press, New York, 1975.

[11] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[12] S. S. C. Burnett, G. Starkschall, C. W. Stevens, and Z. Liao. A deformable-model appraoch to semi-automatic segmentation of CT images demonstrated by application to the spinal canal. *Med. Phys.*, 31(2):251–263, 2004.

[13] A. Can, H. Seng, J. N. Turner, H. L. Tanenbaum, and B. Roysam. Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. *IEEE Trans. Inform. Technol. Biomed.*, 3(2):125–138, 1999.

[14] B. Caputo and H. Niemann. From Markov random fields to associative memories and back: spin-glass Markov random fields. In *Proc. of IEEE workshop on Statistical and Computational Theories of Vision*, Vancouver, 2001.

[15] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Trans. Med. Imag.*, 8(3):263–269, 1989.

[16] O. Chutatape, L. Zheng, and S. Krishman. Retinal blood vessel detection and tracking by matched Gaussian and Kalman filters. In *Proc. IEEE Int. Conf. Eng. and Bio. Soc.*, volume 20, pages 3144–3149, 1998.

[17] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(6):681–685, 2001.

[18] T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham. Active shape models – their training and application. *Comp. Vision and Image Understanding*, 61(1):38–59, 1995.

[19] J. Damon. Generic structure of two-dimensional images under Gaussian blurring. *SIAM Journal of Applied Mathematics*, 59:97–138, 1998.

[20] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[21] R. O. Duda, P. E. Hart, and H. G. Stork. *Pattern classification*. Wiley-Interscience, New York, 2nd edition, 2001.

[22] D. Eberly. *Ridges in image and data analysis*. Kluwer Academic Publishers, Dordrecht, 1996.

[23] D. Eberly. Skeletonization of 2D binary images. `http://www.magic-software.com/Documentation/Skeletons.pdf`, 2001.

[24] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *J. Math. Imaging and Vision*, 4:351–371, 1994.

[25] T. Eguchi, P. Gilkey, and J. Hanson. Gravitation, gauge theories and differential geometry. *Physics Reports*, 66(2):213–393, 1980.

[26] A. Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Ann. der Physik*, 49: 769–822, 1916.

[27] B. Ernst. *Optical illusions*. Taschen, New York, 1996.

[28] L. M. J. Florack. *Image structure*. Kluwer Academic Press, Dordrecht, 1997.

[29] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Images: regular tempered distributions. In *Proc. of the NATO Advanced Research Workshop Shape in Picture - Mathematical description of shape in grey-level images*, volume 126 of *NATO ASI Series F*, pages 651–660. Springer Verlag, 1994.

[30] A. J. Frame, P. E. Undrill, M. J. Cree, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester. A comparison of computer based classification methods applied to the detection of microaneurysms in ophthalmic fluorescein angiograms. *Computers in Biology and Medicine*, 28(3):225–238, 1998.

[31] K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.

[32] L. Gagnon, M. Lalonde, M. Beaulieu, and M.-C. Boucher. Procedure to detect anatomical structures in optical fundus images. In *Proc SPIE Med. Imaging: Image Processing*, volume 4322, pages 1218–1225, 2001.

[33] L. Gang, O. Chutape, and S. M. Krishan. Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter. *IEEE Trans. Biomed. Eng.*, 49(2):168–172, 2002.

[34] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.

[35] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[36] G. Gross, J. Boone, and D. Bishop. Pediatric skeletal age: determination with neural networks. *Radiology*, 195(3):689–696, 1995.

[37] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *Int. Journ. Comp. Vis.*, 20(1/2):113–133, 1996.

[38] E. J. Hall. *Radiobiology for the radiologist*. Williams and Wilkins/Lippincott, 2000.

[39] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a reciever operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[40] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning*. Springer Verlag, 2001.

[41] S. Haykin. *Neural networks*. Prentice Hall International, London, 2nd edition, 1999.

[42] L. Hérault and R. Horaud. Figure-ground discrimination: a combinatorial optimization approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(9):899–914, 1993.

[43] A. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med. Imag.*, 19(3):203–210, 2000.

[44] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.

[45] D. H. Hubel. *Eye, brain, and vision*. Scientific American Library, New York, 1984.

[46] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture in two non striate visual areas (18 and 19) of the cat. *J. Neurophys.*, 28:229–289, 1965.

[47] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of the monkey striate cortex. *J. Physiol. (London)*, 195:215–243, 1968.

[48] A. K. Jain, R. P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):4–37, 2000.

[49] X. Jiang and D. Mojon. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(1):131–137, 2003.

[50] P. Johansen. On the classification of toppoints in scale-space. *J. Math. Imaging and Vision*, 4(1):57–68, 1994.

[51] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

[52] S. N. Kalitzin, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever. Image segmentation and object recognition by Bayesian grouping. In *Proc. ICIP 2000*, Vancouver, 2000.

[53] S. N. Kalitzin, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever. A computational method for segmenting topological point sets and application to image analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(5):447–459, 2001.

[54] S. N. Kalitzin, B. M. ter Haar Romeny, A. H. Salden, P. F. M. Nacken, and M. A. Viergever. Topological numbers and singularities in scalar images. Scale-space evolution properties. *J. Math. Imaging and Vision*, 9(3):253–269, 1998.

[55] S. N. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. On topological deep-structure segmentation. In *Proc. Intern. Conf. on Image Processing*, pages 863–866, Santa Barbara, California, 1997.

[56] Y. Kang, K. Engelke, and W. A. Kalender. A new accurate and precise 3-D segmentation method for skeletal structures in volumetric CT data. *IEEE Trans. Med. Imag.*, 22(5):586–598, 2003.

[57] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int. J. Comp. Vision*, 1(4):321–331, 1987.

[58] R. Keller. *Generic transistions of relative critical sets in parametrized families with application to image analysis*. PhD thesis, University of North Carolina, Department of Mathematics, Chapel Hill, North Carolina, 1999.

[59] G. N. Khan and D. F. Gillies. Extracting contours by perceptual grouping. *Image and Vision Computing*, 10(2):77–88, 1992.

[60] D. Kim, H. Kim, and H. S. Kang. An object-tracking segmentation method: vertebra and rib segmentation in CT images. In *SPIE Proceedings: Medical Imaging 2002*, volume 4684, pages 1662–1671, 2002.

[61] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[62] J. Kittler and J. Illingworth. Relaxation labelling algorithms—a review. *Image and Vision Computing*, 3(4):206–216, 1985.

[63] D. C. Klonoff and D. M. Schwartz. An economic analysis of interventions for diabetes. *Diabetes Care*, 23(3):390–404, 2000.

[64] J. J. Koenderink. The structure of images. *Biol. Cybern.*, 50(5):363–370, 1984.

[65] R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2/3):271–274, 1998.

[66] I. Kokkinos, R. Deriche, P. Maragos, and O. Faugera. A biologically motivated and computationally tractable model of low and mid-level vision tasks. In *8th European Conference on Computer Vision*, pages 506–517, 2004.

[67] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(2):147–160, 2004.

[68] L. Lam, S. W. Lee, and C. Y. Suen. Thinning methodologies–a comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(9):869–885, 1992.

[69] M. Larsen, J. Godt, N. Larsen, H. Lund-Andersen, A. K. Sjølie, E. Agardh, H. Kalm, M. Grunkin, and D. R. Owens. Automated detection of fundus photographic red lesions in diabetic retinopathy. *Invest. Opht. Vis. Sci.*, 44(2):761–766, 2003.

[70] Z. Li. Computational design and nonlinear dynamics of a recurrent network model of the primary visual cortex. *Neural Computation*, 13(8):1749–1780, 2001.

[71] T. Lindeberg. Scale-space behaviour of local extrema and blobs. *J. Math. Imaging and Vision*, 1(1):65–99, 1992.

[72] T. Lindeberg. *Scale-space theory in computer vision*. Kluwer Academic Publishers, Dordrecht, 1994.

[73] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *Int. J. Comp. Vis.*, 30(2):117–154, 1998.

[74] D. Marr. *Vision*. Freeman and Co., New York, 1982.

[75] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computation vision. *J. American Statistical Association*, 82(397):76–89, 1987.

[76] J. L. Marroquin. A Markovian random field of piecewise straight lines. *Biological Cybern.*, 61:457–465, 1989.

[77] M. E. Martínez-Pérez, A. D. Hughes, A. V. Stanton, S. A. Thom, A. A. Bharath, and K. H. Parker. Scale-space analysis for the characterisation of retinal blood vessels. In C. Taylor and A. Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99*, number 1679 in Lecture Notes in Computer Science, pages 90–97. Springer, 1999.

[78] T. McInerney, G. Hamarneh, M. Shenton, and D. Terzopoulos. Deformable organisms for automatic medical image analysis. *Medical Image Analysis*, 6(3):251–266, 2002.

[79] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Med. Imag. Anal.*, 4(2):73–91, 2000.

[80] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21 (6):1087–1092, 1953.

[81] C. E. Metz. Basic principles of ROC analysis. *Seminars in Nucl. Med.*, 8(4):283–298, 1978.

[82] J. Miller. *Relative critical sets in $\mathbb{R}^n$ and application to image analysis*. PhD thesis, University of North Carolina, Department of Mathematics, Chapel Hill, North Carolina, 1998.

[83] M. Morse and S. S. Cairns. *Critical point theory in global analysis and differential topology*. Academic Press, New York, 1969.

[84] M. Nakahara. *Geometry, topology and physics*. Adam Hilger, New York, 1989.

[85] M. Niemeijer, B. van Ginneken, C. Maas, F.J.A. Beek, and M.A. Viergever. Assessing the skeletal age from a hand radiograph: automating the Tanner-Whitehouse method. In *SPIE Medical Imaging*, volume 5032, pages 1197–1205, 2003.

[86] O. F. Olsen. Multi-scale watershed segmentation. In J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors, *Gaussian Scale-Space Theory*. Kluwer Academic Publishers, Dordrecht, 1997.

[87] A. Pais. *Subtle is the Lord: the science and the life of Albert Einstein*. Oxford University Press, New York, 1982.

[88] K. Palágyi. A 3-subiteration 3D thinning algorithm for extracting medial surfaces. *Patt. Recog. Letters*, 23(6):663–675, 2002.

[89] K. Palágyi and A. Kuba. A 3D 6-subiteration thinning algorithm for extracting medial lines. *Patt. Recog. Letters*, 19(7):613–627, 1998.

[90] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover, Mineola, N. Y., 1998.

[91] P. Parent and S. W. Zucker. Trace inference, curvature consistency and curve detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(8):823–839, 1989.

[92] P. Perona and W. T. Freeman. A factorization approach to grouping. In *ECCV (1)*, volume 1406 of *Lecture Notes in Computer Science*, pages 655–670. Springer, 1998.

[93] T. L. Philips and S. A. Leibel. *Textbook of radiation oncology*. Saunders, 1998.

[94] E. Pietka. Computer-assisted bone age assessment based on features automatically extracted from a hand radiograph. *Computerized Medical Imaging and Graphics*, 19(3): 251–259, 1995.

[95] E. Pietka, A. Gertych, S. Pospiech, F. Cao, H. K. Huang, and V. Gilsanz. Computer-assisted bone age assessment:image preprocessing and epiphyseal/metaphyseal ROI extraction. *IEEE Trans. Med. Imag.*, 20(8):715–729, 2001.

[96] E. Pietka, M. McNitt-Gray, M. Kuo, and H. Huang. Computer-assisted phalangeal analysis in skeletal age assessment. *IEEE Trans. Med. Imag.*, 10(4):616–620, 1991.

[97] M. Pilu and R. B. Fisher. Model-driven grouping and recognition of generic object parts from single images. *Journal of Robotics and Autonomous Systems*, 21:107–122, 1997.

[98] A. Pinz, S. Bernögger, P. Datlinger, and A. Kruger. Mapping the human retina. *IEEE Trans. Med. Imag.*, 17(4):606–619, 1998.

[99] S. M. Pizer, P. T. Fletcher, S. Joshi, A. T., J. Z. Chen, Y. Fridman, D. S. Fritsch, A. Graham Gash, J. M. Glotzer, M. R. Jiroutek, C. Lu, K. E. Muller, G. Tracton, P. Yushkevich, and E. L. Chaney. Deformable m-reps for 3D medical image segmentation. *Int. J. of Computer Vision*, 55(2–3):85–106, 2003.

[100] J. P. W. Pluim. *Mutual information based registration of medical images*. PhD thesis, Utrecht University, The Netherlands, 2001.

[101] M. Prokop and M. Galanski, editors. *Spiral and multislice computed tomography of the body*. Thieme, Stuttgart, 2003.

[102] P. Pudil, J. Novovicová, and J. Kittler. Floating search methods in feature selection. *Patt. Recog. Letters*, 15:1119–1125, 1994.

[103] T. Pun. Electromagnetic models for perceptual grouping. In C. Archibald, editor, *Advances in machine vision: strategies and applications*. World Scientific Publishing Co., 1992.

[104] J. A. Rice. *Mathematical statistics and data analysis*. Duxbury Press, 1995.

[105] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, Cambridge, 1996.

[106] A. Robles-Kelly and E. R. Hancock. Perceptual grouping using eigendecomposition and the EM algorithm. In *Proc. 12th Scand. Conf. Im. Anal.*, pages 214–221, 2001.

[107] R. W. Rodieck. *The first steps in seeing*. Sinauer Associates, Inc., Sunderland, Massachusetts, 1998.

[108] A. E. Roth and M. Sotomayor. *Two-sided matching: a study in game theoretic modeling and analysis*. Cambridge University Press, Cambridge, 1990.

[109] L. Schwartz. *Théorie des distributions*, volume I. Hermann et Cie, Paris, 1957.

[110] L. Schwartz. *Théorie des distributions*, volume II. Hermann et Cie, Paris, 1959.

[111] J. A. Sethian. *Level set methods and fast marching methods evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999.

[112] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):888–905, 2000.

[113] J. G. Simmonds. *A brief on tensor analysis*. Springer-Verlag, 2nd edition, 1995.

[114] C. Sinthanayothin, J. F. Boyce, H. L. Cook, and T. H. Williamson. Automated localisation of the optic disc, fovea and retinal blood vessels from digital colour fundus images. *Br. J. Ophthalmology*, 83:902–910, 1999.

[115] C. Sinthanayothin, J. F. Boyce, T. H. Williamson, H. L. Cook, E. Mensah, S. Lal, and D. Usher. Automated detection of diabetic retinopathy on digital fundus images. *Diabetic Medicine*, 19(2):105–112, 2002.

[116] T. Spencer, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester. An image-processing strategy for the segmentation and quantification of microaneurysms in fluorescein angiograms of the ocular fundus. *Computers and Biomedical Research*, 29 (4):284–302, 1996.

[117] R. Strichartz. *A guide to distribution theory and Fourier transforms*. CRC Press, Boca Raton, 1994.

[118] Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

[119] R. M. Summers. Road maps for advancement of radiologic computer-aided detection in the 21st century. *Radiology*, 229(1):11–13, 2003.

[120] J. Tanner and R. Gibbons. A computerized image analysis system for estimating Tanner-Whitehouse 2 bone age. *Hormone Research*, 42(6):282–287, 1994.

[121] J. M. Tanner, M. J. R. Healy, H. Goldstein, and N. Cameron. *Assessment of skeletal maturity and prediction of adult height (*TW3 *method)*. W. B. Saunders, 2001.

[122] B. M. ter Haar Romeny, editor. *Geometry-driven diffusion in computer vision*. Kluwer Academic Publishers, Dordrecht, 1994.

[123] B. M. ter Haar Romeny. *Front-end vision and multi-scale image analysis*. Kluwer Academic Publishers, Dordrecht, 2003.

[124] Y. A. Tolias and S. M. Panas. A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *IEEE Trans. Med. Imag.*, 17(2):263–273, 1998.

[125] T. Walter and J. C. Klein. Segmentation of color fundus images of the human retina: detection of the optic disc and the vascular tree using morphological techniques. In *Medical Data Analysis*, number 2199 in Lecture Notes in Computer Science, pages 282–287. Springer, 2001.

[126] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *IEEE Int. Conf. Comp. Vis.*, pages 975–982, 1999.

[127] A. W. Whitney. A direct method of non parametric measurement selection. *IEEE Trans. Comput.*, 20(9):1100–1103, 1971.

[128] L. R. Williams and D. W. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.

[129] S. Y. Yen, G. D. Rubin, and S. Napel. Fast sliding thin slab volume visualization. In *Proc. Symp. Volume Visualization*, pages 79–86, 1996.

[130] A. Yuille and D. Geiger. A common framework for image segmentation. *Int. J. of Computer Vision*, 6(3):227–243, 1991.

[131] F. Zana and J. C. Klein. A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform. *IEEE Trans. Med. Imag.*, 18(5):419–428, 1999.

[132] F. Zana and J. C. Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Trans. Image Processing*, 10(7):1010–1019, 2001.

[133] S. Zeki. *A vision of the brain*. Blackwell, Oxford, 1993.

[134] A. H. Zemanian. *Distribution theory and transform analysis; an introduction to generalized functions, with applications*. McGraw-Hill, New York, 1965.

Samenvatting

Dit proefschrift gaat over het automatisch detecteren, segmenteren en herkennen van langgerekte structuren in medische beelden. Segmenteren is het verdelen van een beeld in bijelkaar behorende gebieden. Automatisch betekent dat we de computer het werk willen laten doen. En de langgerekte structuren waar we op doelen, zijn bloedvaten en bepaalde botten.

Na een algemene inleiding in hoofdstuk 1, behandelen we een paar basistechnieken die veelvuldig in het proefschrift worden gebruikt.

Langgerekte structuren in beelden kunnen beschreven worden als de ruggen en valleien in een hoogtekaart. In hoofdstuk 2 ontwikkelen we een algemene methode om deze structuren te detecteren. De representatie in termen van ruggen en valleien stelt ons in staat ons te concentreren op de beeldinformatie waar het in dit proefschrift om draait: langgerekte structuren.

De ruggen en valleien uit hoofdstuk 2 worden in het volgende hoofdstuk gegroepeerd in beeldprimitieven, zogenaamde "affiene convex sets", die bij benadering rechte lijnelementjes vormen. Deze beeldprimitieven zijn het basisingrediënt voor de rest van het proefschrift.

Hoofdstuk 4 heeft als onderwerp de automatische detectie en segmentatie van vaatbomen in tweedimensionale kleurenbeelden van het menselijk netvlies. Hiervoor gebruiken we de technieken die in de voorgaande hoofdstukken behandeld en ontwikkeld zijn. Na detectie van de ruggen en het vormen van primitieven, wordt de hulp ingeroepen van de statistische patroonherkenning. In de statistische patroonherkenning wordt gebruik gemaakt van classificatoren, die aan de hand van eigenschappen van een object een uitspraak doen over het type van het object. Zo onderscheiden mensen bijvoorbeeld een appel van een peer door naar vorm en kleur te kijken. Door eigenschappen van de primitieven te berekenen, kunnen we op soortgelijke wijze een uitspraak doen over welke primitieven bij een bloedvat horen en welke niet. Na de classificatie van de primitieven volgt een tweede stap, waarin elk punt in het beeld geclassificeerd wordt. Vergelijking van het automatische systeem met manueel gesegmenteerde beelden laat zien dat het systeem goed presteert.

De standaardclassificatoren uit de statistische patroonherkenning zijn niet geschikt om tegelijkertijd eigenschappen van objecten zelf en eigenschappen tussen objecten onderling te gebruiken. Daarom wordt in hoofdstuk 5 onderzocht of een zogenaamd spin-glass systeem in staat is om zo'n koppeling tot stand te brengen. Spin-glass systemen worden in de natuurkunde gebruikt om de interactie van kleine magneetjes met een extern magneetveld te beschrijven. Aangezien de

magneetjes zelf ook een magneetveld veroorzaken, is er interactie tussen onderlinge magneetjes. Het blijkt dat we de beeldprimitieven kunnen beschrijven als zulke magneetjes. Als gevolg daarvan wordt de classificatie van primitieven die tot de voorgrond behoren verbeterd. Daarnaast profiteren de "zwakke broeders" (voorgrondprimitieven die zonder dit systeem door een classificator niet tot de voorgrond worden gerekend) door de interactie met hun sterke buren.

In hoofdstuk 6 komt alles samen. Met de ontwikkelde technieken zijn we in staat een algemeen raamwerk te definiëren voor de automatische detectie, herkenning en segmentatie van objecten in (medische) beelden. We onderscheiden vijf stappen: (1) detectie van belangrijke beeldstructuur, (2) constructie van beeldprimitieven, (3) classificatie van de primitieven, (4) groepering en herkenning van de geclassificeerde primitieven en (5) volledige segmentatie gebaseerd op de gevonden groepen.

Als een toepassing van dit raamwerk beschouwen we de automatische segmentatie van ribbenkasten in CT-scans en de herkenning van de afzonderlijke ribben. De resultaten voor deze taken zijn bevredigend.

In het laatste hoofdstuk kijken we naar andere toepassingen, zoals vingerdetectie in röntgenbeelden van handen, ruggemergdetectie in CT-scans en detectie van langgerekte botten in CT-scans van het gehele lichaam. Vervolgens filosoferen we over de mogelijke en nodige uitbreidingen van de gevolgde aanpak in dit proefschrift. Onze belangrijkste conclusie is dat op dit moment de grootste uitdaging ligt in de ontwikkeling van algoritmes die in staat zijn om beeldprimitieven te groeperen in betekenisvolle objecten.

Publications

Papers in international journals

J. Staal, B. van Ginneken, M. A. Viergever, "Detection of elongated structures in 3D data applied to automatic rib segmentation and labeling in CT scans", submitted.

J. Staal, S. N. Kalitzin, M. A. Viergever, "A trained spin-glass model for grouping of image primitives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted.

J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, B. van Ginneken, "Ridge based vessel segmentation in color images of the retina", *IEEE Transactions on Medical Imaging*, 2004, vol. 23, pp. 501–509.

B. van Ginneken, A. F. Frangi, J. Staal, B. M. ter Haar Romeny, M. A. Viergever, "Active shape model segmentation with optimal features", *IEEE Transactions on Medical Imaging*, 2002, vol. 21, pp. 924–933.

S. N. Kalitzin, J. Staal, B. M. ter Haar Romeny, M. A. Viergever, "A computational method for segmenting topological point sets and application to image analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, vol. 23, pp. 447–459.

Papers in conference proceedings

M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, M. D. Abràmoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database", in: *SPIE Medical Imaging*, SPIE, 2004, vol. 5370, pp. 648–656.

J. Staal, B. van Ginneken, M. A. Viergever, "Automatic rib segmentation in CT data", in: *Computer Vision Approaches to Medical Image Analysis and Mathematical Methods in Biomedical Image Analysis*, Springer Verlag, 2004, Lecture Notes in Computer Science, vol. 3117.

J. Staal, S. N. Kalitzin, B. van Ginneken, M. D. Abràmoff, T. Berendschot, M. A. Viergever, "Classifying convex sets for vessel detection in retinal images", in: *International Symposium on Biomedical Imaging*, 2002, pp. 269–272.

B. van Ginneken, A. F. Frangi, J. Staal, B. M. ter Haar Romeny, M. A. Viergever, "A non-linear gray-level appearance model improves active shape model segmentation", in: *IEEE workshop on Mathematical Models in Biomedical Image Analysis*, IEEE Society Press, 2001, pp. 205–212.

S. N. Kalitzin, J. Staal, B. M. ter Haar Romeny, M. A. Viergever, "Image segmentation and object recognition by Bayesian grouping", in: *International Conference on Image Processing, IEEE Signal Processing Society*, 2000, pp. 580–583.

S. N. Kalitzin, J. Staal, B. M. ter Haar Romeny, M. A. Viergever, "Frame-relative critical point sets in image analysis", in: *Computer Analysis of Images and Patterns*, Springer Verlag, 1999, Lecture Notes in Computer Science, vol. 1689, pp. 578–587.

J. Staal, S. N. Kalitzin, B. M. ter Haar Romeny, M. A. Viergever, "Detection of critical structures in scale space", in: *Scale-Space Theories in Computer Vision*, Springer Verlag, 1999, Lecture Notes in Computer Science, vol. 1682, pp. 105–116.

B. M. ter Haar Romeny, B. Titulaer, S. N. Kalitzin, G. Scheffer, F. Broekmans, J. Staal, E. te Velde, "Computer assisted human follicle analysis for fertility prospects with 3D ultrasound", in: *Information Processing in Medical Imaging*, Springer Verlag, 1999, Lecture Notes in Computer Science, vol. 1613, pp. 56–69.

Dankwoord

Dat ik ooit begonnen ben aan een promotie is te danken aan Felix Herrmann, die mijn begeleider was gedurende mijn afstuderen. Onder zijn bezielende leiding begon ik te ontdekken dat onderzoek doen misschien niet gemakkelijk maar wel erg leuk is. Felix is er ook de oorzaak van dat ik uiteindelijk, na een mislukt promotie-avontuur in Delft, in Utrecht terecht kwam. Via Felix kwam ik namelijk in contact met Bart ter Haar Romeny, die toentertijd het scale-space-groepje in Utrecht leidde. Op mijn vraag aan Bart of ik misschien in Utrecht kon promoveren, kwam vrij snel een bevestigend antwoord. Ik werd onderzoeker in opleiding en werkte nauw samen met Stiliyan Kalitzin, die post-doc was op hetzelfde project.

Ik vond het leuk om te promoveren, niet alleen omdat ik daardoor onderzoek kon doen, maar ook omdat ik werkte op het Image Sciences Institute. De sfeer binnen het instituut is eigenlijk altijd goed, en er lopen veel mensen met verschillende achtergronden rond. Ik wil al deze mensen bedanken voor hun steun en raad in de afgelopen jaren.

Mijn onderzoek had niet kunnen gebeuren zonder de inhoudelijke medewerking van een heleboel mensen. Ten eerste natuurlijk Stiliyan, die de basis voor dit proefschrift legde door de technieken in hoofdstuk 2 te ontwikkelen. Ook toen je het ISI verliet en in Heemstede aan de slag ging, bleef je betrokken bij het theoretische gedeelte van dit proefschrift. Daardoor is jouw hand ook nog terug te vinden in hoofdstuk 5.

Een zeer speciale rol heeft Bram van Ginneken gespeeld. Toen ik bij het ISI begon was Bram bijna een jaar bezig met zijn promotie-onderzoek. Als jongste promovendi van Bart trokken wij vrij snel samen op. Vanwege onze niet aflatende kritiek op vrijwel alle zaken, maar voornamelijk waar het de software en hardware betrof, noemde iemand ons ooit Bram en Brom. Omdat alles beter moest (en kon), begon jij met het ontwikkelen van iX (imageXplorer), een programma voor beeldverwerking op de PC. Ik haakte vrij snel in en liftte mee op deze succes story. Na mijn ziekte zat ik zonder begeleiding (Stiliyan werkte in Heemstede en Bart professeerde in Eindhoven) en op een moeilijk punt in mijn onderzoek. Jij was inmiddels gepromoveerd en leidde een groepje van een paar promovendi. Je nam ook mij onder je hoede en zorgde dat het theoretische werk dat ik met Stiliyan had gedaan toegepast werd op klinische data. Hoofdstuk 4 en 6 zijn hier getuige van. Zonder jouw vriendschap en steun betwijfel ik of ik mijn promotie had afgerond.

Bart was de grote inspirator van veel promovendi in Utrecht (en is dat nu

in Eindhoven). Altijd goed gehumeurd, optimistisch en overal mogelijkheden ziend. Zonder jou was ik nooit aan dit onderzoek begonnen en had ik niet al die interessante mensen ontmoet.

Max Viergever, beste promotor, dank je wel voor alle steun en dat je deur altijd voor me heeft opengestaan. Ik ben zeer erkentelijk voor jouw inzet en bemoeienis, zeker toen het project na mijn ziekte nieuw leven ingeblazen moest worden. Ik weet niet of je altijd geloofd hebt dat ik de eindstreep zou halen, maar je hebt nooit je twijfels daarover aan mij geuit.

En dan is er het CAD-groepje waarvan ik de afgelopen jaren deel heb uitgemaakt. Ten eerste wil ik Marco Loog noemen, een medestander in deze zware tijden waarin alles klinisch toegepast moet zijn. Met jou kan ik tenminste praten over theoretische en wiskundige problemen. Gelukkig kunnen we ook over allerlei andere onzin praten. Ik hoop dat het jou en Marleen de Bruijne goed zal gaan in Kopenhagen.

Toen ik Meindert Niemeijer voor de eerste keer zag was ik zwaar onder de indruk over zijn hoeveelheid lichaam (toen ik je ooit lange lummel noemde, werd ik meteen afgestraft met kleine kabouter). Inmiddels ben ik ook onder de indruk van jouw vriendschap, goede humeur en de lol die je in het leven hebt. Je bent nauw betrokken geweest bij hoofdstuk 4 en hebt zonder klagen een hoop retinabeelden handmatig gesegmenteerd.

Ivana Išgum, waarschijnlijk ben je niet bij de promotie, maar een kind krijgen lijkt mij ook veel belangrijker. Jij en Rajko zijn goede vrienden geworden. Ik wens jullie heel veel succes met de baby en het afronden van jullie eigen promotie.

En als we het over kinderen krijgen hebben, dan kom ik automatisch bij Arnold Schilham, die weet hoe je dat aanpakt (heb je daar nou al die paperclips voor nodig?). Als vroege starters hebben wij menig kop koffie gedeeld. Dat goede begin van de dag zal ik gaan missen.

Yulia Arzhaeva, je hebt je onsterfelijk gemaakt met de opmerking: “By the way, your wife, she is very beautiful”. Ik ben het natuurlijk van harte met je eens.

Als laatst van dit groepje: Ingrid Sluimer. Samen met Marco ben je vanaf het begin bij het CAD-groepje geweest en we hebben veel lief en leed gedeeld.

Andere mensen die gelieerd zijn (geweest) aan het ISI en inhoudelijk invloed op mijn werk hebben gehad zijn Wiro Niessen, Luc Florack, Alfons Salden, Robert Maas, Alejandro Frangi en Arjan Kuijper.

De volgende mensen droegen misschien niet veel bij op het inhoudelijke vlak, maar wel aan de goede sfeer op het instituut: Josien Pluim, Theo van Walsum, Gerard van Hoorn, Marjan Marinissen, Julia Schabel, Marleen de Bruijne, Erik Meijering, Clemens Bos, Onno Wink, Wilbert Bartels, Evert-jan Vonken, Estia van Raaij, Manon Kluytmans, Romhild Hoogeveen, Rik Stokking, Thijs van Osch, Jan-Henry Seppenwoolde, Remko van der Weide, Cynthia Jongen, Everine van de Kraats, Silvia Olabarriaga, Rashindra Manniesing, Hans Peeters, Chris Kamphuis, Jacqueline Wermers, Paul Smeets en Marius Staring.

Ik wil vier mensen speciaal bedanken. Ten eerste Josien, die ook gedurende mijn ziekte een trouwe vriendin bleek. Je bent een van de weinige vrouwen die me heeft laten zweten en ja, je rare buitelingen bij het squashen zijn vergeven [100,

zie pag. 160]. Verder wil ik je bedanken voor het opofferen van je orchidee om mijn verstervingstheorie te testen.

Dan Theo, C++-guru, hardloper (je gaat veel te snel voor mij) en vriend. Met jou heb ik veel software mogen ontwikkelen en ik heb veel gehad aan jouw commentaar en inzichten. Dat ik aan iX heb kunnen bijdragen, komt door de basis die door jou is gelegd. Daarnaast ben je een van de meest vriendelijke en bescheiden mensen die ik ken. Ik hoop dat we met jou en Paula nog vaak spelletjes mogen spelen.

Ten derde, Gerard, trouwe vriend en de beste systeembeheerder die ik in mijn tijd op het ISI heb mogen meemaken (en ik heb er heel veel mogen meemaken).

En als vierde Marjan, het zonnetje in huis op het secretariaat (en daarbuiten) en die alles snel voor je regelt.

One of the great virtues of doing a PhD is the possibility to visit conferences, where you meet people who work on similar subjects. I have fond memories of the second Scale-Space Conference at Corfu in 1999. Here I met many of the Danes of the Copenhagen group: Mads Nielsen, Peter Johansen, Ole Fogh Olsen, Jon Sporring, and Kim Steenstrup Pedersen. I believe that Martin (Grumse) Lillholm was there too, but the amount of alcohol that those Danes poured into me has *blurred* my memories a little.

The Danes organized a summer school in 2000, where I met some other wonderful people: Erik Dam and Artur Pece (the only real Viking there).

Finally, at the ECCV in Prague this year I met Kaleem Siddiqi and Maxime Descoteaux from McGill University, Montreal, Canada. Their work and mine are complementing each other and for a while it looked like that I was going to do a post-doc in Montreal. Kaleem offered me a position for which I am still very grateful. I hope you do not mind too much that I took the offer by TNO-TPD.

Naast congressen waren er ook nog cursussen en de leukste daarvan was de patroonherkenningscursus van Bob Duin in Delft. Bob was ook een van de trouwe leden van de STW-gebruikerscommissie. Jouw deur stond altijd voor me open, zelfs als ik precies 48 uur te laat was. De enige consequentie die dat had, was dat je mij de kleur van de vloerbedekking wilde laten kiezen.

Wat de wetenschappelijke bedankjes betreft, wil ik tot slot Michael Abràmoff noemen. Afgezien van Felix heb ik nooit iemand ontmoet die zo intens met zijn werk bezig is en daar duidelijke visies over heeft. Doordat je zeer geïnteresseerd was in mijn methodes voor detectie van langgerekte structuren, betrok je me, samen met Ton Smit en Maria Suttorp, bij het opstarten van een internetbedrijf voor automatische screening van diabetische retinopathie. Helaas heb ik door mijn ziekte niet veel bij kunnen dragen, maar ik heb het een interessante en leerzame ervaring gevonden. Jij bent ook de oorzaak van hoofdstuk 4 en leverde de beelden voor de DRIVE-database.

Promoveren doe je niet alleen, je familie en vrienden promoveren gratis mee. De belangrijkste persoon in mijn leven is Nicole, die mij door de zwaarste periode in mijn leven sleepte toen ik in 2000 door ziekte niet meer in staat was om te werken. Onze relatie is daardoor nog hechter geworden dan zij al was en ik hoop dat we nog heel lang samen mogen zijn, vele reizen gaan maken, lekkere wijnen

drinken en genieten van alle andere leuke dingen in het leven (bij de weg, *jij* bent de allerliefste).

Eva en Edo en (inmiddels) Jens, wat zouden we zonder jullie moeten? Het is jammer dat jullie nu in Noorwegen wonen, maar gelukkig zien we elkaar toch nog vaak.

Liesbeth en Frédéric, jullie wonen gelukkig wat dichterbij. Ik ben blij dat Frédéric me introduceerde bij TNO-TPD.

Niet te vergeten zijn natuurlijk mijn ouders, tante Nel, tante Fred en tante Corrie. Ook Nel was tijdens mijn ziekte een grote steun. In dit rijtje horen Joop en Wien, schoonpâh en stiefschoonmoeder, natuurlijk ook thuis.

Mirjam en Karel hebben we dankzij Edo en Eva ontmoet en sindsdien zijn we een paar beste vrienden rijker. We kunnen altijd logeren op de boerderij, jullie huis verbouwen, met alle beesten spelen (soms mogen we zelfs oppassen!), lekker eten en drinken en onder het genot van een goed glas whisky de wereld verbeteren. We voelen ons bij jullie net zo thuis als in ons eigen huis in Leiden. Jullie zijn de trotse bazen van Pippi en Scooter aan wie ik mijn proefschrift heb opgedragen. Toen ik begon te herstellen van mijn ziekte en nodig het huis uitmoest, mocht ik een paar daagjes bij jullie logeren. De wandelingen op de hei met Pippi en alle vrijpartijen met haar hebben zeker positief aan het herstel bijgedragen. Later kwam Scooter erbij, en als we in Leiden een tuin hadden gehad van het formaat dat Scooter nu heeft, dan had ze misschien wel bij ons gewoond. Voor iedereen die nu nieuwsgierig is geworden naar deze twee dames, zie de foto's hieronder.

Pippi

Scooter

Een heel bijzonder iemand die ik een paar jaar geleden heb ontmoet is Francien. Jij leerde me ruimte vinden, zowel in mijn gewrichten als in mijn hoofd (en daarbuiten). Je weet me altijd (letterlijk en figuurlijk) op het verkeerde been te zetten. De lessen Alexandertechniek die jij me geeft zijn heel waardevol, maar de koppen koffie met Leids gebak van bakker Jacobs zou ik ook niet willen missen.

Annet, voor jou een hoop hongi's. We hebben vele reizen met je gemaakt en hopelijk gaan we er nog veel meer maken. En wie weet kopen we ooit nog een huis met zijn drieën (met hond).

Jacob en Hanneke, het is goed jullie te kennen. Jacob heeft mij vanuit Delft en Schiedam altijd met advies terzijde gestaan op het moment dat ik cruciale beslissingen moest nemen. Jullie hartelijkheid en zorgzaamheid is hartverwarmend. Ik

vind het een eer dat Jacob in de commissie zit. Nicole en ik hopen nog lang van jullie gezelschap te genieten.

Michael en Tanja, ik kende Michael een beetje als die enthousiaste vriendelijke jongen uit Leiden. Ik leerde hem pas echt goed kennen toen ik ziek was. Jullie zijn in die tijd een enorme steun geweest en gaven jullie onvoorwaardelijke vriendschap. Daarvoor ben ik nog steeds heel dankbaar.

Alex en Annamaria ken ik al uit mijn afstudeerperiode in Delft, toen ik altijd met Alex mee mocht rijden vanuit Rotterdam. Ik hoopte, nu jullie ook in Leiden wonen, dat ik weer mee kon rijden naar de TPD. Jammer genoeg vertrekken jullie voor een tijdje naar de Verenigde Staten. We gaan jullie en de kleine Nicole enorm missen.

Bernard en Ingeborg, spelletjes doen met jullie en lekker eten en drinken zijn een welkome afwisseling geweest in de drukke tijden van het promoveren. Bernard, wanneer gaan we een nieuwe Frasersjaal halen en de whiskytour doen?

Toen Eva en Edo naar Noorwegen vertrokken, zijn we met Radboud en Liesbeth na-Groenesteegse-opvang begonnen. Ik vind dat nog steeds een erg goed initiatief.

Bergen beklimmen in plaatjes, zoals ik in dit proefschrift heb gedaan, is een leuke bezigheid, maar echte bergen hebben toch mijn voorkeur. Ik bewaar goede herinneringen aan de tochten die ik met Peter en Jasmien heb mogen maken en ik hoop dat ik (en dan met helm) nog een keertje mee mag.

Dan zijn we tot slot aanbeland bij de paranymphen. De meeste mensen doen zo'n vier tot vijf jaar over hun promotie en hebben dan twee nymphen. Ik vind dat ik er na zeven jaar (acht als je Delft meetelt) wel drie mag hebben. Al lang geleden vroeg ik Robert en Remi of zij deze taak op zich wilde nemen. Sorry dat ik jullie zo lang heb laten wachten. Ik hoop dat er nu eindelijk tijd komt om weer eens muziek met elkaar te maken. De derde nymph, Miriam, vroeg ik later. Miriam, ik heb met jou al vrij snel nadat je bij het ISI kwam een speciale band gehad en ik vond het ook leuk om iemand van het instituut als paranymph te hebben. Wat dat laatste betreft gooide je helaas roet in het eten door halverwege te stoppen met je promotie omdat de invulling van het onderwerp je niet lag. Ik ben blij dat jullie drieën me op 13 oktober gaan ondersteunen.