Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/1832
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: langs@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-79 22. Mai 2003

Automatic Quantification of Destructive Changes Caused by Rheumatoid Arthritis [1]

Georg Langs and Horst Bischof and Philipp L. Peloschek

Abstract

Rheumatoid arthritis is an incurable disease affecting predominantly peripheral joints of the appendicular skeleton. It can lead to severe disabling mutilations and even to the complete destruction of the joints. The accurate and reproducible quantification of the progression of the disease and of the destructive changes caused to the joints is a decisive factor during therapy and during clinical trials. The manual quantification methods are time consuming and lack accuracy as well as reproducibility.

In this thesis an alternative approach to automatically quantify the destructive changes caused by rheumatoid arthritis is proposed. Based on a hand radiograph the positions of the bones and joints are determined by *local linear mapping nets*. They learn the visual appearance as well as the anatomical structure of the hand during a training phase. The ability to learn makes a straightforward transfer of the method to other anatomical structures possible.

Based on the coarse position estimates of the bones, the contour is identified with *active shape models* and *snakes*. When these methods are combined in the *ASM driven snakes* algorithm, a control of *a priori* knowledge utilized during the search for the pathologically changed contour is possible. The resulting description and visual information of the bone contour and its surroundings are used for point wise classification of the contour with respect to the question whether or not individual points are affected by rheumatoid arthritis.

The automatic determination of the extent of erosions in the joint region allows for an accurate and operator independent quantification of the disease progression.

[1]This research has been partly supported by Austrian Science Fund (FWF) under grant P14445-MAT and P14662-INF

Synopsis

An introduction is given in Chapter 1, during which the medical background of the present work is outlined. After this introduction the thesis is roughly divided into 3 parts.

In Chapter 2 an approach to locate joints in hand radiographs is introduced. It is based on *local linear mapping nets* and *Gabor jets*. The approach relies on knowledge about the actual image and the anatomical structure, both learned by the algorithm during a training phase. It is therefore transferable to other anatomical structures in a straightforward way.

Chapter 3 deals with the *active shape model driven snakes* algorithm. It gives an exact localization of the bone contour which is the object of investigation for the quantification of the disease.

Chapter 4 explains ways to extract and classify features from the bone boundary. In order to determine the extent of the destruction, standard classification methods like *neural network classifiers* and *support vector machine classifiers* are applied to the features extracted from the contour description resulting from the ASM driven snakes algorithm.

All Chapters start with an overview of existing methods used giving explanations necessary for understanding as well as references to helpful literature. They proceed with detailed explanations of the individual approaches. All methods have been implemented with MATLAB and the chapters end with experimental results.

Contents

List of Figures

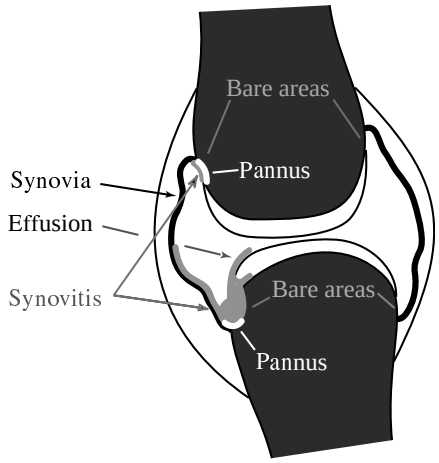


Figure 1.1: Destruction of the joint.

Chapter 1

Introduction

1.1 Rheumatoid Arthritis

Rheumatoid arthritis (RA) is an incurable disease if not treated leading to severe disabling mutilations of synovial joints. RA affects predominantly the synovium of peripheral joints in the appendicular skeleton. It is one of the leading reasons for disability, 17% of the cases of disability among people aged 15 and older are caused by RA. The prevalence is $1\% - 2\%$. In Austria approximately 100000 people suffer from this disease. In approximately 10% of the cases RA leads to death directly or indirectly. A recent study estimated the total cost to the North American economy caused by arthritis and its related effects to amount $ 64 billion.

1.1.1 Progression of the Disease

Although origin and triggers of rheumatoid arthritis are still subject of research, they are assumed to be a combination of endogenous (genetic) and exogenous (infectious) factors. The disease starts with *synovitis*, an inflammation of the *synovia* that coincides with the development of an aggressive effusion.

The synovium develops to a so called *pannus* that undergoes three states. During the *tumor like progression* (TLP) clusters of highly active synoviogenetic cells emerge. This is the phase of the *infiltrative destruction*. During the second phase the TLP-pannus collapses and macrophages appear in a *benign*

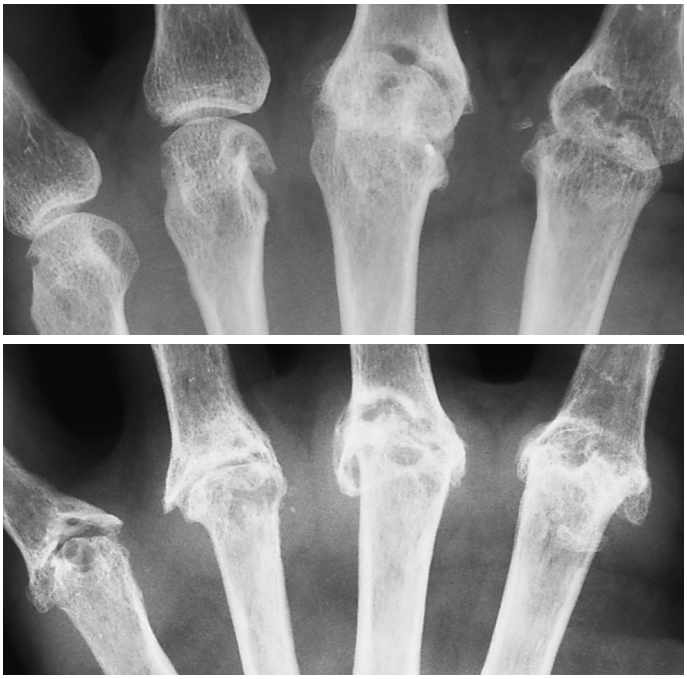


Figure 1.2: Two radiographs of hands affected by RA. The upper image shows a concave erosion on the first finger from the left besides more advanced destructions on the other joints. The lower image shows severe destructions on all joints.

granulation tissue. In the third phase the *scar pannus* is built by *progressive collagensynthesis*. Fig. 1.1 shows a schematic drawing of the joint region.

This development takes place predominantly in the *bare areas*, areas in the joint region that are neither protected by cartilage nor by capsule tissue. The pannus grows on the surface of the cartilage as well as into the cartilage and the bone. It causes destruction in three different ways:

1. Pannus on the surface of the cartilage grows successively over the cartilage surface starting from the bare area. Mainly cartilage is destroyed and among other effects *joint space narrowing* is caused.

2. Pannus in the bone marrow grows through the bare area into the bone and spreads predominantly into the bone marrow. A sub cartilage migration of the pannus is possible too and can lead to separation of the cartilage from the underlying bone.

3. the *forceps-like* destruction is a combination of (1) *chondrale* and (2) *osseous* destruction.

Fig. 1.2 shows two clippings from radiographs of hands affected by RA. The upper image shows a concave clearly visible erosion on the first finger from the left which is the small finger MCP joint. The other joints are in a more severe state of destruction. The lower image shows a hand with almost completely destroyed joints on all fingers.

The *erosions* caused by the pannus affecting the bone substance are subject of radiological quantification based on radiographs. Hand radiographs depicting the wrist and fingers provide information about the progression of the disease. The extent of the erosions i. e. the degree of destruction caused to the joint region of the bone is an indicator for the progression of the disease. Medicines that can slow down or almost stop the progression exist. An effective therapy is possible but depends heavily on information about the temporal development of the destruction.

Please refer to [52, 67] for a detailed explanation of the disease, its progression and established quantification methods.

1.2 The State of the Art: Disease Progression Quantification

As stated above the accurate quantification of the progression of the erosive changes to the bone caused by rheumatoid arthritis is a decisive factor during its treatment and the development of appropriate medicines. The effects of different agents reducing the radiographic progression in rheumatoid arthritis have to be compared across clinical trials [64, 54]. Until now manual quantification procedures are mainly utilized. They are time consuming and lack reproducibility as well as accuracy. Among others these restrictions have severe adverse effects to clinical trials and to continuous therapy of patients. Severe limitations to manual quantification are additionally posed by the following issues:

1. **The choice of the scoring methodology** *Scoring systems* are methodologies that standardize the way information is retrieved from the patient data and its combination to a *score* that describes a present state of the destruction caused to the joints by RA. There exist several scoring systems to assess joint radiographs. These methods evaluate different features, assess different joints and have different scoring ranges. The most widely used methods are the Larsen and Sharp scores [39, 58]. They include information about erosive damage, joint narrowing, a global grading system or comparisons of the radiograph with reference films. Due to the differences in the scoring ranges and in the abnormalities included in the assessment, results obtained with different scores cannot be directly compared.

2. **Reader reliability** During clinical trials one or two observers evaluate each radiograph. Inter observer reliability i. e. the consistency of the scoring results given by the same expert is high. However the absolute scores from various readers may be significantly different. This further complicates the comparison if different readers score across trials.

   The equivalent situation during patient treatment is a change of the therapist. Due to different *reading levels* of different observers the quantification of the disease progression may become inconsistent or at least lack accurate information about the temporal development of RA.

3. **Radiograph scoring sequence** The *follow-up* scoring of radiographs taken at successive time instances for a single patient are an important part of clinical trials.

   It turns out that the order in which radiographs of a sequence are scored by a radiologist has a significant effect on the measurement error of scores and the ability to capture disease progression accurately [65].

   This implies that a method capable of including or excluding knowledge about scoring results obtained from other radiographs of a sequence in a controlled way when scoring a single film is desired.

4. **Subtractive radiology** Digital Subtraction Radiography was introduced in [48]. It is an attempt to make the exact quantification of disease progression directly visible. Radiographs taken at different times are compared by subtracting gray values of overlapping images. The method requires accurate manual adjustment of the images. Due to variations of finger positions readjustment is necessary even for individual bones. This method lacks precision and reproducibility caused by positioning variance. It did not find its way to clinical practice.

5. **Joint space measurement** Relatively early in the history of computational analysis of plain film radiographs the semiautomated measurement of joints space narrowing was introduced. It is based on artificial neural networks [1, 32]. However destructions caused to the bone have to be taken into account in order to provide reliable data.

Part of the above listed issues are due to the time limits posed by clinical practice to the manual quantification task, part of them are inherent in measurements performed by human operators and the consequential character of the scoring systems.

1.3 The Aim of this Thesis

The aim of this thesis is the proposal of a new method to perform the quantification of rheumatoid arthritis. It is aimed at a scoring methodology based on accurate measurements of the destruction caused to the joints and is meant to provide more reliable results by transferring those steps affected by inherent human weaknesses to the computer. As computers provide results the radiologists are able to concentrate on the interpretation of these results and the planning of more effective therapies during their work. We expect advantages in particular in the light of the development of new medicines.

It is not an aim to replace the radiologist with the computer as being in charge for decisions and finally responsible for the therapy, but to provide a significantly better basis for this function.

Chapter 2

Locating Joints in Radiographs

The method to locate joints in radiographs proposed in this thesis (see also [37, 53]) is motivated by natural vision and the behavior of neurons in the primary visual cortex [33]. It is similar to an approach from visual gesture recognition [50] but extends its stability to radiographs.

Reliable features from the visual input are extracted by applying Gabor jets. These responses are further processed by applying an artificial neural net. This is necessary to approximate the nonlinear dependencies between the image content and the positions of the points the algorithm detects, namely the joint positions.

The input to the system is a hand radiograph. No manual annotations are added except that the algorithm does not determine whether the radiograph shows a left or a right hand. This has to be indicated by an operator. A tool be able to perform this task automatically is expected to be simple but it lies outside the scope of this work.

The quantification algorithm discussed in Sec. 4.2 analyzes the *MCP joints* of the hand with respect to the extend of erosions on their visible contour. Therefore the methods in this chapter are described with a focus on these regions. They are transferable to other structures in a straightforward way. Fig. 2.1 gives an overview of the anatomy of the hand. MCP joints are joints between the *metacarpals* and the *proximal phalanges*. *CMC joints* are located on the proximal end of the metacarpals i. e. on the end closer to palm of the hand. The finger tips are also referred to as *DIP joints*.

In order to perform a fine analysis of the contour first the coarse positions of the joints and bones in the hand have to be detected. During experiments we located 12 joints. For each finger i. e. *index finger*, *middle finger*, *ring finger* and *small finger DIP-*, *MCP-* and *CMC* joints were located. This allows for an estimation of the position of the bones that are subject to further refinement by methods described in Sec. 3.3.

2.1 State of the Art

The need to detect the individual position of the fingers in hand posture recognition for tasks like virtual reality and sign language has resulted in various approaches. These approaches deal with the determination of hand postures supported by features like silhouette [69] or by detecting anatomical landmarks like fingertips [68, 42, 59, 51, 50] in monocular images. An exact identification of CMC joint locations in hand/wrist radiographs of patients with mild to moderate rheumatoid arthritis i.e. reproducible joint spacing was proposed [19] as a first step for further investigation of these joints. A limitation of this method is the constraint of non-overlapping joint contours and its high methodological specialization on the properties in specific regions of the hand. It is based on results from an algorithm introduced in [18]. An approach to locate joint spaces with artificial neural network based approaches has been introduced in [4].

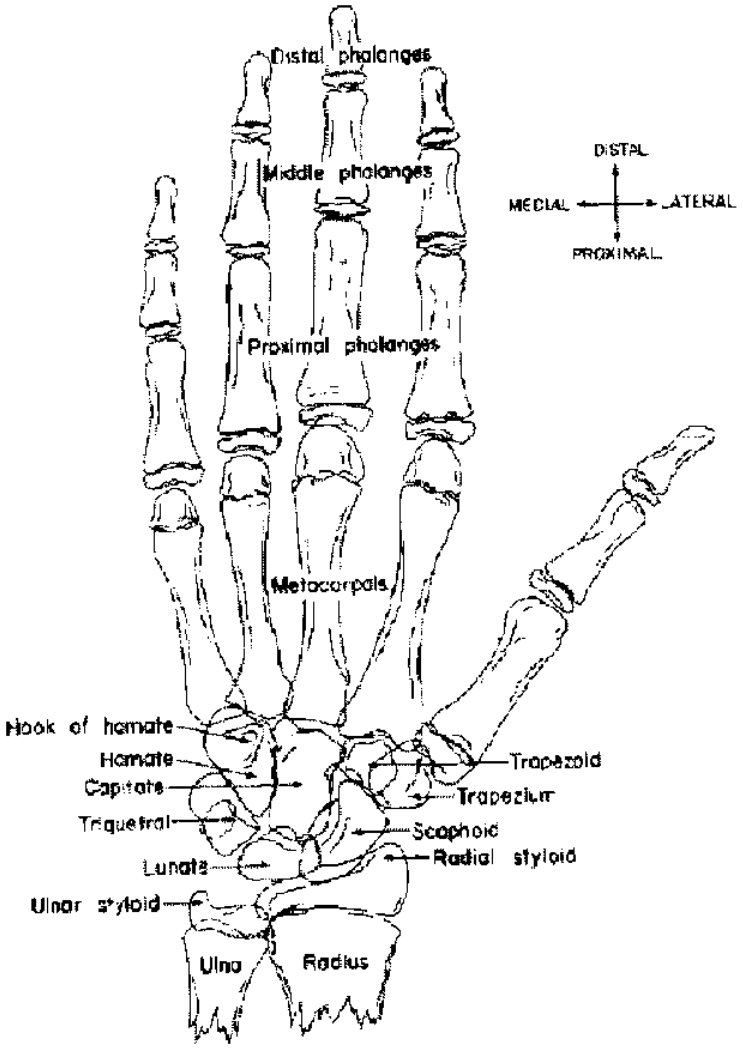


Figure 2.1: Scheme of the hand (from [21]).

2.2 A different Approach

In this chapter a different approach will be proposed. Features are extracted from the input image with sets of Gabor filters that are sensitive to direction. They show behavior similar to structures in the visual cortex of the brain and give a new compact representation of the image content. These features are interpreted by local linear mapping nets (LLM-nets) in order to extract knowledge about joint positions. In addition information about estimates of joint positions are used in an iterative process that pays account to the anatomical structure of the hand.

After an outline of the concepts behind Gabor filters and LLM-nets the algorithm to approximate joint positions from radiographs will be explained in the following chapter.

2.3 Gabor Jets

Gabor jets model the behavior of *hypercolumns*, cell modules in the primary visual cortex [33, 46]. These modules are sensitive to directions and movements for a certain region in the field of view from both eyes. Gabor jets consist of a set of Gabor filters. The *receptive field function* that describes the response of the hypercolumns to a small spot of light as a function of its position, can be fitted well by *Gabor functions* [55].

By splitting a Gabor function into its real and imaginary part two filter kernels are derived, these can extract information about direction and the phase of an input image.

2.3.1 The Gabor function

Definition 2.1 Gabor function *Given the spatial coordinates* (x, y) *the Gabor function is defined by*

$$g(x,y) = \frac{1}{\pi} e^{-(x^2+y^2)+i\pi x}. \tag{2.1}$$

Based on this function a family of functions can be generated by parameterizing the *dilation* by an integer value $i \in \mathbb{Z}$ and rotations by an angle ϕ:

$$g_{j,\phi}(x,y) = \frac{\alpha^{2j}}{\pi} e^{-\alpha^{2j}(x'^2+y'^2)+i\pi\alpha^j x'}, \tag{2.2}$$

where

$$x' = x\cos\phi + y\sin\phi \tag{2.3}$$
$$y' = -x\sin\phi + y\cos\phi. \tag{2.4}$$

The oscillation of the resulting function is due to the harmonic wave factor

$$e^{i\pi\alpha^j x'} \tag{2.5}$$

with a wavelength

$$\lambda_j = \frac{2}{\alpha^j} \tag{2.6}$$

and a wave vector of orientation ϕ and magnitude

$$k_j = \pi\alpha^j. \tag{2.7}$$

The Gaussian factor

$$e^{-\alpha^{2j}(x'^2+y'^2)} \tag{2.8}$$

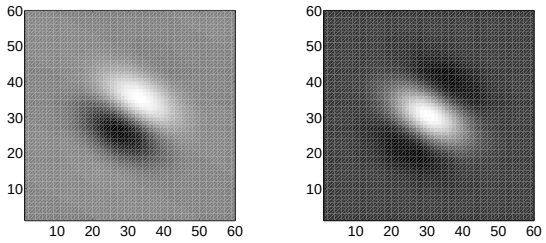


Figure 2.2: The imaginary and real part of a Gabor filter.

causes the function to be negligible for $|(x, y)| > \lambda_j$. In order to split the expression into imaginary and real part two variables are defined:

$$u_0 := \alpha^j \cos\phi \tag{2.9}$$
$$v_0 := \alpha^j \sin\phi \tag{2.10}$$

Thereby the expression is reformulated

$$g_{j,\phi}(x,y) = \frac{\alpha^{2j}}{\pi} e^{-\alpha^{2j}(x'^2+y'^2)+i\pi\alpha^j x'} = \tag{2.11}$$
$$= \frac{\alpha^{2j}}{\pi} e^{-((u_0x+v_0y)^2+(-v_0x+u_0y)^2)+i\pi\alpha^j(u_0x+v_0y)} = \tag{2.12}$$
$$= \frac{\alpha^{2j}}{\pi} e^{-\alpha^{2j}(x^2+y^2)+i\pi\alpha^j(u_0x+v_0y)} = \tag{2.13}$$
$$= \frac{\alpha^{2j}}{\pi} e^{-\alpha^{2j}(x^2+y^2)} e^{i\pi\alpha^j(u_0x+v_0y)} \tag{2.14}$$

and split into real and imaginary part,

$$\alpha^{3j} e^{-\alpha^{2j}(x^2+y^2)} \cos(u_0x + v_0y) + i\alpha^{3j} e^{-\alpha^{2j}(x^2+y^2)} \sin(u_0x + v_0y)). \tag{2.15}$$

Now the Gabor filter kernels

$$g^i_{j,\phi}(x,y) = \alpha^{3j} e^{-\alpha^{2j}(x^2+y^2)} sin(u_0x + v_0y) \tag{2.16}$$
$$g^r_{j,\phi}(x,y) = \alpha^{3j} e^{-\alpha^{2j}(x^2+y^2)} cos(u_0x + v_0y) \tag{2.17}$$

can be defined. Gabor filters are applied as pairs consisting of $g^i_{j,\phi}(x, y)$ and $g^r_{j,\phi}(x, y)$.

2.3.2 Gabor Jets

Gabor jets model the behavior of *hypercolumns*, cell modules in the primary visual cortex. These modules are sensitive to directions and movements for a certain region in the field of view from both eyes [60, 24]. Cells with common orientation selectivity and eye preference are distributed in clusters. By moving horizontally through the cortex cells and gradually changing direction sensitivity can be found (Fig. 2.4). Horizontal connections allow the cells to mediate communication between columns of similar orientation. The spacing between clusters or the width of a *hypercolumn* for orientation is $\approx 750\mu m$. Note that the interconnectivity of these modules is very dense, thus enabling them to integrate contextual information from other cell modules specializing among other things on other visual or auditive stimuli or body position.

The algorithmic counterpart, a *Gabor jet* is built by a set of Gabor filter pairs differing in orientation and frequency applied onto the same part of the image.

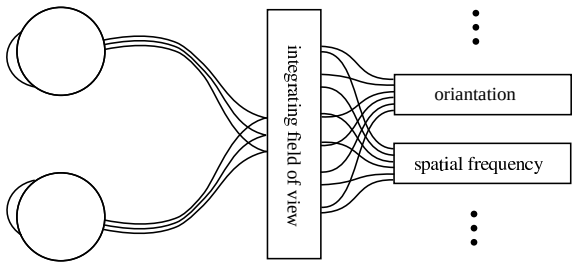


Figure 2.3: Systematic model of the visual system

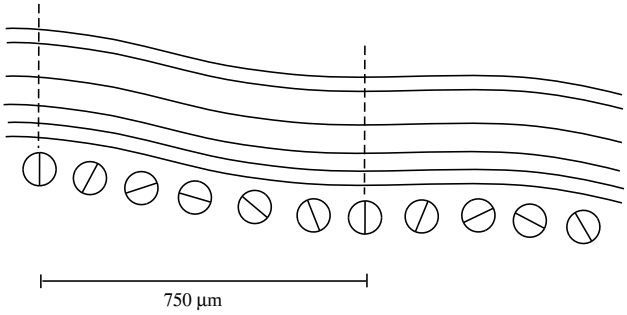


Figure 2.4: Cross section of cortex: cells with orientation preference shifting in a systematic fashion. A full 180° cycle is covered in a distance of approximately $750\mu m$. A segment indicated by dashed lines is referred to as *hypercolumn* for orientation. [24]

Definition 2.2 Gabor Jet *Let g_j^r be the real and g_j^i $j = 1, 2, \ldots, p$ be the imaginary parts of a series of pairs of Gabor filter kernels. Then a Gabor Jet* $\mathbf{G}$ *is defined by*

$$\mathbf{G} =< g_1^r, g_1^i, \ldots, g_p^i > . \tag{2.18}$$

2.3.3 Feature Extraction with Gabor Jets

In order to calculate the response of a Gabor jet $\mathbf{G} =< g_1^r, g_1^i, \ldots, g_p^i >$ from an input image $\mathbf{P}(x, y)$ for each filter kernel pair, the *cross correlation* of $\mathbf{P}$ and $g_j^i(x, y)$ and $g_j^r(x, y)$ is calculated.

Definition 2.3 Continuous Cross Correlation Coefficient *Let $g_1(y_1, y_2)$ and $g_2(y_1, y_2)$ be functions $\mathcal{R}^2 \to \mathcal{R}$. Then the cross correlation coefficient is defined as follows:*

$$r(s) = \frac{\int_{-\infty}^{\infty} g_1(\vec{x}')g_2(\vec{x} - \vec{s})d^2x'}{(\int_{-\infty}^{\infty} g_1^2(\vec{x})d^2\vec{x} \int_{-\infty}^{\infty} g_2^2(\vec{x} - \vec{s})d^2\vec{x})} \tag{2.19}$$

...

Definition 2.4 Discrete Cross Correlation Coefficient *Let $g_1(y_1, y_2)$ and $g_2(y_1, y_2)$ be functions $(1, 2, \ldots, m) \times (1, 2, \ldots, n) \to \mathcal{R}$. Then the cross correlation coefficient is defined as follows:*

$$r(g_1, g_2) = \frac{\sum_{x=1}^{m} \sum_{y=1}^{m} (g_1(x, y) - \bar{g_1})(g_2(x, y) - \bar{g_2})}{((\sum_{x=1}^{m} \sum_{y=1}^{m} (g_1(x, y) - \bar{g_1}))^2 (\sum_{x=1}^{m} \sum_{y=1}^{m} (g_2(x, y) - \bar{g_2}))^2)^{1/2}} \tag{2.20}$$

A feature vector is achieved by calculating cross correlations of the input image and the Gabor filter kernels of the Gabor jet.

$$\mathbf{x}_{feat} = \begin{pmatrix} r(\mathbf{P}, \mathbf{g}_1^r) \\ r(\mathbf{P}, \mathbf{g}_1^i) \\ \vdots \\ r(\mathbf{P}, \mathbf{g}_p^r) \\ r(\mathbf{P}, \mathbf{g}_p^i) \end{pmatrix} \in \mathbb{R}^{2p}. \tag{2.21}$$

By applying one or more Gabor jets of appropriate size on an image $\mathbf{P}$ the initial high dimensionality of the input data can be reduced, while a reasonable amount of information can be preserved. This is dependent on the structure of the data and the choice of Gabor jets.

2.3.4 Choice of Filter Kernels

The selection of filter kernels for each window during the training phase is based on the assumption, that filters with high responses on a patch in the training set provide relevant information about this patch. The algorithm applies filters $\mathbf{g}_{j,\phi}$ with $\phi = 1, 2, \ldots, 180$ on the training patches. The p highest maxima of

$$c_i(\phi) = corr(\mathbf{g}_{j,\phi}, \mathbf{win}_i) \tag{2.22}$$

are used for feature extraction.

2.4 Local Linear Mapping Nets

A local linear mapping net or LLM-net is a smooth mapping from an n-dimensional input space $\mathbb{R}^n$ into a m-dimensional output space $\mathbb{R}^m$.[51]

A nonlinear function $f : \mathbb{R}^n \to \mathbb{R}^m$ is approximated by a set of linear functions $f_i : D_i \to \mathbb{R}^m$ each being valid only on domains $D_i \subset \mathbb{R}^n$ where $\bigcup D_i = \mathbb{R}^n$ i. e. the set of domains forms a partition of the input space $\mathbb{R}^n$.

LLM-nets determine the linear functions f_i as well as the partition of $\mathbb{R}^n$ during a training phase. A set of input units $w_i^{in} \in \mathbb{R}^n$ defines the partition of $\mathbb{R}^n$:

$$\forall D_i : x \in D_i :\Leftrightarrow \forall D_j, i \neq j : \|x - w_i^{in}\|_2 < \|x - w_j^{in}\|_2 \tag{2.23}$$

where $\|\|_2$ is the n-dimensional two-norm. For each input unit w_i^{in} a corresponding output unit $w_i^{out} \in \mathbb{R}^m$ and a matrix $A_i \in \mathbb{R}^{m \times n}$ define the local linear mapping from $\mathbb{R}^m n$ to $\mathbb{R}^m$. The LLM-net has a fixed number n of units:

$$w_i^{in} \in \mathbb{R}^n \tag{2.24}$$
$$w_i^{out} \in \mathbb{R}^m \tag{2.25}$$
$$A_i \in \mathbb{R}^{m \times n} \quad \text{with} \quad i = 1, \ldots, n \tag{2.26}$$

2.4.1 Training of a LLM-net

During the training of the LLM-net based on training examples in the form of pairs

$$(x, y) \in \mathbb{R}^n \times \mathbb{R}^m. \tag{2.27}$$

where $y = f(x)$ an updating process of w_i^{in}, w_i^{out} and A_i is performed.

1. $w_i^{in} \in \mathbb{R}^n$, $w_i^{out} \in \mathbb{R}^m$ and $A_i \in \mathbb{R}^{m \times n}$ are initialized randomly.
2. For each training example (x, y) the domain D_i it lies in is determined by calculating the Euclidean distance of the input vector x to all reference vectors w_i^{in}. The closest reference vector w_s^{in} is chosen according to Eq. 2.23.

$$w_s^{in} \quad \text{with} \quad d_s = min(d_i) \quad \text{where} \quad d_i = \|x - w_i^{in}\|_2 \tag{2.28}$$

3. The reference vectors and the matrix A_s are updated according to

$$y^{net} = w_s^{out} + A_s(x - w_s^{in}), \tag{2.29}$$
$$\Delta w_s^{in} = \epsilon_1(x - w_s^{in}), \tag{2.30}$$
$$\Delta w_s^{out} = \epsilon_2(y - y^{net}) + A_s \Delta w_s^{in}, \tag{2.31}$$
$$\Delta A_s = \epsilon_3(d_s^2)^{-1}(y - y^{net})(x - w_s^{in})^T, \tag{2.32}$$

4. The process is iterated by choosing a new training example (x, y) and proceeding with Step 2.

The learning step sizes $\epsilon_i, (i = 1, 2, 3)$ decrease slowly and the input reference vectors accumulate in regions with high training example density in $\mathbb{R}^n$. The matrices A_i are adapted to the training data.

Note that the distribution of the units is influenced only by the topology of the training examples in the input space. Methods to stay abreast of topology in the output space are discussed in [23].

2.4.2 Approximating a Function with a LLM-net

To apply the trained LLM-net $< w_i^{in}, w_i^{out}, A_i >$ to an input vector $\mathbf{x}_{in}$ the input unit $\mathbf{w}_s^{in}$ with lowest Euclidean distance d_s to the input vector $\mathbf{x}_{in}$ is chosen. The output of the net is determined by

$$\mathbf{y}^{net} = \mathbf{w}_s^{out} + A_s(\mathbf{x}_{in} - \mathbf{w}_s^{in}). \tag{2.33}$$

Example 1

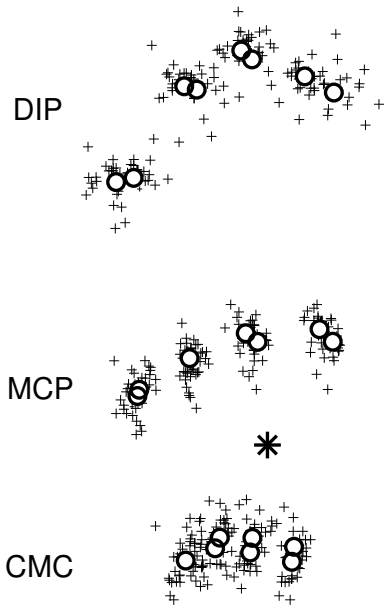


Figure 2.5: Training sets for 12 LLM-nets trained on the tips, MCP- and CMC joints. For each net, pairs of circles represent output units, the star represents the center of gravity.

Let $f : \mathbb{R}^1 \rightarrow \mathbb{R}^1$ be a function with one dimensional input and one dimensional output space. f is approximated by a LLM-net with two units:

$$LLM :< \{\mathbf{w}_1^{in}, \mathbf{w}_2^{in}\}, \{\mathbf{w}_1^{out}, \mathbf{w}_2^{out}\}, \{k_1, k_2\} > \tag{2.34}$$

Fig. 2.6 shows a plot of f and the piecewise approximations by f_1 and f_2. They are defined by

$$f_1(x) = k_1 x + \mathbf{w}_1^{out} \tag{2.35}$$
$$f_2(x) = k_2 x + \mathbf{w}_2^{out} \tag{2.36}$$

For an input $x \in \mathbb{R}$ first

$$d_1 = \|x - \mathbf{w}_1^{in}\|_2 \tag{2.37}$$
$$d_2 = \|x - \mathbf{w}_2^{in}\|_2 \tag{2.38}$$
$$s = i : d_i = min(d_1, d_2) \tag{2.39}$$

are calculated. The output of the net is

$$y = \mathbf{w}_s^{out} + k_s(x - \mathbf{w}_s^{in}). \tag{2.40}$$

2.5 An Algorithm that Locates Joints

As mentioned in the beginning of this chapter the method that locates joints should fulfill the following conditions:

- It should rely on knowledge obtained during a supervised training phase.
- It should locate landmarks $\mathbf{J}_i$ based on visual appearance of the hand radiograph, as well as based on knowledge about anatomical configurations obtained during the training phase.

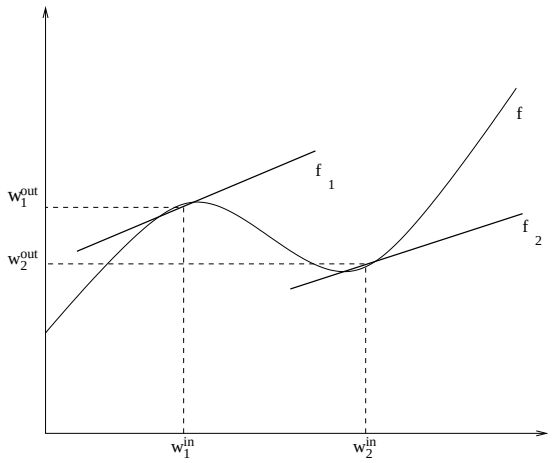


Figure 2.6: Example of an LLM-net with two units approximating a function $f : \mathbb{R}^1 \rightarrow \mathbb{R}^1$

- It should be general in the sense that it should be possible to transfer the method to other anatomical locations in a straightforward way. Therefore specific strategies concerning anatomy should be learned during training.

After a preprocessing step during where radiographs are oriented in order to show the thumb on a defined side the algorithm locates joints in the hand radiograph

$$\mathbf{I}(x, y), \quad x \in \{1, 2, \ldots, m\} \quad y \in \{1, 2, \ldots, mn\} \tag{2.41}$$

in the following way:

1. **Covering the Region of Interest**

   The center of gravity of the gray values in $\mathbf{I}(x, y)$ is determined. Based on this position a configuration of square windows is generated in the image. Fig. 2.7 shows 12 windows covering the hand. Note that the windows are overlapping to a high extent. The basic configuration is chosen before training and has to cover the hand or the region of interest in the image $\mathbf{I}(x, y)$. During experiments a dependency between overlap of the windows in a certain region and the accuracy of the position approximation of points lying within the region was observed. Due to the iterative strategy of the algorithm this has no severe impact on the final localization result.

   The center of gravity provides a coarse estimate of the position of the hand and is therefore used as anchor point for the window configuration. From each of the windows a patch $\mathbf{win}_i$ of the image is extracted:

   $$\mathbf{win}_i(x, y) \subset \mathbf{I}(x, y) \tag{2.42}$$

   Fig. 2.8 shows 12 extracted patches corresponding to the configuration of windows depicted in Fig. 2.7.

2. **Extracting Features from Patches**

   Each of the patches $\mathbf{win}_i$ is subject to a feature extraction procedure as described in Sec. 2.3.3. By applying Gabor jets on the patches a representation of the image content that has lower dimensionality than the original image, but provides relevant information about the joint positions is found. This corresponds to the arrangement of hypercolumns in the cortex where hypercolumns represent regions in the field of view.

   For each patch $\mathbf{win}_i$ a Gabor jet

   $$\mathbf{G}_i =< g^r_{i1}, g^i_{i1}, \ldots, g^i_{ip} > \tag{2.43}$$

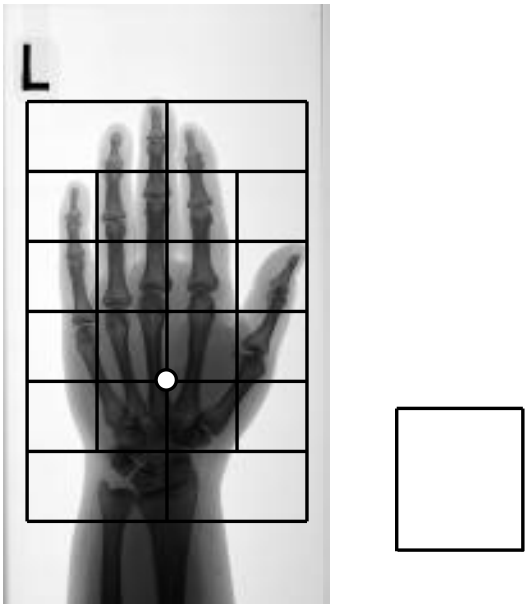


Figure 2.7: Configuration of 12 square windows on which individual Gabor jets are applied. On the right one window is depicted.

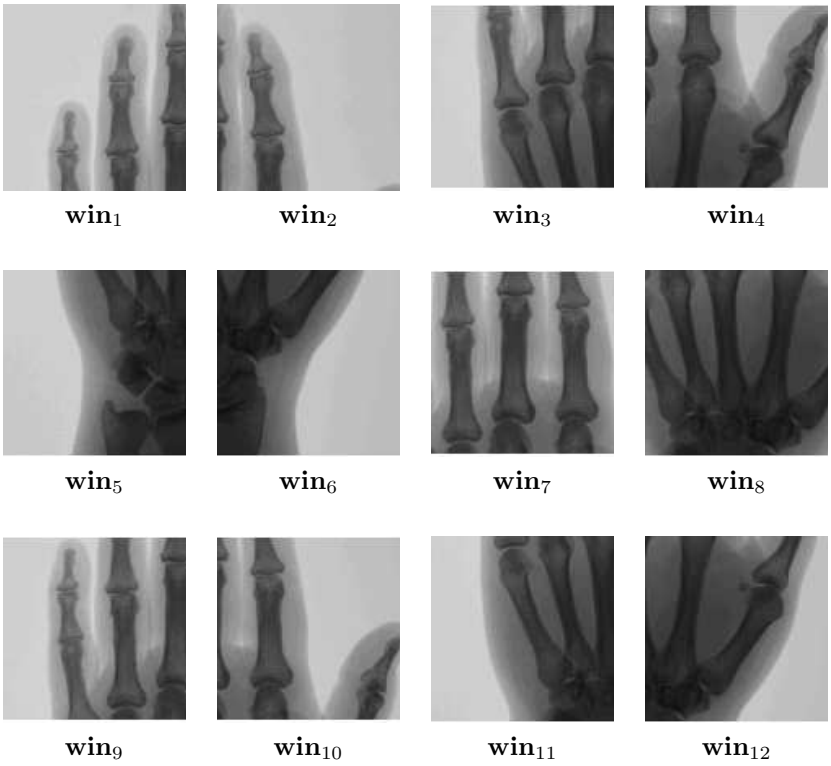


Figure 2.8: Extracted patches.

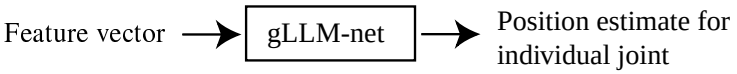


Figure 2.9: Input and output of the gLLM-net.

consisting of p Gabor filter kernel pairs has been selected during the first phase of training (Sec. 2.3.4).

The utilized Gabor jets consist of p Gabor filter kernel pairs $\mathbf{g}^r_{ij}, \mathbf{g}^i_{ij}$, $j = 1, 2, \ldots, p$ and an additional Gaussian filter

$$\mathbf{g}_{gau}(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2.44}$$

They are applied to the set of windows by calculating the cross correlation coefficients (Eq. 2.20) of the Gabor filters kernels and the image patches

$$r(\mathbf{win}_i, \mathbf{g}^{r,i}_{ij}). \tag{2.45}$$

This yields an intermediate feature vector

$$\mathbf{x'}_{feat} = \begin{pmatrix} r(\mathbf{win}_1, \mathbf{g}^r_{11}) \\ r(\mathbf{win}_1, \mathbf{g}^i_{11}) \\ r(\mathbf{win}_1, \mathbf{g}^r_{12}) \\ r(\mathbf{win}_1, \mathbf{g}^i_{12}) \\ \vdots \\ r(\mathbf{win}_w, \mathbf{g}^r_{wp}) \\ r(\mathbf{win}_w, \mathbf{g}^i_{wp}) \\ r(\mathbf{win}_w, \mathbf{g}_{gau}) \end{pmatrix} \in \mathbb{R}^{w(2p+1)}. \tag{2.46}$$

The feature vector $\mathbf{x}_{feat}$ is derived from $\mathbf{x'}_{feat}$ by converting the Cartesian representation of the Gaussian plane i. e. *real* and *imaginary* part of the filter responses to the trigonometric representation i. e. *radius* and *angle*.

3. **Calculating Position Based on Visual Appearance** The feature vector $\mathbf{x}_{feat}$ is input to local linear mapping nets (for simplicity of reading referred to as *gLLM-nets*) ([50]), each of them trained to identify an individual joint location $(x, y)_{J_j}$ from this information (Fig. 2.9). The gLLM-net that approximates the position of $\mathbf{J}_j$ is therefore defined by

$$gLLM_j =< w^{in}_i, w^{out}_i, A_i > \quad , \quad i = 1, 2, \ldots, n \tag{2.47}$$

with

$$w^{in}_i \in \mathbb{R}^{w(2p+1)} \quad , \quad w^{out}_i \in \mathbb{R}^2 \quad , \quad A_i \in \mathbb{R}^{2\times w(2p+1)}. \tag{2.48}$$

The output $\mathbf{y}^{gLLM_j}$ of $gLLM_j$ calculated according to Eq. 2.33 is an estimate of the coordinate vector of the individual joint location $\mathbf{x}_{J_j} = (x, y)_{J_j}$ of joint $\mathbf{J}_j$

$$\hat{\mathbf{x}}^1_{J_j} := \mathbf{y}^{gLLM_j} \in \mathbb{R}^2 \tag{2.49}$$

4. **Calculating Position Based on Anatomical Knowledge**

The hand or an arbitrary anatomical configuration is represented by a graph (Fig. 2.10). In the case of the hand it consists of vertices $\mathbf{J}_i$ with $i \in \{1, 2, \ldots, 12\}$ representing the joints and undirected edges $E_{ij} \in \mathbf{E} \subseteq \{1, 2, \ldots, 12\}^2$ where $E_{ij} \in \mathbf{E} \Rightarrow E_{ij} \in \mathbf{E}$. The edges establish neighborhood relations $N(\mathbf{J}_i) = \{\mathbf{J}_j : E_{ij} \in \mathbf{E}\}$ between landmarks.

In Fig. 2.10 a graph generated manually is depicted. The vertices represent DIP, MCP and CMC on small-, ring-, middle- and index finger. The connecting edges were chosen to correspond to known anatomy i. e. connections by a series of bones or the palm of the hand. The edges connecting $\{\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3, \mathbf{J}_4\}$ do not fit into this criteria, but were selected to compare the improvement of

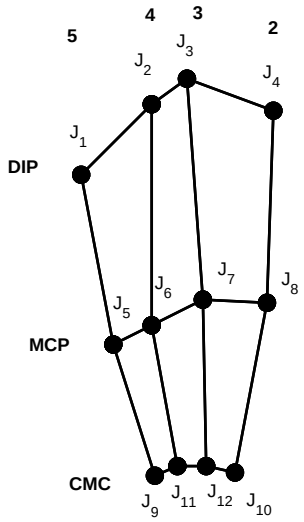


Figure 2.10: Graph Representation of the hand.

accuracy, obtained by this strategy in more rigid regions like the palm of the hand and regions with high flexibility like the finger tips.

This graph can be result of a Delaunay triangulation [40], too. This allows for automatic generation. The refinement of joint positions $\hat{\mathbf{x}}_{J_i}^k$ is based on preceding estimates

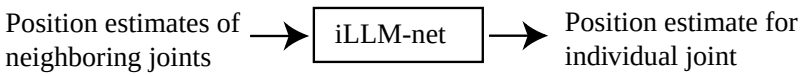


Figure 2.11: Input and output of the gLLM-net.

$$\{\hat{\mathbf{x}}_{J_i}^{k-1} | i \in I\} \tag{2.50}$$

For each joint $\mathbf{J}_i$ the coordinates of the neighbors $\mathbf{x}_{J_j} : \mathbf{J}_j \in N(\mathbf{J}_i)$ are input, the coordinates of the joint $\mathbf{x}_{J_i}$ itself are output of an LLM-net that will be referred to as *iLLM-net*, i. e.

$$\mathbf{x}_{in} := \begin{pmatrix} \hat{\mathbf{x}}_{J_{j_1}}^{k-1}(1) \\ \hat{\mathbf{x}}_{J_{j_1}}^{k-1}(2) \\ \vdots \\ \hat{\mathbf{x}}_{J_{j_n}}^{k-1}(1) \\ \hat{\mathbf{x}}_{J_{j_n}}^{k-1}(2) \end{pmatrix} \quad \begin{pmatrix} \hat{\mathbf{x}}_{J_i}^{k}(1) \\ \hat{\mathbf{x}}_{J_i}^{k}(2) \end{pmatrix} := \mathbf{y}^{net} \tag{2.51}$$

with

$$j_r, r = 1, 2, \ldots, n : \mathbf{J}_{j_r} \in N(\mathbf{J}_i) \tag{2.52}$$

resulting in

$$\{\hat{\mathbf{x}}_{J_i}^{k} | i \in I\}, \tag{2.53}$$

a set of new coordinate estimates for $\mathbf{J}_i, i \in I$. (Fig. 2.11)

Example 2

The position estimates for the joints $J_1, J_2, \ldots, J_{12}$: $\{\hat{\mathbf{x}}_{J_1}^{k-1}, \hat{\mathbf{x}}_{J_2}^{k-1}, \ldots, \hat{\mathbf{x}}_{J_{12}}^{k-1}\}$ are given. Let iLLM be an iLLM-net trained to determine a refined position estimate of the Joint J_6 from the coordinates of its neighbors according to Fig. 2.10:

$$n(J_6) = \{J_2, J_5, J_6, J_7, J_{11}\} \tag{2.54}$$

iLLM has one unit and is defined by

$$w^{in} \in \mathbb{R}^{10} \tag{2.55}$$
$$w^{out} \in \mathbb{R}^{2} \tag{2.56}$$
$$A \in \mathbb{R}^{2\times 10}. \tag{2.57}$$

Where A can be seen as a block matrix containing linear mappings from individual joint positions that are superimposed to calculate the position of J_6.

$$A = \begin{pmatrix} a_2 & b_2 & a_5 & b_5 & a_6 & b_6 & a_7 & b_7 & a_1 1 & b_1 1 \\ c_2 & d_2 & c_5 & d_5 & c_6 & d_6 & c_7 & d_7 & c_1 1 & d_1 1 \end{pmatrix} \tag{2.58}$$

This allows for an update of $\hat{\mathbf{x}}_{J_6}^{k-1}$ to

$$\hat{\mathbf{x}}_{J_6}^{k} = \sum_{i\in\{2,5,6,7,11\}} \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_{J_i}^{k-1}(1) \\ \hat{\mathbf{x}}_{J_i}^{k-1}(2) \end{pmatrix}. \tag{2.59}$$

5. **Iteration**

   In order to increase accuracy, feature extraction and position estimation is iterated with filter configurations on image regions with decreasing size for each joint. They are initiated on the preceding position estimations. This yields higher accuracy, while the algorithm focuses only on interesting areas.

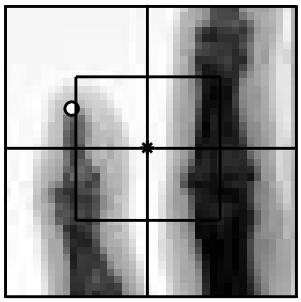


Figure 2.12: Window configuration around the first estimate for DIP5 (finger tip of the small finger) and true position. Star: $\hat{\mathbf{x}}_{J_i}^{k-1}$, Circle: $\mathbf{x}_{J_i}$.

For each joint J_j the preceding position estimate $\hat{\mathbf{x}}_{J_j}^{k-1}$ defines the center of a *search patch* $P_{J_j}^k$ for $\hat{\mathbf{x}}_{J_j}^k$ of size s_k.

$$P_{J_j}^k(\hat{\mathbf{x}}_{J_j}^{k-1}) = \mathbf{I}(x, y) \tag{2.60}$$

with

$$x \in \mathbb{N} \cap [\hat{\mathbf{x}}_{J_j}^i(1) - s_{i+1}/2, \hat{\mathbf{x}}_{J_j}^i(1) + s_{i+1}/2] \tag{2.61}$$
$$y \in \mathbb{N} \cap [\hat{\mathbf{x}}_{J_j}^i(2) - s_{i+1}/2, \hat{\mathbf{x}}_{J_j}^i(2) + s_{i+1}/2] \tag{2.62}$$

From this patch features are extracted by applying Gabor jets on a configuration of 5 windows as depicted in Fig. 2.12.

$$P_{J_j}^k(\hat{\mathbf{x}}_{J_j}^{k-1}) \rightarrow \mathbf{x}_{feat} \tag{2.63}$$

The vector of filter responses $\mathbf{x}_{feat}$ is input to a gLLM-net trained to determine a refinement of the position estimate for the joint.

$$\mathbf{x}_{feat} \rightarrow gLLM^k_{J_i} \rightarrow \Delta_k \hat{\mathbf{x}}_{J_i} \tag{2.64}$$

The set of new estimates

$$\hat{\mathbf{x}}^k_{J_j} = \hat{\mathbf{x}}^{k-1}_{J_j} + \Delta_k \hat{\mathbf{x}}_{J_i} \qquad j = 1, 2, \ldots, 12 \tag{2.65}$$

is input to different iLLM-nets $iLLM^{k+1}_{J_j}$, each trained to determine a refined position from the coordinates of the neighbors of a joint as described in Step 4. The scheme of the entire iteration process is depicted in Fig. 2.13.

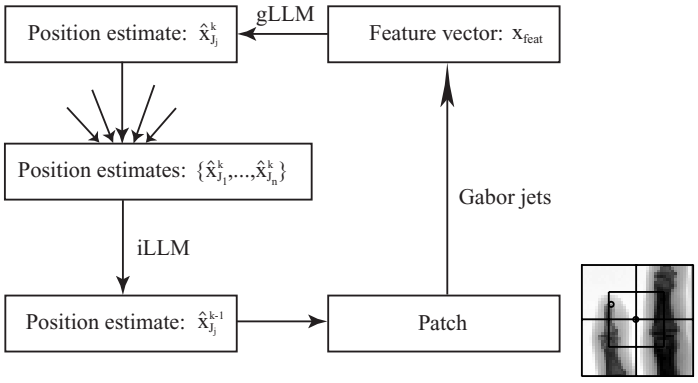
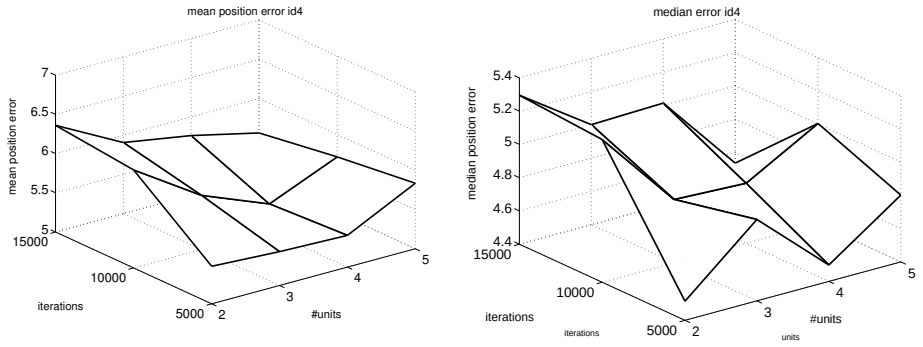


Figure 2.13: Scheme of the iteration.

2.6 Experiments

Experiments were performed on a set of 40 hand radiographs. They were obtained from a PACS (picture archiving and communication system)[62, 30]. The size of the radiographs was reduced to 10% of originally 2617×2047 pixels with 300 dpi resolution. The size of a pixel in the training and test sets is $0.847 \times 0.847mm$ The color depth was 3600 gray values. All radiographs are oriented so that hands appear as depicted in Fig. 2.7 i. e. the thumb is on the right side.



(a) (b)

Figure 2.14: (a) Mean and (b) median position error (pixels) for different LLM-nets with 3 Gabor filter pairs per window.

During the experiments the accuracy of the algorithm was evaluated. In particular the following results were examined:

- Accuracy of $\hat{\mathbf{x}}^1_{J_j}$ i. e. one gLLM-net working with one feature vector based on 12 initial windows (Fig. 2.7)
- Accuracy after iterating the gLLM search with decreasing window size
- Accuracy when iLLM and gLLM-nets were applied at each iteration

Furthermore the third approach was evaluated regarding the differences of improvements achieved by iLLM-nets when used in addition to gLLM-nets on different joints, in particular on DIP and MCP.

The data was divided into a training set containing 30 images and a test set containing the remaining 10 images. Ground truth joints were indicated manually. The performances of LLM-nets derived from training with different numbers of units and iterations were compared. Figure 2.14 shows the mean position error after identifying joint locations with a gLLM-net based on features extracted with Gabor jets consisting of 1 Gaussian filter and 3 Gabor filter pairs for each window. The orientations were chosen according to Eq. 2.22.

Orientation optimization. The optimization of the orientations of Gabor jets with 3 filter pairs reduced the mean position error of $\hat{\mathbf{x}}^1_{J_j}$ $(j = 1, 2, \ldots, 12)$ to 91% of the error achieved with Gabor filters with fixed orientations i. e. 0, 60 and 120 degrees.

The number of units influences the result more significantly than the choice of orientation. Best results for $\hat{\mathbf{x}}^1_{J_j}, j = 1, 2, \ldots, 12$ were achieved with a gLLM-net with 4 units that was trained in 10000 steps on optimized Gabor jets (3 filter pairs). The mean position error is 5.34 pixels. Figure 2.14 shows the mean position error for $\hat{\mathbf{x}}^1_{J_j}, j = 1, 2, \ldots, 12$ for different gLLM-nets and 3 Gabor filter pairs per window.

Iteration. By iterating the gLLM algorithm, results can be improved considerably. On the test set the mean error for $\hat{\mathbf{x}}^2_{J_j}, j = 1, 2, \ldots, 12$ is 4.45 pixels with search patches $P^{i+1}_{J_j}$ of size $s_2 = 40$. The training was performed on a set of 200 artificially generated search patches. They were displaced randomly Gaussian distributed with $\sigma = 8$.

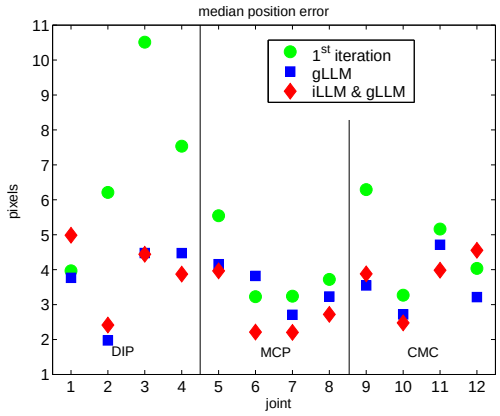


Figure 2.15: Mean error (pixels) for 12 joint locations.

DIP	only gLLM	gLLM&iLLM
Mean Position error:	5.8491	5.9197
Median Position error:	3.3113	3.7331
MCP		
Mean Position error:	3.6101	3.3278
Median Position error:	3.7461	2.5417
Mean X coord error:	1.4769	1.2824
Median X coord error:	0.95283	0.922663
Mean Y coord error:	2.9569	2.7174
Median Y coord error:	2.4391	1.8884
CMC		
Mean Position error:	3.9193	3.677
Median Position error:	3.5482	3.5868

Table 2.1: Mean/median position errors (pixels).

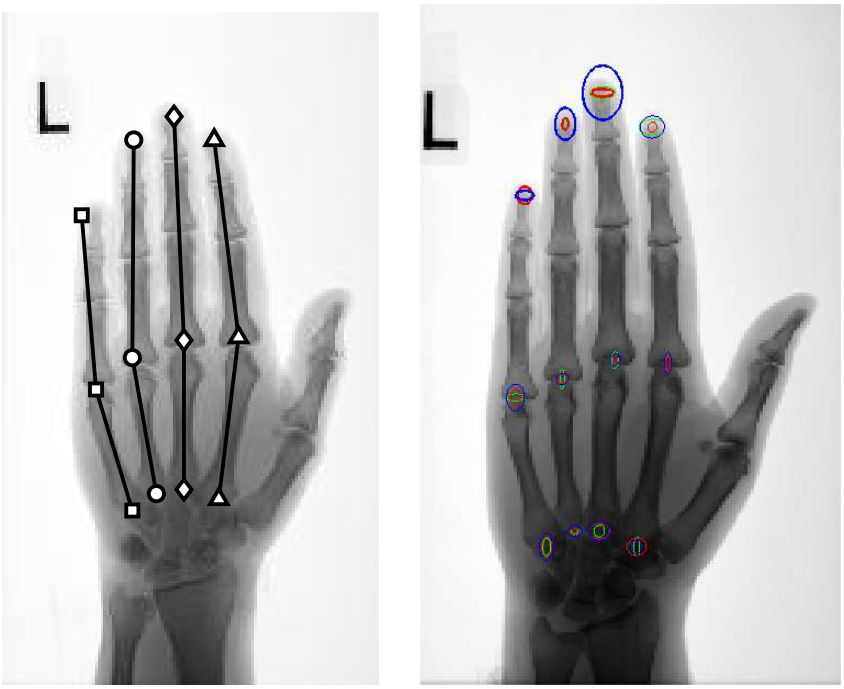
Results on different joints. The enhancement of results achieved by applying *iLLM-nets* at each iteration is significantly different on DIP and MCP joints (Fig. 2.15):

DIP joints: There is no improvement over results without iLLM nets. This is due to the high flexibility of the DIP joints. This means that the degree of dependence between finger tips is not high enough to be reasonably modeled. Due to high contrast the Gabor filter responses provide enough information to detect the finger tips sufficiently accurate. Due to the small training set the accuracy of the approach is lowest on the DIP joints. Note that no positioning aid was used during exposure of the radiograph.

MCP joints: The median error is reduced by 33% with iLLM nets. Lower contrast and soft tissue structures cause poorer performance of gLLM-nets based only on Gabor jets. On the other hand the structure of the MCP joints is more rigid since they are connected by the palm of the hand. The improvement is mainly caused by the reduced position error parallel to the finger (down to 1.88 pixels, Tab. 2.1).

80% of the DIP, 95% of the MCP and 97.5% of CMC joints that are automatically are within a circle with radius of 8 mm around the true positions.

The median position error for $\hat{\mathbf{x}}^1_{J_j}$ for all joints is 5.17 pixels (4.37mm), after two iterations i. e. $\hat{\mathbf{x}}^2_{J_j}$ with gLLMs it is 3.53 pixels (2.98mm), with iLLMs and gLLMs it is 3.28 pixels (2.77mm).



(a) (b)

Figure 2.16: (a) Result of the algorithm: 12 joint positions. (b) Ellipses representing the localization error for (blue) $\hat{\mathbf{x}}^1_{J_j}$, (green) $\hat{\mathbf{x}}^2_{J_j}$ after gLLM and (red) $\hat{\mathbf{x}}^3_{J_j}$ after iLLM and succeeding gLLM localization.

The iLLM-nets create high robustness in the case of poor accuracy of $\hat{\mathbf{x}}^1_{J_j}, j = 1, 2, \ldots, 12$: A mean position error of 11.13 pixels (9.42mm) could be reduced to 6.32 pixels (5.35mm) by gLLM-nets and to 4.71 pixels (3.98mm) by additional iLLM-nets for $\hat{\mathbf{x}}^2_{J_j}$.

Note that the accuracy values are also influenced by the quality of the training as well as the test data. Since the localization of the joints in the radiographs is meant to be an initialization step for more accurate methods like active shape models and snakes no particular emphasis was put in the optimization of the results. A subsequent inspection of the ground truth data revealed that the accuracy corresponds approximately to the manual indication variations. It is likely that accuracy improves with more careful training and higher resolutions. In particular the accuracy improvement obtained by iteration is expected to increase.

Chapter 3

Locating the Contour of the Bone

After an initial coarse localization of the bones in the radiograph a fine determination of the bone contour is the way to go for a detailed analysis with respect to rheumatoid arthritis assessment. The contour is subject to deformations of various forms caused by the disease. State of the art algorithms like active shape models or pixel based segmentation techniques are limited and hinder a direct application. Although the usage of automatically derived a priori knowledge of *active shape models* (ASM) is one of the most important advantages since it makes stable and accurate identification of structures in images possible, it creates difficulties during observation of pathologically changed bones. Active contours on the other hand adapt to new structures that do not appear during the training of the system, but suffer from high sensitivity at initialization. Complex appearance with low contrast and overlapping structures in radiographs avoid a pure pixel based approach with no a priori knowledge about the entity to search for.

In this chapter an approach is proposed which combines active shape models and active contours in order to accurately determine the contour of the bone [38]. Besides improvements of the results of the ASM method it is able to segment even severe erosive changes in the joint regions of the bone. Active shape models and active contours are discussed in detail. Their advantages and shortcomings are highlighted. A method that addresses the problems while preserving the strong capabilities of both approaches is introduced.

3.1 Active Shape Models

Active shape models (ASM) have been introduced in 1992 by [11]. They fit in the context of the discussion about whether the process of perception should be modeled as an active or a passive one. In the context of object or shape recognition this can be related to the importance assigned to *a priori knowledge*. The way to obtain this knowledge is of importance, since the costs of building models by human operators are high.

Active shape models construct a model of the data to be identified during a training phase. Manually annotated examples of shapes are input to an algorithm that derives a model of shape and texture variation automatically.

One usually considers the outline (2D) or the border surface (3D or nD) of an object as the shape. Since we have to deal with shapes in a mathematically comprehensible way the first step of this investigation is restricted to finite sets of points or *landmarks*. Still in Sec. 3.3 the concept is extended to continuous outlines of two-dimensional objects.

Landmarks and forms. The definition of *shape* is based on a set of landmarks i. e. a finite set of points in a vector space. In the context of this work the vector space is the spatial domain of the image $\mathbb{R}^2$. First a **form** is defined as

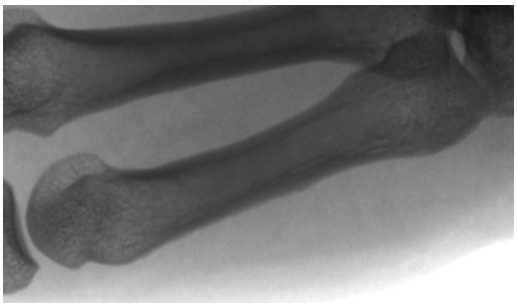


Figure 3.1: Metacarpal bone.

Definition 3.1 (Form) *A vector of points of*

$$\mathbf{x} = (x_1, x_2, \ldots, x_m) \quad \forall i : x_i \in \mathbb{R}^n \tag{3.1}$$

is called a form in $\mathbb{R}^n$.

The m landmarks x_i building a form are identified by their index i and therefore can be matched between different forms. Note that $\mathbf{x}$ contains no information about connectivity between landmarks. A feasible representation of a form in $\mathbb{R}^n$ is a vector built by concatenation of the coordinate vectors of the landmark positions

$$\mathbf{x} = (x_1^1, x_1^2, \ldots, x_1^n, x_2^1, x_2^2, \ldots, x_2^n, \ldots, x_m^1, x_m^2, \ldots, x_m^n) \in \mathbb{R}^{mn} \tag{3.2}$$

Shape and Pose. Throughout the work we will strictly differentiate between *shape* and *pose*.

Definition 3.2 (Shape) *Given a set of points in* $\mathbb{R}^n$

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \tag{3.3}$$

then the shape is defined as the equivalence class $[\mathbf{X}]$ *in the set of point sets with equal cardinality with respect to the relation*

$$\exists T : \quad \mathbf{X} \sim \mathbf{Y} :\Leftrightarrow \mathbf{X} = T(\mathbf{Y}) \tag{3.4}$$

where T *is a similarity transformation.*

Shape is all geometrical information that remains when location, scale and rotational effects i. e. Euclidean similarity transformations are filtered out from the object.

Definition 3.3 (Pose) *The pose of an object is the geometrical information about location, scale and rotational effects.*

These definitions [61, 11] enable us to treat pose and shape of an object separately.

3.1.1 Obtaining Shape

3.1.1.1 Landmarks

Landmarks should be chosen in a way that makes a consistent localization on different images possible i. e. they should be perceptually significant points which are common to all examples. Unfortunately in most cases there are not enough of such optimal landmarks to give a dense definition of an object. A metacarpal bone (Fig. 3.1) does not show significant points on large parts of its outline. The following section will address solutions that lead to a dense sampling of the shapes in the training set. These solutions are reasonably accurate.

One can distinguish roughly between 3 types of landmarks in the context of radiographs.

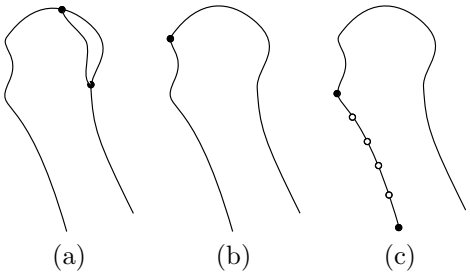


Figure 3.2: (a) Anatomical landmarks, (b) Mathematical landmark, (c) Pseudo landmarks between two landmarks

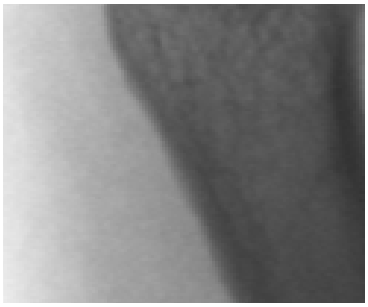


Figure 3.3: Section of a bone contour from a radiograph.

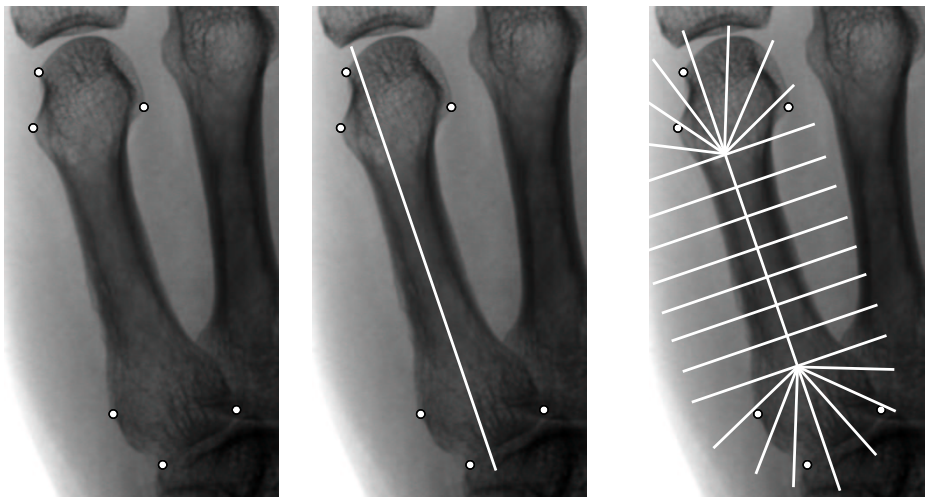
- **Anatomical landmarks** are points in anatomical structures that can be identified clearly by an human expert.
- **Mathematical landmarks** are points lying on the outline of an object that can be described by means of mathematical features like the maximum of curvature or an inflection point.
- **Pseudo landmarks** are points that have been constructed based on other landmarks and information about the object. Sampling the outline of an object between landmarks equidistantly is the most common method to obtain pseudo landmarks.

3.1.1.2 Annotation of Landmarks During Training

During training the positions of landmarks have to be located in the images of the training set. This can be done either manually, semi-automated or automated.

Manual annotation. For manual annotation of landmarks a trained human expert i. e. a expert that is familiar with the anatomy indicates the positions of the landmarks in every image of the training set. This is a very time consuming task. A severe limitation of this method is the lack of accuracy. In Fig. 3.3 a section of a bone contour is depicted. The outline is not visible as sharp edge, but as a smooth transition between 2 gray values. An accurate annotation of landmarks by a human expert is likely to suffer from variations orthogonal to the outline that are not present in the data.

Semi-automated annotation. Several methods have been proposed to facilitate the task of manual landmark annotation. After initial manual training with only a small set of training samples, new shapes can be fitted with the already built ASM and are subject to manual correction. Still if no adjustment is made to the landmarks proposed by the sparse model no additional knowledge is added to the ASM, since the shape is already represented.



(a) (b) (c)

Figure 3.4: Scheme of landmark annotation: (a) 3 anatomical landmarks on proximal and distal joint each, (b) axis through centers of gravity of landmark triplets (c) resulting support grid.

Most of semi-automatic landmarks annotation methods are based on segmentation approaches that do not rely on a priori knowledge. The lack of robustness is compensated by manual adjustment of misplaced landmarks. Most of them require that a densely sampled object outline is given. [57, 21, 20]

Supported manual annotation. An accurate definition of the bone contour is crucial in rheumatoid arthritis assessment. Our approach to obtain a training set of shapes relies on two inputs:

- Highly accurate suggestions for landmarks provided by the algorithm. These suggestions are obtained from the image by a grid overlaid over the bone position (Fig. 3.5) and a search for zero crossings of the second derivate of the gray value distribution on the lines of the grid. In Fig. 3.4 the generation process of the grid is depicted.

  The landmark-candidates are consistently positioned orthogonal to the outline since they lie on the maximum of the gradient image. A consistent position parallel to the bone contour is dependent on the quality of the grid.

- A human expert who chooses from the landmark candidates in order to cope with ambiguities caused by the highly structured image. The time consumption is still high, but besides high accuracy this method provides a convenient and less tiresome way to build training and test sets of shapes during our experiments. Fig. 3.5 shows the finished grid and some candidates presented to the human operator annotating the landmarks.

Based on 6 anatomical landmarks (Fig. 3.4(a)) 3 of which are lying on the proximal joint region and 3 in the distal joint region define an axis by their centers of gravity (Fig. 3.4(b)). Based on this axis a grid is overlaid on the image (Fig. 3.4(c)). Along the lines of the grid local maxima of the image gray level gradient are marked as candidates for landmarks (Fig. 3.6). In order to gain more robustness against noise a Gaussian smoothing kernel is convoluted with the gray level profile.

The expert has to choose from candidates. This procedure can be facilitated by automatically selecting a candidate to be the first suggestion of the algorithm. If an appropriate Gaussian filter kernel has been used, the highest maximum is a reliable estimate of the point where the edge crosses the line. This is correct in most of the cases.

The drawback of this method is its sensitivity to a consistent generation of the grid. Due to the variations of the anatomical landmarks chosen to define the center axis in relation to the entire bone the

Figure 3.5: Grid and candidates during supported manual annotation of landmarks.

compactness of the resulting model (an exact definition is given in Sec. 3.1.2) of shape variation suffers. In Fig. 3.11 deformations represented by the second mode of the shape model are depicted. A considerable part of the variation is caused by movements of landmarks parallel to the bone contour.

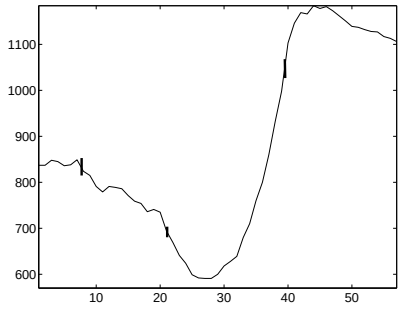


Figure 3.6: Gray values along a line of the grid crossing the bone boundary and some maxima of the first derivate.

3.1.1.3 Alignment of the Landmarks

Given a set of forms, as is the result of the landmark annotation procedure described in the preceding section, the pose i. e. effects of scale, translation and rotation have to be filtered out, in order to obtain a set of shapes according to Def. 3.2. This is subject to the landmark alignment. The pose can be viewed as a common reference coordinate system to which the forms have to be aligned. For each equivalence class $[\mathbf{x}_i]$ a representative has to be found. Thus, if two shapes differ only in pose, they are equal after alignment. The quality of alignment of the forms has severe effects in the resulting model and its compactness. Let us define alignment as follows:

Definition 3.4 (Alignment of forms) *Considering a set of forms*

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \tag{3.5}$$

then a mapping that assigns each form $\mathbf{x}$ *a unique representative of the shape equivalence class* $[\mathbf{x}_i]$*, where* $\mathbf{x} \in [\mathbf{x}_i]$ *, is called an alignment of the set of forms.*

A common approach to perform alignment of this set is by using the *Procrustes distance* [25, 10].

Procrustes shape distance. The Procrustes shape distance of two forms $\mathbf{x}_1$ and $\mathbf{x}_2$ (Fig. 3.7 1^{st} column) is calculated by first determining the centroid for both sets of points: $\mathbf{c}_{\mathbf{x}_1}$ and $\mathbf{c}_{\mathbf{x}_2}$ (Fig. 3.7 2^{nd}

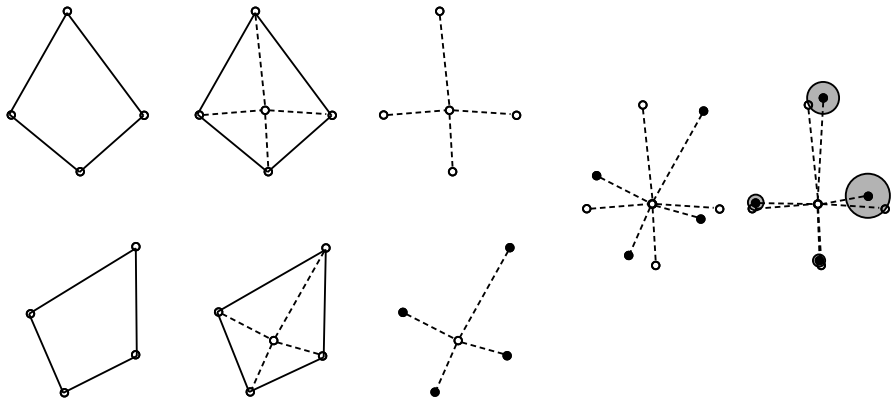


Figure 3.7: Calculation of Procrustes distance.

column). Both sets are scaled so that the sum of squares of distances to its centroid (Fig. 3.7 dashed lines in 3^{rd} column) is of unit length

$$\forall i : \sum_j |\mathbf{x_i}_j - \mathbf{c}_{\mathbf{x}_i}|^2 = 1. \tag{3.6}$$

The forms are translated

$$\mathbf{x}'_1 = T_1\mathbf{x}_1 \tag{3.7}$$
$$\mathbf{x}'_2 = T_2\mathbf{x}_2 \tag{3.8}$$

so that the centroids are identical (Fig. 3.7 4^{th} column) and lie in the origin $c_{\mathbf{x}'_1} = c_{\mathbf{x}'_2} = O$. Finally the sets are rotated

$$\mathbf{x}''_1 = R\mathbf{x}'_1 \tag{3.9}$$
$$\mathbf{x}''_2 = \mathbf{x}'_2 \tag{3.10}$$

so that

$$d_p(\mathbf{x_1}'', \mathbf{x_2}'') = \sum_j |\mathbf{x_1}''_j - \mathbf{x_2}''_j|^2, \tag{3.11}$$

where $\mathbf{x_i}''_j$ denotes the j^{th} point in the i^{th} form, becomes minimal. In Fig. 3.7 5^{th} column, d_p is proportional to the area of the gray circles. Then d_p is the Procrustes distance between the sets of points and RT_i is an alignment according to Def. 3.4.

Original space and shape space. In order to align a set of forms according to the Procrustes distance all shapes have to be transformed by *similarity transformations* (rotation, scale and translation) so that

$$\forall i = 1, \ldots, N : d_p(\mathbf{x_i}, \bar{\mathbf{x}})|^2, \tag{3.12}$$

i. e. the Procrustes distance of all shapes to the mean shape becomes minimal.

[35] explains that only the metric induced by the Procrustes distance is consistent with the relation of the Euclidean distance in the original plane to a distance in shape space. This means that it equals zero if and only if two forms differ only by a similarity transform.

Kendall shape space. This property of the Procrustes distance leads to the definition of the *Kendall shape space* [35]. The set of equivalence classes or shapes becomes a Riemannian manifold i. e. a Riemannian submersion

$$f : \mathbb{R}^{Nn} \to \mathbb{R}^{Nn-N-1-\frac{k(k-1)}{2}} \tag{3.13}$$

called the *Kendall shape space*.

Lemma 3.1 *The Procrustes shape distance* $d_p(\mathbf{x}, \mathbf{y})$ *induces a proper metric* $m_p(\mathbf{x}_1, \mathbf{x}_2)$ *in the Kendall shape space by defining*

$$m_p(\mathbf{x}_1, \mathbf{x}_2) := d_p(\mathbf{x}, \mathbf{y}) \quad \textit{with} \quad \mathbf{x} \in \mathbf{x}_1, \mathbf{y} \in \mathbf{x}_2 \tag{3.14}$$

Proof: A proof is given in [35]. □

Furthermore the Procrustes distance allows the straightforward definition of the mean of a set of shapes [3]. Given a set of equivalence classes the *Fréchet mean* is defined as the equivalence class about which the shapes have the global minimum of summed squared Procrustes distances.

Alignment of a set of forms. The alignment of a set of forms according to the Procrustes analysis is still poorly defined unless the alignment of the mean shape is constraint by some additional criteria. In [10] an iterative method to find a mean shape while aligning a set of forms is proposed:

1. Translate each form so that its center of gravity is at the origin
2. Choose an example as an initial estimate of the mean shape $\bar{\mathbf{x}} = \mathbf{x}_i$ and scale so that
$$|\bar{\mathbf{x}}| = 1. \tag{3.15}$$
3. Set $\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}$
4. Align all forms $\mathbf{x}_i$ to the current estimate of the mean shape $\bar{\mathbf{x}}$ resulting in $\mathbf{x}_i^2 i = 1, \ldots, N$
5. Calculate the mean shape $\bar{\mathbf{x}}_1$ of $\mathbf{x}_i^2, i = 1, \ldots, N$.
6. Align $\bar{\mathbf{x}}_1$ with $\bar{\mathbf{x}}_0$ and scale so that $|\bar{\mathbf{x}}| = 1$.
7. If $|\bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_i|$ is above a certain threshold return to 4.

The tangent space. An alternative to scale normalization given solely by Eq. 3.15 is the transformation of each shape into the *tangent space* $t_{\bar{\mathbf{s}}}$ to the mean. The tangent space $\bar{\mathbf{s}}$ is the hyperplane of vectors normal to $\bar{\mathbf{s}}$ passing through $\bar{\mathbf{s}}$. Hence

$$\mathbf{x} \in t_{\bar{\mathbf{s}}} \Leftrightarrow (\bar{\mathbf{s}} - \mathbf{x}) \cdot \bar{\mathbf{s}} = 0 \tag{3.16}$$

and

$$|\mathbf{x} \cdot \bar{\mathbf{s}}| = 1 \quad \text{if} \quad |\bar{\mathbf{s}}| = 1. \tag{3.17}$$

Alignment is done by first aligning a form to the mean shape by scaling and rotating followed by a projection into the tangent space by scaling $\mathbf{x}$ by $1/|\mathbf{x} \cdot \bar{\mathbf{s}}|$. The advantages of this approach concerning the compactness of the resulting representation are explained in [10].

3.1.2 Modeling Shape

3.1.2.1 Principal Component Analysis

Principle Component Analysis (PCA), a basic form of the *Karhunen-Loève Transform* [66] was introduced in 1933 by Harold Hotelling [61]. It can be used as a compression method for data of arbitrary dimension. By performing an appropriate transform to a coordinate system that adapts to the data distribution, a representation that describes the data in a more compact way is found. This is done by means of representing a high proportion of the variation present in the data within a less dimensional subspace. It is done by rotating the basis of the original space in order to maximize the variance along its basis vectors.

Fig. 3.8 shows a first data distribution in a two dimensional space spanned by the basis vectors $\{\mathbf{x}_1, \mathbf{x}_2\}$. The new basis $\{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2\}$ found by PCA consists of the two *principal components* of the data

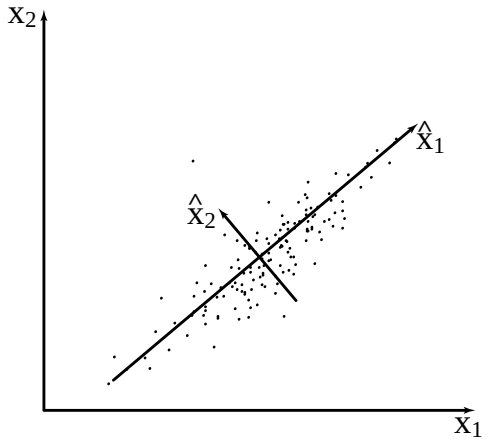


Figure 3.8: PCA as variance maximizing rotation of the basis.

set. Points can be represented only by a part of the new basis e.g. coordinate with respect to $\hat{\mathbf{x}}_1$ while a maximum of data variance is captured in the resulting description.

To describe the *principal component analysis* consider a set of points in an r dimensional space

$$\mathbf{S} = \{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_N\} \subset \mathbb{R}^r. \tag{3.18}$$

To calculate the principal components of $\mathbf{S}$ the centroid of the data has to be determined

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}'_i \tag{3.19}$$

The data is centered by subtracting $\bar{\mathbf{x}}$

$$\forall i : \mathbf{x}_i := \mathbf{x}'_i - \bar{\mathbf{x}} \tag{3.20}$$

resulting in a set of vectors of which the covariance matrix is calculated. Fig. 3.9 shows a covariance matrix calculated from 40 vectors representing the aligned training samples of a shape. Each shape in the training set is built by 42 landmarks. The (x, y) coordinates of the landmarks are concatenated to

$$\mathbf{x}_i = (x_1, x_2, \ldots, x_n, y_1, y_2 \ldots, y_n) \tag{3.21}$$

in order to obtain a vector representation of each shape. White corresponds to high black to negative covariance values between coordinates. The diagonal i. e. $cov(x_i, x_i)$ shows the variance of the individual values in the training set.

Definition 3.5 (Covariance Matrix) *The first-order* covariance matrix $\mathbf{V}$ *of a set of vectors* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^r$ *is defined by*

$$\mathbf{V}_{ij} = \langle (\mathbf{x}_i - \mu_i), (\mathbf{x}_j - \mu_j) \rangle \tag{3.22}$$

where μ_i *is the mean value of the elements in the vector* $\mathbf{x}_i$ *and* $\langle , \rangle$ *denotes the scalar product.*

This is based on the definition of *covariance.*

Definition 3.6 (Covariance) *Given n sets of variates denoted* $\{x_1\}, \ldots, \{x_n\}$ *then the covariance of* x_i *and* x_j *is defined by*

$$cov(x_i, x_j) \equiv \langle (x_i - \mu_i)(x_j - \mu_j) \rangle \tag{3.23}$$

$$\equiv \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle \tag{3.24}$$

where $\langle\ \rangle$ *denotes the* mean *and* μ_i *and* μ_j *are the means of* x_i *and* x_j *respectively.*

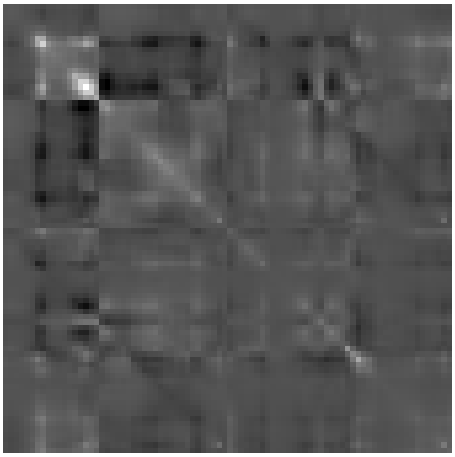


Figure 3.9: Covariance matrix for a set of 40 aligned shapes with 42 landmarks each.

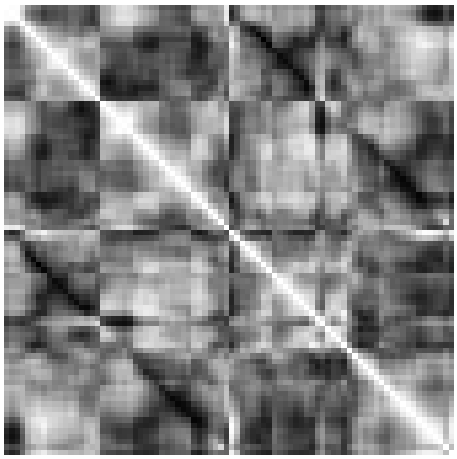


Figure 3.10: Correlation matrix for a set of 40 aligned shapes with 42 landmarks each.

The *covariance* is closely related to the *correlation*.

Definition 3.7 (Correlation) *Given* 2 *variates* x_1 *and* x_2 *then the correlation is defined by*

$$corr(x_1, x_2) :\equiv \frac{cov(x_1, x_2)}{\sigma_{x_1}\sigma_{x_2}} \tag{3.25}$$

where σ_{x_i} *denotes the standard deviation of* x_i *and* $cov(x_1, x_2)$ *denotes the covariance.*

The correlation can be seen as a covariance normalized by the individual variances of the variables. Fig. 3.10 shows a matrix of correlations for the same vector sets used in Fig. 3.9. The correlation gives the strength of the relationship of the variables. If $x_1 \equiv x_2$, $corr(x_1, x_2) = 1$ and for arbitrary x_1, x_2 it holds that

$$-1 \leq corr(x_1, x_2) \leq 1. \tag{3.26}$$

Since the diagonal elements of the covariance matrix are

$$\begin{aligned} \mathbf{V}_{ii} &= \langle (x_i - \mu_i)(x_i - \mu_i) \rangle & (3.27) \\ &= \langle x_i^2 \rangle - \langle x_i \rangle^2 = \sigma_{x_i}^2 & (3.28) \end{aligned}$$

the correlation can also be expressed as

$$corr(x_1, x_2) \equiv \frac{cov(x_1, x_2)}{\sqrt{\mathbf{V}_{ii}\mathbf{V}_{jj}}} \equiv \frac{\mathbf{V}_{ij}}{\sqrt{\mathbf{V}_{ii}\mathbf{V}_{jj}}} \tag{3.29}$$

The principle axes of the set of points $\mathbf{S}$ are defined as the *eigenvectors* or *modes* ϕ_k where $k = 1, 2, \ldots, N$ with corresponding *eigenvalues* λ_k of the covariance matrix $\mathbf{V}$. They are sorted so that $\forall i > j \lambda_i \leq \lambda_j$. The eigenvalue λ_k is equal to the variance of the data in the direction of ϕ_k.

Lemma 3.2 *The coefficients of the new description are uncorrelated.*

Proof: A proof is given in [66]. □

If

$$\mathbf{\Phi} = [\phi_1 | \phi_2 | \ldots | \phi_N] \tag{3.30}$$

is the matrix built column wise by the eigenvectors, then the data can be represented with respect to the new coordinate system built by the eigenvectors with the origin in the centroid of the data by a coordinate transform:

$$\hat{\mathbf{x}} = \mathbf{\Phi}^\top (\mathbf{x} - \bar{\mathbf{x}}) \tag{3.31}$$

The coordinates $\hat{\mathbf{x}}$ are often referred to as the parameters or coefficients of an instance $\mathbf{x}$ with respect to the model. To calculate or *reconstruct* the instance with respect to the original basis

$$\mathbf{x} = \mathbf{\Phi}(\hat{\mathbf{x}}) + \bar{\mathbf{x}} \tag{3.32}$$

has to be performed. This gives the possibility to reconstruct the original coordinates based only on a subset of parameters i. e.

$$\mathbf{x} \approx \tilde{\mathbf{x}} = \mathbf{\Phi}_K(\hat{\mathbf{x}}) + \bar{\mathbf{x}} \tag{3.33}$$

where

$$\mathbf{\Phi}_K = [\phi_1 | \phi_2 | \ldots | \phi_K] \quad \text{with} \quad K < N. \tag{3.34}$$

In general an *instance* of the model can be constructed with respect to a parameter vector $\mathbf{b}$.

Definition 3.8 (Instance) *If* $\mathbf{\Phi}_K = [\phi_1 | \phi_2 | \ldots | \phi_K]$ *is a matrix built by* K *eigenvectors of the covariance matrix of a training set* $\{\mathbf{x}_i | i = 1, \ldots, N\}$ *with mean* $\bar{\mathbf{x}}$ *then*

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}_K \mathbf{b} \tag{3.35}$$

is called an instance of the model $\mathbf{\Phi}_K$*, where* $\mathbf{b} \in \mathbb{R}^K$*,*

Since the variance of the data along ϕ_k is given by λ_k the fraction of the variance represented by $\mathbf{\Phi}_K$ is

$$\frac{\sum_{k=1}^{K} \lambda_k}{\sum_{k=1}^{N} \lambda_k} \tag{3.36}$$

K is called the number of *modes* of the model built by $\mathbf{\Phi}_K$ and $\bar{\mathbf{x}}$. The dependencies between the number of modes that is equal to the dimension of the subspace spanned by the used eigenvectors, the dimension of the original space and properties of the training set i. e. the set of points in the original space, will be explained in detail later. Since for some applications the number of training samples is likely to be far lower than the dimensionality of the original space a method to efficiently retrieve the desired eigenspace under such circumstances is introduced briefly in the following paragraph.

Applying PCA to small training sets. If PCA is applied to a set of vectors

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \quad \text{with} \quad \mathbf{x}_i \in \mathbb{R}^n \tag{3.37}$$

with $N < n$, then only N eigenvalues are nonzero. Therefore it makes sense to perform the eigenvalue decomposition not on the $n \times n$ covariance matrix but on a smaller $N \times N$ matrix derived from the data. The nonzero eigenvalues as well as the corresponding eigenvectors of the $n \times n$ covariance matrix can be retrieved easily from the eigenvalues and eigenvectors of the small matrix. This can save significant amount of computing time, since the complexity of the eigenvalue decomposition increases with the cube of the size of the matrix.

Theorem 3.1 *Given vectors $\mathbf{x}_i$ with $i = 1, 2, \ldots, N$ and a matrix $\mathbf{D}$ consisting of the centered vectors*

$$\mathbf{D} = [(\mathbf{x}_1 - \bar{\mathbf{x}})|(\mathbf{x}_2 - \bar{\mathbf{x}})| \ldots |(\mathbf{x}_N - \bar{\mathbf{x}})] \tag{3.38}$$

where $\bar{\mathbf{x}}$ denotes the centroid of $\{\mathbf{x}_i | i = 1, 2, \ldots, N\}$. Then the covariance matrix can be written as

$$\mathbf{V} = \frac{1}{N}\mathbf{D}\mathbf{D}^\top \in \mathbb{R}^{n \times n}. \tag{3.39}$$

and for the matrix

$$\mathbf{T} = \frac{1}{N}\mathbf{D}^\top\mathbf{D} \in \mathbb{R}^{N \times N} \tag{3.40}$$

the eigenvectors $\mathbf{e}'_i$ and eigenvalues λ'_i ordered so that $\forall i = 1, 2, \ldots, (N-1) : \lambda_i > \lambda_{i+1}$ are related to the accordingly ordered eigenvectors $\mathbf{e}_i$ and eigenvalues λ_i of $\mathbf{V}$ in the following way:

$$\forall i : \mathbf{e}_i = \mathbf{D}\mathbf{e}'_i \quad \textit{and} \quad \lambda_i = \lambda'_i \tag{3.41}$$

for $j > N : \lambda_j = 0$ [36].

Proof: The proof is divided into two parts, the first part shows that Eq. 3.39 is a direct consequence of Def. 3.5, then the more complex proposition in Eq. 3.41 is proven.

1. For each element of the covariance matrix

$$\mathbf{V}_{ij} = \frac{1}{N}\langle(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_j)\rangle \tag{3.42}$$

   according to Def. 3.5. The same holds for $\frac{1}{N}\mathbf{D}\mathbf{D}^\top$ following basic matrix multiplication rules.

2. The validity of Eq. 3.41 still needs to be proven. Consider the eigenvectors $\mathbf{e}'_i$ of $\mathbf{D}^\top\mathbf{D}$, then

$$\mathbf{D}^\top\mathbf{D}\mathbf{e}'_i = \lambda'_i\mathbf{e}'_i \quad \text{holds,} \tag{3.43}$$

$$\Rightarrow \mathbf{D}\mathbf{D}^\top\mathbf{D}\mathbf{e}'_i = \lambda'_i\mathbf{D}\mathbf{e}'_i \tag{3.44}$$

   and $\mathbf{D}\mathbf{e}'_i$ is an eigenvector of $\mathbf{D}\mathbf{D}^\top$ with corresponding eigenvalue λ'_i.[63]

□

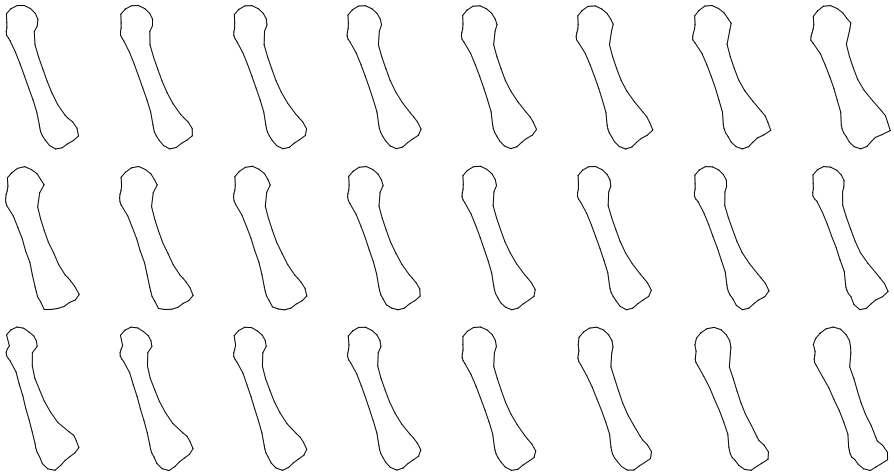


Figure 3.11: First 3 modes of a training set consisting of 30 bones.

3.1.2.2 Statistical Model of Shape Variation

A shape is represented by n landmarks. In order to model the variation of the shapes, the coordinates of the landmarks are considered as a vector

$$\mathbf{x} \in \mathbb{R}^{2n}. \tag{3.45}$$

Pose information has been removed during an alignment procedure, therefore remaining variations are due to deformations of the shape. PCA is performed on a set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ of shape representations resulting in a model with mean shape $\bar{\mathbf{x}}$ and $m = min(2n, t)$ modes.

$$\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m \quad \forall i > j : \lambda_i > \lambda_j \tag{3.46}$$

are eigenvectors of the covariance matrix of the shape training data. An instance of the shape model can be constructed by

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b}, \tag{3.47}$$

where

$$\mathbf{\Phi} = [\mathbf{e}_1|\mathbf{e}_2|\ldots|\mathbf{e}_m] \tag{3.48}$$

and $\mathbf{b}$ is the vector built by the coefficients of $\mathbf{x} - \bar{\mathbf{x}}$ with respect to $\mathbf{\Phi}$. Each of the modes is a deformation vector of the mean shape. Fig. 3.11 shows the first 3 modes obtained from a training set of 30 metacarpal bones of the small finger. Each row shows a sequence of 8 mean shapes deformed by a mode according to different parameters.

Choice of number of modes. The number of modes influences not only the compactness of the model but also the danger of over-fitting the model to noise in the training set. It has to be chosen according to properties of the training set in several ways.

1. The number of modes t is chosen so that a given fraction f_v of the total variation in the training set is represented. If $\lambda_i, i = 1, 2, \ldots, n$ are the eigenvalues ordered according to Eq. 3.46 of the covariance matrix then the total variance in the training set is the sum of all eigenvalues

$$v_T = \sum \lambda_i. \tag{3.49}$$

   t is chosen so that

$$t = min\{\tilde{t} | \sum_{i=1}^{\tilde{t}} \lambda_i \geq f_v v_T\}. \tag{3.50}$$

2. If noise of variance σ^2_{noise} is expected in the measurements of the point positions, t can be chosen to be
$$t = max\{\tilde{t}|\lambda_t > \sigma^2_{noise}\} \tag{3.51}$$

3. In order to approximate any training sample with given accuracy by reconstruction based on a subset of t modes, t has to be determined during a test procedure. Either reconstruction accuracy is tested for all training samples after building the model based on the entire training set or a model based on all training samples except one is used to reconstruct this sample. For each iteration t_i is chosen as the minimum of all $\tilde{t}$ that fulfill the required claim. After performing this test for all samples, a reliable estimate for the required number of modes $t = min(\{t_1, t_2, \ldots, t_t\})$ can be found.

4. In the *broken stick method*, eigenvectors are considered interpretable if their eigenvalues exceed a threshold based on random data. The i^{th} eigenvector $\mathbf{e}_i$ is retained if for the corresponding eigenvalue
$$\lambda_i > \sum_{k=i}^{p} \frac{1}{k} \tag{3.52}$$
holds.

A comparative study of methods can be found in [22] and [31]. The most common approach is to choose the number of modes according to 1.

Constraining Parameter Variation. The instances synthesized from the model $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{\Phi b}$ have to be similar to the examples in the training set. Therefore the parameters in $\mathbf{b}$ have to be chosen according to the distribution in the training set. Let
$$p(\mathbf{b}) \tag{3.53}$$
be a probability distribution over the space of parameters that assigns each parameter set a certain value learned from the training set. A new parameter set $\mathbf{b}'$ is considered *plausible* if
$$p(\mathbf{b}) \geq p_t \tag{3.54}$$
where p_t is a threshold so that some proportion of the training set passes the plausibility test. Again the choice of threshold is affected by the expected noise in the training set.

If one assumes that the parameters $\mathbf{b} = (b_1, b_2, \ldots, b_m)$ are independent (Lemma 3.2 supports this claim), then
$$log(p(\mathbf{b})) = -0.5 \sum_{i=1}^{m} \frac{b_i^2}{\lambda_i} + c, \tag{3.55}$$
where c is constant. $\mathbf{b}$ can either be constraint by thresholds for each element b_i (for example $\forall i : b_i \leq 3\sqrt{\lambda_i}$) or by surrounding it with a hyperellipsoid claiming that
$$\left(\sum_{i=1}^{m} \frac{b_i^2}{\lambda_i} \right) \leq M_t \tag{3.56}$$
where M_t is a threshold chosen using the χ^2 distribution [10].

An instance of a model $\mathbf{\Phi}, \mathbf{x}$ is called a *plausible instance* if the parameter vector $\mathbf{b}$ meets the requirements stated above.

3.1.3 Modeling Local Texture

Until now the model contains only information about shape variation, the image content was left out until now. In this section a way to include information about local image content in the vicinity of the landmarks the shapes are built from is explained. This information is used during image interpretation to adapt to the actual image content and will be the second driving force during ASM search.

Training. During training positions of landmarks $x_j^i \in \mathbb{R}^2$ are annotated in the training images. (Sec. 3.1.1.2 gives a more detailed explanation) Hence in each of N_t training images n points were indicated resulting in a set of forms

$$\{\mathbf{x}_1, \mathbf{x}_1, \dots \mathbf{x}_{N_t}\} \tag{3.57}$$

where

$$\forall i = 1, \dots, N_t : \mathbf{x}_i = [x_1^i, \dots, x_n^i] \in \mathbb{R}^{2\times n} \tag{3.58}$$

For each of the corresponding landmarks x^i in all forms

$$x_j^i, j = 1, \dots, N_t \tag{3.59}$$

gray values on a line with length $2k + 1$ orthogonal to the shape are extracted on both sides of the landmark (Fig. 3.12).

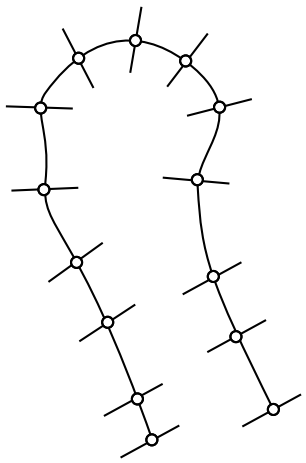


Figure 3.12: Gray value profiles orthogonal to the shape.

More precisely for a landmark $x_j^i = (p^i, p^i)$ with neighboring landmarks $x_j^{i-1} = (p^{i-1}, p^{i-1})$ and $x_j^{i+1} = (p^{i+1}, p^{i+1})$, gray values are sampled at positions

$$\mathbf{g}_j'^i(s) = x_j^i + (s - k - 1)\mathbf{n}_j^i \quad s = 1, \dots, 2k + 1 \tag{3.60}$$

with

$$\mathbf{n}_j^i = \frac{2x_j^i - x^{i-1} - x_j^{i+1}}{\|2x_j^i - x^{i-1} - x_j^{i+1}\|_2}. \tag{3.61}$$

Fig. 3.13 shows sample positions neighboring a landmark along a line orthogonal to the shape. To reduce effects of changing global illumination the 1^{st} derivative of the gray level vector $\mathbf{g}_j'^i$ is calculated. The resulting profile will be denoted by $\mathbf{g}_j^i \in \mathbb{R}^{2k+1}$ and $\mathbf{g}_j^i$ is normalized by

$$\mathbf{g}_j^i \to \frac{\mathbf{g}_j^i}{\sum_{s=1}^{2k+1} |\mathbf{g}_j^i(s)|}. \tag{3.62}$$

For each model point this yields a set of vectors which are assumed to be distributed as a multivariate Gaussian. Mean $\bar{\mathbf{g}}^i$ and covariance $\bar{\mathbf{S}}^i$ of this distribution are estimated. This gives a statistical gray level model for each model point. [13].

Ordering of landmarks. Note that we assume that the landmarks are ordered following a natural boundary in the image. The order of the landmarks can either be determined during annotation e. g. in the case of the boundary of a bone there is a natural order of landmarks traveling the boundary in clockwise or counter-clockwise direction as used in this thesis. An alternative would be an automatic establishment of this order. However this is of minor interest, since active appearance models (AAM) generalize ASMs with respect to texture representation and are therefore a more feasible way to model complex objects without a unique boundary.

3.1.4 Image Interpretation with Active Shape Models

Given an ASM $\mathbf{\Phi}, \mathbf{x}$ it is possible to locate forms that can be approximated by a translated and rotated instance $\mathbf{s}$ of the shape model. During an ASM search, model parameters $\mathbf{b}$, according to the instance with respect to $\mathbf{\Phi}, \mathbf{x}$ and pose information T and R according to pose of the form found in the present input image are determined. The method is iterative and fits the image content with a form starting from a possibly coarse initialization. The form is constraint by the ASM ensuring that it is a valid instance. Intuitively, this corresponds to a statement like 'it looks like a bone'. The search is driven by two influences, one originating from the model, the other from the image content.

1. Start with an initialization of the instance in the image.
2. Update all points of the instance according to the image content.
3. Update all points of the new instance according to the model.

Steps 2 and 3 are repeated until the instance converges. In the following the concepts underlying these steps are explained in detail.

1. Find an initialization instance. Many ways to find pose and model parameters at which the search and refinement of the instance can initiate have been proposed. In [9] a method based on multiple hypotheses is proposed for the search of structures in medical images. A number of hypotheses each consisting of pose information and model parameters are produced. They give approximate locations of the model points and are subject to refinement and final selection. In [28] a method to generate a set of hypotheses working with genetic algorithms is explained.

The hand is made up of many similar bones thus producing similar shapes in hand radiographs. This brought about a method based on a global model which was used during this thesis. This method is described in detail in Chap. 2. It gives approximate results of positions of bones of the hand. Results are given by the estimates of the joint locations, these allow the positioning of an initial instance of the shape, in particular the mean shape, so that its outermost landmarks are identical with the joint location estimates (Fig. 3.14)

Hence, given two joint positions j_{prox} and j_{dist} for the proximal (closer to the body) and distal (closer to the finger tips) joint of the bone and corresponding landmarks of the model x_1 and x_2 without loss of

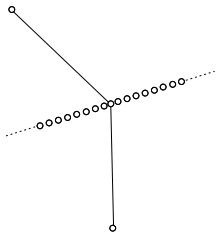
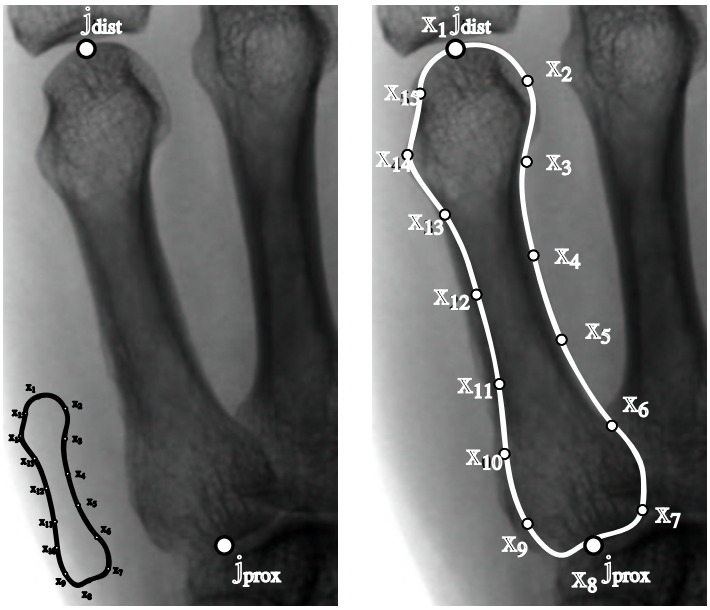


Figure 3.13: Sample points along the profile line.



(a) (b)

Figure 3.14: (a) Joint positions and mean shape, (b) initialization instance with $x_1^{init} = j_{dist}$ and $x_8^{init} = j_{prox}$.

generality then the points x_i of the mean shape have to be translated, scaled and rotated by

$$x_i^{init} = Rx_i + T \quad \forall i = 1, \ldots, n \tag{3.63}$$

with

$$R = \begin{pmatrix} \frac{x_2^d j_2^d + x_1^d j_1^d}{(x_1^d)^2 + (x_2^d)^2} & -\frac{x_1^d j_2^d - x_2^d j_1^d}{(x_1^d)^2 + (x_2^d)^2} \\ \frac{x_1^d j_2^d - x_2^d j_1^d}{(x_1^d)^2 + (x_2^d)^2} & \frac{x_2^d j_2^d + x_1^d j_1^d}{(x_1^d)^2 + (x_2^d)^2} \end{pmatrix} \tag{3.64}$$

$$T = j_1 - Rx_1 = j_2 - Rx_2 \tag{3.65}$$

where

$$x^d = \begin{pmatrix} x_1^d \\ x_2^d \end{pmatrix} = x_1 - x_2 \quad \text{and} \quad j^d = \begin{pmatrix} j_1^d \\ j_2^d \end{pmatrix} = j_{prox} - j_{dist} \tag{3.66}$$

in order to generate the pose of the initialization instance of the bone model.

Note that R is a similarity transform and therefore

$$R = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}. \tag{3.67}$$

Given a coarse position estimate of the shape $\hat{\mathbf{s}}^0$ instance the search algorithm proceeds as follows.

2. Update the instance according to image content. In order to adapt the instance $\hat{\mathbf{s}}^i$ to the image content an updating step has to be performed for each landmark $x_j \in \hat{\mathbf{s}}^i$. According to the statistical model of the local texture obtained during training the landmark is translated along a line orthogonal to the shape (Fig. 3.15). A vector $\mathbf{g}_{input}^j \in \mathbb{R}^{2K+1}$ of gray values with length $K \geq k$ is extracted from the image at the positions described in Eq. 3.60. The 1^{st} derivative is determined and compared to the previously defined local texture vector $\bar{\mathbf{g}}^j$. In Fig. 3.15 the process is depicted. By shifting $\bar{\mathbf{g}}^j$ stepwise against $\mathbf{g}_{input}^j$ (Fig. 3.15) and computing the *Mahalanobis distance* [44] between the overlapping parts of

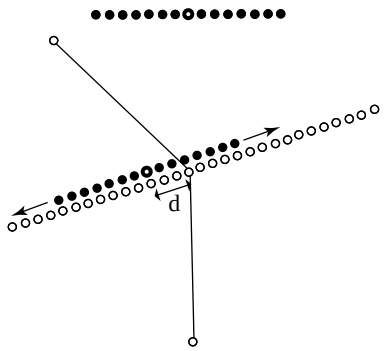


Figure 3.15: By shifting a window along a line orthogonal to the shape, extracted gray value vectors $\mathbf{g}^j_{input}(d)$ are compared to $\bar{\mathbf{g}}^j$.

the vectors, a shift of the landmark with highest probability of fitting to the local texture model $(\bar{\mathbf{g}}^j, \bar{\mathbf{S}}^j)$ can be found.

Definition 3.9 (Mahalanobis distance) *Given two Gaussian distributions with equal covariance matrix* $\mathbf{S}$ *the Mahalanobis distance between two samples* $\mathbf{m}_1$ *and* $\mathbf{m}_2$ *is defined as*

$$d_M^2(\mathbf{m}_1, \mathbf{m}_2) = (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{S}^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \tag{3.68}$$

For each shift the Mahalanobis distance gives the quality of the fit of the sub interval of the present texture vector and the model

$$f(d) = (\mathbf{g}^j_{input}(d) - \bar{\mathbf{g}}^j)^T (\bar{\mathbf{S}}^j)^{-1} (\mathbf{g}^j_{input}(d) - \bar{\mathbf{g}}^j) \tag{3.69}$$

where

$$\mathbf{g}^j_{input}(d) := \mathbf{g}^j_{input}(K + 1 + d - k : K + 1 + d + k). \tag{3.70}$$

$f(d)$ decreases as the fit improves. The shift with lowest Mahalanobis distance is chosen and the landmark is moved accordingly. The result of this step is a set of new landmarks

$$\hat{\hat{\mathbf{s}}}^i = \{x_1, x_2, \ldots, x_n\}. \tag{3.71}$$

that are not necessarily a plausible instance of the model.

3. Update the instance according to the model (Pose and shape parameters). In order to deform $\hat{\hat{\mathbf{s}}}^i$ into a form of a plausible shape instance $\hat{\mathbf{s}}^{i+1}$ so that it can be represented in the ASM that was obtained during training, the set of landmarks has to be adapted to the model.

$\hat{\mathbf{s}}^{i+1}$ is claimed to be a form based on a proper instance of the model, hence there exists a translation T_{X_t,Y_t} , rotation and scaling $R_{s,\theta}$ and an ASM shape to express

$$\hat{\mathbf{s}}^{i+1} = T_{X_t,Y_t} + R_{s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi b}). \tag{3.72}$$

By minimizing the sum of squared distances

$$|\hat{\hat{\mathbf{s}}}^i - S_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi b})|^2 \tag{3.73}$$

between corresponding points in $\hat{\hat{\mathbf{s}}}^i$ and $\hat{\mathbf{s}}^{i+1}$ a fit can be found. $S_{X_t,Y_t,s,\theta}$ is a similarity transform consisting of T and R in Eq. 3.72. The search is realized by an iteration [10]:

1. Initialize the shape parameters $\mathbf{b}$ with $\mathbf{0}$.

2. Generate the model instance $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi b}$.

3. Find pose parameters X_t, Y_t, s, θ which best map $\mathbf{x}$ to $\hat{\hat{\mathbf{s}}}^i$.

4. Invert these pose parameters and use them to project $\hat{\hat{\mathbf{s}}}^i$ to $\hat{\hat{\mathbf{x}}}^i$ into the model coordinate frame i. e. to align it to the shapes in the model.

$$\hat{\hat{\mathbf{s}}}^i_M = S^{-1}_{X_t,Y_t,s,\theta}(\hat{\hat{\mathbf{s}}}^i) \tag{3.74}$$

5. Project $\hat{\hat{\mathbf{x}}}^i$ into the tangent plane to $\bar{\mathbf{x}}$ by scaling it by $1/(\langle \hat{\hat{\mathbf{x}}}^i, \bar{\mathbf{x}} \rangle)$.

6. Update the model parameters to match $\hat{\hat{\mathbf{x}}}^i$

$$\mathbf{b} = \mathbf{\Phi}^T(\hat{\hat{\mathbf{x}}}^i - \bar{\mathbf{x}}) \tag{3.75}$$

7. Apply constraints on $\mathbf{b}$. According to the plausibility constraints defined in Eq. 3.54 or Eq. 3.55 the parameters $\mathbf{b}$ can simply be truncated so that

$$\forall i : |b_i| \leq 3\sqrt{\lambda_i}. \tag{3.76}$$

   For arbitrary plausibility models $p(\mathbf{b})$ the nearest $\mathbf{b}$ has to be found so that Eq. 3.54 is fulfilled.

8. Go to step 2 until the search converges.

Convergence can be either be defined as a state when an iteration produces no significant changes to the parameters or alternatively a maximum number of iterations can be given. Usually the algorithm converges after a few iterations.

3.2 Active Contour Models

Active contour models or *snakes* were introduced in [34] as energy minimizing splines guided by constraint forces and influenced by image forces. The image forces move the spline towards features such as lines or edges. One can distinguish between *parametric active contours* [34] and *geometric active contours* [45]. In this work we will concentrate on parametric active contours.

3.2.1 Traditional Snakes

Consider a curve

$$\mathbf{v}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \quad s \in [0,1], \tag{3.77}$$

that moves through the 2 dimensional spatial domain of an image

$$\mathbf{I}(x,y), \tag{3.78}$$

then its energy functional $\mathbf{E}^*_{snake}$ puts the forces driving the movement of the snake in a more analytical form:

$$\mathbf{E}^*_{snake} = \int_0^1 \mathbf{E}_{snake}(\mathbf{v}(s))ds \tag{3.79}$$

$$= \int_0^1 \mathbf{E}_{int}(\mathbf{v}(s)) + \mathbf{E}_{ext}(\mathbf{v}(s))ds \tag{3.80}$$

where $\mathbf{E}_{int}$ represents the internal energy of the spline due to bending and elasticity while $\mathbf{E}_{ext}$ represents the forces caused by image content. The constraint forces are left out since they are irrelevant in the context of this work. They are discussed in detail in [34].

The snake $\mathbf{v}(s)$ moves to minimize the energy functional where the internal energy is

$$\mathbf{E}_{int} = \frac{1}{2}(\alpha|\mathbf{v}'(s)|^2 + \beta|\mathbf{v}''(s)|^2) \tag{3.81}$$

with $\mathbf{v}'(s)$ and $\mathbf{v}''(s)$ denoting first and second derivate with respect to s. α and β are weighting parameters that control the tension and rigidity of the snake.

The external energy $\mathbf{E}_{ext}$ is derived from image content. Its smaller values lie on the regions of interest. A typical external energy is designed to drag the snake towards edges in the image, hence

$$\mathbf{E}_{ext}(x, y) = -|\nabla \mathbf{I}(x, y)|^2. \tag{3.82}$$

This energy becomes effective only in the vicinity of edges and can be changed to an energy that is based on a map where each pixel is assigned a value corresponding to the distance of the closest edge. Let this map be $\mathbf{I}_2(x, y)$, then the energy is

$$\mathbf{E}_{ext}(x, y) = -|\nabla \mathbf{I}_2(x, y)|^2. \tag{3.83}$$

A snake has to satisfy the Euler equation

$$\alpha\mathbf{v}''(s) - \beta\mathbf{v}''''(s) - \nabla\mathbf{E}_{ext} = 0 \tag{3.84}$$

which can be viewed as a *force balance equation*

$$\mathbf{F}_{int} + \mathbf{F}_{ext} = 0 \tag{3.85}$$

where

$$\mathbf{F}_{int} = \alpha\mathbf{v}''(s) - \beta\mathbf{v}''''(s) \tag{3.86}$$

$$\mathbf{F}_{ext} = -\nabla\mathbf{E}_{ext}. \tag{3.87}$$

Iterative active contour search. In order to find a iterative solution to Eq. 3.85 $\mathbf{v}$ is treated as a function not only of s but also of time t i. e. $\mathbf{v}(s, t)$. The partial derivative of $\mathbf{v}$ with respect to t is given by

$$\frac{\delta\mathbf{v}}{\delta t}(s, t) = \alpha\mathbf{v}''(s) - \beta\mathbf{v}''''(s) - \nabla\mathbf{E}_{ext}. \tag{3.88}$$

The snake stabilizes when $\delta\mathbf{v}/\delta t = 0$ while during iteration a factor is multiplied with $\delta\mathbf{v}/\delta t$ to control the temporal step-size. Fig. 3.16(a) shows an image of a line, in Fig. 3.16(b) the distances to the closest line pixel is visualized by gray values where white maps to the biggest distance and black to zero. In Fig. 3.16(c) and (d) resulting gradient force fields are depicted. Fig. 3.16(c) depicts the gradients of the edge map in (a) and Fig. 3.16(d) shows a gradient field resulting from the map (b) where the value of each pixel represents its distance to the closest edge.

3.2.2 Gradient Vector Flow

Force fields. The force fields depicted in Fig. 3.16 have a drawback with respect to the local minima of the force balance equation (Eq. 3.85) in the proximity of boundary concavities as shown in Fig. 3.16(a). These local minima are caused by diametrically opposed force directions in the external force field as depicted in Fig. 3.16(d). Fig. 3.17(a) shows a part of the force field where the effect becomes visible. The arrows point towards the lines on the side of the sine resulting in a local configuration inside of the concavity consisting exclusively of horizontal forces. The snake is not dragged into the concavity, which therefore it remains undetected.

Various approaches to overcome this and other shortcomings of the snake method have been proposed. Among these are dynamic external force fields, that change during the iteration process. Multi resolution snakes were proposed in [43] and pressure forces were used in balloons [8]. Both suffer from added

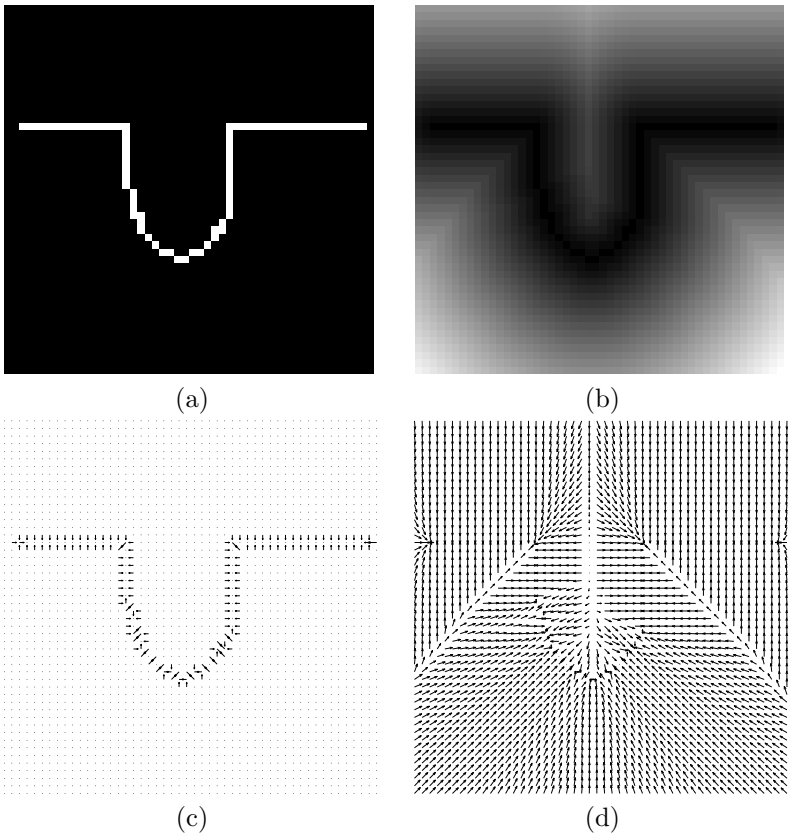


Figure 3.16: (a) Line, (b) distance to the line, (c) gradient vector field resulting from (a), (d) gradient vector field resulting from (b).

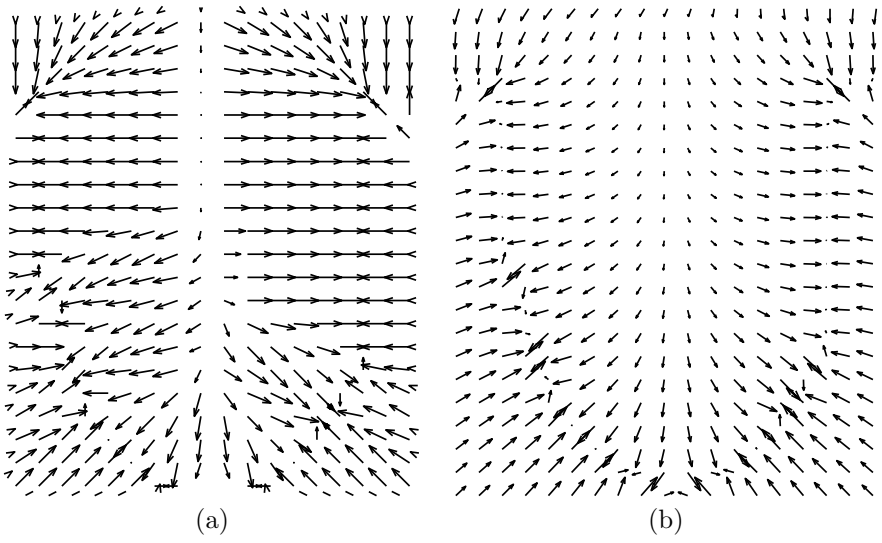


Figure 3.17: (a) conservative force-field, (b) non-conservative GVF force-field .

complexity to the problem, resulting in the need of fine tuning and careful initialization of snakes. These snakes are based on pressure forces or problems during the change of resolution for multi resolution snakes, although the latter broaden the capture range of snakes.

A promising alternative for the static external force field is the *gradient vector flow field*. In contrast to the static force fields described above in Eq. 3.82 and Eq. 3.83 this is a *non-conservative* force field.

Definition 3.10 (Conservative force field) *A force field* $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ *is called conservative if for paths* $\mathbf{v}(s)$ *with* $s \in [0, 1]$, $a = \mathbf{v}(0)$ *and* $b = \mathbf{v}(1)$

$$\int_a^b \mathbf{F}(\mathbf{v}(s))ds \tag{3.89}$$

is independent from the chosen path $\mathbf{v}(s)$.

Example for a conservative force field are central forces or previously used snake force fields formulated using a standard energy minimizing framework. Forces for particular points in a space can be determined directly for conservative force fields.

Non-conservative force field. By their nature non-conservative force fields lack the property of direct calculation of local forces.

Definition 3.11 (Non-conservative force field) *A force field* $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ *that is not a conservative force field is called non-conservative force field.*

Irrotational and solenoidal component. The Helmholtz theorem [49] states that the most general static vector field can be decomposed into an *irrotational* and a *solonoidal* component. This is equivalent to splitting the force field up into a *conservative* (irrotational) and a *non-conservative* (solonoidal) component. Since traditional snakes rely on fields based on gradients of potential functions it enters Eq. 3.85 as an irrotational or conservative field.

3.2.2.1 Gradient Vector Flow Snakes

In [70] it is assumed that the inability to move a snake into boundary concavities is a property of all conservative forces. A more general approach to generate a force field is directly by a force balance condition as formulated in Eq. 3.85.

In order to have both irrotational and solenoidal components, the static external force field $\mathbf{F}_{ext}$ is defined as a so called *gradient vector flow field* (GVF field)

$$\mathbf{F}_{ext} = \mathbf{v}(x, y) \tag{3.90}$$

hence the dynamic formulation of the snake (Eq. 3.88) becomes

$$\frac{\delta \mathbf{v}}{\delta t}(s, t) = \alpha \mathbf{v}''(s) - \beta \mathbf{v}''''(s) + \mathbf{v}(x, y). \tag{3.91}$$

The field is generated in 2 steps.

Edge map. An edge map $f(x, y)$ is derived directly from the image $\mathbf{I}(x, y)$. Closer to the edges, or other features it has to search for, the value of the edge map becomes higher. More precisely the edge map has to fulfill 3 properties:

- The gradient ∇f of an edge map has a vector pointing towards the features to detect, presumably edges in the image.

- These vectors have large a magnitude only in the immediate vicinity of the edges.
- In homogeneous regions, where $\mathbf{I}(x, y)$ is nearly constant ∇f is nearly zero.

If the gradient of the edge map is used as an external force like with traditional snakes these properties result in the following behavior of the snake: It converges to edges if it is initialized close to these edges due to the first property. Due to the second the capture range of the external force field will be very small and due to the third in homogeneous regions there will be no external force at all. One goal of a more general formulation of the external force is to extend the gradient map further from the edges into homogeneous regions.

Gradient vector flow. With gradient vector flow, this is realized by a computational diffusion process that keeps the first property but changes the second and third so that even in homogeneous regions the forces are pointing into reasonable directions with respect to distant edges. The gradient vector flow field is defined as follows:

Definition 3.12 (Gradient vector flow field) *A vector field*

$$\mathbf{v}(x, y) = [u(x, y), v(x, y)] \tag{3.92}$$

that minimizes the energy

$$\mathbf{E}_v = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dxdy \tag{3.93}$$

is called a gradient vector flow field.

This definition ensures that in homogeneous regions i. e. if ∇f is small the energy is dominated by the squared partial derivatives of the vector field $\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2)$. When ∇f is large the second term $|\nabla f|^2 |\mathbf{v} - \nabla f|^2$ dominates and is minimized if $|\mathbf{v} - \nabla f|^2$ is close to zero i. e. if $\mathbf{v} = \nabla f$. This yields a smooth vector field slowly varying in homogeneous regions as depicted in Fig. 3.17(b). μ is a parameter controlling the balance between smoothing and data driven influences on the vector field. A more detailed discussion and interesting connections to other research fields can be found in [70]. The resulting vector field can be expected to have both irrotational and solenoidal components since the integrand in Eq. 3.93 corresponds to an equal penalty of divergence and curl similar to optical flow [26].

Iterative generation of a GVF field. Since the gradient vector flow field cannot be calculated for each pixel in a direct way but is dependent on global properties of the field an iterative method is necessary for generation. It can be shown [14] that the GVF field is a solution to the Euler equations

$$\mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \tag{3.94}$$
$$\mu\nabla^2 u - (v - f_y)(f_x^2 + f_y^2) = 0 \tag{3.95}$$

where ∇^2 denotes the Laplacian operator. These equations can be solved iteratively by viewing u and v as functions of time

$$u_t(x, y, t) = \mu\nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y)) \cdot (f_x(x, y)^2 + f_y(x, y)^2) \tag{3.96}$$
$$v_t(x, y, t) = \mu\nabla^2 v(x, y, t) - (v(x, y, t) - f_y(x, y)) \cdot (f_x(x, y)^2 + f_y(x, y^2)) \tag{3.97}$$

The equations Eqs. 3.96 and 3.97 are known in physics as *general diffusion equations*. They are decoupled partial differential equations and can therefore be solved separately in u and v.

For iterative solution consider three constants independent from t

$$\begin{aligned} b(x,y) &= f_x(x,y)^2 + f_x(x,y)^2 && (3.98)\\ c^1(x,y) &= b(x,y)f_x(x,y) && (3.99)\\ c^2(x,y) &= b(x,y)f_y(x,y), && (3.100) \end{aligned}$$

then Eqs. 3.96 and 3.97 can be rewritten as

$$\begin{aligned} u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - b(x,y)u(x,y,t) + c^1(x,y) && (3.101)\\ v_t(x,y,t) &= \mu\nabla^2 v(x,y,t) - b(x,y)v(x,y,t) + c^2(x,y). && (3.102) \end{aligned}$$

b, c^1 and c^2 are computed in advance and remain fixed. The spacing between the image pixels is Δx and Δy, respectively. The time step is Δt while x, y and t are replaced by i, j and n. The partial derivatives can be approximated as

$$\begin{aligned} u_t &= \frac{1}{\Delta t}(u_{i,j}^{n+1} - u_{i,j}^n)\\ v_t &= \frac{1}{\Delta t}(v_{i,j}^{n+1} - v_{i,j}^n) \end{aligned}$$

and

$$\begin{aligned} \nabla^2 u &= \frac{1}{\Delta x \Delta y}(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j})\\ \nabla^2 v &= \frac{1}{\Delta x \Delta y}(v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i,j-1} - 4v_{i,j}). \end{aligned}$$

Hence by substitution in Eqs. 3.96 and 3.97 the iteration is defined by

$$\begin{aligned} u_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)u_{i,j}^n + r((u_{i+1,j}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n) + c_{i,j}^1\Delta t\\ v_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)v_{i,j}^n + r((v_{i+1,j}^n + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n) + c_{i,j}^2\Delta t \end{aligned}$$

where

$$r = \frac{\mu\Delta t}{\Delta x \Delta y}. \tag{3.103}$$

The convergence of this process is shown in [2]. It is stable if b, c^1 and c^2 are bounded and and $r \leq 1/4$. This condition is trivial for b, c^1 and c^2 since they are fixed and Δt has to be restricted by

$$\Delta t \leq \frac{\Delta x \Delta y}{4\mu} \tag{3.104}$$

considering Eq. 3.103.

3.3 ASM Driven Snakes

The concepts of active shape models and active contour models that have been introduced by now have some inherent shortcomings besides their valuable properties that make them suitable for medical image analysis even with complex visual input. In particular some properties of the ASM approach prevent it from identifying pathologic changes to bones as being present in rheumatoid arthritis accurately. The erosions caused by the disease are small local deformations of the bone contour visible in a radiograph.

Considering the disadvantages of both approaches it becomes clear that some of them are quite complementary. In the following section properties of ASMs and snakes will be discussed with respect to enhancement. A method is proposed that overcomes some of the main drawbacks of ASMs by the succeeding use of a snake. While it still inhibits some of the a priori knowledge from the ASM, it is able to free itself from the inherent constraints of the model based approach in a controlled way.

An outline of the drawbacks of ASMs and a detailed explanation of the *ASM driven snakes* algorithm to overcome some of the drawbacks are highlighted.

3.3.1 The Drawbacks of Active Shape Models

For the detection of erosive changes of the bone surface a very accurate identification of the contour in the radiographs is necessary. Active shape models are able to detect anatomic structures in a stable way, still they are subject of several drawbacks. (See [61] for a detailed discussion).

Need for distinct features ASMs deal with a finite number of corresponding landmarks. Therefore they are dependent on distinguishable features present during training and application (see Sec. 3.1.1.1). A continuous identification of a contour is derived by interpolation. This approach is unable to accurately describe fine structure in the contour between landmarks.

Training set The performance of an ASM is heavily dependent on the choice of the training set. The variation of forms an ASM is able to detect is limited by the variety of the shapes used during training. The benefit of restricting the search to valid shapes turns into a drawback if the achievable size of the training set is limited. This can be due to either the vast amount of time needed to manually assign training shapes or a inherent characteristic of the detection task to perform.

Although the size of the training set has been addressed in [12] where finite element models are used to generate additional training examples severe pathological changes like erosions caused by RA cannot be modeled accurately by ASMs due to their highly irregular and local nature. This problem is also due to the next point.

Linearity The linearity of PCA, the statistical model used to describe the shape variation introduces certain limitations to the class of shapes that can be represented. They assume a Gaussian distribution of the shape parameters. Although this is a valid assumption for healthy shapes it no longer holds for pathologic deformations. In particular erosive changes which are small growing concavities on various locations on the bone contour cannot be modeled efficiently.

In order to address these drawbacks it is necessary to investigate alternative extensions to the ASM approach. In this thesis an approach will be introduced that combines the advantages, namely the reliability of ASMs with the flexibility of snakes. An alternative would have been the inclusion of a sufficient amount of bones showing different states of destruction in the ASM training set.

Instead the ASM training was performed only with healthy bones and a strategy different from ASMs was applied to detect the exact contour of bones that eventually show destructions. The fine search for the bone contour is performed by snakes which are guided by the ASM, but provide greater flexibility. This choice is done due to several reasons:

1. ASMs are considered not to be a suitable tool to model fine local details which are independent from the rest of the shape. During the PCA compression the necessary information would get lost, if an appropriate number of eigenvalues are chosen to model the overall contour.

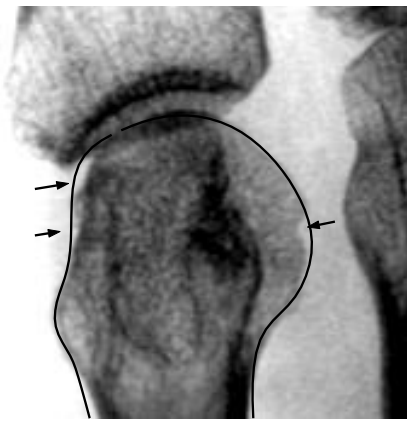


Figure 3.18: Metacarpal bone with erosive changes (arrows) and result of an ASM search.

2. The size of the necessary training set increases drastically if the ASM has to cover deformations of the joint regions caused by RA. Under clinical circumstances it is not possible to collect an accordingly high number of training samples, since the time consumption of manually locating the boundary line is rather high.

3. An ASM based only on healthy bone shapes provides the possibility to identify a reference shape assumed to be close to the contour of the bone in a healthy state. It makes a parametrization of the bone contour with respect to an assumed healthy state possible.

The proposed method combines the advantages of ASMs and snakes. *ASM driven snakes* enhance the result of ASMs and provide high accuracy and robustness, while the size of the necessary set of samples for ASM training can be decreased. The method not only enhances the accuracy of landmark detection of a standard ASM but identifies contour lines in between ASM landmarks. The approximation of fine shape details is restricted only by parameters of the inner force field of a snake. This is in contrast to [27]. A priori knowledge is used only with respect to the gray level neighborhood of the landmarks while the tie of the final result to the shape model is gradually decreased. The snakes are sensitive to accurate initialization but can move more freely trough the image domain. They adapt even to fine structure and provide a theoretically continuous spline description. These are crucial properties in detection of pathologic changes like bone erosions, since they cannot be modeled accurately with ASMs. In Fig. 3.18 a bone in a hand radiograph showing erosive destructions and the result of an ASM search are depicted. The ASM result does not adapt to the erosions.

3.3.2 ASM Driven Snakes Algorithm

After a coarse initialization of the bone position has been achieved by LLM-networks as described in Chap. 2 the bone contour is identified by a standard ASM search. The algorithm proceeds in the following main steps:

1. Initialization by ASM search. The ASM has been trained on a set of shapes of intact bones. Each shape is represented by a set of N landmarks

$$\mathbf{x} = (x_1, x_2, \ldots, x_N). \tag{3.105}$$

An ASM search on the input image $\mathbf{I}(x, y)$ results in estimates

$$\hat{\mathbf{x}}^{\mathbf{1}} = (\hat{x}_1^1, \hat{x}_2^1, \ldots, \hat{x}_N^1) \tag{3.106}$$

of the landmark positions.

2. Spline interpolation. The landmarks are interpolated by a cubic spline in order to obtain a continuous approximation of the bone contour. Splines are piecewise polynomial curves. Their approximation properties are controlled by the grade of the polynomials and the size of the curve partitions. Given a curve $\mathbf{c}(s)$ parameterized on an interval $s \in [a, b,]$ and a partition Δ

$$a = s_0 < s_1 < \ldots < s_n = b \tag{3.107}$$

of this interval a spline can be defined a follows.

Definition 3.13 (Spline) *A function* $\mathbf{s} : [a, b,] \to \mathbb{R}^n$ *is called a spline of grade* p *and smoothness* q *denoted by* $\mathbf{s} \in \mathcal{S}_q^p(\Delta)$*, if* $\mathbf{s} \in C^q[a, b,]$ *i. e. the* q^{th} *derivative of* $\mathbf{s}$ *exists and is continuous, and for* $j = 1, \ldots, n$

$$\mathbf{s}|_{[s_{j-1}, s_j]} \in \mathcal{P}_p([s_{j-1}, s_j]) \tag{3.108}$$

where $\mathcal{P}_p([s_{j-1}, s_j])$ *denotes the polynomials of grade* $\leq p$ *on the interval* $[s_{j-1}, s_j]$*.* Δ *is a partition of* $[a, b,]$ *as described in Eq. 3.108 and* $C^{-1} := [a, b]^{\mathbb{R}^n}$*.*

A spline $\mathbf{s} \in \mathcal{S}_2^3$ is called a cubic spline and is sufficient for the approximation of the intermediate bone contour.

The landmarks define a partition of a closed curve

$$[\mathbf{s}(\hat{x}_1^1), \mathbf{s}(\hat{x}_2^1)[, \ldots, [\mathbf{s}(\hat{x}_N^1), \mathbf{s}'(\hat{x}_1^1)[. \tag{3.109}$$

The resulting spline $\mathbf{s}_{\hat{\mathbf{x}}}(s)$ allows for a parameterization of the bone contour approximation by its arclength $s \in [0, S]$. The arclength is approximated by the square root of chord-length as described in [41]. By defining a step size Δs the continuous curve is discretized resulting in a series of points

$$\hat{\mathbf{s}}_{\hat{\mathbf{x}}}(i), \quad i = 0, \ldots, n \tag{3.110}$$

where $n = max\{k \in \mathbb{N} : k \leq S/\Delta s\}$. The parameter determining the position of landmark x_j is denoted by s_j,

$$\mathbf{s}_{\hat{\mathbf{x}}}(s_j) = x_j. \tag{3.111}$$

3. Gray value profile extraction. For all points $\hat{\mathbf{s}}_{\hat{\mathbf{x}}}(i)$ a line orthogonal to the spline $\mathbf{s}_{\hat{\mathbf{x}}}$ can be constructed by

$$p_i(\lambda) := \hat{\mathbf{s}}_{\hat{\mathbf{x}}}(i) + \lambda\sigma \left(\frac{\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s)}{\| \frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s) \|_2} \right), \quad \frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s) \neq 0 \tag{3.112}$$

and $p_i(\lambda) := p_{i-1}\sigma(\lambda)$ if $\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s) = 0$. σ is iteratively defined by

$$\sigma_i = \begin{cases} \sigma_{i-1}, & sign(\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s)) = sign(\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i - 1\Delta s)) \\ -\sigma_{i-1}, & sign(\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i\Delta s)) \neq sign(\frac{\delta^2 \mathbf{s}_{\hat{\mathbf{x}}}}{\delta s^2}(i - 1\Delta s)) \\ 1, & i = 0 \end{cases} \tag{3.113}$$

σ ensures that the direction of the parametrization λ of the lines always points to the same side of the contour.

Gray value profiles are extracted from the image $\mathbf{I}(x, y)$ on the positions $p_i(\lambda)$:

$$\mathbf{I}(p_i(\lambda)), \quad i = 1, \ldots, n, \quad \lambda = -m, \ldots, m \tag{3.114}$$

yielding a set of profiles

$$\mathbf{P} = \langle \mathbf{p}_1 | \mathbf{p}_2 | \ldots | \mathbf{p}_n \rangle \in \mathbb{R}^{(2m+1)\times n} \tag{3.115}$$

where

$$\mathbf{p}_i = (\mathbf{I}(p_i(-m)), \ldots, \mathbf{I}(p_i(m)))^\top \in \mathbb{R}^{(2m+1)}, \quad i = 1, \ldots, n. \tag{3.116}$$

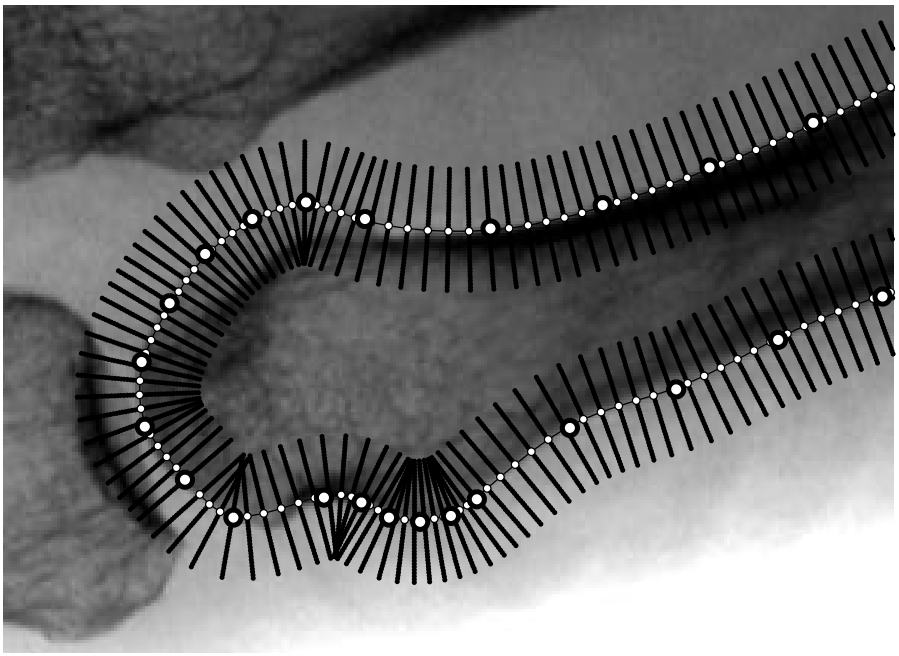


Figure 3.19: Lines orthogonal to the spline interpolation. Large circles mark landmarks.

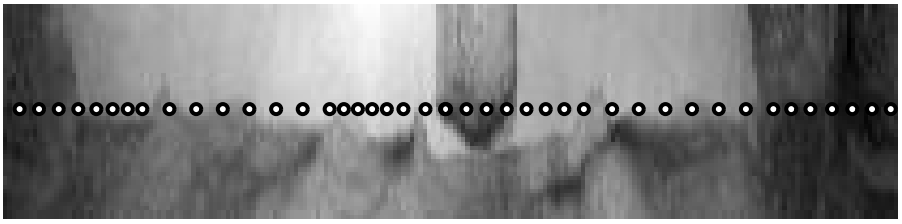


Figure 3.20: **P** extracted from a radiograph. Circles mark positions of landmarks in the original image.

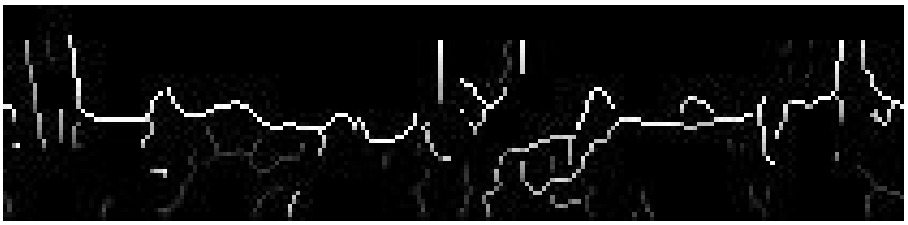


Figure 3.21: Edge map.

The extraction of gray values is done by interpolating gray values of neighboring positions. Fig. 3.19 shows a radiograph of a metacarpal of the small finger. The spline interpolation of the landmark positions (large circles) is discretized (small circles) and orthogonal lines along which gray values are extracted are depicted. Fig. 3.20 shows the resulting matrix $\mathbf{P}$. The spline is mapped to a horizontal line in the center of the spatial domain of $\mathbf{P}$

$$\hat{\mathbf{s}}_{\hat{\mathbf{x}}}(i) \mapsto^{\mathbf{P}} \hat{\mathbf{s}}^1_{\hat{\mathbf{x}}}(i) := (0, i). \tag{3.117}$$

4. Calculating the edge map. Based on the profile set $\mathbf{P}$ a weighted edge map is calculated.

$$f(x, y) : \{1, \ldots, n\} \times \{-m, \ldots, m\} \rightarrow [0, 1] \tag{3.118}$$

In order to stay abreast of the image content as well as the a priori knowledge represented in the ASM $f(x, y)$ is generated by a formula that controls the influence of both.

$$f(x, y) := sign(c(x, y)) *' (\tau c(x, y) + \eta(\mathbf{a}_b(x) \star d\mathbf{P}(x, y)/dy) + \gamma o(x, y)) \tag{3.119}$$

where $*'$ is the pixel-wise multiplication.

- $c(x, y)$ is the accumulation of outputs of a sequence of Canny edge detectors [6] $C(x, y, \sigma, t)$ with pairs of varying hysteresis thresholds $\mathbf{t} = \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_c$ applied to $\mathbf{P}(x, y)$.
- $\mathbf{a_b}(x)$ are generated by interpolating values of neighboring ASM texture profiles g_i and g_j with $i \leq x < j$. The convolution $(\mathbf{a}_b(x) \star d\mathbf{P}(x, y)/dy)$ corresponds to the similarity of the derivatives of profiles extracted from the input image and an interpolation of derivatives of profiles achieved during ASM training.
- The binary function $o(x, y) \in \{0, 1\}$ equals 1 iff the point (x, y) is part of an edge that crosses $\mathbf{s}_1$.

The snake parameters α and β steer the bond between each new estimate and the initial ASM estimate $\hat{\mathbf{x}}_1$. Increasing α and β causes a higher similarity in direction but prevent the fitting of fine structures. The parameter γ controls the effect of the estimate $\hat{\mathbf{x}}_i$ on $\hat{\mathbf{x}}_{i+1}$. γ determines the force dragging $\mathbf{s}_{i+1}$ to connected edges that cross $\mathbf{s}_i$. It becomes important if the previous estimate can be expected to be close to the true contour. η steers the influence of the visual appearance information stored in the ASM. τ controls the influence of an edge detector with gradually lower weighted branchings to the external force field corresponding to the search for a continuous contour. In Fig. 3.21 an example for a resulting weighted edge map is depicted. It corresponds to Fig. 3.20.

5. Searching for the contour with a GVF snake. A GVF vector field $\mathbf{v}$ (Fig. 3.22) is generated from f and the standard iterative snake algorithm is applied. This results in a new approximation of the transformed shape and contour positions in between landmarks in $\mathbf{P}$.

The active contour is initialized either on the position of $\hat{\mathbf{s}}_1$ or at the maxima of each column $\mathbf{p}_i$ in $\mathbf{P}$. It converges to a new estimation of the contour and the set of landmarks ${}^{\mathbf{P}}\hat{\mathbf{s}}^2_{\hat{\mathbf{x}}}$. For each point of ${}^{\mathbf{P}}\hat{\mathbf{s}}^1_{\hat{\mathbf{x}}}$ a vertical correction value is contained in this description. No horizontal corrections i. e. corrections parallel to the initial contour estimation, are performed.

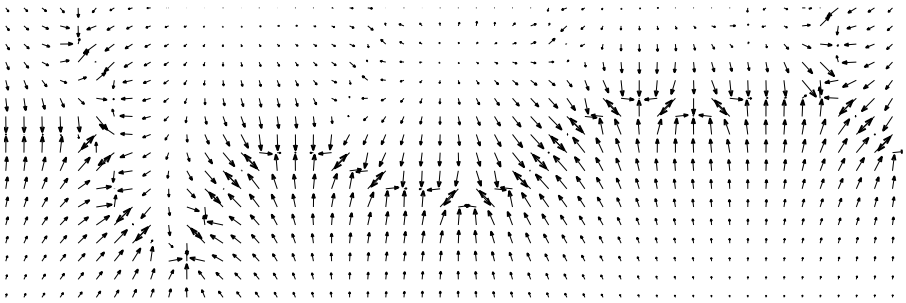


Figure 3.22: Part of the resulting GVF force field.

6. Iteration with different parameters. The accuracy of the contour detection can be improved by iterating the snake algorithm with different parameter sets. In addition this allows for a controlled adaption to fine details while the tie to the initial spline interpolation of ASM landmarks is relaxed.

The result of every iteration is a new estimate for the outline of the bone in $\mathbf{P}$:

$$^{\mathbf{P}}\hat{\mathbf{s}}_{\hat{\mathbf{x}}}^{i+1}(j) = (y_j, j), \quad j = 1, \ldots, n \tag{3.120}$$

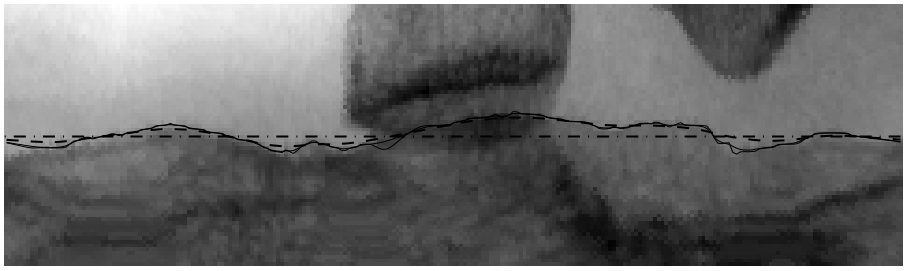


Figure 3.23: Three snakes with different elasticity. Dash-dotted line: ASM initialization, Dashed: snake result with low elasticity, Continuous: two snakes with high elasticity.

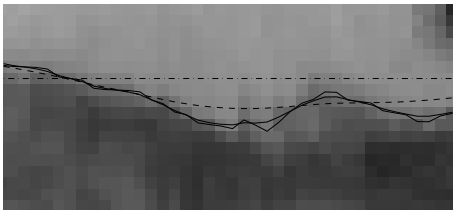


Figure 3.24: Detail of the contour: three snakes with different elasticity. dash-dotted line: ASM initialization, dashed: snake result with low elasticity, continuous: two snakes with high elasticity.

7. Projecting the contour back to the image. The final estimate is projected back into the spatial domain of the original radiograph

$$^{\mathbf{P}}\hat{\mathbf{s}}_{\hat{\mathbf{x}}}^{i_{fin}}(j) \mapsto \hat{\mathbf{s}}_{\hat{\mathbf{x}}}^{i_{fin}}(j) \quad j = 1, \ldots, n \tag{3.121}$$

with respect to the gray value extraction positions, hence

$$\hat{\mathbf{s}}_{\hat{\mathbf{x}}}^{i_{fin}}(j) = p_j(y_j), \quad j = 1, \ldots, n \tag{3.122}$$

is the final estimate for the bone contour on which the landmarks are situated at interpolated positions

$$\hat{x}_j^{i_{fin}} = \hat{\mathbf{s}}_{\hat{\mathbf{x}}}^{i_{fin}}(s_j). \tag{3.123}$$

Generalization to arbitrary dimensions. Note that the concept of GVF snakes as well as ASMs can be generalized to 3 dimensions [70].

3.4 Experiments

The contour identification method was evaluated with respect to (1) improved accuracy of the identified landmarks gained by ASM driven snakes in contrast to standard ASM search and (2) accuracy of the contour located in between landmarks.

3.4.1 Experimental Setup

Tests were performed on radiographs of hands with and without erosive changes. The edge length of a pixel in the data set is 0.0846mm. The ASMs were initialized randomly with varying initialization error. Experimental data consists of Metacarpals of the small finger with manually annotated landmarks. The performance of *ASM* and *ASM driven snakes* were evaluated with two ASM training sets T_{15} (15 training samples) and T_{30} (30 training samples). Each bone was represented by 42 landmarks. The test set for landmark accuracy evaluation consisted of 10 accordingly manually annotated shapes different from the training shapes. The test set for contour identification accuracy in between landmarks consisted of 3 metacarpals affected by rheumatoid arthritis.

3.4.2 Accuracy Improvement of Landmark Positions

Error orthogonal to the contour. The ASM driven snakes decrease position error only orthogonally to the contour. With T_{30} the median orthogonal landmark position error was reduced from the original 3.7 pixels to 1.33(36%) pixels by 3 succeeding snakes with increasing elasticity. (Tab. 3.1)

training set	error	ASM	ASM driven snakes
T_{30}	mean	5.52 (0.467mm)	3.60 (0.305mm)
	median	3.71 (0.314mm)	1.33 (0.113mm)
T_{15}	mean	6.39 (0.541mm)	4.49 (0.380mm)
	median	3.78 (0.320mm)	1.46 (0.124mm)

Table 3.1: Errors (pixels) orthogonal to the bone contour for T_{30} and T_{15}.

In Fig. 3.25 true landmarks, landmarks found by an ASM and landmark positions improved by ASM driven snakes are depicted. On the lower contour of the bone, one can see clearly that the ASM driven snakes landmark estimates denoted by circles, adapt to the true landmark positions far better then the standard ASM landmarks. However no improvement parallel to the contour is made, since the ASM driven snakes algorithm only uses knowledge about gray level profiles orthogonal to the contour.

Reliability evaluated by means of error corridors. The percentage of landmarks lying within a 3 pixel error corridor around the true shape was increased from 40% to 75% by ASM driven snakes. Fig. 3.26 shows the distribution of landmark position errors orthogonal to the contour. Tab. 3.2 and Tab. 3.3 list more detailed observations for both training sets T_{30} and T_{15}.

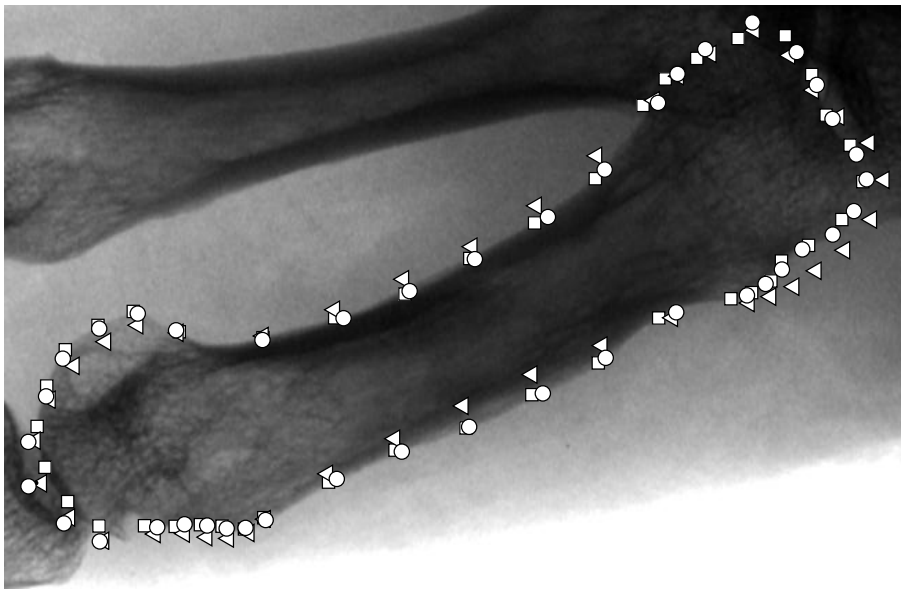


Figure 3.25: Comparison of contour estimates. □ : Manually assigned landmarks. (a) Landmarks detected by an △ ASM and by ∘ ASM driven snakes;

error corridor	ASM	ASM driven snakes
3 pixel (0.25mm)	40.0 %	74.6 %
1 pixel (0.085mm)	13.2 %	40.23 %
0.5 pixel (0.043mm)	6.3 %	21.66 %
0.1 pixel (0.008mm)	1.5 %	4.16 %

Table 3.2: Percentage of landmarks in error corridors for T_{30}.

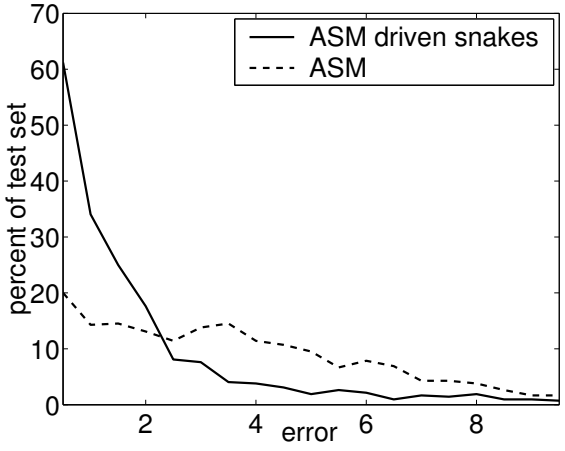


Figure 3.26: Orthogonal error distribution (pixels) for landmarks found by ASM and ASM driven snakes.

error corridor	ASM	ASM driven snakes
3 pixel (0.25mm)	40.2 %	73.2 %
1 pixel (0.085mm)	12.9 %	37.7 %
0.5 pixel (0.043mm)	5.4 %	19.0 %
0.1 pixel (0.008mm)	1.3 %	3.8 %

Table 3.3: Percentage of landmarks in error corridors for T_{15}.

training set	ASM	ASM driven snakes
T_{30}	3.26 (0.276mm)	1.55 (0.131mm)
T_{15}	3.56 (0.301mm)	1.67 (0.141mm)

Table 3.4: Mean errors (pixels) orthogonal to the bone contour for T_{30} and T_{15}. Only landmark estimates with an ASM position error < 10 pixels were included into the statistic.

Effect of training set size. The median error of an ASM trained with T_{15} is 3.78 pixels, training with T_{30} decreased the error to 3.71 pixels, ASM driven snakes based on T_{15} achieve a median error of 1.45 pixels. Thus ASM driven snakes allow for reduction of training set size. In Tab. 3.1 a comparison of mean and median errors achieved with the different training sets is listed.

Limits of ASM driven snakes. The snakes achieve no improvement on parts of the true bone contour that are outside the capture range of the gray level profiles extracted to form $\mathbf{P}$. In Tab. 3.4 statistics are given for points that were located by the standard ASM with an error smaller than 10 pixels orthogonal to the contour. The improvement of position accuracy for this part of the data is better then on the entire test set.

If the bone contour is detected very poorly by the ASM due to structures inside of the bone that have higher contrast than the true bone contour, the ASM driven snakes algorithm can fail to correctly identify the true bone contour. This drawback can be overcome only to some extent during generation of the edge map by using larger orthogonal texture profiles that suppress internal high contrast structures with poor similarity to the trained gray value profile.

3.4.3 Accuracy of Contour Between Landmarks

Evaluation of the bone contour identification accuracy between landmarks was done by visual inspection of the results. A golden standard would be desirable, but manual generation of such data that should provide sufficient accuracy (Note that mean landmark error of the method is $\approx 0.305mm$) is a challenging task not in the scope of this work. Fig. 3.27 shows the result of an ASM (dashed line) compared to the result with ASM driven snakes (solid line) on the joint region of a bone. The ASM driven snake adapts well to the erosion. In the overlapping region of the joint the snake adapts to the true bone contour of the right bone. This is due to the a priori knowledge influencing the GVF vector field, since the gradient direction of the contour is modeled by the ASM.

In Fig. 3.28 a more detailed image of the region around the erosion is depicted. The results of an ASM search (dash-dotted line) and of two snakes with different elasticity parameters (dashed and solid line) are drawn. Both snakes perform equal on the healthy bone contour but differ at the erosion. The possibility to control the adaption capability to fine detail during contour identification enables succeeding algorithms to derive qualitative features from the obtained description by comparing these results.

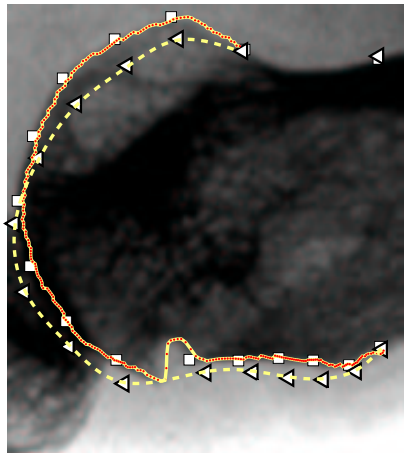


Figure 3.27: Contour in between landmarks. Dashed line: detected by ASM; solid line: detected by ASM driven snakes.

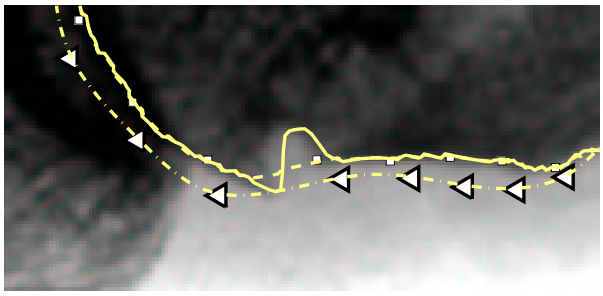


Figure 3.28: Results. Dash-dotted line: ASM search; solid line: ASM driven snake with high elasticity; dashed line: low elasticity.

Chapter 4

Quantification of Destructive Changes

The last step in the automatic assessment of destructive changes caused by rheumatoid arthritis is the automatic quantification of their magnitude in a predefined area of the bone contour. The ASM driven snakes algorithm results in a detailed description of the contour. It therefore allows for direct feature extraction in order to subsequently classify separate segments or points on the contour. The classification is performed with respect to two classes:

- Point is affected by erosive changes,
- Point is not affected by erosive changes.

The scheme of the process is drafted in Fig. 4.1. This chapter is divided into three parts. In Sec. 4.1 the feature extraction procedure is described. It is based on 4 succeeding estimates of the bone contour that differ in accuracy with respect to fine detail, deviations from a healthy reference bone contour generated by the ASM and the image content in the vicinity of the estimates. In Sec. 4.2 the classification of contour points is explained. It is based on standard classification methods. After an outline of the procedure the classification methods used during our experiments are explained briefly. Finally experiment results are reported in Sec. 4.3.

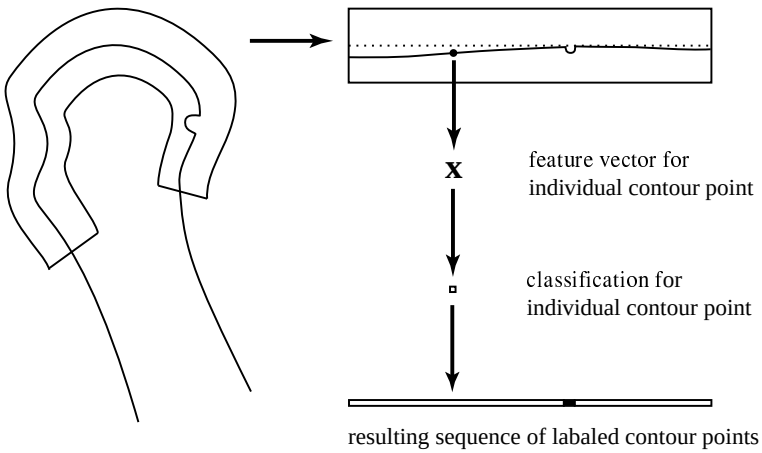


Figure 4.1: Scheme of the quantification process.

4.1 Feature Extraction

Features are extracted from the profile set $\mathbf{P}$ and the identified Contour. For contour identification 3 steps of snake fitting with changing parameters that increase the elasticity of the snake are performed. The contour identification adapts to details with an increasingly fine structure. The estimates are denoted by

- $\hat{\mathbf{s}}_1$ for the initial estimate given by the ASM result.
- $\hat{\mathbf{s}}_2$ for a refinement by a snake with initialization on maxima in the edge map based on $\hat{\mathbf{s}}_1$. The snakes elasticity is rather low. This results in an adaption to smooth deviations from the ASM result. They may be caused by an insufficient training set size of the ASM.
- $\hat{\mathbf{s}}_3$ for a refinement by a snake with initialization on the position of $\hat{\mathbf{s}}_2$. The elasticity of the snake is increased over the preceding snake. This causes the snake to adapt to fine structure while loosening its ties to the initial ASM result.
- $\hat{\mathbf{s}}_4$ for a refinement by a snake with initialization on maxima in the edge map, based on $\hat{\mathbf{s}}_3$. The elasticity of the snake is further increased over the preceding snake.

Note that the procedure is a result of experimental series and can therefore be subject to further improvement. The main property of succeeding snake estimates is that they should adapt to increasingly fine structures, while fulfilling the claim that the influence of the a priori knowledge inherent in the ASM decreases and the assumption that every estimate is closer to the true contour than its predecessor.

Features are extracted for each point of the snake separately, taking into account contour estimates, texture in the vicinity of the point and relations to neighboring points on the estimate. The feature extraction is parameterized with respect to the initial ASM estimate.

4.1.1 Features Used for Classification

The following features are extracted for a point x on the contour. They are considered as experimental tools or 'trial balloons' that provide a certain bandwidth of information to choose from during subsequent steps performed by algorithms.

1. Distance between $\hat{\mathbf{s}}_1(x)$ and $\hat{\mathbf{s}}_2(x)$.
2. Distance between $\hat{\mathbf{s}}_2(x)$ and $\hat{\mathbf{s}}_3(x)$.
3. Distance between $\hat{\mathbf{s}}_3(x)$ and $\hat{\mathbf{s}}_4(x)$.
4. $d(\hat{\mathbf{s}}_3(x) - \hat{\mathbf{s}}_2(x))/dx$.
5. $d^2(\hat{\mathbf{s}}_3(x) - \hat{\mathbf{s}}_2(x))/dx^2$.
6. $d(\hat{\mathbf{s}}_4(x) - \hat{\mathbf{s}}_3(x))/dx$.
7. $d^2(\hat{\mathbf{s}}_4(x) - \hat{\mathbf{s}}_3(x))/dx^2$.
8. Convolution with a kernel $[0.25, 0.25, 0.25, 0.25, 0, -0.25, -0.25, -0.25, -0.25]$ orthogonal to the contour $\hat{\mathbf{s}}_2(x)$ at position $\hat{\mathbf{s}}_2(x)$.
9. Distance between $\hat{\mathbf{s}}_1(x)$ and $\hat{\mathbf{s}}_3(x)$.
10. $abs(d(\hat{\mathbf{s}}_3(x) - \hat{\mathbf{s}}_2(x))/dx - d(\hat{\mathbf{s}}_4(x) - \hat{\mathbf{s}}_3(x))/dx)$
11. Output of a vertical Sobel operator applied around position $\hat{\mathbf{s}}_3(x)$.
12. Gray value at $\hat{\mathbf{s}}_3(x)$.
13. Variance of $\hat{\mathbf{s}}_2(x) - \hat{\mathbf{s}}_3(x)$ in a neighborhood with size 5 of $\hat{\mathbf{s}}_3(x)$.

14. Sum of gray values in a neighborhood with size 5 of $\hat{\mathbf{s}}_3(x)$.

In Fig. 4.2, Fig. 4.3, Fig. 4.4 and Fig. 4.5 the features are visualized. In Fig. 4.6 distributions for the values of the features extracted from the training and test data are plotted. Each plot corresponds to one feature. The two curves are the result of a Parzen estimate of the feature value density function for the two classes of points affected by erosions and healthy points.

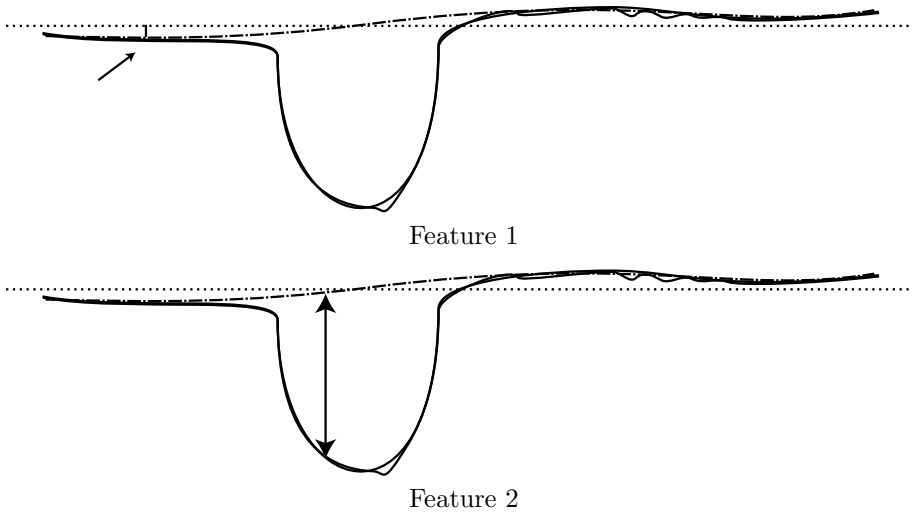


Figure 4.2: Features 1 and 2.

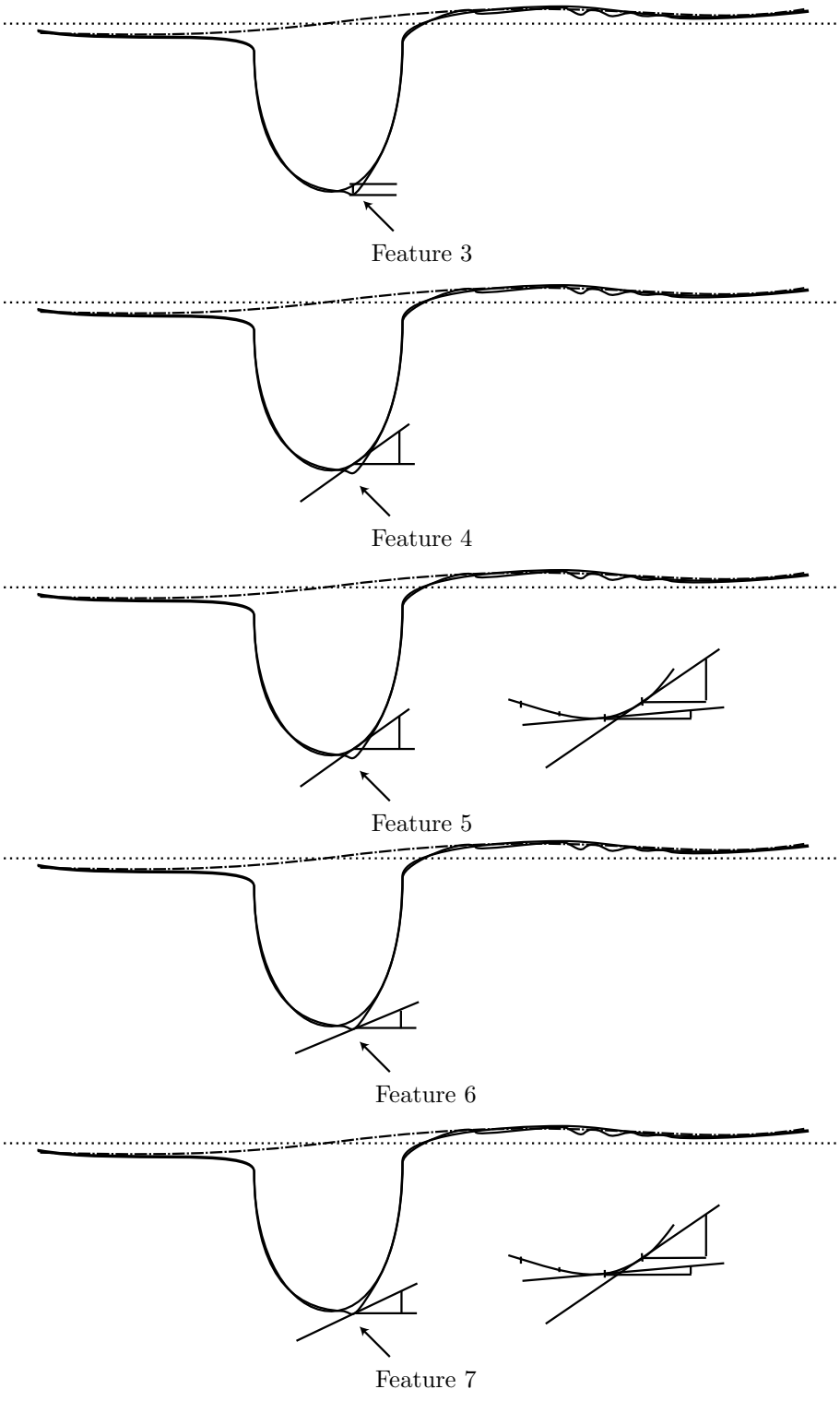
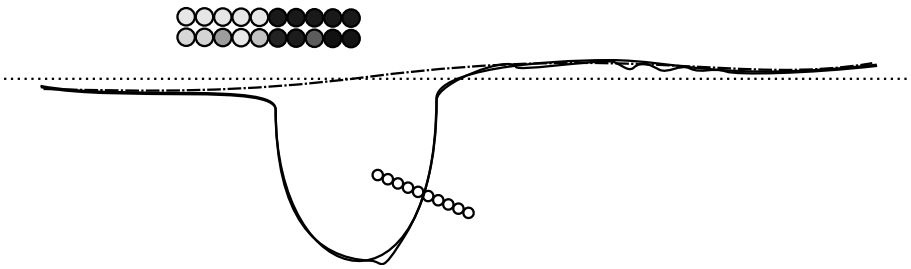
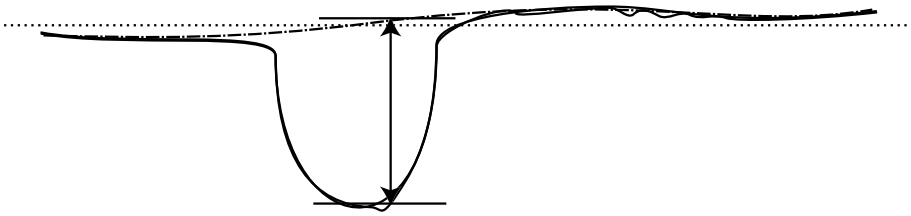


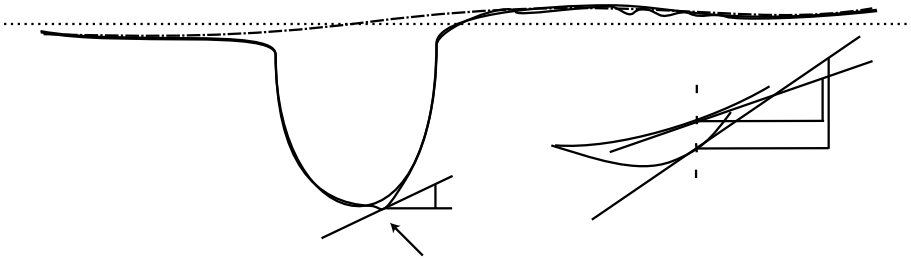
Figure 4.3: Features 3 - 6.



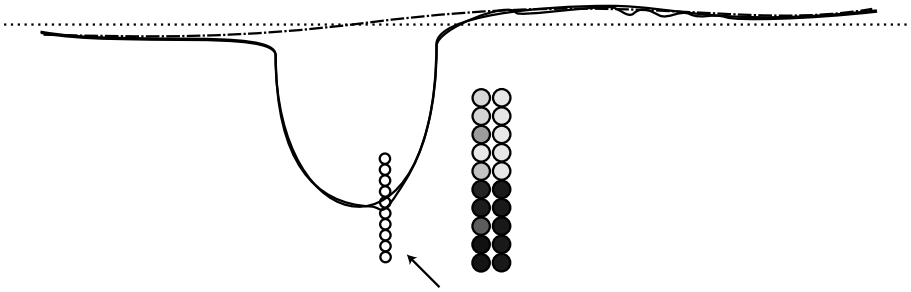
Feature 8



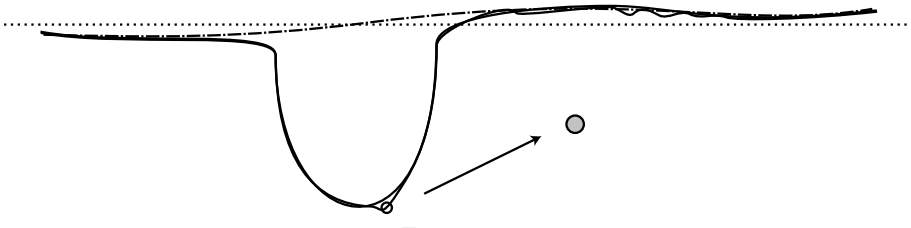
Feature 9



Feature 10



Feature 11



Feature 12

Figure 4.4: Features 8 - 12.

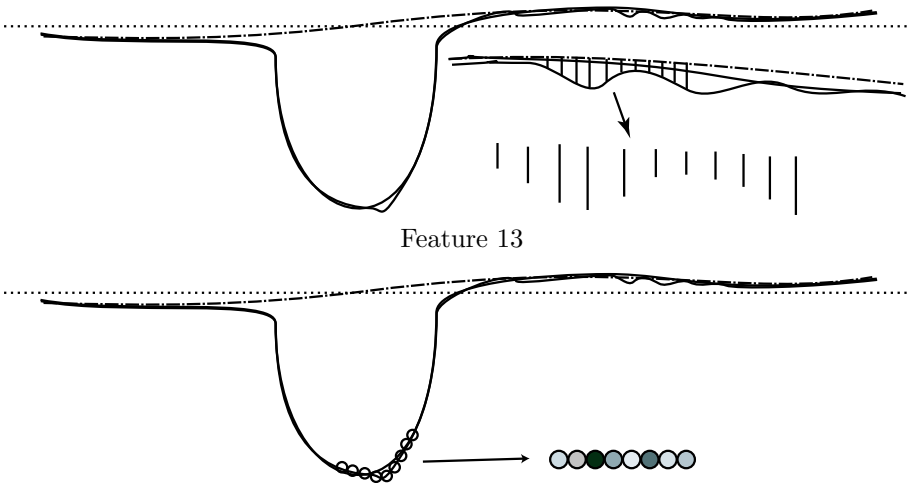


Figure 4.5: Features 13 and 14.

4.2 Classification of the Contour Points

The classification procedure was implemented in MATLAB based on PRTOOLS [17]. Different classifiers were evaluated against each other. Among these were *linear* and *quadratic discriminant functions*, *Parzen classifiers*, *artificial neural nets* and *support vector machines*. The methods will be explained briefly in the following sections although for reasons of compactness references will supersede a detailed discussion.

[66] and [16] give a good overview on methods of statistical pattern recognition and pattern classification including linear and quadratic discriminant functions as well as support vector machines and neural nets.

Classification procedure. As noted in the preceding section features are extracted for each point on the contour of the bone taking into account its properties as well as relations to neighboring points. Based on the representations of the contour points in the feature space they are subject to individual classification. Hence the result is a sequence of points labeled with respect to the two classes τ_1 with points exhibiting erosive destructions and τ_2 with points that are not affected by rheumatoid arthritis.

The resulting sequence of labels assigned to the contour points is enhanced by morphologic operations that obey to prevalent properties of erosive changes namely the unlikeliness of extremely small erosions of the size of one pixel and erosions intermitted by accordingly small intact contour pieces.

This corresponds to the quantification of erosions caused by rheumatoid arthritis usually performed by radiologists. After estimating the extent of the erosions it is put into relation with the entire contour length in the joint region. The sequence of automatically labeled contour points allows for straightforward quantification of the extent of destruction in the joint region parameterized by the ASM.

In the following sections an overview of the utilized standard classifiers is given. Their properties are explained restricted to the *two class* case adequate to RA quantification.

4.2.1 Discriminant functions

Discriminant functions partition the *feature space* $\mathbb{F}^{14}$ i.e. the space were each feature representation of a contour point is a point $\mathbf{x} \in \mathbb{R}^{14}$. The points are labeled according to the class they belong to. In this

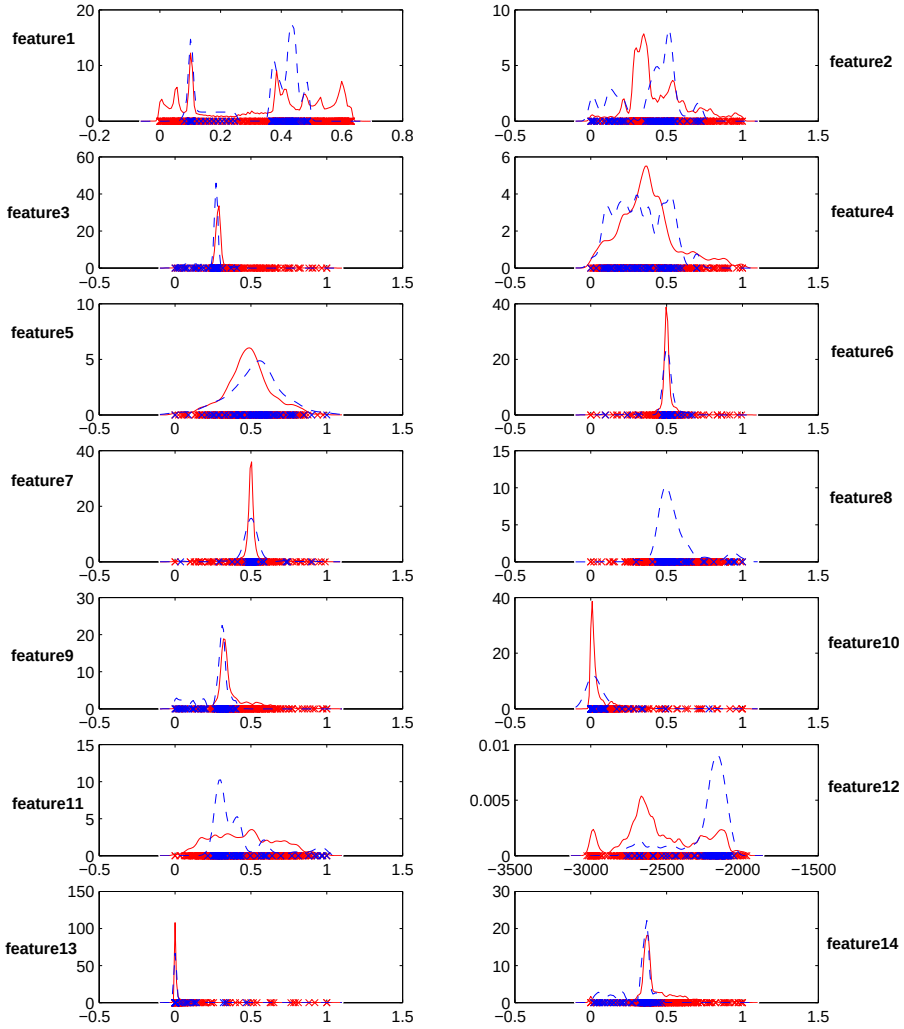


Figure 4.6: Distribution of features. Each plot shows distribution for a feature. Solid line for points within erosions and dashed line for points on intact contour.

text the labels 1 and -1 are denoted by $\mathbf{x} \in \mathbf{C}_1$ and $\mathbf{x} \in \mathbf{C}_2$ respectively. The partition of the feature space that determines regions that contain points belonging to only one class is defined by discriminant functions. For the general classification case there are $c-1$ discriminant functions. For the *two-class* case corresponding to the differentiation between eroded points (τ_1) represented by $\mathbf{C}_1 \subset \mathbb{F}^{14}$ and intact points (τ_2) represented by $\mathbf{C}_2 \subset \mathbb{F}^{14}$ one discriminant function is needed:

$$g : \mathbb{R}^{14} \rightarrow \mathbb{R}. \tag{4.1}$$

$g(\mathbf{x}) = 0$ defines a decision surface and the classifier is implemented by the rules

$$g(\mathbf{x}) > 0 \quad \Rightarrow \quad \mathbf{x} \in \mathbf{C}_1 \tag{4.2}$$
$$g(\mathbf{x}) < 0 \quad \Rightarrow \quad \mathbf{x} \in \mathbf{C}_2. \tag{4.3}$$

Linear discriminant function. A *linear discriminant function* can be written as

$$g(\mathbf{x}) = w_0 + \mathbf{w}^t\mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i \tag{4.4}$$

where $\mathbf{w}$ is a weight vector and w_0 is the *bias* or *threshold weight*. If $g(\mathbf{x})$ is linear then the decision surface is a hyperplane separating the feature space into two parts. It is defined by the $1 + d$ coefficients of $g(\mathbf{x})$. In Fig. 4.7 the scheme of a linear classifier based on the linear discriminant function is depicted. 5 *input units* transfer the input values $x_1, \ldots, x_5$ without alteration. The results are multiplied by the factors $w_1, \ldots, w_5$. In the *output unit* they are summed up together with the bias w_0 and classification is done by the signum function of the result, being 1 for $\mathbf{C}_1$ and -1 for $\mathbf{C}_2$.

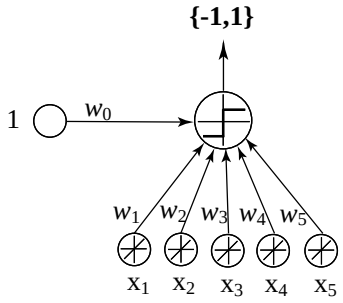


Figure 4.7: Scheme of a linear classifier for a 5 dimensional feature space.

Quadratic discriminant function. A *quadratic discriminant function* can be written as

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i \sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij} x_i x_j. \tag{4.5}$$

$x_i x_j = x_j x_i$ therefore we can assume without loss of generality that $w_i w_j = w_j w_i$. Thus the quadratic discriminant function has $1 + d + d(d+1)/2$ coefficients.

Training of a linear discriminant function. The training of a *two-class linearly separable case* will be explained. For other cases please refer to the literature.

In order to *train* a discriminant function i. e. to determine a function separating the feature space according to manually labeled training examples different methods have been proposed. The coefficients or weights of $\mathbf{g}$ are treated as a *weight vector* $\mathbf{a}$ in a *weight space*. A set of samples $\mathbf{y}_i, \quad i = 1, \ldots, n$ with known classification into the classes $\mathbf{C}_1$ and $\mathbf{C}_2$ is given. These samples are used to determine a linear discriminant function $g(\mathbf{y}) = \mathbf{a}^\top \mathbf{y}$. If a weight vector exists that defines a hyperplane separating the training samples correctly into $\mathbf{C}_1$ and $\mathbf{C}_2$ the set is said to be *linearly separable* and the weight vector is called a *separating vector* or *solution vector*. Thus a solution vector suffices $\mathbf{a}^\top \mathbf{y} > 0$ if $\mathbf{y} \in \mathbf{C}_1$ and $\mathbf{a}^\top \mathbf{y} > 0$ if $\mathbf{y} \in \mathbf{C}_2$.

By replacing all $\mathbf{y}_i \in \mathbf{C}_2$ by their negatives the solution vector has to satisfy

$$\forall i = 1, \ldots, n: \quad \mathbf{a}^\top \mathbf{y}_i > 0. \tag{4.6}$$

Each training sample $\mathbf{y}_i$ places constraints on the possible location of the solution vector $\mathbf{a}$ in the weight space by Eq. 4.6. Thus each training sample restricts the location of the weight vector to a half space and the solution vector has to lie within the intersection of all half spaces called the *solution region*. Therefore in the general case the solution vector is not unique.

To find a solution vector the set of linear inequalities $\mathbf{a}^\top \mathbf{y}_i > 0$ has to be solved. In order to transfer this problem to a minimization problem of a scalar function a *criterion function* $\mathbf{J}(\mathbf{a})$ has to be chosen in a way so that a weight vector that minimizes the criterion function is a solution vector. Thereby the training task can be performed by iterative minimization algorithms like *gradient descent procedures*.

Other algorithms are the *Newton descent* or the *Levenberg-Marquardt method* [47]. The basic gradient descent algorithm updates a randomly initialized weight vector $\mathbf{a}(1)$ by

$$\mathbf{a}(k+1) \leftarrow \mathbf{a}(k) - \eta(k)\nabla\mathbf{J}(\mathbf{a}(k)) \tag{4.7}$$

where $\nabla\mathbf{J}(\mathbf{a})$ denotes the gradient vector of $\mathbf{J}$ on the position $\mathbf{a}$ and $\eta(k)$ is a positive factor called the *learning rate* that determines the step size of the update process. The update is iterated until the gradient falls below a certain predefined threshold ϕ. A good choice of $\eta(k)$ is crucial for the convergence of the process. Small values $\eta(k)$ decrease the speed while high values increase the danger of divergence.

An obvious candidate for the criterion function $\mathbf{J}(\mathbf{a}, \mathbf{y}_1, \ldots, \mathbf{y}_n)$ would be the number of samples not classified correctly by $\mathbf{a}^\top\mathbf{y}$, but since it is piecewise constant gradient search methods fail. A more suitable criterion function is the *perceptron criterion function*

$$\mathbf{J}_p(\mathbf{a}) = \sum_{\mathbf{y}\in\mathcal{Y}_a} (-\mathbf{a}^\top\mathbf{y}) \tag{4.8}$$

where $\mathcal{Y}_a$ is the set of samples not classified correctly by $\mathbf{a}^\top\mathbf{y}$. $\mathbf{J}_p(\mathbf{a})$ is proportional to the sum of distances of the misclassified samples to the boundary of the decision region. It equals 0 iff $\mathbf{a}$ is in the solution region.

The concept of linear discriminant functions - although it is powerful in certain tasks - is not general enough to cope with the majority of real world problems. Namely, the constraint of decision boundaries formed by hyperplanes is insufficient for demanding applications.

4.2.2 Artificial Neural Networks

The restrictions posed by the use of linear discriminant functions can be overcome if nonlinear components are involved in the building process of decision regions. Assuming a proper choice of nonlinear discriminant functions arbitrary decision regions can be obtained. A straightforward approach would be to start with a complete basis of nonlinear functions, like polynomials, and determine the parameters of the solution during training. Due to the large number of parameters compared to the limited number of training samples this is not possible even if the order of the polynomial basis is restricted to some extend. An alternative way to restrict the choice of nonlinearity would be a priori knowledge about the nature of the classification problem. However in the general case were the classifier training has to rely purely on the training samples no such knowledge is available.

The multi layer perceptron. *Multi layer perceptrons (MLPs)* or *multi layer neural networks* implement linear discriminants in a space where the inputs have been mapped nonlinearly and learn parameters governing linear discriminants and nonlinear mapping at the same time.

In principle MLPs can be seen as an extension to the linear discriminant functions as depicted in Fig. 4.7. Lets view a linear discriminant function as a 2-layer network with linear mappings in the input layer. A 3-layer MLP introduces a third layer in between the input and output layer. It is called the *hidden layer* since its units do not communicate directly to the world out of the MLP but only to other layers in the network. General MLPs allow nonlinear mappings in the units. In Fig. 4.8 a 3-layer MLP is depicted. The nonlinear mappings are indicated by sigmoid function. Multi layer neural networks with a sufficient number of hidden units of a general type can simulate any given function.

The training of a multi layer perceptron. One of the key powers provided by these networks is the possibility to use simple training algorithms. Some of them are closely related to the gradient descent in error learning known from linear discriminant functions, while being able to deal with very complex models. The most popular training algorithm is the *back propagation algorithm* or *generalized delta rule*.

An issue of high importance in neural networks is the task of regularization i. e. the selection of the complexity of the network. It involves a trade off between an adequate learning ability for certain training

data making a minimum complexity necessary and the danger of *over learning* i. e. poor generalization of a trained network. Although methods have been proposed to address this problem, knowledge of the data and application domain is still necessary for the appropriate design of a neural network solution.

Please refer to [16] and [29] for a detailed and broad discussion of neural network methods.

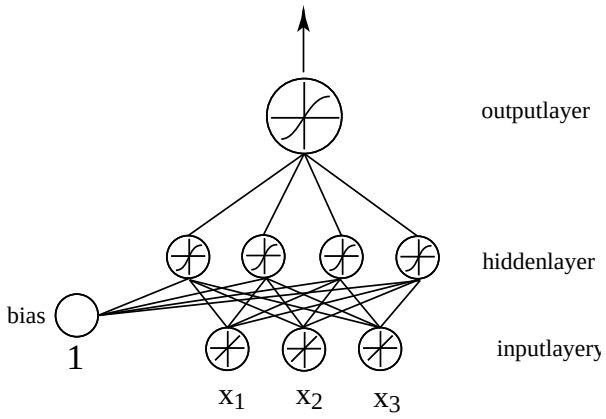


Figure 4.8: Scheme of a neural network with one hidden layer, three input units and one output unit.

Feed forward neural network classification. In Fig. 4.8 a neural network with an input layer consisting of 3 *input units*, a hidden layer with 4 *hidden units* and an output layer with one *output unit* is depicted. A bias is represented by a fourth constant input unit. The units of succeeding layers are connected and a weight is assigned to each of these connections. Each unit builds the sum of the inputs weighted by the corresponding weights and maps it according to a nonlinear mapping. The output of the network is the result of the single output unit denoted by

$$g(\mathbf{x}) = g(x_1, x_2, x_3) \tag{4.9}$$

In general a neural network can exhibit multiple output units. For a 3-layer network the outputs g_k where k is the index of the output unit, n_H is the number of hidden units and d is the number of input units can be expressed by

$$g_k(\mathbf{x}) = f\left(\sum_{j=1}^{n_H} w_{2,kj} f\left(\sum_{i=1}^{d} w_{1,ji} x_i + w_{j0}\right) + w_{k0}\right). \tag{4.10}$$

$w_{1,ji}$ are the weights assigned to the connections between input unit i and hidden unit j, $w_{2,kj}$ are the weights between hidden unit j and output unit k.

The nonlinear mappings or *activation function* assigned to the network units in a back propagation networks are typically step functions or sigmoid functions.

Radial basis function networks. *Radial basis function networks* or *RBF networks* [5, 56] have been introduced motivated by the locally tuned response observed in neurons in the visual or auditory system. AN RBF network an alternative to the back propagation multi layer perceptron differing in the nonlinear mappings assigned to the network units.

Radial basis functions are linear combinations of radially symmetric nonlinear basis functions [66]. In RBF networks they are considered as neuronal activation functions or nonlinear mappings in the network units. Although the architecture of connections is the same as in back propagation networks, RBF network units do not compute their output from the weighted sum of their inputs. The vector of input weights is regarded as a point in the *input space*. The output of a unit is commuted by calculating the distance of the input vector to the weight vector and mapping the result by an RBF.

4.2.3 Support Vector Machines

The idea behind *support vector machines* for a 2 class problem is to find a hyperplane in the feature space that separates the classes optimally. A hyperplane is called optimal in this context if the distance of the closest samples of either of the two classes to the hyperplane called the *margin* is maximal. Fig. 4.9 gives an example in a 2-dimensional feature space. The samples that have a distance to the *separating hyperplane* equal to the margin are called *support vectors*. In Fig. 4.9 they are represented by filled rectangles and circles.

Given 2 classes τ_1 and τ_2 with corresponding labels $y_i = 1$ and $y_i = -1$, let $\omega^\top \mathbf{x} + \omega_0 = 0$ be the optimal separating hyperplane. All points $\mathbf{x}_i$ in the training set have a distance to the hyperplane larger than the margin $b/|\omega|$ or

$$y_i(\omega^\top \mathbf{x} + \omega_0) \geq b \tag{4.11}$$

Then the *canonical hyperplanes* are

$$H_1 \quad : \quad \omega^\top \mathbf{x} + \omega_0 = 1 \tag{4.12}$$

$$H_2 \quad : \quad \omega^\top \mathbf{x} + \omega_0 = -1 \tag{4.13}$$

and for all samples

$$\omega^\top \mathbf{x} + \omega_0 \quad \geq \quad 1 \quad \text{for } y_i = 1 \tag{4.14}$$

$$\omega^\top \mathbf{x} + \omega_0 \quad \leq \quad -1 \quad \text{for } y_i = -1 \tag{4.15}$$

holds. The support vectors lie on the canonical hyperplanes.

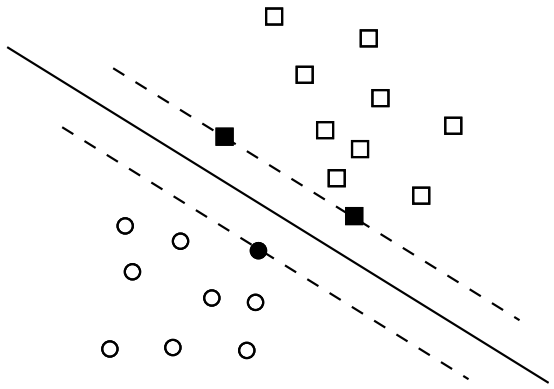


Figure 4.9: Separating hyperplane (solid) and canonical hyperplanes (dashed) in a 2-dimensional space with 2 classes.

Dealing with linear non-separability. Similar to linear discriminant functions the separating hyperplane with maximal margin is restricted to cases were the data is linearly separable. In order to deal with linearly non-separable data two alternatives are possible. Either the data is mapped nonlinearly to a high dimensional space were linear separability is reached or the constraints in Eq. 4.14 and Eq. 4.15 are relaxed. In general the first approach can lead to over-fitting and poor generalization ability [66]. The second approach is performed by introducing *slack variables* $\xi_1, \ldots, \xi_n$ for all n training samples. The constraints are changed to

$$\omega^\top \mathbf{x} + \omega_0 \quad \geq \quad 1 - \xi_i \quad \text{for } y_i = 1 \tag{4.16}$$

$$\omega^\top \mathbf{x} + \omega_0 \quad \leq \quad -1 + \xi_i \quad \text{for } y_i = -1 \tag{4.17}$$

with $\xi_i \geq 0,\ i = 1, \ldots, n$. For misclassified samples $\xi_i > 1$ is part of a cost term that yields an optimal hyperplane by minimization. This corresponds to allowing for misclassification on the training set but penalizing it by the cost function. This causes robustness against outliers to the training algorithm.

Please refer to [66] and [16] for a detailed discussion of the support vector machine approach.

4.2.4 Parzen Classifier

Likelihood ratio classifier. Let τ_i denote class i, P_i denote the a priori probability of τ_i, $p_i(\mathbf{x}) = p(\mathbf{x}|\tau_i)$ the conditional density of τ_i and $p(\mathbf{x}) = \sum_{i=1}^{L} P_i p_i$ the density function, where L is the number of classes. According to Bayes theorem

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{4.18}$$

the a posteriori probability of τ_i given a vector $\mathbf{x}$ can be expressed as

$$q_i(\mathbf{x}) = \frac{P_i p_i(\mathbf{x})}{p(\mathbf{x})}. \tag{4.19}$$

If $p_i(\mathbf{x})$ and P_i are known, classification algorithms or *hypothesis tests* determine the class of a vector $\mathbf{x}$ by minimizing the probability of a classification error depending on q_i. The so called *Bayes classifier* or *Bayesian decision rule for minimum error* assigns the vector $\mathbf{x}$ the class with the higher a posteriori probability. Hence for a two-class problem τ_1, τ_2 the classifier decides

$$q_1(\mathbf{x}) > q_1(\mathbf{x}) \rightarrow \tau_1 \tag{4.20}$$

$$q_1(\mathbf{x}) < q_1(\mathbf{x}) \rightarrow \tau_2. \tag{4.21}$$

The *likelihood ratio classifier* is an alternative formulation of this rule. The *likelihood ratio* l_r leads to the implication

$$l_r = \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} > \frac{P_2}{P_1} \quad \rightarrow \quad \tau_1 \tag{4.22}$$

and vice versa. Since the density function of the training sample distribution is not known explicitly, in general, a way to estimate it from the set of training samples still has to be found.

Parzen density estimate. For a set of n samples $\{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ the density of the distribution can be estimated by the *Parzen density estimate*. The value of the density function at position $x \in \mathbb{R}^m$ is approximated by

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \kappa(x - x_i), \tag{4.23}$$

where κ is called a *kernel function* that is chosen according to the assumed nature of the distribution. In practice it is limited to either Gaussian or uniform distributions in high dimensional feature spaces. In [7] a more detailed discussion of density function estimation is given.

Parzen classifier. The Parzen density estimate can be substituted into the likelihood ratio classifier resulting in the so called *Parzen classifier*

$$\hat{l}_r = \frac{\hat{p}_1(x)}{\hat{p}_2(x)} = \frac{\frac{1}{n}\sum_{i=1}^{n} \kappa_1(x - x_i^{(1)})}{\frac{1}{n}\sum_{i=1}^{n} \kappa_2(x - x_i^{(2)})} \tag{4.24}$$

that classifies a vector $\mathbf{x}$ based on a given set of training samples $\{x_1, \ldots, x_n\}$ with known classification. The classification rule is analogue to the likelihood ratio classifier

$$\hat{l}_r > \frac{P_2}{P_1} \quad \rightarrow \quad \tau_1 \tag{4.25}$$

$$\hat{l}_r < \frac{P_2}{P_1} \quad \rightarrow \quad \tau_2. \tag{4.26}$$

4.2.5 Enhancement of results

The result of the classifying procedure is a sequence of contour points each classified into one of the classes *erosion* or *no erosion* as stated above. Additional improvement of the result can be gained by applying rather simple morphologic operations that reflect knowledge about the nature of erosions to these sequences. In Fig. 4.10 two such operations are depicted. They were designed manually and were applied during our experiments. They enhanced the rate of proper classified contour points. We expect a larger

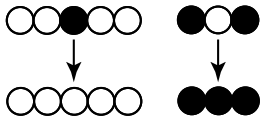


Figure 4.10: Final alterations of classification result.

training set of joints affected by the disease to make more sophisticated methods possible. An interesting approach would by to apply machine learning techniques for sequential data [15] on the training contours.

4.3 Experiments

4.3.1 Experimental Setup

Data. Experiments were performed on 4 profile sets extracted from radiographs which depict bones affected by rheumatoid arthritis. After the contour of the bones had been determined by ASM search, gray value profiles were extracted between predefined landmarks that represented the boundaries of the region of inspection on the joints. The distance of the extracted profiles each orthogonal to the ASM estimate was approximately 1 pixel or 0.0846mm as estimated by the square root of the cord-length on the ASM spline interpolation.

Feature extraction procedure. Based on the initialization $\hat{\mathbf{s}}_1$ resulting from the ASM, the ASM driven snakes algorithm was applied and 3 succeeding approximations of the exact bone contour were generated:

- $\hat{\mathbf{s}}_2$: $\tau = 1$, $\eta = 50$, $\gamma = 70$, snake rigidity:10, initialization on maximal values of edge map in each profile.
- $\hat{\mathbf{s}}_3$: $\tau = 1$, $\eta = 50$, $\gamma = 30$, snake rigidity: 0.8, initialization on $\hat{\mathbf{s}}_2$
- $\hat{\mathbf{s}}_4$: $\tau = 1$, $\eta = 200$, $\gamma = 100$, snake rigidity: 0.01, initialization on maximal values of edge map in each profile.

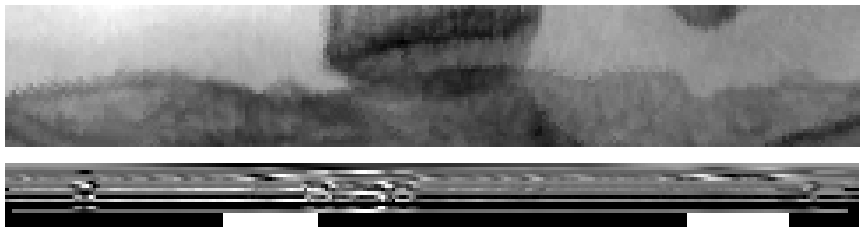
Features were extracted from the resulting contour descriptions for each profile in the profile set. The inspected contour was parameterized according to the initial contour description $\hat{\mathbf{s}}_1$ and each point on the contour was classified regarding to the classes *erosion* and *no erosion*.

Training and succeeding tests were performed on each of the 4 profile sets individually and on the catenation of them in order to evaluate the performance of the classifiers with respect to different characters of erosions. The profile sets were classified manually for ground truth.

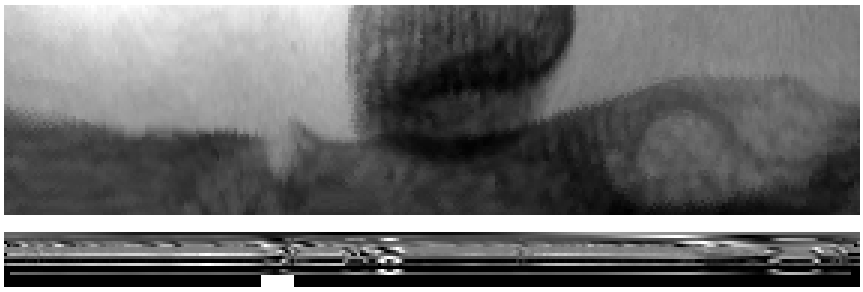
In Fig. 4.11 the profile sets in the experiment data are depicted. Beneath each profile extracted features are visualized as gray values where each column corresponds to one contour point and each row to one feature. The lowest thick lines represent the labels of the contour points in each sub figure. White corresponds to τ_1 or *erosion*, black corresponds to τ_2 or *no erosion*.

Classifiers tested. Different classifiers were tested on the profile sets. Results are reported regarding to the following 12 classifiers.

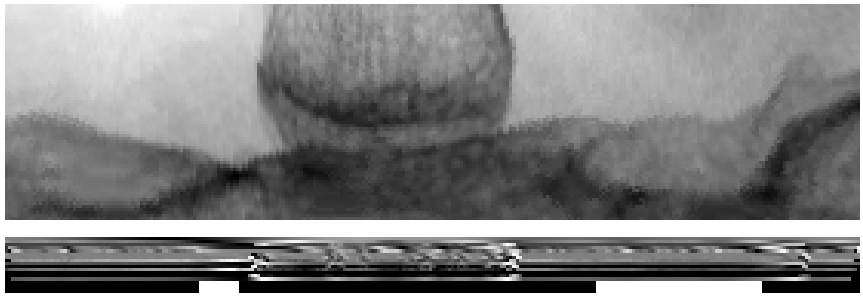
- (c1) Linear discriminant function,
- (c2) Quadratic discriminant function,
- (c3) Parzen classifier,
- (c4) Levenberg-Marquardt Feedforward neural network with 3 hidden units,
- (c5) Radial basis function neural network with 4 hidden units,
- (c6,c7,c8) Back propagation feed forward neural networks with 3, 4 and 5 hidden units,
- (c9) Support vector machines with possible overlap, polynomial proximity mappings and
- (c10,c11,c12) radial basis proximity mappings with different distance scalings.



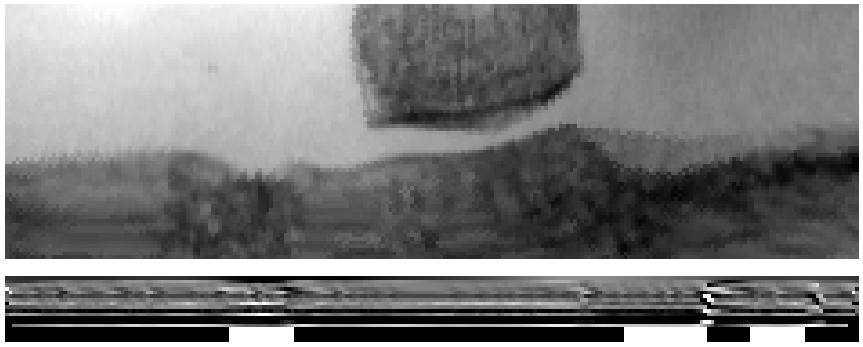
(a)



(b)



(c)



(d)

Figure 4.11: Profiles and corresponding features in the experiment data visualized by gray values. In each image a row corresponds to a feature, a column to a point of the contour in the joint region. Manual assigned labels are represented by white (τ_1) and black (τ_2). They were used for training and for ground truth during tests.

Classifier	False positives	False negatives	Misclassified
c1	5.08	2.54	7.61
c2 -	0	4.06	4.06
c3	0	3.55	3.55
c4 -	0	4.06	4.06
c5	0	3.55	3.55
c6	0	3.05	3.05
c7	0	2.03	2.03
c8	0.51	2.54	3.05
c9	0	2.03	2.03
c10 -	0	4.06	4.06
c11 -	0	4.06	4.06
c12 -	0	4.06	4.06

Table 4.1: Experiments on profile Fig. 4.13b with smooth concave erosion covering 4.06% of the profile.

4.3.2 Experiment Results

Classification results on individual profiles. The classifiers were trained on 30% of the individual profiles. The testing was performed on the remaining 70% of the profile. Fig. 4.13 shows the profiles in the experiment data corresponding to Fig. 4.11. Instead of the features, classification results are depicted for the tested classifiers. Dots denote points belonging to an erosion. The lowest line in each subplot shows the result of manual annotation of labels as used for ground truth.

In Tab. 4.1 classification errors (percentage) for a sharp concave erosion as depicted in Fig. 4.13b are reported. The total number of contour points was 281 of which 84 were used for training while the remaining points formed the test set. Classifiers that did not detect the erosion are marked with a dash. The neural net classifier c7 and support vector machines classifier c9 perform best. Fig. 4.12 shows a projection of points representing the contour in the feature space onto the plane spanned by the first and second feature. Dots represent the contour points positioned within an erosion. The distance between the ASM result and the first two snakes provide relevant information about the position of the erosion. This would imply good performance of even a linear discriminant function (c1). Due to the extremely small number of erosion points contained in the training set the performance results varied to a very high extent. Classifier c1 showed particularly high differences between different training/test cycles. c10 to c12 fail to detect the erosion.

In Tab. 4.2 classification errors (percentage) are given for a joint exhibiting a rough erosion of larger size. The profile set and classification results are depicted in Fig. 4.13d. The total number of contour points was 233 of which 70 were used for training, while the remaining points formed the test set. Similar to Fig. 4.13b the support vector machines c11 and c12 fail to detect the erosion. Linear and quadratic discriminant functions (c1 and c2) perform extremely poor. Best results are obtained by a neural net (c6) and the Parzen classifier (c3).

Classification results on the entire set of profiles. The catenated profiles exhibits 1026 profile points of which 18.5% are affected by erosive damages. Training was performed on 30% or 205 points and 70% or 718 of the data. The training set exhibits equal fractions of points belonging to each of the two classes. The remaining points form the test set.

In Fig. 4.14 manual and automatic classification results are depicted. Fig. 4.14a shows plain classification results, Fig. 4.14b shows results after the labels in Fig. 4.14a were enhanced by morphologic operations. Each vertical line of points corresponds to the results of one classifier. Dots indicate that the point is within an *erosion* according to the classifier. Manual ground truth is depicted in the vertical line on the right of both plots. The classifiers tested are listed in the legend. All of them were provided with the same set of features as described above.

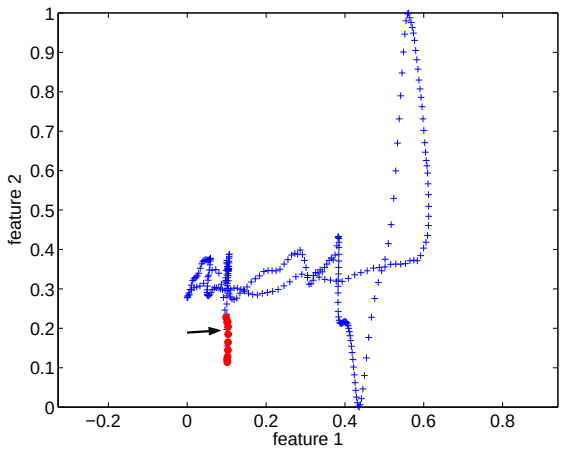
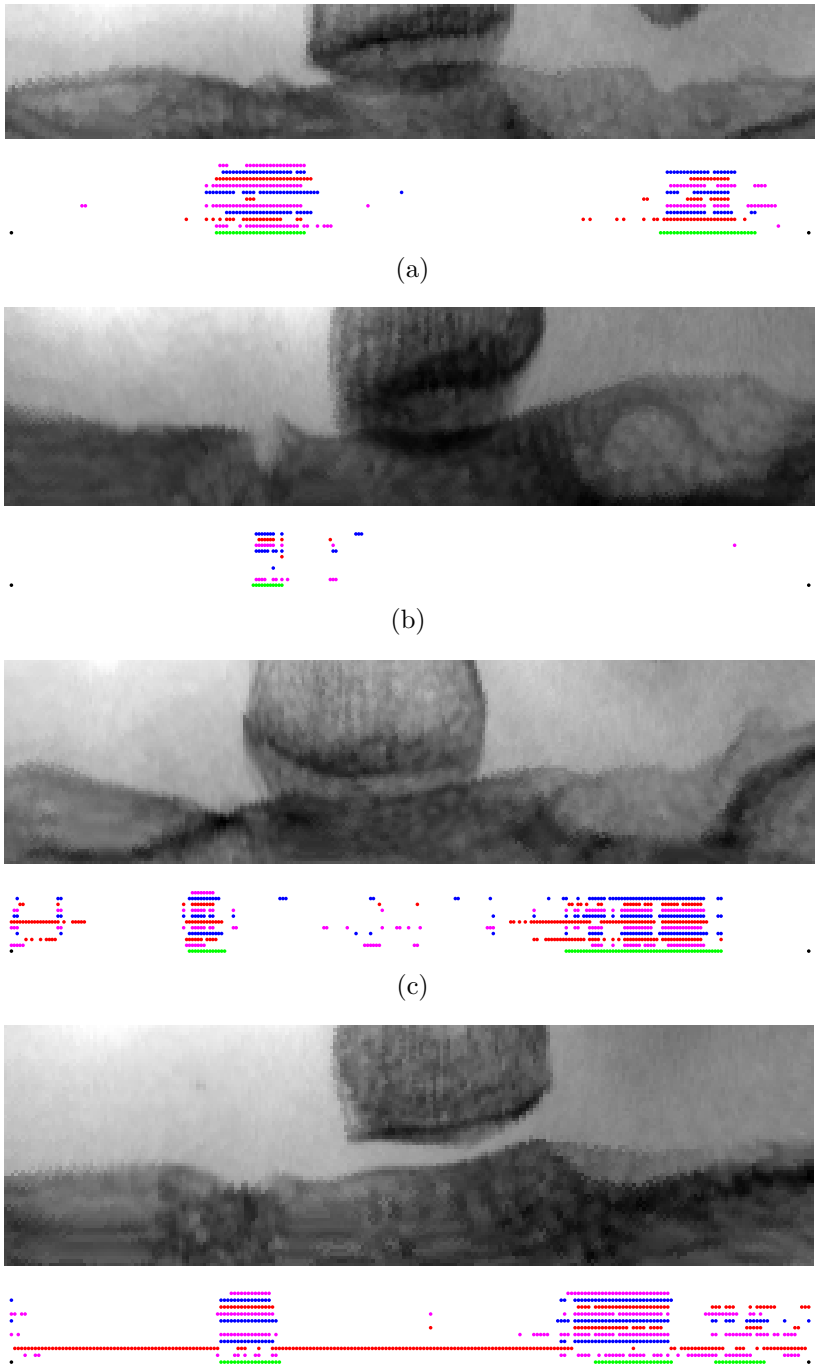


Figure 4.12: Scatter plot of features 1 and 2 for an erosion as depicted in Fig. 4.11 2^{nd} radiograph. Dots (arrow) represent points in the feature space labeled as erosion in the radiograph.

Classifier	False positives	False negatives	Misclassified
c1	7.98	15.34	23.31
c2	72.39	1.84	74.23
c3	3.07	7.36	10.43
c4	5.52	9.2	14.72
c5	6.75	12.88	19.63
c6	5.52	3.68	9.2
c7	8.59	6.75	15.34
c8	7.98	7.98	15.95
c9	3.68	7.36	11.04
c10	0.61	19.02	19.63
c11 -	0	23.93	23.93
c12 -	0	23.93	23.93

Table 4.2: Experiments on profile Fig. 4.13d with rough erosion covering $23,93\%$ of the profile.



(a)

(b)

(c)

(d)

Figure 4.13: Profiles and corresponding classification of contour points. Dots indicate *erosive changes* according to classification. Lowest line: manual indication of erosions; other lines: automated classification. See text for detailed discussion.

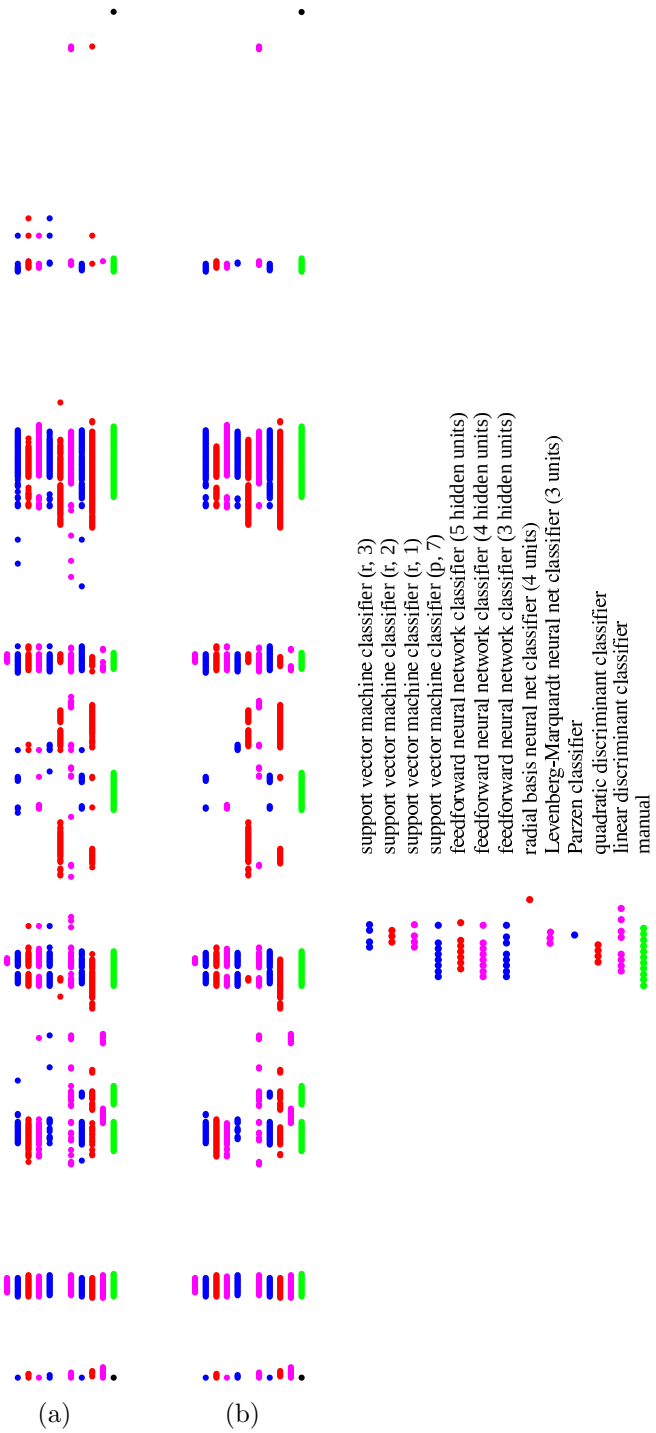


Figure 4.14: Classification on whole dataset. 30% of the data was used for training. (a) shows classification results without enhancement and (b) with morphological enhancement.

Classifier	False pos.	False neg.	Misclassified
Training set size 30%			
c1	2.65	16.85	19.5
c2	17.83	3.2	21.03
c3	1.81	4.74	6.55
c4	5.57	6.82	12.4
c5	12.81	11.56	24.37
c6	2.92	8.5	11.42
c7	2.92	6.96	9.89
c8	3.2	8.5	11.7
c9	1.81	6.13	7.94
c10	0	17.27	17.27
c11 -	0	18.52	18.52
c12 -	0	18.52	18.52
Training set size 70%			
c1	1.3	15.58	15.58
c2	16.23	3.25	3.25
c3	1.62	3.9	3.9
c4	2.92	6.49	6.49
c5	13.96	11.69	11.69
c6	1.3	8.44	8.44
c7	2.6	6.17	6.17
c8	2.27	6.49	6.49
c9	1.3	4.55	4.55
c10	0	13.31	13.31
c11 -	0	18.51	18.51
c12 -	0	18.51	18.51

Table 4.3: Experiments on catenated profiles. 18.52% of the contour points were affected by erosive changes.

In Tab. 4.3 the percentage of false positives, false negatives and misclassified contour points are listed for different classifiers. The Parzen classifier (c3), a neural net classifier with 5 hidden units (c8) and a support vector machine (c9) perform best.

Enhancement of classification results. After classification of the contours, morphological operations were applied on the resulting sequences as described above. Fig. 4.14a shows the sequences before and Fig. 4.14b after the morphological enhancement.

In Tab. 4.4 the numbers of contour points not classified correctly are listed. Plain classifier results and enhanced results are compared for 3 data sets. Two individual profiles A1 (Fig. 4.13b), A2 (Fig. 4.13d) and A, the catenation of 4 profiles are compared.

Morphological operations can improve the performance reasonably since the dimension of the erosions or a discontinuation is typically more than one pixel. Shortcomings of the classification algorithm, that result in a 'noisy' classification can be corrected.

classifier	A1	A1 enhanced	A2	A2 enhanced	A	A enhanced
c1	15	5	38	31	140	133
c2	8	8	121	124	151	121
c3	7	8	17	8	47	20
c4	8	8	24	15	89	44
c5	7	8	32	21	175	138
c6	6	6	15	5	82	47
c7	4	4	25	17	71	28
c8	6	4	26	17	84	41
c9	4	3	18	10	57	28
c10	8	8	32	34	124	130
c11	8	8	39	39	133	133
c12	8	8	39	39	133	133

Table 4.4: Misclassified contour points for two individual profiles (A1 and A2) and the catenation of all profiles (A).

Chapter 5

Conclusion and Outlook

In this thesis an approach to automatically quantify erosive destructions caused by rheumatoid arthritis has been proposed. It is meant to serve as a starting point for a new scoring methodology providing a more accurate and reliable basis to the diagnostic finding process radiologists have to perform during therapy of patients and during clinical trials. Among the main advantages of such an automated method are enhanced reproducibility and accuracy and a reduction of time consumption.

The method consists of three main steps. The first one is the coarse localization of the joints in hand radiographs with local linear mapping nets and Gabor jets. It is based on knowledge about visual appearance and anatomical structure, both of which are learned by the system during a training phase. In this sense the method is transferable to other structures in a straightforward way.

The second step, namely the exact identification of the bone contour is accomplished by an integration of active shape models and snakes. This strategy allows for a control of the influence of a priori knowledge present in ASMs by means of an iterative convergence towards the pathologically deformed bone contour which would be hard or impossible to model. Besides the capability to detect this extremely fine structure the chance to decrease the size of the ASM training set is featured.

The final step comprehends the extraction of features from small sections of the bone contour in a predefined region and the succeeding classification of these sections with respect to the question whether they are affected by erosive changes or not. This data i. e. the destroyed fraction of the joint makes direct quantification of the disease progression possible.

Future work. Future work will concentrate on two issues. On the theoretical side it is planned to explore possible extensions of the concept of active appearance models. On the side of clinical practice future work aims at an implementation of the methods described above and the ones still to be developed in the procedure of clinical trials and patient therapy. This will take place after an extensive testing period during which the capabilities of the automatic quantification approach and a corresponding scoring system has to be evaluated.

After all high interest exists in the process of diagnostic finding performed by human radiologists that accomplish a highly demanding visual task. Parallels between the criteria used by radiologists for a long time and the extracted features that proved most reliable during the automatic detection of erosions are already promising.

Bibliography

[1] E. Allander, P.O. Forsgren, H. Petterson, and P. Seideman. Comuterized assessment of radiological changes of the hand in rheumatic diseases. *Scand J Rheumatology*, 18:291–296, 1989. dealing with joint space measurement. 10

[2] W.F. Ames. *Numerical Methods for Partial Differential Equations*. Academic, New York, 1992. 51

[3] F.L. Bookstein. Two shape metrics for biomedical outline data: Bending energy, procrustes distance, and the biometrical modeling of shape phenomena. In *Proceedings International Conference on Shape Modeling and Applications*, pages 110 –120, 1997. 35

[4] J.M. Boone et al. Neural networs in radiology: An introduction and evaluation in a signal detection task. *Med. Phys.*, 17:234–241, 1990. 12

[5] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988. 71

[6] F. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Machine Intelligence*, 8(6):679–698, 1986. 56

[7] C.H. Chen, L.F. Pau, and P.S.P. Wang. *Handbook of Pattern Recognition and Computer Vision*. World Scientific, 1999. 73

[8] L.D. Cohen. On active contour models an ballons. *CVGIP: Image UNderstanding*, 53:211–218, 1991. 47

[9] T. F. Cootes, A. Hill, Christopher J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–366, 1994. 43

[10] T. F. Cootes and C. Taylor. Statistical models of appearance for computer vision, 1999. 33, 35, 41, 45

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 266–275, Berlin, 1992. Springer. 29, 30

[12] T.F. Cootes and C.J. Taylor. Combining point distribution models with shape models based on finite-element analysis. *IVC*, 13(5):403–409, June 1995. 52

[13] T.F. Cootes and C.J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, 2001. 42

[14] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience, New York, 1953. 50

[15] Thomas G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops*, Lecture Notes in Computer Science, pages 15–30, 2002. 74

[16] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley and Sons, 2001. 67, 71, 72

[17] R.P.W. Duin. Prtools, matlab toolbox for pattern recognition, January 2000. 67

[18] J. Duryea et al. Automated algorithm for the identification of joint space and phalanx margin locations on digitized hand radiographs. *Med. Phys.*, 26:453–461, 1999. 12

[19] J. Duryea, S. Zaim, and F. Wolfe. Neural network based automated algorithm to identify joint locations on hand/wrist radiographs for arthritis assessment. *Med. Phys.*, 29(3):403–411, 2002. 12

[20] Nicolae Duta, Anil K. Jain, and Maria-Pierre Dubuisson-Jolly. Automatic construction of 2d shape models. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 2001. 32

[21] N. Efford. Knowledge-based segmentation and feature analysis of hand-wrist radiographs, 1994. 6, 13, 32

[22] L. Ferré. Selection of components in principal components analysis: a comparison of methods. *Computational Statistics and Data Analysis*, 1995. 41

[23] B. Fritzke. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, 7, 1995. 18

[24] Charles D. Gilbert. *The Cognitive Neurosciences*, pages 73–90. MIT Press, 1995. 15, 16

[25] C. Goodall. Procrustes methods in teh statistical analysis of shape. *Journal fo the Royal Statistical Society, Series B*, 1991. 33

[26] S.N. Gupta and J.L. Prince. Stochastic models for div-curl optical flow methods. *IEEE Signal Processing Letters*, pages 32–35, 1996. 50

[27] G. Hamarneh and T. Gustavsson. Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images. *IEEE Computers in Cardiology*, 27:115 –118, 2000. 53

[28] A. Hill, C.J. Taylor, and T.F. Cootes. Object recognition by flexible template matching using genetic algorithms. In *Proceedings of European Conference on Computer Vision*, pages 852–856. Springer Verlag, 1992. 43

[29] R.M. Hristev. *The ANN Book*. GNU Public License, 1998. 71

[30] H. Imhof et al. Change in process managment by implementing ris, pacs and flat-panel detectors. *Radiologe*, 42(5):344–350, May 2002. 26

[31] D.A. Jackson. Stopping rules in principal component analysis: a comparison of heuristical and statistical approaches. *Ecology*, 1993. 41

[32] Michael F. James, Goeff Heald, John H. Schorter, and Robert A. Turner. Joint space measurement on hand radiographs using computerized image analysis. *Arthritis and Rheumatism*, 38 (7):891–901, 1995. 10

[33] J.P. Jones and L.A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987. 12, 14

[34] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision*, 1:321–331, 1988. 46

[35] D.G. Kendall. Shape-manifolds, procrustean metrics and complex projective spaces. *Bulletin of the London Mathematical Society*, 1984. 34, 35

[36] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990. 39

[37] Georg Langs, Philipp Peloschek, and Horst Bischof. Locating joints in hand radiographs. In *Proceedings of Computer Vision Winter Workshop CVWW*, pages 97–102, 2003. 12

[38] Georg Langs, Philipp L. Peloschek, and Horst Bischof. Asm driven snakes in rheumatoid arthritis assessment. In *Proceedings of the 13th Scandinavian Conference on Image Analysis SCIA 2003*, Lecture Notes in Computer Science. Springer, 2003. 29

[39] A. Larsen, K. Dale, and M. Eek. Radiographic evaluation of rheumatoid arthritis and related conditions by standard reference films. *Acta Radiol Diagn*, 18:481 – 491, 1977. 10

[40] D.T. Lee and B.J. Schachter. Two algorithms for constructing a delaunay triangulation. *Int. J. Computer Information Sci*, 9:219–242, 1980. 23

[41] Eugene T.Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21:363–370, 1989. 54

[42] J. Lee and T. Kunii. Model-based analysis of hand posture. *IEEE Computer Graphics and Applications*, 15:77–86, 1995. 12

[43] B. Leroy, I.Herlin, and L.D. Cohen. Multi-resolution algorithms for active contour models. In *Proc. 12th Int. Conference on Analysis and Optimization of Systems*, pages 58–65, 1996. 47

[44] P.C. Mahalanobis. On tests and measures of groups divergence i. *Journal of the Asiatic Society of Benagal*, 26:541, 1930. 44

[45] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:158–175, 1995. 46

[46] S. Marcelja. Mathematical description of the response of simple cortical cells. *Journal of Optical Society of America*, 70(11):1297–1300, 1980. 14

[47] T. Masters. *Advanced Algorithms for Neural Networks: A C++ Sourcebook*. John Wiley and Sons, 1995. 70

[48] Larry W. Moreland, Michael S. Reddy, William J. Koopman, Richard L. Webber, Graciela S. Alarcon, and Marjorie K. Jeffcoat. Digital subtraction radiography for the assessment of bone changes in rheumatoid arthritis. *The Journal of Rheumatology*, 19:1697–1703, 1992. 10

[49] P.M. Morse and H. Feshbach. *Methods of Theoretical Physics*. New York: McGraw-Hill, 1953. 49

[50] C. Noelker and H. Ritter. Detection of fingertips in human hand movement sequences. In *Proceedings of the International Gesture Workhop, LNCS1371*, pages 209–218, 1997. 12, 22

[51] Claudia Noelker and Helge Ritter. GREFIT: Visual recognition of hand postures. In *Gesture Workschop*, pages 61–72, 1999. 12, 17

[52] Philipp L. Peloschek. *Computerunterstützte radiologische Quantifizierung der rheumatoiden Arthritis*. PhD thesis, University of Vienna, 2000. 9

[53] P.L. Peloschek, G. Langs, H. Bischof, F. Kainberger, C. Krestan, M. Uffmann, and H. Imhof. Automated radiological quantification of hand lesions in rheumatoid arthritis: Accuracy and precision of joint localization (abstract). *European Radiology*, 13(Suppl. 1):201, 2003. European Congress of Radiology 2003, Vienna Austria. 12

[54] P.L Peloschek, J. Sailer, F. Kainberger, K. Boegl, and H. Imhof. Radiologische quantifizierung von gelenkveränderungen. ein methodischer uberblick. *Radiologe*, 40(12):1154–1162, 2000. 10

[55] N. Petkov and T. Lourens. Interacting cortical filters for object recognition. In *Proc. of Asian Conference on Computer Vision, Osaka, Japan*, pages 583–586, Nov 1993. 14

[56] Tomaso Poggio and Federico Girosi. A theory of networks for approximation and learning. Technical Report AIM-1140, Massachusetts Institue of Technology, 1989. 71

[57] S. Sclaroff and A.P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 1995. 32

[58] J.T. Sharp et al. How many joints in the hands and wrists should be included in a score of radiologic abnormalities used to assess rheumatoid arthritis? *Arthritis Rheum*, 28:1326–1335, 1985. 10

[59] N. Shimada et al. Hand gesture estimation and model refinement using monoculat camera - ambiguty limitation by inequality constraints. In *Proc. Of the 3rd Conf. On Face and Gesture Recognition*, pages 268–273, 1998. 12

[60] Eero P. Simoncelli and David J. Heeger. A model of neuronal responses in visual area mt. *Vision Research*, 38:743–761, 1998. 15

[61] M. B. Stegmann. Active appearance models: Theory, extensions and cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, aug 2000. 30, 35, 52

[62] W.M. Tellis and K.P. Andriole. Finding the optimal archiving and communication system (pacs) architecture: a comparison of three pacs designs. *Journal of Digital Imaging*, 14(2 Suppl 1):72–76, 2001. 26

[63] Matthew Turk and Alex Pentland. Eigenface for recognition. *Journal of Cognitive Neuroscience*, 1991. 39

[64] Désirée van der Heijde. Structural damage in rheumatoid arthritis as visualized through radiographs. *Arthritis Res*, 4(2):29–33, 2002. 10

[65] Désirée van der Heijde, A. Boonen, M. Boers, P. Kostense, and S. van der Linden. Reading radiographs in chronological order, in pairs or as single films has important implications ofor the discriminative power of rheumatoid erthritis clinical trials. *Rheumatology*, 38:1213–1220, 1999. 10

[66] Andrew Webb. *Statistical Pattern Recognition* 2^{nd} *Edition*. Wiley, 2002. 35, 38, 67, 71, 72

[67] F. Wolfe. The natural history of rheumatoid arthritis. *Journal of Radiology*, suppl. 44(23):13–22, 1996. 9

[68] Ying Wu and Thomas S. Huang. Capturing articulated human hand motion: A divide-and-conquer approach. In *ICCV (1)*, pages 606–611, 1999. 12

[69] Ying Wu, John Lin, and Thomas Huang. Capturing natural hand articulation. In *Proc. Of IEEE Int. Conference on Computer Vision*, pages 426–432, 2001. 12

[70] Chenyang Xu and Jerry L. Prince. Snakes, shapes and gradient vector flow. *IEEE Transactions on image Processing*, 7(3):359–369, March 1998. 49, 50, 58