Quality Estimation and Segmentation of Pig Backs

Mads Fogtmann Hansen

Kongens Lyngby 2005
Master Thesis IMM-Thesis-2005-100

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Abstract

This thesis explores the possibility of using CT scans of pork bodies to estimate the quality of the pig product "18cm back". It presents the necessary tools for deriving the measures, which are needed to perform a quality estimation. This includes finding the ribs, extracting the 18cm back from the pork middle, sectioning the 18cm back into four parts and finding the meat-fat percentage in the 18cm back.

Pure intensity based classification is an obvious approach for determining the meat-fat percentage as the intensities in a CT scan is given in a relative scale (Hounsfield). The possibility of performing a meat-fat segmentation with a trained linear or quadratic classifier is examined in this thesis. However, pure intensity based classification might function poorly under the presence of intensity inhomogeneities introduced by the CT scanner. A small investigation was conducted in to the matter, and it revealed the presence of scanner introduced artifacts in the used data set.

As an alternative approach, the possibility of performing a shape guided segmentation of the pig back is investigated. An implicit parametric shape model is presented which does not rely on corresponding landmarks. The shape model is later integrated into a region based segmentation framework.
The extensive number of muscles and the small separation between the muscles in a pig back demand for the models to be coupled. A couple of initial attempts of modelling the coupling of the models in to the region based framework is likewise presented.

The basic segmentation framework is tested and compared against an Active Appearance Model on a set of MR images of the Corpus Collasum, while the coupled segmentation framework is tested on a set of CT scans of the pork middle.

Keywords: The Virtual Slaughterhouse, Quality estimation of meat, Rib removal, Radial basis functions, Region based segmentation, Region of interest, Shape models, Implicit surfaces, Level sets, Coupling shape models, CT.

Resume

Denne afhandling undersøger muligheden for at bruge CT scanninger af midterstykket af en gris til at vurdere kvaliteten af produktet "18cm back". De nødvendige værktøjer, som skal bruges for at kunne vurdere kvalititen af en 18cm back, bliver præsenteret. Det inkluderer at finde ribbenene, at lokalisere 18cm back'en i midterstykket, at inddele 18cm back'en i fire dele og at finde kød-fedt procenten i 18cm back'en.

Ren intensity baseret klassifikation er et oplagt valg til at bestemme kød-fedt procenten, da intensiteterne i en CT scanning er givet i en relativ skala (Hounsfield). Dog kan tilstedeværelsen af scanner introducerede inhomogeniteter påvirke en ren intensitet baseret klassifikation så meget, at resultatet bliver utroværdigt. En eksistens undersøgelse blev foretaget på det benyttede datasæt, som påviste tilstedeværelsen af scanner artifakter. Alligevel, gav en lineær og en kvadratisk diskriminant analyse udemærkede resultater på datasættet.

Som et alternativ til den rene intensitet baserede klassifikation bliver det undersøgt, hvorvidt man kan bruge form modeller til at segmentere en griseryg. En implicit form model bliver præsenteret, som ikke behøver punkt korrespondance. Form modellen bliver senere integreret i et region baseret framework. Antallet af muskler og den ringe separation af disse muskler gør det nødvendigt at koble de benyttede modeller. Et par initielle forsøg på at modellere denne kobling ind i det region baserede framework bliver ligeledes præsenteret.

Det basale segmenterings framework bliver testet og sammenlignet med en Ac-

tive Appearance Model på et sæt af MR billeder af Corpus Collasum, mens det koblede segmentaterings framework bliver testet på et sæt CT scanninger af midterstykket på en gris.

Nøgleord: Det Virtuelle Slagteri, Kvalitets estimering af kød, , Radial basis functions, Region baseret segmentering, Region of interest, Form modeller, Implicitte flader, Level sets, Koblede form modeller, CT.

Preface

This thesis has been prepared over nine months at the Section for Image Analysis, Department of Mathematical Modelling, IMM, at The Technical University of Denmark, DTU, in partial fulfillment of the requirements for the degree Master of Science in Engineering, M.Sc.Eng. The extent of the thesis is equivalent to forty ETCS points.

It is assumed that the reader has a basic knowledge in the areas of statistics and image analysis.

Mads Fogtmann Hansen, November 2005

Acknowledgements

Many people have contributed to work done in this thesis either by assisting with theoretical issues, by commenting on the thesis or by providing data.

The pig data sets as well as the outline of the project were provided by the Danish Meat Research Institute (DMRI). I thank all the employees at DMRI, who have helped me, during the course of the project. I special thank goes to Lars Bager Christensen, who has been my contact person at DMRI.

I thank Charlotte Ryberg and Egill Rostrup, Danish Research Center for Magnetic Resonance for providing the MRIs of the Corpus Callosum.

Naturally, a big thank goes to my academic supervisor Associate Professor Rasmus Larsen for the growing support and encouragement throughout the thesis - certainly the thesis would be lacking in quality without your help.
My co-supervisor Bjarne Ersbøll, I thank for his advice and an encouragement in the weekly group meetings.

A well deserved thanks go to all the people, with whom I have shared an office, for providing a pleasant atmosphere and a nice working / study environment. Especially, my good friend Jens Fagertun, who (besides from debugging my code) has helped me with solving many critical issues.

My brother Lars Fogtmann Hansen for proofreading the entire thesis and correcting many of the errors, which had infected the thesis. I really appreciate you for taking the time in a period, where I know, you have been extremely busy. Besides from my brother, numerous people have been forced to comment on parts of the thesis. These people are Lars Bager Christensen, Jens Fagertun, Martin Vester-Christensen, Søren Erbou and Rasmus Engholm - I thank for the many valuable comments.

At last, a thank to my family and all my friends for the support you have given me. I am fully aware that I have neglected you during the course of the thesis (especially towards the end), and I hope to make it up in the near future.

Mads Fogtmann Hansen, November 2005.

Contents

Chapter 1

Introduction

This thesis project is part of a larger study initiated by Danish Meat Research Institute (DMRI) in collaboration with Technical University of Denmark (DTU) - known as the Virtual Slaughterhouse (VSH) among the parties. Before going into the core details about the VSH a small description of DMRI and its purposes is in its place.

1.1 Danish meat research institute

DMRI was established back in 1957 by the Danish Bacon and Meat Council, which is owned by the Danish pig producers. Their main objective is to:

- Become the world leading knowledge center within the meat and slaughter technology
- Provide a link between national as well as international research and the Danish meat industry.

Today, DMRI employs 160 researchers and technicians and is situated at Maglegaardsvej in Roskilde, Denmark.

The research activities of DMRI are divided in to four strategic areas:

- Product quality
- Product safety
- Automation
- External environment

For more information please refer to [7].

1.2 The virtual slaughter house

The virtual slaughter house originated from the "Pig Carcass Classification in the EU project" (EUPIGCLASS) which aims to *improve methodologies to measure, test and monitor the carcass quality throughout the EU* ([8]).

Within the EU, the lean meat content is used as the common reference for the carcass quality and determined by dissection. Unfortunately, dissection is a manual and expensive process why automatic methods for determining the lean meat content with similar precision are being investigated and developed.

One possible approach is to use the non-invasive acquisition technique Computed Tomography (CT) to determine the lean meat content. Obviously, the determination of the lean meat content should be based on quantitative image analysis. To investigate the possibilities of CT the DMRI bought a CT-scanner (baptized Scannerborg) with help from the foundation "Norma & Frode S. Jacobsens Fond".

The primary goal of the VSH is to provide a platform for calibrating measures of meat quality derived from CT against other measures, e.g. dissection and ultrasound. In addition to classification, CT might also bring valuable information to the following areas:

- **Product planning and development**.
  The use of CT as a decision base for the cutting of carcasses in to the final

products will undoubtedly prove profitable both in terms of money and quality, as the amount of waste products and low quality products will be minimized.

- **Automation**:
  When designing slaughter robots the diversity of pigs is a major challenge. CT generated virtual pig models can give valuable information of this diversity.

1.3 Computed tomography

CT was the first acquisition technique to construct a three dimensional image of the internals of an object. The word tomography arises from the greek words tomos (slice) and graphy (describing) and refers to the fact a three dimensional CT image is represented as a series of two dimensional X-ray images.
The theoretical foundation of the CT system was publicized by Allan McLeod Cormack back in 1963 and 1964 in two articles . However, little attention was given to the technique until Godfrey Newbold Hounsfield build the first CT scanner in 1972 at his EMI lab unaware of the work by Allan McLeod Cormack. Both scientist received the nobel prize in 1979 for their contribution to medical imaging.

The two dimensional X-ray images (slices) are generated by rotating a X-ray source around the object. The transmitted radiation from the X-ray beam are picked up by an array of sensors on the opposite site of the source. The sensor data are converted into slice image through a process known as tomographic reconstruction. The voxel intensities of a CT image are given in the Hounsfield scale, which is a relative radio density scale ranging between -1024 and infinity. More precisely, the attenuation of a tissue type is measured relative to the the attenuation of distilled water and air, which are given the values 0 and -1000, respectively.

For more information about CT consult [20].

1.4 The main objective of the thesis

DMRI has already developed semi-automatic methods for a subjective evaluation of the quality of the product 18cm back, which is one of the most profitable products. The method relies on photographic images taken of the 18cm back after it has been cut in to slices.

The primary objective of the thesis is develop an algorithm, which can extract the measures needed to assess the quality of an 18cm back from a CT scan. This involves:

- The development of a method, which can locate the 18cm back in a CT scan.
- The identification of a suitable method for the meat-fat segmentation of the 18cm back as well as possible problems such as the existence of artifacts in the images.
- Finding the dissecting lines which divide the 18cm back in to the loin part and the three tail parts.

1.5 Quality estimation of 18cm pig backs

It is out of scope of this report to discuss quality estimation of 18cm backs in depth. Nevertheless, it is necessary to present the measures, which are needed to perform a quality estimation, to understand the requirements of this project. The name “18cm back” comes from the way it is cut out. From the splitting line of a pork middle, a distance of 18cm is measured perpendicular to the longitudinal axis of the pig. At this distance a right angle cut is made through the middle part to the ribs. The ribs are removed from the final product ”18cm back”. An 18cm back with the dissecting lines is shown in Figure 1.1.

In order to perform the quality estimation, the “18cm back” is divided into the loin and three tail parts. The quality is estimated upon measures obtained from these parts. The measures in question are listed below:

- The total area of the 18cm back.
- The total area of the tale.

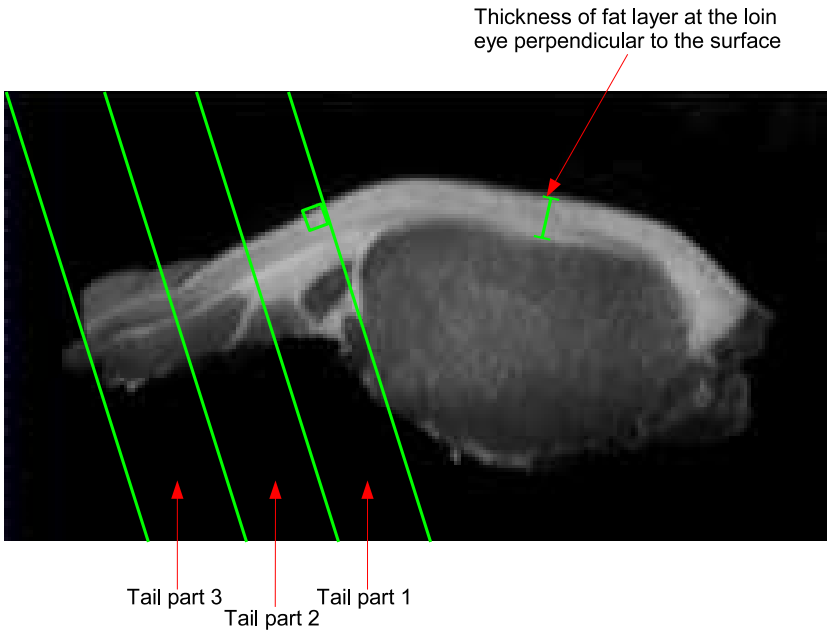


Figure 1.1: Sectioning a pig back for quality estimation. The right most green line is perpendicular to the skin surface, and the remaining lines are parallel to the right most line. The lines are evenly spaced.

- The thickness of the outer fat layer.
- Meat percentages of tail part 1.
- Meat percentages of tail part 2.
- Meat percentages of tail part 3.
- The total meat area.
- The total fat area.

1.6 Thesis overview

The thesis has been divided into three parts which should be read in sequence.

Part I: Quality estimation of 18cm back Introduces methods for finding the ribs, locating the 18cm back and the dissecting lines, and performs an intensity based meat-fat classification. In addition, it conducts a small examination in to the possible existence of inhomogeneities in the used data set.

Part II: Implicit parametric shape model for segmentation Formulates a shape model, which does not rely on point correspondence, and integrates it into a region based segmentation framework.

Part III: Discussion Rounds off the thesis by proposing new ideas for future work and commenting on the results.

1.7 Mathematical notation

This thesis employs typical linear algebraic notation. The essentials are listed below:

Vectors are as default column vectors and denoted with a non-italic lowercase boldface letter: $\mathbf{v} = [a, b, c]^\top$

Vector functions are typeset in non-italic boldface: $\mathbf{f}(\mathbf{v}) = \mathbf{v} + \mathbf{v}$

Matrices are typeset in non-italic uppercase boldface: $\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

1.8 Image symbolics and operators

In the remaining of the thesis all images will denoted with the capital and non-boldface letter I. If there are more than one image in the same context a subscript will be added, e.g. I_a and I_b or I_1 and I_2.
Further, the image value at an pixel s is given by $I(s)$. If there is a need to reference more than one pixel at the time subscripts will be added (e.g. $s_1, s_2, \ldots, s_n$) or the set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ is introduced. The Cartesian coordinates of a pixel s_i will be given the lowercase bold letter $\mathbf{v}_i$, and the coordinates can be indexed individually with the ()-operator, e.g. $\mathbf{v}_i(1)$ denotes the first coordinate of the pixel s_i. The notation $\mathbf{v} = (x, y)$ or $\mathbf{v} = (x, y, z)$ can be used instead of indexing. The {}-operator is used to interpolate an image in a point, e.g. $I\{\mathbf{p}\}$ interpolates the image I at the point $\mathbf{p}$. An image region is denoted with the capital letter R and can, in principle, be represented by any

spatial object or by a level set. Pixels within a region might not necessarily be connected through a four-connectivity neighborhood.

1.9 Nomenclature

I An image.

$E(\cdot)$ An energy function.

N The image dimension.

$\mathbf{\Sigma}$ The covariance / dispersion matrix.

$\mathbf{\Lambda}$ A diagonal matrix of eigenvalues in decreasing order.

$\mathbf{U}$ A matrix of eigenvector ordered after decreasing egienvalues.

λ_i The ith eigenvalue.

$\mathbf{u}_i$ The ith eigenvector.

$\mathbf{\Psi}$ A level set function / signed distance map.

$\mathbf{\Phi}$ A generated level set function.

s A pixel.

$\mathbf{v}$ The Cartesian coordinates of the pixel s.

1.10 Image axes

To avoid confusion, when referring to a specific axis of a 3D image with the letters x, y and z, this section will briefly define the x-axis, the y-axis and the z-axis in a 3D image:

x: The horizontal axis.

y: The vertical axis.

z: The scan axis.

The x, y and z axes are marked in Figure 1.2.

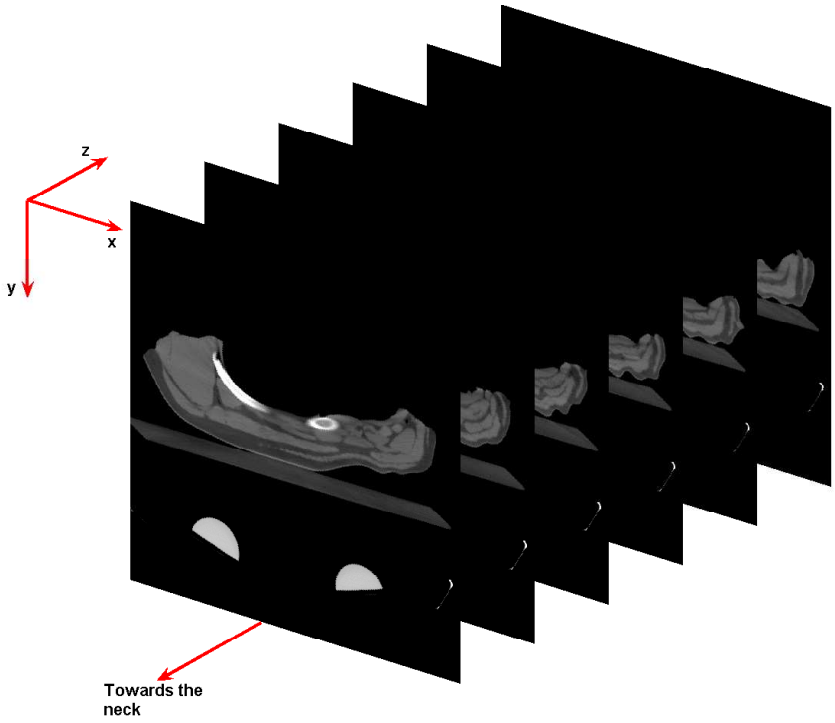


Figure 1.2: The axes in a CT scan.

Chapter 2

Data

Two data sets were available to this thesis project. The first data set was acquired on an older Siemens Somatom Plus scanner from Kemiteknisk at DTU and was available from the beginning of the project. A second data set arrived around 3 month in to the project and was acquired on DMRIs own scanner, a Siemens Somatom ART scanner.
The original idea was to use the first data set for investigation and model building and the second data set for testing. With the arrival of the second data set, it became clear, that this idea was not appropriate as they differed too much. The most important differences were:

- The first data set was made from half pig bodies while the second data set was made from the pork middle. As a direct consequence the pig backs in the second data set are more deformed by gravity than the backs from first data set.
- The quality of the second data set is superior to the quality of the first data set. E.g. the majority of the slice images in the first data set have reconstruction errors in the center.

Naturally, this created a dilemma as whether to use both data set or to keep one them. As the quality of the second data set was superior to the quality of

the first, and as possible new data set would have a quality similar to the second data set, it was decided with approval from DMRI to discard the first data set.

2.1 The second data set

The second data set, which was used in in this project, was acquired on a Siemens Somatom ART scanner from DMRI. The data set consisted of 22 CT scans of pig backs with 100 to 140 slices in each.

Moreover, the slice thickness and slice spacing are 5mm, and the resolution of the slices are 512×512. Partial printout of pig 7 from the data set is shown in Figure 2.1. It is observed that the pig back is not deformed by gravity in the first slices. After a few slices it slowly begins to deform as the rib area becomes smaller.

2.1.1 Reduction of data set

Unfortunately, at the scan time some of the pig backs were not completely inside the field of view of the scanner. Consequently, part of the back is missing in some of the scans. To make matters worse, the missing parts are in the loin area which is located inside the 18cm back. As a result six scans were pruned from the data set. The scan left are 3, 4, 6, 7, 8, 9, 12, 13, 14, 16, 17, 18, 19, 20, 21 and 22.

Further, all the scans have been cropped such that they consist of exactly 60 slices. The primary reason for cropping the data set was to reduce the manual labor of outlining the muscles in the scans[1]. Furthermore, the 18cm back is not visible in all of the slices. From the remainder of the report the non-cropped slices will be indexed from 1 to 60.

2.1.2 Preprocessing

An example of a slice from pig 7 with the corresponding histogram is shown in Figure 2.2.

In the histogram, there are four visible peaks at the intensities -450, -75, 50 and

[1] Explanation follows in the last part of the chapter.

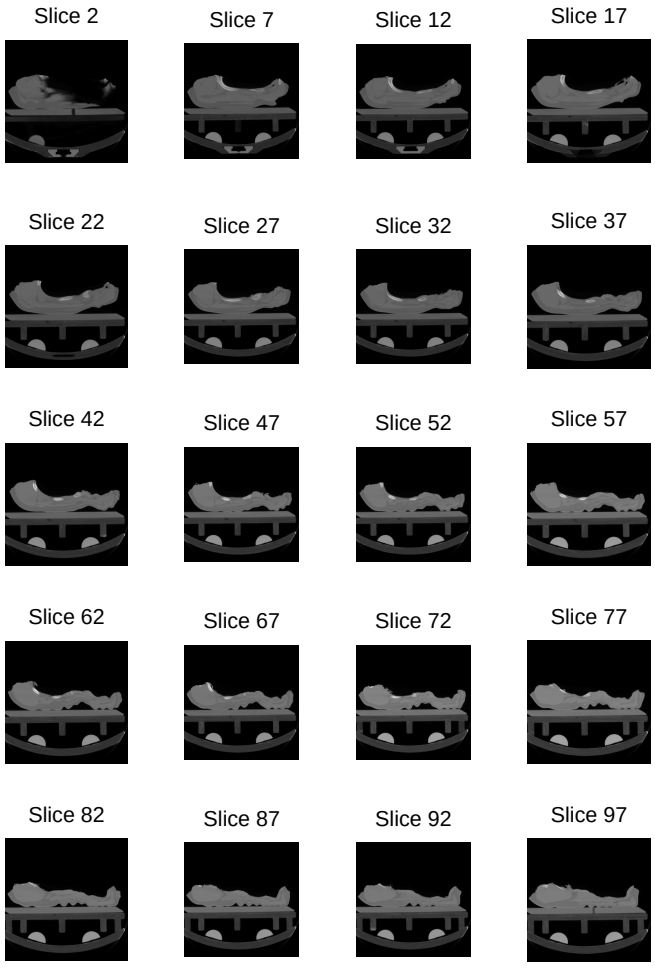


Figure 2.1: Partial print of pig 7 from the second data set.

300, approximately. The peaks at -75 and 50 correspond to meat tissue and fat tissue, respectively. Note, that the peak at 300 is not bone but the plastic rails located at the bottom of the slices. To enhance the contrast between fat, meat and bone in the image, the image is thresholded between -200 and 700. Figure 2.3 displays the resulting image with the corresponding histogram.

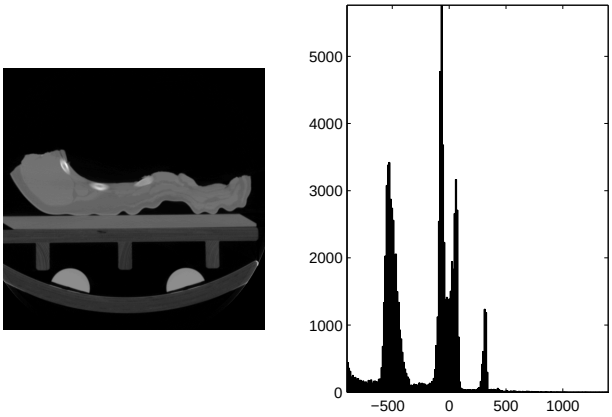


Figure 2.2: Slice image from pig 7 and corresponding histogram (second data set).

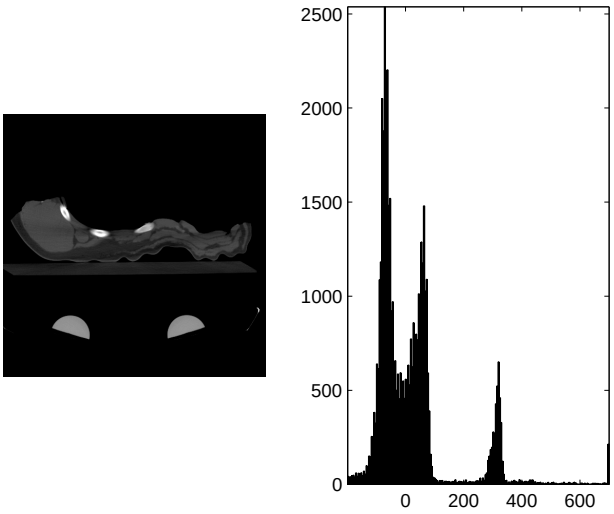


Figure 2.3: Thresholded slice image ([-200, 700]) and corresponding histogram (second data set).

From Figure 2.3, it is observe that there all still "non-pig" objects present in the scan; a plastic board on which the pig back lay and two plastic rails. These objects have intensities in the same range as fat, meat and bone why they cannot be thresholded away. Fortunately, all the "non-pig" objects are situated below the pig back and can consequently be removed, if the row corresponding to the top of the plastic board can be found.

By exploiting the fact, that the top edge of the plastic board has an almost perfect horizontal alignment an algorithm for the removal of the "non-pig" objects can be derived. Figure 2.4 shows the result of convolving a horizontal sobel

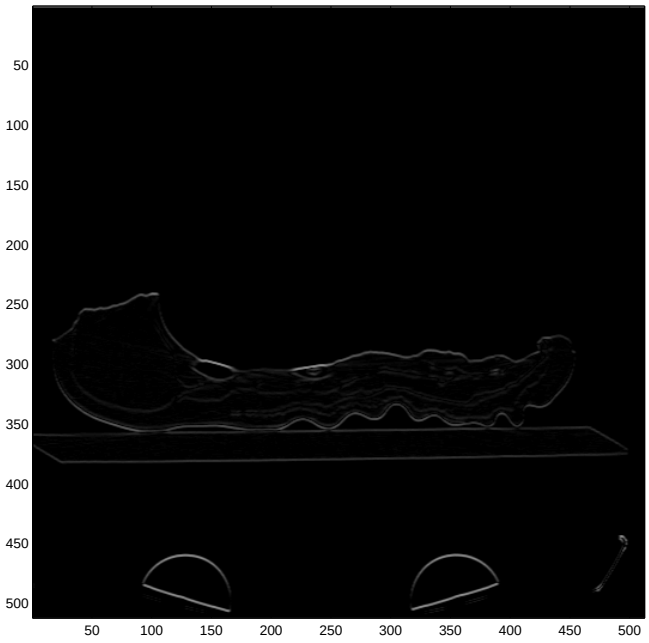


Figure 2.4: Example of pig slice applied with a horizontal sobel filter. The sobel filter used has the size 3×3.

Two relative thin white horizontal lines appears in Figure 2.4. These lines correspond to the top and bottom edge of the plastic board. Consequently, the row sum of the absolute pixel values in a slice image convoluted with a horizontal sobel filter will be highest in the rows corresponding to the top and bottom edge of the plastic board. The algorithm is written formally beneath:

1. Compute the horizontal sobel gradient image.

2. For every row i compute the sum of the absolute pixel values in the row:

$$rs_i = \sum_{j=1}^{N_c} abs\left(\frac{\partial I}{\partial y}(i,j)\right),$$

where N_c is the number of columns.

3. The row t corresponding to the top of the plastic board is found by

$$t = \min(\arg\max_i(\mathcal{R}), \arg\max_i(\mathcal{R} \setminus \max(\mathcal{R}))),$$

where $\mathcal{R} = \{rs_i\}$

In practise, it is recommendable to replace step 2 with

$$rs_i = \sum_{k=i-1}^{i+1} \sum_{j}^{N_c} abs\left(\frac{\partial I}{\partial y}(k,j)\right).$$

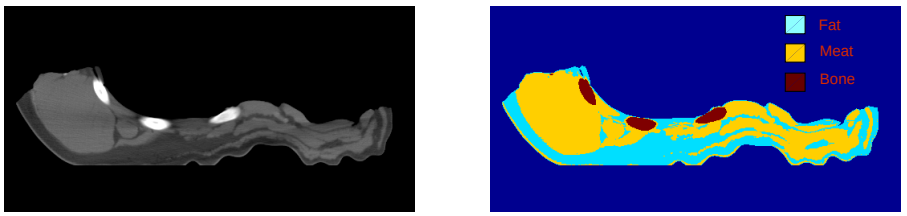
This slight modification adds robustness in cases where the top edge of the plastic board has a slight slope.

At last, a simple segmentation based on visual determination of the optimal intensity thresholds between fat, meat and bone will be performed. From the histogram in Figure 2.3 it seems, that the optimal threshold between meat and fat is just below 0, say -10, and the optimal threshold between meat and bone is just above 100, say 110. Figure 2.5 shows the result of the segmentation. Clearly, the segmentation result shown in Figure 2.5 harmonizes with the one of the human vision system, which indicates that a purely intensity based segmentation might be plausible.

2.1.3 Outlined muscles

A total of six muscles has been outlined by the author in half of the CT scans - the CT scans in question are pig 4, 7, 9, 12, 14, 18, 20 and 22. The incitement is to use the shapes for model building as well as a measure of the ground truth. The database is described in details in Appendix A[2].

[2]The database consists of approximately 1750 shapes.



(a) Preprocessed image (b) Histogram segmented image

Figure 2.5: Segmentation of a slice from pig 7 (Second data set).

The muscles in question are listed below and will for remainder of thesis be denoted by the number at the left of the Latin name:

42 Iliocostalis.

48 Latissimus Dorsi.

50 Longissimus (Loin).

89 Rectus Femoris.

90 Rectus Thoracis.

106 Trapezius.

The muscles above are labelled in Figure 2.6 and a 3D visualization of the outlined muscles of pig 7 is shown in Figure 2.7.

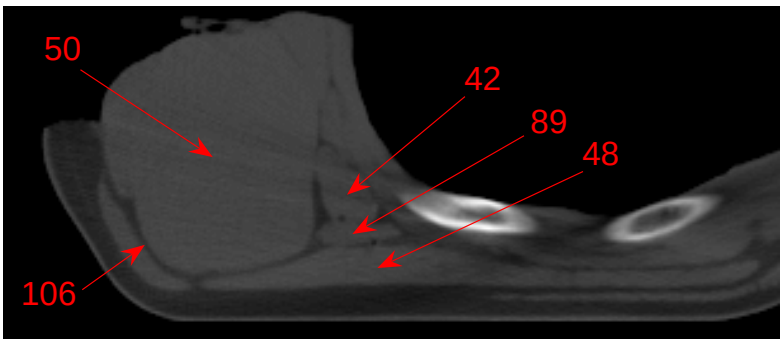
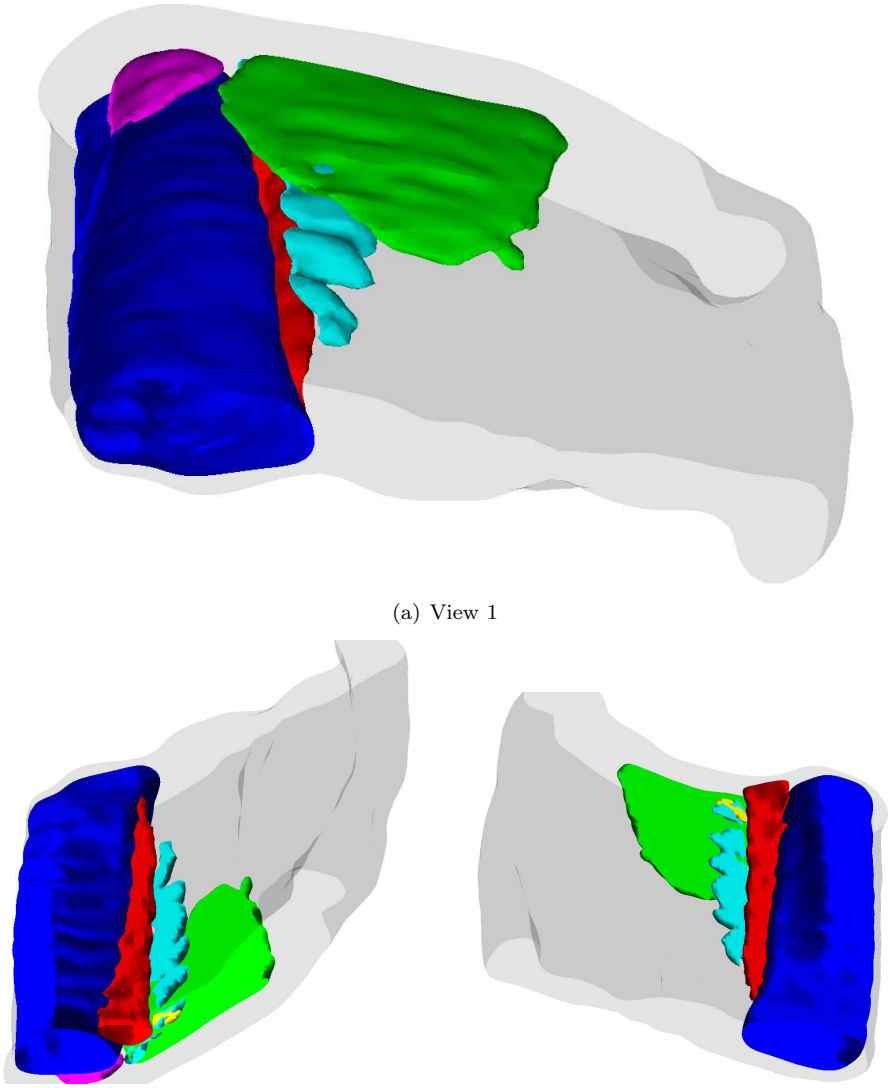


Figure 2.6: The outlined muscles in a 18cm back except from 90, which looks like 89 and is positioned in the same position as 89. However, they never appear in the same slice.



(a) View 1

(b) View 2

(c) View 3

Figure 2.7: 3D visualization of pig 7. Muscle color code: 106=magenta, 90=cyan, 89=yellow, 50=blue, 48=green, 42=red.

Chapter 3

Background theory

This chapter presents common methods used in image analysis and applied mathematics. If the reader is already acquainted with these methods this chapter can be skipped and be used as a reference when needed. The methods, which are covered in this chapter, are:

- Principal components analysis.
- Linear and quadratic discriminant analysis.
- Image warping.
- Radial basis functions.
- Parzen windows.

3.1 Principal component analysis

Principal component analysis (PCA) or Hottelling transformation (named after Harrold Hoteling, who introduced the method in 1933) is a linear orthogonal transformation which rotates the coordinate system such that the maximum

variability of the data is projected onto the axes. Thus, it can be used to reduce the dimensionality of a data set, keeping the subspace which explains the largest amount of variance.
The first principal component is the projection which accounts for the most variance. More formally, the first principal component of a data set with an empirical mean equal to zero is defined by

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} E((\mathbf{w}^\top \mathbf{x}))^2. \tag{3.1}$$

The remaining principal components (PCs) can iteratively be defined with

$$\mathbf{w}_k = \arg\max_{\|\mathbf{w}\|=1} E((\mathbf{w}^\top \hat{\mathbf{x}}_{k-1})^2), \tag{3.2}$$

where

$$\hat{\mathbf{x}}_{k-1} = \mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i \mathbf{w}_i^\top \mathbf{x}. \tag{3.3}$$

Hence, the ith principal components is given by $s_i = \mathbf{w}_i^\top \mathbf{x}$ and $\mathbf{w}_i^\top \mathbf{w}_j = 0$ for $i \neq j$. Further, the direction $\mathbf{w}_i$ is known as the ith principal axis.

In geometrically sense, PCA can be viewed as an rotation of coordinate system maximizing the variance projected onto the axes. This illustrated in Figure 3.1.

(3.1) and (3.2) formally define the principal components, however it is hardly a practical way to determine the PCs. Let $\mathbf{X}$ be a random vector with an empirical mean equal to zero. We seek an orthonormal projection matrix $\mathbf{U}$ such that $\mathbf{Y} = \mathbf{U}^\top \mathbf{X}$ has a diagonal covariance matrix[1]. The covariance matrix of $\mathbf{Y}$ becomes

$$\mathrm{COV}(\mathbf{Y}) = E[\mathbf{Y}\mathbf{Y}^\top] = E[(\mathbf{U}^\top \mathbf{X})(\mathbf{U}^\top \mathbf{X})^\top] = \mathbf{U}^\top E[\mathbf{X}\mathbf{X}^\top]\mathbf{U} = \mathbf{U}^\top \mathrm{COV}(\mathbf{X})\mathbf{U}. \tag{3.4}$$

By multiplication of $\mathbf{U}$ on either side[2] the following equation emerges

$$\mathbf{U}\mathrm{COV}(\mathbf{Y}) = \mathbf{U}\mathbf{U}^\top \mathrm{COV}(\mathbf{X})\mathbf{U} = \mathrm{COV}(\mathbf{X})\mathbf{U}. \tag{3.5}$$

[1] The covariance matrix must be diagonal as the principal components are independent.
[2] $\mathbf{U}^\top = \mathbf{U}^{-1}$ as $\mathbf{U}$ is orthonormal.

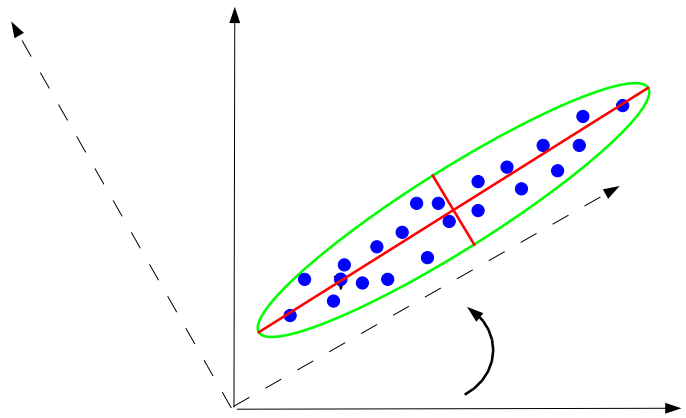


Figure 3.1: PCA illustration in 2 dimension

This is clearly an eigenvalue problem, why the columns $\mathbf{u}_i$ of $\mathbf{U}$ is the eigenvectors of the covariance matrix $\mathrm{COV}(X)$. Consequently, the eigenvectors with non-zero eigenvalues correspond the the principal axis. Further, the eigenvalues λ_j is the projected variance on to the eigenvector $\mathbf{u}_j$.

3.1.1 Selecting the number of principal components

A crucial step in the dimensionality reduction of a data set is the selection of suitable number of principal components. There is no universal answer or solution to the selection of an appropriate number of principal components as it to a large extent depends on the application. Nevertheless, a couple empirical proven rules should be mentioned.

The most intuitive way of selecting a suitable number of principal components k is to select k such that the k most significant components explain some percentage p of the original variance.
Recall, that the λ_i corresponds to the variance along the ith principal axis. Hence, given some suitable p we can find the corresponding k by solving the equation

$$\sum_{i=1}^{k} \lambda_i = \frac{p}{100} \sum_{i=1}^{N} \lambda_i. \tag{3.6}$$

Perhaps, a more qualified way of selecting an appropriate k is to examine how well the original observation can be reconstructed from model. Since, this approach has not been adopted in the project, it will not be described. The reader is referred to [4].

3.2 Classical discriminant analysis

Discriminant analysis (DA) deals with the assignment of an individual to one of a number of known populations or classes. This section will describe linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) - LDA is a special case of QDA. QDA and LDA assume that the populations are normally distributed. LDA further assumes that the population have the same covariance matrix.

3.2.1 The Bayes solution

Let us assume that we know the distribution of the population $\pi_1, \pi_2, \dots \pi_k$. With this knowledge, we want to classify an individual based on a measurement of a set variables

$$\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix}. \tag{3.7}$$

Let $f_i(\mathbf{x})$ and p_i denote the frequency function and prior probability of π_i. Further, the loss of a misclassification is $L(j, i)$. Given a measurement $\mathbf{X}$ of an individual the discriminant score of the ith population can be computed by

$$S_i^*(\mathbf{x}) = -\sum_{j=1}^{k} L(j,i) p_j f_j(\mathbf{x}) = -\sum_{\substack{j=1 \\ j \neq i}}^{k} L(j,i) p_j f_j(\mathbf{X}). \tag{3.8}$$

The Bayes' solution to the decision problem is to choose the population π_v with largest discriminant score $S_v = \max S_i^*$. This is quickly realized by examining the expected loss with respect to posterior distribution $k(\pi_i|\mathbf{x}) = \frac{p_i f_i(\mathbf{x})}{\sum_{j=1}^{k} p_i f_i(\mathbf{x})}$

of π_i:

$$E_i(\mathbf{x}) = \sum_{j=1}^{k} L(j,i)k(\pi_j|\mathbf{x}) \tag{3.9}$$

$$= \sum_{j=1}^{k} L(j,i)\frac{p_i f_i(\mathbf{x})}{\sum_{j=1}^{k} p_i f_i(\mathbf{x})} \tag{3.10}$$

$$= -S_i^*(\mathbf{x})\frac{1}{\sum_{j=1}^{k} p_i f_i(\mathbf{x})}. \tag{3.11}$$

Clearly, selecting the minimal E_i is equivalent to selecting maximum S_i^*. Under the assumption of equal losses ($L(j,i) = 1$)the discriminant score

$$S_i'(\mathbf{x}) = p_i f_i(\mathbf{x}) \tag{3.12}$$

can be chosen instead of (3.8).

3.2.2 Discriminating between several normal distributed population

Let $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ be the distribution of the population π_i. The frequency function of the population π_i becomes

$$f_i(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^p}\frac{1}{\sqrt{\det \boldsymbol{\Sigma}_i}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right). \tag{3.13}$$

By inserting (3.13) into (3.12) and and by discarding the normalization term $\frac{1}{\sqrt{2\pi}^p}$ from (3.13) the following discriminant score appears:

$$S_i(\mathbf{x})^q = S_i'(\mathbf{x}) = -\frac{1}{2}\ln(\det \boldsymbol{\Sigma}_i) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln(p_i). \tag{3.14}$$

This score is known the quadratic discriminant function.

If it is assumed that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_i$ for all populations π_i (equal covariance matrices) the term $-\frac{1}{2}\ln(\det \boldsymbol{\Sigma}) - \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}$ will be common in all S_i''s why it can be

omitted. The resulting discriminant function

$$S_i(\mathbf{x})^l = \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln p_i \tag{3.15}$$

is known as the linear discriminant function.

3.3 Image warping

A full survey of the suggested warping methods in literature is beyond the scope of the thesis. For more detailed information about the subject refer to [9]. This chapter will present a basic image warping technique called affine warping.

Image warping is the task of resampling a source image $\mathbf{I}$ into a target image $\mathbf{I}'$ given a set of n control mapping points $\{(\mathbf{x}_1, \mathbf{x}'_1), \ldots, (\mathbf{x}_n, \mathbf{x}'_n)\}$. More formally $W : \mathbf{I} \in \mathbb{R}^{N_1 \times N_2} \mapsto \mathbf{I}' \in \mathbb{R}^{N_1 \times N_2}$.

From the set of control mapping points we seek a continuous vector function $\mathbf{f} : \mathbf{x}_i \mapsto \mathbf{x}'_i$ which projects any point $\mathbf{x}_i$ in to a new position $\mathbf{x}'_i$. Since $\mathbf{f}$ is continuous the mapping of pixels or voxels will most likely introduce interpolation problems and holes in new image $\mathbf{I}'$, so in practise we look for the reverse mapping $\mathbf{f}' \approx \mathbf{f}^{-1}$.

3.3.1 Piece-wise affine

This is properly the most simple warping function as it assumes that $\mathbf{f}$ is linear locally. Given a triangulation of the point set $\{\mathbf{x}_i\}$ consisting of m triangles $\{t_i = (g_{i1}, g_{i3}, g_{i3})\}$ we can write $\mathbf{f}$ as the sum of m continuous functions $\mathbf{f}_i$:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{m} \mathbf{f}_i(\mathbf{x}), \tag{3.16}$$

where

$$\mathbf{f}_i(\mathbf{x}) = \begin{cases} \alpha_i(\mathbf{x})\mathbf{x}_{g_{i1}}\prime + \beta_i(\mathbf{x})_i\mathbf{x}_{g_{i2}}\prime + \gamma_i(\mathbf{x})\mathbf{x}_{g_{i3}}\prime & \text{if } \mathbf{x} \text{ is inside } t_i. \\ b & \text{if } \mathbf{x} \text{ is outside } t_i. \end{cases} \tag{3.17}$$

In most cases, it is satisfactory to triangulate the convex hull of the point set $\{\mathbf{x}_i\}$ with a suitable triangulation method such as the Delaunay triangulation

in 2D[3]. However, some times one might only want to warp a concave polygon. In such cases a constrained Delaunay triangulation is the best way to perform the triangulation. Consult Section 10.1.2 for more information about Delaunay triangulation.

The question remains how do we find the functions α_i, β_i and γ_i and how do we determine whether a point is inside or outside of a triangle
Let $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ denote the three vertices in a triangle. Any point inside the triangle can then be written as

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_1 + \beta(\mathbf{x}_2 - \mathbf{x}_1) + \gamma(\mathbf{x}_3 - \mathbf{x}_1) && (3.18) \\ &= \alpha\mathbf{x}_1 + \beta\mathbf{x}_2 + \gamma\mathbf{x}_3. && (3.19) \end{aligned}$$

where $\alpha = 1 - \beta - \gamma$ and consequently $\alpha + \beta + \gamma = 1$. For a point $\mathbf{x}$ to be inside we further require that $0 \leq \alpha, \beta, \gamma \leq 1$. For the transformation to be affine the relative position of $\mathbf{x}$ in the target image $\mathbf{I}'$ must be

$$\mathbf{x}' = \alpha\mathbf{x}'_1 + \beta\mathbf{x}'_2 + \gamma\mathbf{x}'_3. \quad (3.20)$$

Hence, the value of the functions α_i, β_i and γ_i in a point $\mathbf{x}$ are found by solving a linear system of equations with two unknowns and two equation.

$\mathbf{f}$ might be continuous however it is not differential. Consequently, a straight line, which crosses the boundary between to triangles, might be kinked in the mapped space - see Figure 3.2.

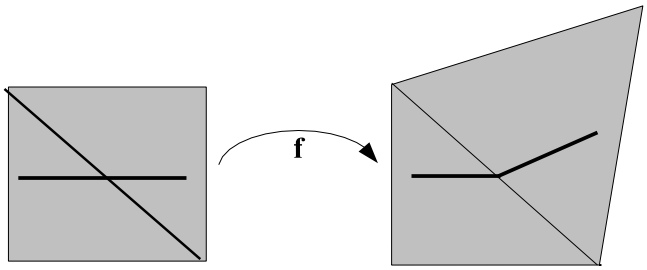


Figure 3.2: Affine warping might introduce kinks in straight lines, which cross triangle edges.

Naively, one can iterate trough all pixel or voxels in an image in order to perform the warping process. Clearly, (3.16) stated that no warp will occur if a pixel or

[3]Note, the Delaunay triangulation does extend to 3D but the complexity increases significant.

voxel is outside any for the triangles. Hence, it is much better to warp all pixel inside a given triangle to the corresponding triangle $\mathbf{I}'$ at ones. Algorithm 1 does exactly that. The function `ROUND`($\mathcal{P}$) takes a set of points $\mathcal{P}$ as argument and convert the set of points to indices by rounding, the function `SORT_ACCENDING`($\mathbf{v}$) sorts a vector $\mathbf{v}$ in accenting order, and finally `WARPLINE`(i,$ends$,...) warps the part of the ith horizontal line in the target image, which is between $ends$.

Algorithm 1 Warp 2D image

Require: A set of triangles $\mathcal{T}$, two sets of points $\mathcal{P}$ and $\mathcal{P}'$
1: ROUND($\mathcal{P}'$)
2: **for all** $t_i \in \mathcal{T}$ **do**
3: Get vertices $\mathbf{x}_1 = \mathcal{P}'(t_i(1))$, $\mathbf{x}_2 = \mathcal{P}'(t_i(2))$, $\mathbf{x}_3 = \mathcal{P}'(t_i(3))$
4: Rearrange $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ such that $\mathbf{x}_1(2) \leq \mathbf{x}_2(2) \leq \mathbf{x}_3(2)$
5: Find the intersection point $\mathbf{x}_{int}$ between the horizontal line that goes through $\mathbf{x}_2$ and the line trough $\mathbf{x}_1$ and $\mathbf{x}_3$
6: Calculate $\mathbf{v}_1 = \mathbf{x}_2 - \mathbf{x}_1$ and $\mathbf{v}_2 = \mathbf{x}_{int} - \mathbf{x}_1$
7: Normalize $\mathbf{v}_1$ and $\mathbf{v}_2$ such that $\mathbf{v}_1(2) = 1$ and $\mathbf{v}_2(2) = 1$
8: **for** $i = 1$ to $p_2(2) - p_1(2)$ **do**
9: $ends(1) = p_1(1) + i * v_1(1)$ and $ends(2) = p_1(1) + i * v_2(1)$
10: SORT_ACCENDING($ends$)
11: WARPLINE($p_1(1) - i$, end,t_i, $\mathcal{P}$, $\mathcal{P}'$)
12: **end for**
13: Calculate $\mathbf{v}_1 = \mathbf{x}_2 - \mathbf{x}_3$ and $\mathbf{v}_2 = \mathbf{x}_{int} - \mathbf{x}_3$
14: Normalize $\mathbf{v}_1$ and $\mathbf{v}_2$ such that $\mathbf{v}_1(2) = 1$ and $\mathbf{v}_2(2) = 1$
15: **for** $i = 1$ to $p_3(2) - p_2(2)$ **do**
16: $ends(1) = p_1(1) - i * v_1(1)$ and $ends(2) = p_1(1) - i * v_2(1)$
17: SORT_ACCENDING($ends$)
18: WARPLINE($p_3(1) - i$, end,t_i, $\mathcal{P}$, $\mathcal{P}'$)
19: **end for**
20: **end for**

3.4 Radial basis functions

A radial basis function (RBF) approximation takes the form

$$F(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi(\|\mathbf{x} - \boldsymbol{\mu}_i\|), \tag{3.21}$$

where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n$ are the set of *centers* in $\mathbb{R}^d$, $\|.\|$ is the Euclidian distance and $\phi(r)$ is the basis function. Common choices of basis functions or kernel

functions are:

- Piecewise linear spline: $\phi(r) = r$.
- Thin plate spline: $\phi(r) = r\log(r)$.
- Cubic spline: $\phi(r) = r^3$.
- Gaussian: $\phi(r) = e^{-\left(\frac{r}{c}\right)^2}$
- Multiquadric: $\phi(r) = \sqrt{(r^2 + c^2)}$

In many cases (3.21) is extended with a linear and constant term:

$$F(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi(\|\mathbf{x} - \boldsymbol{\mu}_i\|) + \boldsymbol{\alpha}^\top \mathbf{x} + c. \tag{3.22}$$

3.4.1 Interpolation with RBF

Given a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ and a set of interpolation values $\{y_1, y_2, \ldots, y_m\}$, the points can be interpolated in the values with (3.21) by solving the linear system of equation[4]

$$\begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m,1} & \phi_{m,2} & \cdots & \phi_{m,m} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$
$$\boldsymbol{\Phi}\mathbf{w} = \mathbf{y}, \tag{3.23}$$

where $\phi_{i,j} = \phi(\|\mathbf{x}_j - \mathbf{x}_i\|)$.

Similar, the point set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ can be interpolated in set of interpolation values $\{y_1, y_2, \ldots, y_m\}$ with (3.22) by solving the following set of equations

$$\begin{bmatrix} \boldsymbol{\Phi} & \mathbf{S}^\top \\ \mathbf{S} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\alpha} \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \tag{3.24}$$

[4] All points becomes *centers*.

where $\mathbf{S} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$.

3.4.2 Approximation with RBF

Many cases interpolation is infeasible as it requires the inversion of an m-by-m symmetric matrix. Consequently, in cases with a large m approximation is used instead.
Given a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, a set of values $\{y_1, y_2, \dots, y_m\}$ and a set of centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_m\}$, the points can be approximated in the values with (3.21) by solving the over-determined linear system of equation

$$\begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,n} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m,1} & \phi_{m,2} & \cdots & \phi_{m,n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$
$$\boldsymbol{\Phi}\mathbf{w} = \mathbf{y}, \tag{3.25}$$

where $\phi_{i,j} = \phi(\|\mathbf{x}_i - \boldsymbol{\mu}_j\|)$. Since the system of linear equations is over-determined it is solved using the pseudo-inverse.

A similar set of equations can be derived for (3.22) given the point set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, a set of values $\{y_1, y_2, \dots, y_m\}$ and a set of centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_m\}$:

$$\begin{bmatrix} \boldsymbol{\Phi} & \mathbf{S}_c^\top \\ \mathbf{S}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\alpha} \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \tag{3.26}$$

where $\mathbf{S}_c = \begin{bmatrix} \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 & \dots & \boldsymbol{\mu}_m \\ 1 & 1 & \dots & 1 \end{bmatrix}$ and $\mathbf{S}_d = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_m \\ 1 & 1 & \dots & 1 \end{bmatrix}$.

The most common way of choosing the centers is to use a grid with some predetermined spacing. The minimum and maximum value of the grid are determined in every dimension by calculating the minimum and maximum values of the set of points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ in every dimension.
If the points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ have a tendency to group together clustering is often a very effective strategy. The approach is often used in neural networks.

3.4.3 Using RBF for modelling implicit surfaces

An implicit surface is defined by $\{\mathbf{x} : f(\mathbf{x}) = 0\}$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$. To approximate the embedded function f a set of constraint points $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ and a set of values $\{r_1, r_2, \ldots, r_n\}$ need to be specified. Constraint points can either be:

- **Boundary points** are points on the surface of the object. Thus, they receive the value 0.
- **Interior points** are points inside the object. Thus, the receive negative values.
- **Exterior points** are points outside the object. Thus, they receive positive values.

Surely, (3.21) and (3.22) can be used to interpolate f. More information about implicit surfaces can be found in [27].

3.4.4 Regularization

The presence of noise in the data points might lead to the interpolation or approximation of a non-smooth surface. To avoid non-smooth surfaces it might be necessary to penalize high curvature - this is known as regularization.
The regularized version of (3.24) is

$$\begin{bmatrix} \mathbf{\Phi} + \lambda\mathbf{I} & \mathbf{S}^\top \\ \mathbf{S} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\alpha} \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \tag{3.27}$$

where λ control the weight between the fitness to the data points and smoothness of the surface. For a more in depth description of regularization the reader can consult [11].

3.5 Non-parametric density estimation

Many signal or image processing application requires knowledge about the density of the classes. Often, the form of the density function (e.g. normal distri-

bution) is known or at least assumed to known, and in these cases a parametric estimation scheme is used. Given n samples a non-parametric estimate of a pdf can be estimated with the parzen window estimator

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi \left(\frac{x - x_i}{h_n} \right), \tag{3.28}$$

where φ is a kernel function.

Part I

Quality estimation of 18cm backs

Chapter 4

Introduction to quality estimation of pig backs

In Chapter 2, we saw evidence that a purely intensity based classification into meat and fat might be plausible with the second data set. If this is the case the task of producing the measurements needed to perform a quality estimation of an 18cm back simply becomes a matter of (i) finding the 18cm back, (ii) dissecting the back into the loin part and the three tail parts and (iii) performing a simple intensity thresholding. Nevertheless, it is a well known fact that many acquisition devices for non-invasive imaging introduce artifact in form of intensity inhomogeneities. This may or may not be depended on the anatomy of the object. In MR imaging, finding and removing bias fields is a major area. Since inhomogeneities can have a large degree of impact in the performance and result of an intensity based segmentation, a small investigation in to the issue is presented in this part. This part contains the following chapters:

- Locating and removing the ribs and other bone.
- Locating the 18cm back and the three dissection lines.
- Investigation of inhomogeneities.
- Estimating the meat-fat percentages.

Chapter 5

Removal of ribs and bone fragments

Correct removal of bone from the back is essential to the quality estimation of the 18cm back for several reasons:

- Bone is not present in the final 18cm back product, why it should not be allowed any effect on the estimation of the quality.
- In order to locate the 18cm back, it necessary is to find the outside surface of the rib cage.

Certainly, the removal of all bone from the back could be done by intensity thresholding. In practise however, when the ribs are separated from the back the areas between the ribs are removed as well.

5.1 Locating point on the ribs

The task of locating points on ribs can be done easily with a half circle scan from a base point. This is illustrated in Figure 5.1. The half circle is divided

into a number of sample directions. Along each sample direction the method searches for bone and marks the spot where it hits the bone for the first time, and the spot where it leaves the bone for the first time. The latter is assumed to be an outside point on the ribs.

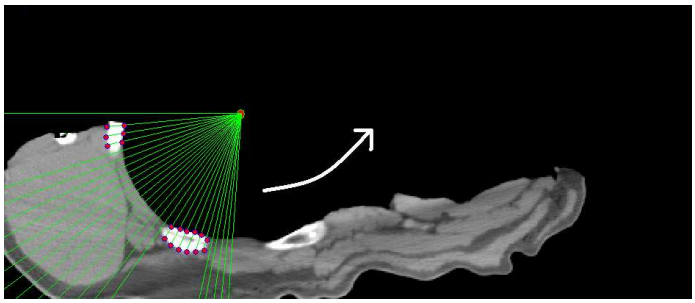


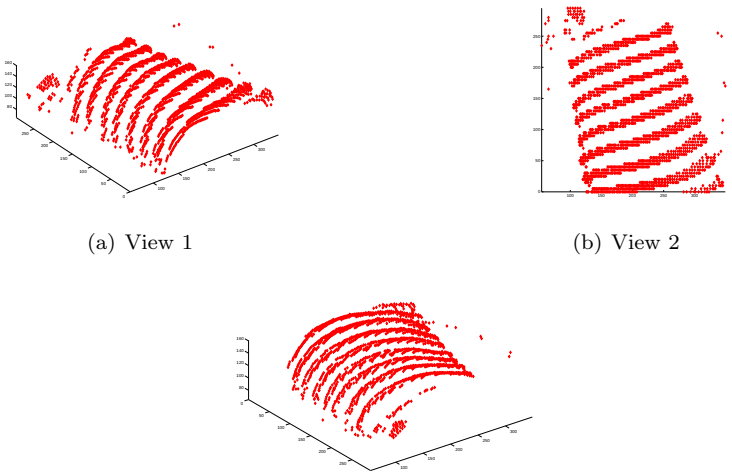
Figure 5.1: Locating points on the ribs

5.2 Pruning of outliers based on *K*-nearest neighbor clustering

Even though, the large majority of bone in a pig backs is ribs, there are often small fragments of bone from other bones such as the spine. These fragments introduce outliers to the point set. As non-rib fragments of bone in a pork middle are small and relative far from the ribs the outliers will normally be in small groups far away from the true rib points. Consequently, the outliers can be identified by K-nearest neighbor clustering (KNNC) with a cutoff radius. In KNNC a point $\mathbf{x}_1$ is a neighbor to $\mathbf{x}_2$ in a point set $\mathcal{P}$ iff $d = \|\mathbf{x}_1 - \mathbf{x}_2\| \leq r$ and $K > \left(\sum_{x \in \mathbf{P} \setminus \{x1, x2\}} \|\mathbf{x} - \mathbf{x}_2\| \leq d\right)$, where r is the cutoff radius. This is also illustrated in Figure 5.3:

A point is clustered together with its neighbors, its neighbors' neighbors and so on. After the clustering, all points belonging to a cluster with less than some number of points are discarded as outliers.

A good selection of the cutoff radius r, the number of neighbors K and the minimum cluster size cs is essential for the success of the pruning, and depended on the density of the samples. Fortunately, good and robust choices of r, K and cs can easily be obtained by visual inspection of the rib points and common sense.

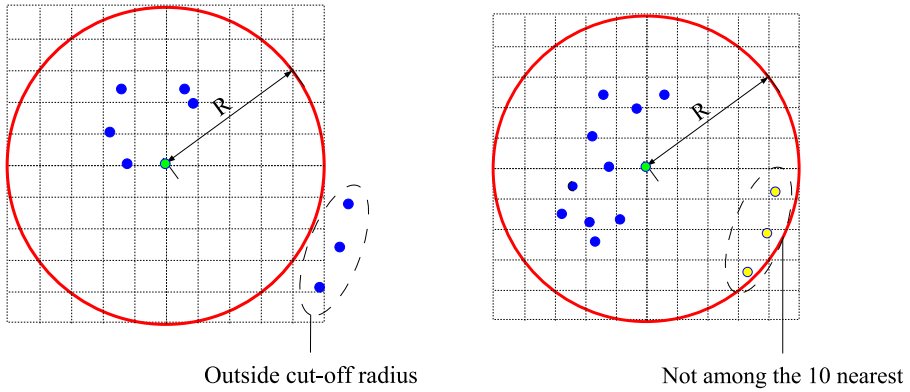


(a) View 1

(b) View 2

(c) View 3

Figure 5.2: The result of a rib search on pig 7 from three different views. The angle spacing between the sample direction was chosen to be $\frac{0.05}{\pi}180$ degrees



(a) $d = \|\mathbf{x}_1 - \mathbf{x}_2\| \leq r$

(b) $K > \left(\sum_{x \in \mathbf{P} \setminus \{x1, x2\}} \|\mathbf{x} - \mathbf{x}_2\| \leq d\right)$

Figure 5.3: Identifying neighbors in KNNC where $K = 10$.

First, realize that r and cs are directly depended on the sample density while K is not. Trivial, r should be chosen equal to the expected maximum distance between two neighboring points. Obviously, this distance depends on the sample density. Likewise, the sizes of the clusters are proportional with the sample density . K is only a significant parameter, if it is expected that some clusters are separated with a distance less than r. If this is not case simply chose K large - a too large K will slow the computation.

It is time to chose pruning parameters for point set plotted in Figure 5.2. By inspection of Figure 5.2 we realize:

- It is unlikely that the distance between a rib point and the nearest point on the same rib is larger than 15, why r is chosen to be 15.
- The distances between the ribs and between ribs and the outliers are larger than r why K is chosen equal to 15 (large).
- The compactness of points in a rib makes it unlikely, that there can be less than 50 points in a cluster, why cs is chosen equal to 50.

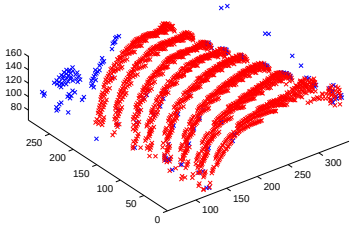
Figure 5.4 shows the result of pruning the point set from Figure 5.2 with the above parameter selection. $r = 15$, $K = 15$ and $cs = 50$ works well in all of the CT scans in the second data set.

5.3 Fitting the ribs

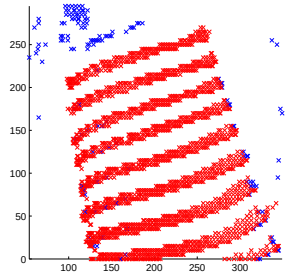
Given a set of outside rib points the outside surface of the rib cage can easily be fitted with a RBF as described in Chapter 3.4. Note, the y-values of the outside rib points are chosen to be the set of interpolation values as every point above the outside surface of the rib cage is either a part of the ribs or the background. The result of fitting the rib points shown Figure 5.4 using (3.26) is shown in Figure 5.5.

5.4 The ends of the ribs

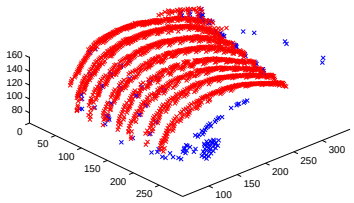
It is an established fact, that it is hard to predict and control the behavior of a fitted function in areas relative far from any data point. For this reason, it



(a) View 1

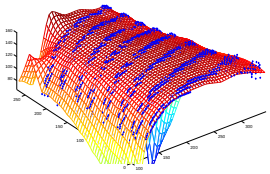


(b) View 2

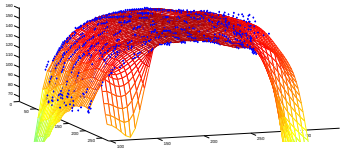


(c) View 3

Figure 5.4: The result of applying K-nearest neighbor pruning with a cutoff radius on the point set in Figure 5.2. The following values were used $K = 15$, $r = 15mm$ and $cs = 50$.



(a) View 1



(b) View 2

Figure 5.5: The result of fitting a piece-wise linear spline to the point set in Figure 5.4. A 30mm spaced grid was used to sample the centers.

is necessary to locate the beginning and the end of the ribs in every slice with respect to the x-axis. The following algorithm can be used:

1. Construct the matrix $\mathbf{F} = \begin{bmatrix} x_1^F & x_2^F & \dots & x_n^F \\ z_1^F & z_2^F & \dots & z_n^F \end{bmatrix}$ and the matrix $\mathbf{L} = \begin{bmatrix} x_1^L & x_2^L & \dots & x_n^F \\ z_1^L & z_2^L & \dots & z_n^L \end{bmatrix}$, where (x_i^F, z_i^F) is given by the rib point $(x_i^F, \_, z_i^F)$ with the smallest x-value in ith slice, and (x_i^L, z_i^L) is given by the rib point $(x_i^L, \_, z_i^L)$ with the largest x-value in ith slice.

2. For all points (x_i^F, z_i^F) in $\mathbf{F}$ find the lines which satisfy:

$$l_{jk}^F(z_q^F) \leq x_q^F \text{for all } x_q^F, \ : \max(0, i-\alpha) \leq j,k,q \leq \min(n, i+\alpha) \quad (5.1)$$

   where l_{jk}^F is the straight line which goes through the points (x_j^F, z_j^F) and (x_k^F, z_k^F), and α is a constant.

   Choose the line $l_{j'k'}^F$ which receives the minimum distance score:

$$d_{l_{jk}^F}^s = \sum_{q=\max(0,i-\alpha)}^{\min(n,i+\alpha)} d([x_q^F \ z_q^F]^\top, l_{jk}^F)^2 \quad (5.2)$$

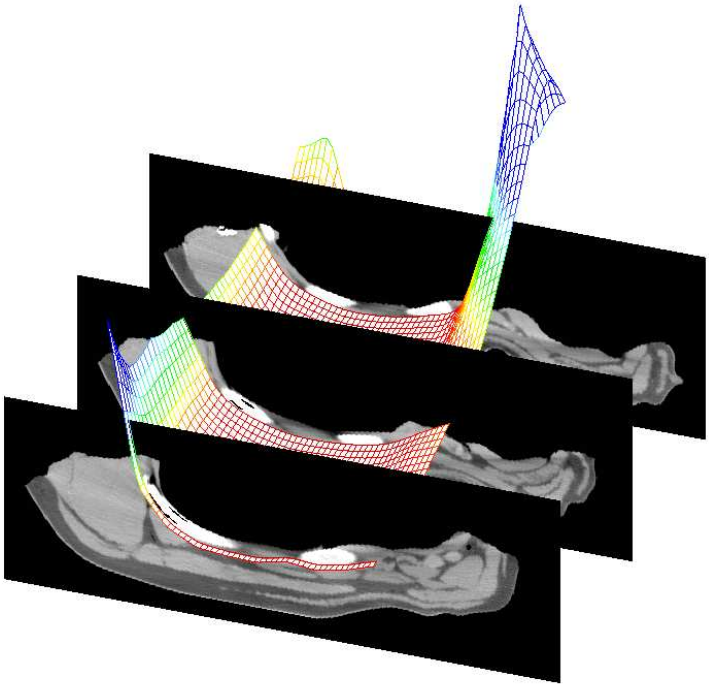
   where $d(\mathbf{x}, l)$ is the euclidian distance between the point $\mathbf{x}$ and the line l. The lower limit of x in the ith slice is then $l_{j'k'}^F(z_i^F)$.

3. Perform step 2 for all the points in $\mathbf{L}$ where "$\leq$" in (5.1) is replaced with "$\geq$".

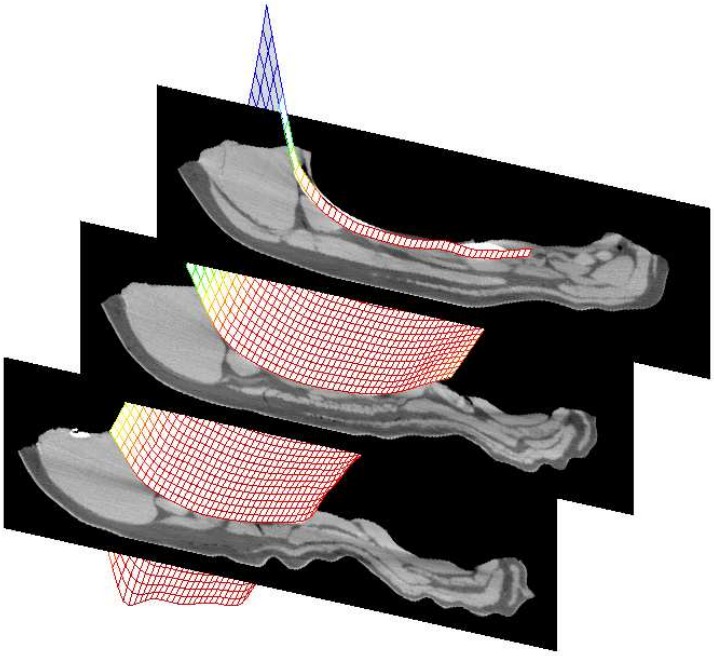
5.5 Verification of the fit and results

In order to verify the quality of the rib fit a visual verification is performed. In Figure 5.6, the fitted function from Figure 5.5 is plotted with three slices taken form the beginning, the middle and the end of pig 7. Clearly, the piece-wise linear spline fits tightly to the ribs.

By removing every voxel, which is above the fitted function, it is possible to remove the ribs. The result is shown in Figure 5.7.



(a) View 1



(b) View2

Figure 5.6: Verification of the rib fit in Figure 5.5.

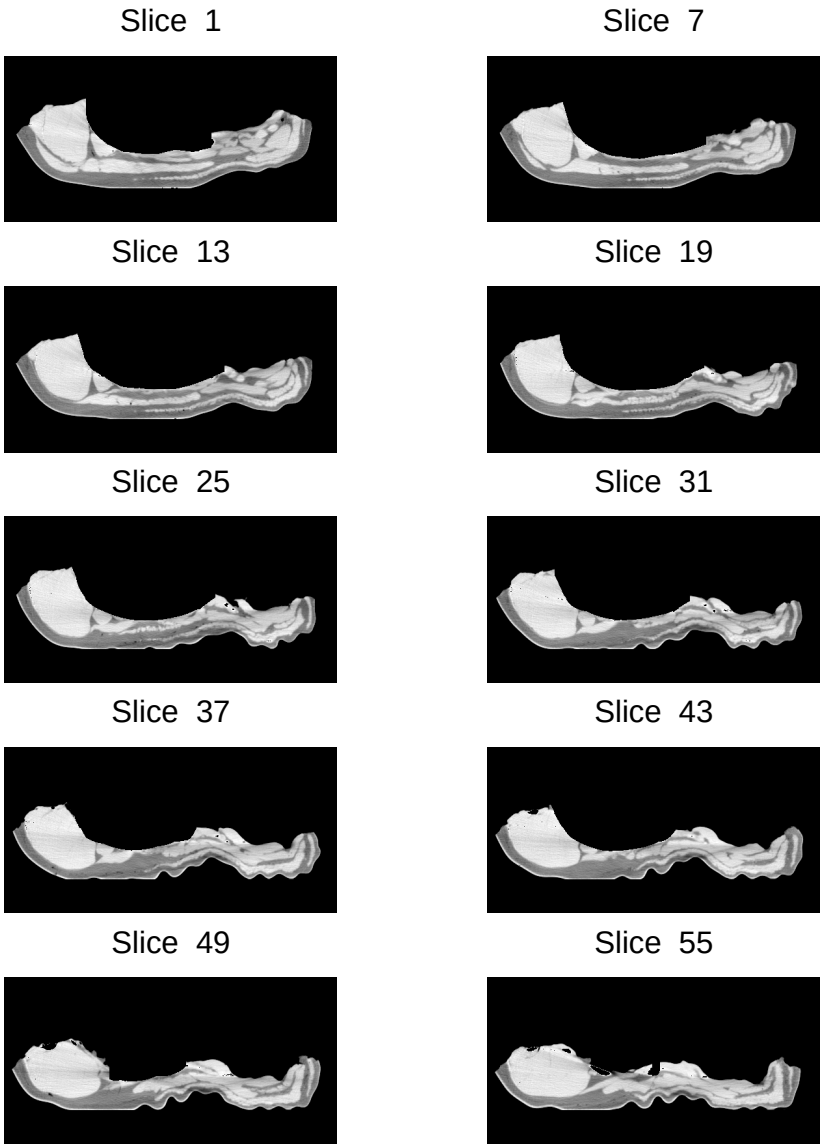


Figure 5.7: Rib and bone removed back from pig 7.

5.6 Discussion

In this chapter a simple but robust method for finding and removing the ribs was proposed. It requires the selection for quite a few parameters; the angle spacing between the sample direction, the selection of a base point, the cutoff radius, the number of neighbors and the minimum cluster size. As a consequence, one might fear that it require a lot of parameter tuning. In this chapter, a parameter selection, which works on all the scans of the second data set, was found.

Chapter 6

Locating and dissecting the 18cm back

In the last chapter, it was described how the ribs could be removed from a back similar to way a butcher removes the ribs. Given a rib and bone removed back, it is a rather trivial task to find the 18cm back as one just needs to:

- measure a distance of 18cm along the skin from the beginning of the skin near the loin,
- make a cut perpendicular to skin surface,
- and follow the edge of the rib and bone removed back, back to the initial point.

Selected slices of the extracted 18cm back of pig 7 are shown in Figure 6.1

It is a more challenging job to locate the three lines, which dissect the back into the loin part and the three tail parts.
Naturally, as soon as the line, which separates the loin part from the tails parts, is found, it is simple to find the remaining two lines.

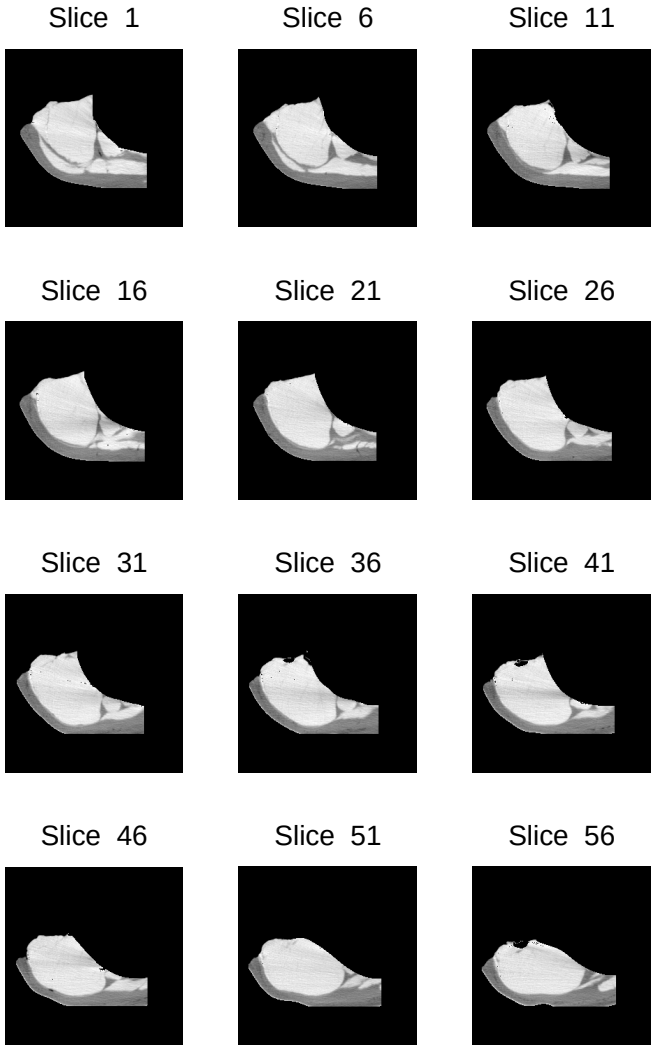


Figure 6.1: The 18cm back of pig 7.

Given a predetermined meat-fat threshold, it is possible record the estimates of the meat percentages in a number of scan line perpendicular to the skin surface. The scan lines are in illustrated in Figure 6.2.

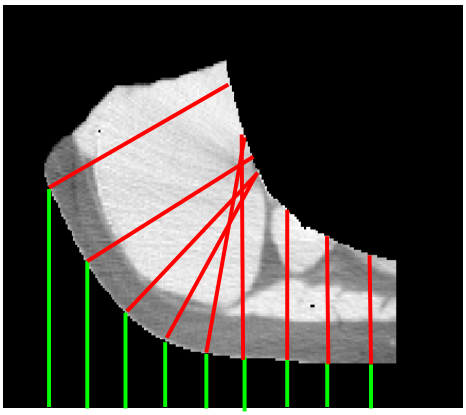


Figure 6.2: Illustration of the scan lines.

Figure 6.3 and Figure 6.4 display the meat percentage in a sequence of scan lines taken from seven consecutive slices. From the two figures a strong characteristic pattern emerges - the meat percentage is relative high in the two thirds of the sequence, and then it drops rapidly. It is the authors opinion that the scan line, which corresponds to the end of the drop, is the line, which separates the loin part from the tail part. This assumption yields the result shown in Figure 6.5 for pig 7.

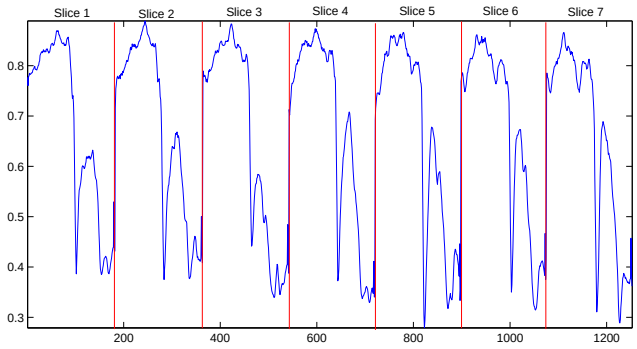


Figure 6.3: The meat percentage in a sequence of scan line taken from the seven first slices of the extracted 18cm back of pig 7.

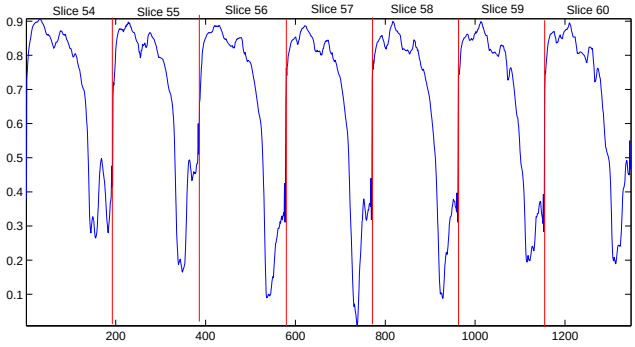


Figure 6.4: The meat percentage in a sequence of scan line taken from the seven last slices of the extracted 18cm back of pig 7.

6.1 Discussion

In this chapter, it was shown how the 18cm back can be extracted from the pork middle. Further, a method for the location of the dissection lines was likewise presented. It is the authors opinion that it might be possible to estimate the quality of an 18cm back purely on the estimated meat percentages of scan lines, as they carry much of the same information as the measures described in Section 1.5

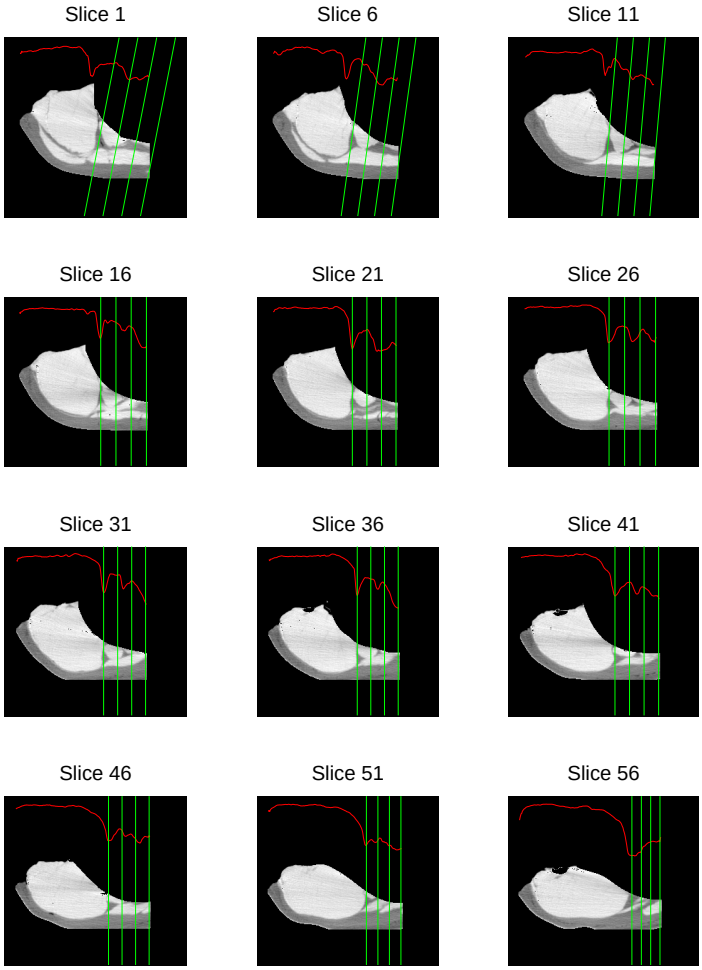


Figure 6.5: Dissection into loin part and tail part for pig 7.

Chapter 7

Investigation in to the existence of inhomogeneities

As mention earlier the possible existence of artifact introduced by the scanner might have an unwanted effect on an intensity based classification. This chapter will conduct an examination on the second data set with the aim of identifying possible intensity inhomogeneities.

Scanner introduced artifacts are often detected by scanning fantom objects. These objects are often build out of materials which have similar intensities to known tissue type. No such trials have been performed - at least not intentionally. Nevertheless, every pig back (except 22) have been placed on a plastic board which has intensities in the Hounsfield spectra similar to fat tissue. Further, there are two plastic rails in the bottom of the scans, which have intensities in the bone range. These objects will be used as fantom objects, and there intensities will be tracked through the slices and matched between scans. Figure 7.1 marks the samples areas.

Figure 7.2 and Figure 7.3 shows the average intensities in the sample areas through the slices for the pigs 4, 7, 9, 12, 14, 18, 20 and 21[1]. In general, the average intensity in area 1 is slightly higher than the average intensity in area 2,

[1] Pig 21 has replaced pig 22 since pig 22 does not have a plastic board.

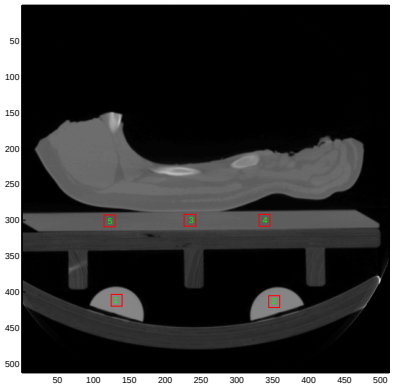


Figure 7.1: Intensity sample areas.

and there is a slight intensity increase through the slices. Further, no significant difference in intensity between the scans is visible. Combining these observations with the fact that the plastic rails are relative far away from the back, it seems reasonable to conclude that there is no evidence in Figure 7.2 which discourages pure intensity based classification.

Figure 7.3 displays more intensity variation than Figure 7.2. A small pattern is visual in the figure as the average intensity in area 3 is higher than the average intensity in area 4, and the average intensity in area 4 is higher than the average intensity in area 5, in the first slices but vanishes towards the last slices. The plastic board is not a real fantom object. Thus, it is not a guarantee that it is homogeneous. As a consequence, it is hard to interpret whether the variation is due to inhomogeneities in the plastic, scanner artifacts or both.

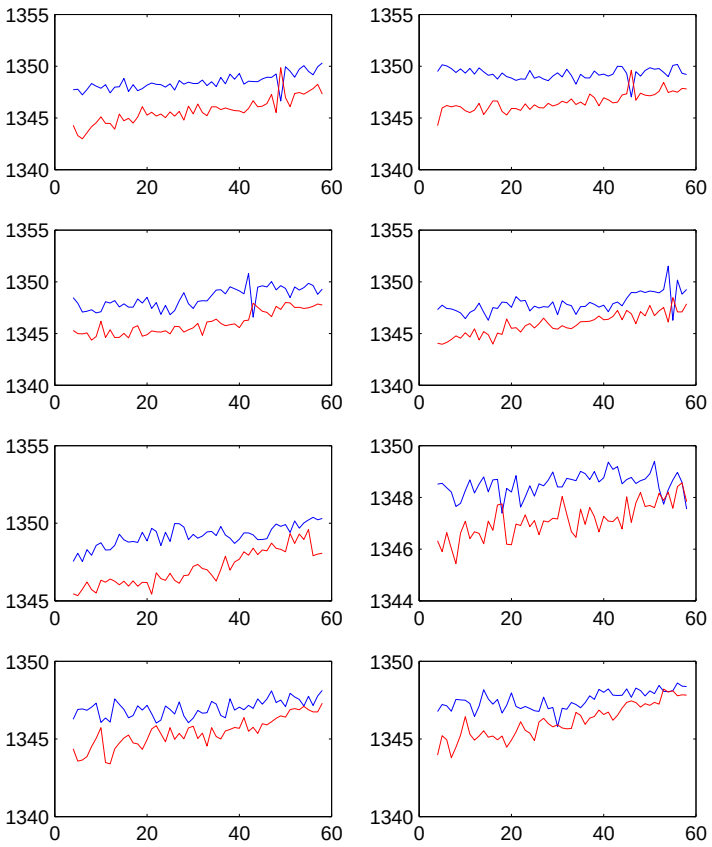


Figure 7.2: Average intensity plot for sample area 1 (blue) and 2 (red).

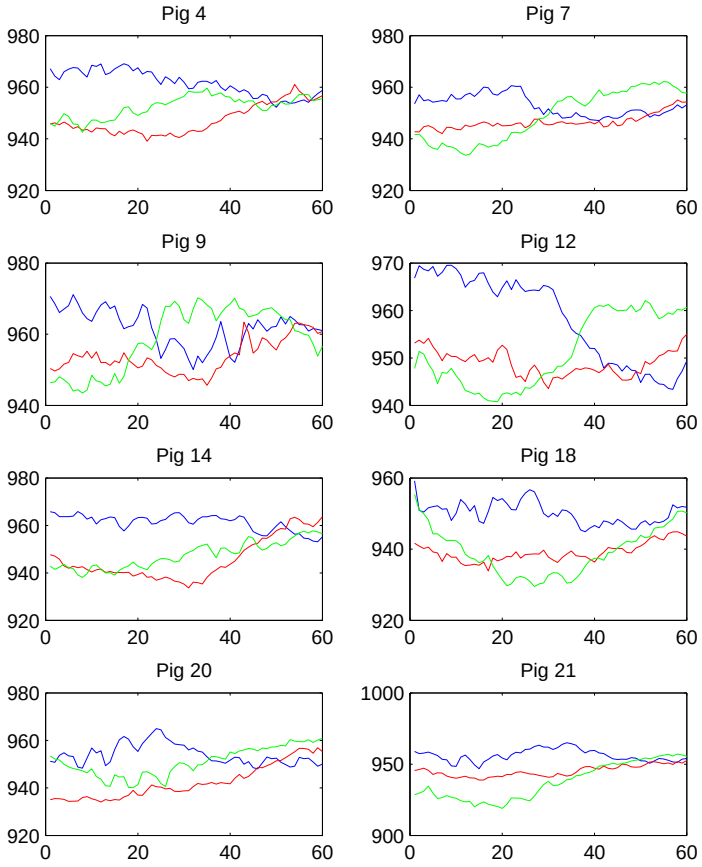


Figure 7.3: Average intensity plot for sample area 3 (blue), 4 (red) and 5 (green).

A last intensity test was performed which tracks the average intensity in the loin through the slices by extracting a 20-by-20 pixel sample area centered at the center of mass of the loin[2]. The result is shown in Figure 7.4, where a clear oscillating wave pattern emerges. The phenomena is also visible in the image domain. Figure 7.5 displays the slice image of pig 7 defined by the plane $x = 100$, which is a vertical cut through the middle of the loin. From Figure 7.5, it seems likely that oscillating pattern is due to shading by the ribs.
Further, seven of the eight intensity plots in Figure 7.4 reveal a relative large drop in intensity from the first slice to the last slice - on average approximately 15. Whether, this phenomena is caused by the scanner or whether it has an anatomical explanation, e.g. a change in the density of the loin muscle, is hard to conclude as all of the backs has been scanned from neck and downwards.

7.1 Conclusion

This investigation has revealed a few artifact which might have a considerable effect on a pure intensity based segmentation.

[2]The center of mass is calculated upon the outlined shapes described in Appendix A

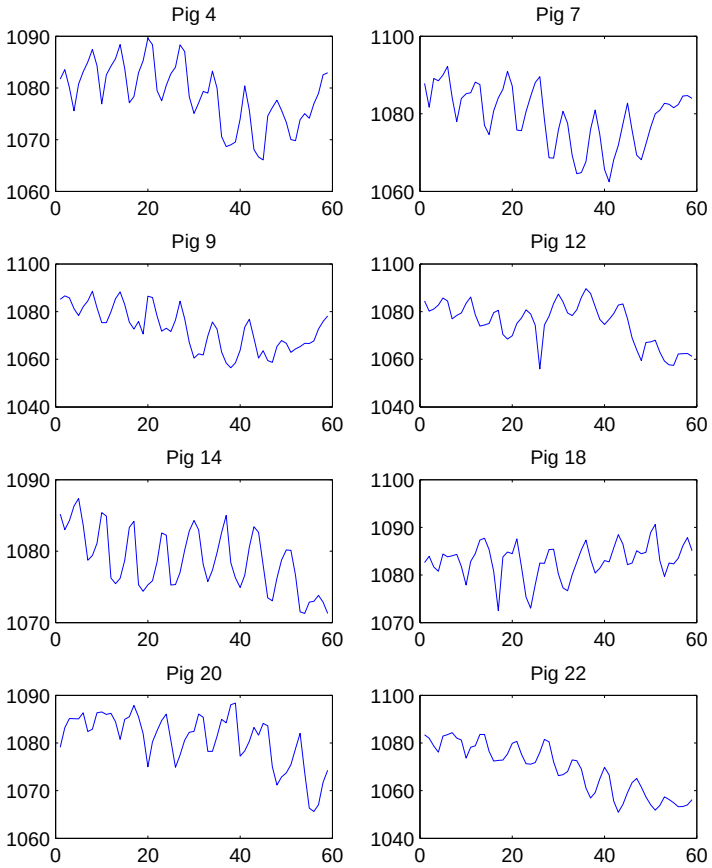


Figure 7.4: The average intensity plot for loin sample area.

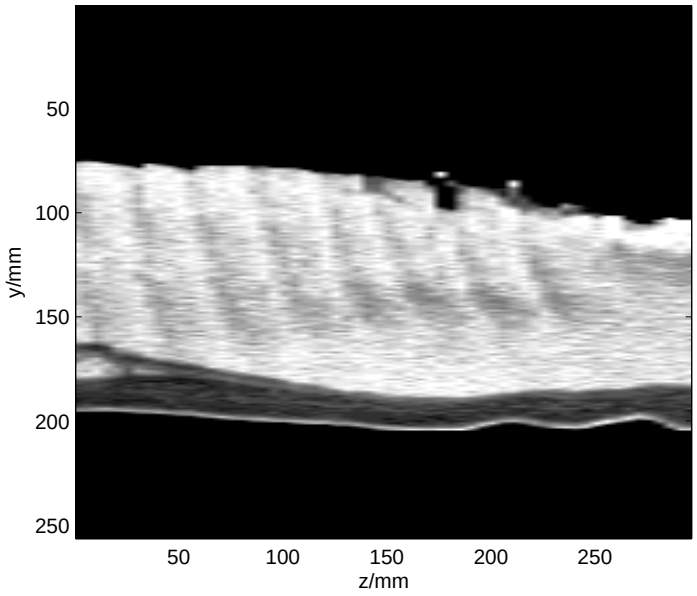


Figure 7.5: The slice image of pig 7 defined by the plane $x = 100$. Histogram equalized to enhance the visibility the rib shades.

Chapter 8

Estimating the meat-fat percentages

The last chapter proofed the existence of intensity inhomogeneities in the second data set. In this chapter, the performance and stability of classical discriminant analysis will be tested on the second data set using the leave-one-out methodology.

Since the Hounsfield scale is defined relative to the radiodensity of distilled water and air, the mean intensity of a tissue type should in principle be independent on scanner and reconstruction algorithm. Christensen et al [3] investigated, how large an effect the choice of reconstruction algorithm has on the spectrum of a CT scan. They applied four commercial reconstruction algorithms to the same raw data and compared the resulting tomogram spectra. The spectra are shown in Figure 8.1.

Figure 8.1 indicates, that the distributions of meat and fat voxels are depended on the choice reconstruction algorithm. Consequently, a supervised classifier might work poorly on CT scans which have been reconstructed with a different algorithm than the training set. Christensen substantiate this claim by predicting the weight of five products with the four different reconstruction algorithms from a prediction model build on the tomogram spectrum of one of the re-

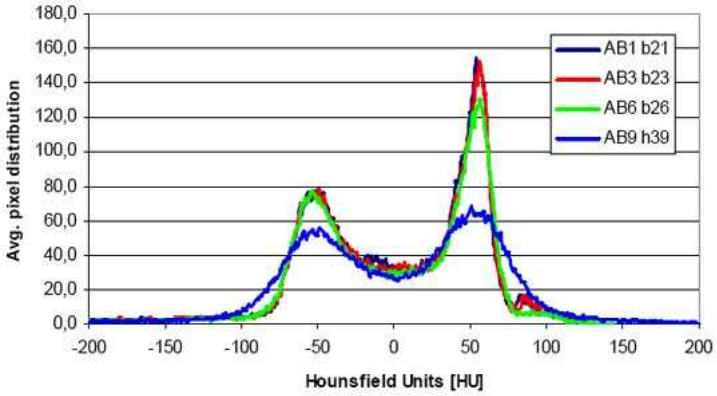


Figure 8.1: An example of reconstructed tomogram spectra of the same belly product (Weight: 1023 gram), using four different commercial reconstruction algorithms ([3]).

construction algorithms. The experiment showed the estimated weights were depended on the reconstruction algorithm.

The leave-one-out test will use the outlined shapes described in chapter 2 as "ground truth". It is assumed that the volume of the 18cm back, which is not inside any of the outlined shapes, consists of pure fat. This assumption is not entirely correct as there are a few muscles inside the 18cm back, which has not been outlined. This is primarily a concern in the 15-20 last slices, why the segmentation only will be done on the first 40 slices in the CT scans[1]. Anyway, it should be noted that the estimates of the performance of the classification schemes are conservative.

The quality of the segmentation will be measured in term of the voxel wise correspondence between the "ground truth" and the segmentation result. This give us the four measures shown in Figure 8.2 - in principle only three.

The result of the leave-one-out test is shown in Table 8.1. In all of the eight trials, the two discrimination methods yielded close to identical results, which could indicate that the variances of the intensities of meat and fat are equal. The hypothesis was rejected by a Bartlett test with a significance level $\alpha = 0.01$. As there are meat in the fat regions of the ground truth the result of the Bartlett

[1]This is rather unfortunate as it becomes harder to observe possible effects due to intensity inhomogeneities.

		Segmentation	
		Fat	Meat
Ground truth	Fat	FF	FM
	Meat	MF	MM

Figure 8.2: Measuring the correspondence between a segmentation result and the ground truth.

test is questionable. Furthermore, the discrimination methods assigned the same label to a voxel as the "ground truth" in approximately 94% (FF+MM) of the time for every trial. FM and MF were in six out of eight trial approximately equal despite the existence of wrongly classified meat voxels in the "ground truth". The explanation for this is that the wrongly classified meat voxels in the "ground truth" pull the estimated fat distribution towards the meat distribution.

8.1 Discussion

It seems plausible to obtain a good segmentation of the 18cm back from classical discriminant analysis, if the quality of the CT scan is similar to the quality of the second data set. Nevertheless, the next part of the thesis will focus on using shape models as a priori knowledge in a segmentation scheme.

The presence of two or more reconstructions of the same scan might raise the quality of the discrimination as there will be more measures to discriminate on.

Method	Measure \ Pig	4	7	9	12	14	18	20	22
LDA	MM	0.6688	0.6370	0.6305	0.6620	0.7182	0.5869	0.6483	0.6876
	FF	0.2733	0.3104	0.2925	0.2805	0.2194	0.3529	0.2917	0.2519
	FM	0.0281	0.0280	0.0373	0.0283	0.0316	0.0197	0.0219	0.0180
	MF	0.0298	0.0246	0.0397	0.0293	0.0308	0.0405	0.0381	0.0425
	Total	1.0000	1.0000	1.0000	1.0001	1.0000	1.0000	1.0000	1.0000
QDA	MM	0.6677	0.6356	0.6292	0.6611	0.7172	0.5859	0.6474	0.6876
	FF	0.2740	0.3117	0.2933	0.2812	0.2201	0.3534	0.2922	0.2519
	FM	0.0275	0.0266	0.0365	0.0276	0.0309	0.0192	0.0214	0.0180
	MF	0.0308	0.0260	0.0410	0.0301	0.0318	0.0415	0.0390	0.0425
	Total	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 8.1: Leave-one-out results for classical discriminant analysis.

Part II

Implicit parametric shape model for segmentation

Chapter 9

Introduction to shape model segmentation

In the last chapter, 18cm backs were segmented using LDA and QDA with reasonable success. This part will investigate the possibility of using shape models as a priori knowledge in a segmentation scheme.

A shape modelling approach by Leventon [15], which does not require corresponding landmarks in the training set, is presented and afterwards integrated in a region based segmentation framework. The region based segmentation framework is an extension to the one presented in [23].

The existence of multiple muscles in a pig back implies the need for multiple shape models. Further, the muscles in a pig back are in some cases in a close proximity to each other - often only separated by a thin fat membrane. Thus, the following two requirements arises:

- The shape models need a good initialization with regard to pose.
- The shape models need to be coupled such that they do not overlap.

A few attempts to accommodate these requirements in the region based framework are likewise presented.

One key consideration, when building a statistical shape model, is to ensure that the model captures all significant shape variance. If the training set is not representative, the model will consequently be over constrained. In Chapter 2, it was established that the pig backs in the second data set were deformed by gravity. This deformation introduces non-anatomical variance to the muscle structures which is impossible or at least extremely difficult to capture in a shape model. The first chapter in this part presents a method for estimating this deformation and removing it.

Chapter 10

Stretching of pigs

The initial idea behind pig stretching was to remove the deformation caused by gravity such that a shape guided segmentation can be done more easily. For this reason, it was not a strict goal that the estimate of the non-deformed back needed to be close to identical to the ground truth. Essential, as long as the deformation correction removes enough non-anatomical variance for a shape guided segmentation to applied with success, the method performs adequately as the segmentation can be warped back to the deformed back.

10.1 Representing the deformation of a slice

It is challenging task to make a realistic model of the deformation of a pig back as it is a complex system of forces effecting each other. Even trying to model the deformation of a shape with an homogenous interior is a difficult task. Hence, in order to have a simple representation of the deformation we need a simple representation of a shape. Figure 10.1 display the central path (CP) trough a pig back. Clearly, most of the deformation of the pig back is visually present in this line. So, an estimate of the deformation of this path must be an overall good estimate of deformation of the back.
The process of skeletonization deals with the reduction of the foreground pixels

in a binary shape to a skeleton while trying to preserve the connectivity and the extent.

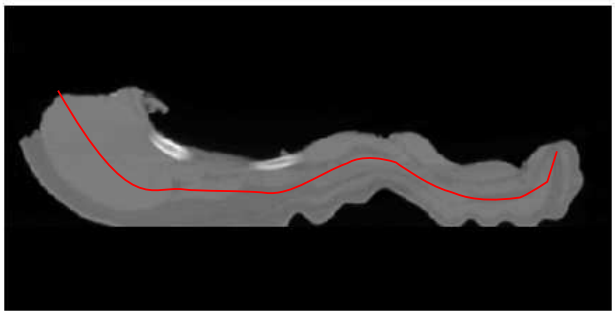


Figure 10.1: The central path of a pig back

10.1.1 Skeletonization

Figure 10.2 exhibits the expected result of a skeletonization method on a solid T.

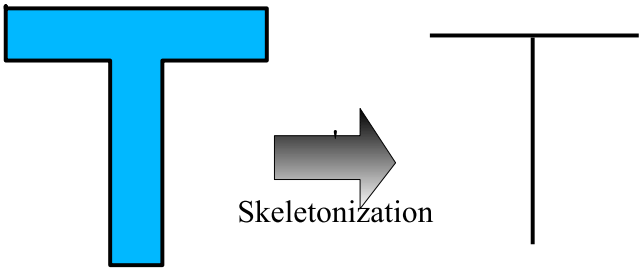
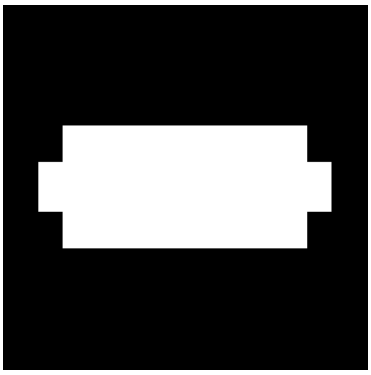
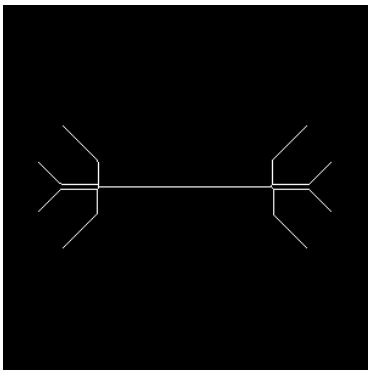


Figure 10.2: Skeletonization of the letter T.

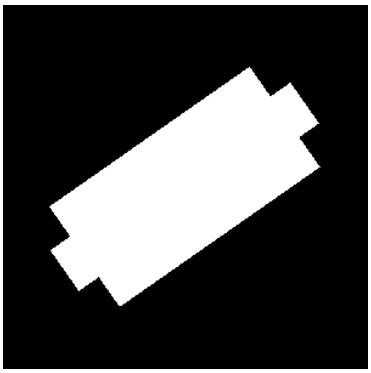
It can be argued that it is possible obtain an approximation to the result shown in Figure 10.2 simply by applying the morphological thinning operator, iteratively. Unfortunately, this approach is not robust. Sure, the thinning approach preserve the topology of the shape but it is not invariant with respect to transformations such as rotation and scaling. That, thinning is not invariant with respect to rotation is shown in Figure 10.3.



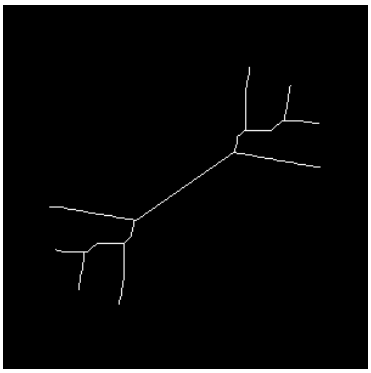
(a) Simple rectangle (SR)



(b) Skeletonization with thinning operator on SR



(c) Simple rotated rectangle (RR)



(d) Skeletonization with thinning operator on RR

Figure 10.3: Thinning is not invariant with respect to rotation.

The word skeletonization is often interchanged with the medial axis transformation - Hence, it is the most accepted definition of skeletonization. Vermeer [24] defines the medial axis of a shape as:

Definition 10.1
The medial axis (MA) of an object, also referred to in the literature as the skeleton, is the closure of the locus of the centers of all maximal inscribed discs.

Figure 10.4 shows an example of a medial axis.

The MA or skeleton will most likely have several branches which will be more or

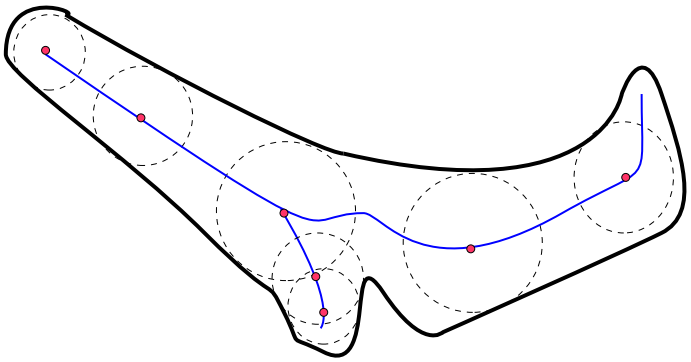


Figure 10.4: The medial axis of a given shape.

less significant with regard to structure of the shape. Hence, pruning is needed in order to get one non-branching line. The most intuitive choice of measure for the significance of a branch is properly the Euclidian distance - it also matches fine with the central path in Figure 10.1. The central path can now formally be defined:

Definition 10.2
The central path of a shape is the longest path in the medial axis or skeleton of the shape.

10.1.2 Chordal axis transformation

Even though, the MAT works fine in a conceptual sense, it has many drawbacks which makes it unsuitable for practical use. Prasad [19] mentions the following drawbacks

- In general, it is not possible to obtain the boundary of a shape directly by inverting its MAT.
- The MAT is a nonlocal transform, in that a small feature on the boundary of a shape may induce a skeletal feature in the MAT, that is far away from the boundary feature, and boundary features may be greatly exaggerated or underplayed by their skeletal counterparts.
- Computationally expensive.

- The MAT does not extend to a discretely represented shape[1].
- The MAT of a shape may have several small branches and spurs induced by minor undulations or noise present in the boundary of the shape.

In [19] another transformation, the chordal axis transformation (CAT), is proposed which does not suffer from the drawbacks of the MAT - except from the first and the last drawback. Before the chordal axis transformation can be defined the maximal chord of tangency needs to be defined:

Definition 10.3
A maximal chord of tangency of a maximal disc is a chord of the bounding circle of a maximal disc, such that i) it connects two points of tangency of the maximal disc with the boundary of the shape, and ii) at least one of the two arcs of the bounding circle subtended by the chord is free of points of tangency with the shape's boundary.

Armed with this definition the CAT can be defined:

Definition 10.4
The Chordal Axis Transform of a non-degenerate planar shape[a] *is the set of all ordered pairs* (p, δ)*, where* p *and* δ *are either the midpoint and half the length, respectively, of a maximal chord of tangency, or the center and radius, respectively, of a maximal disc with three maximal chords of tangency, that form an acute angled triangle.*

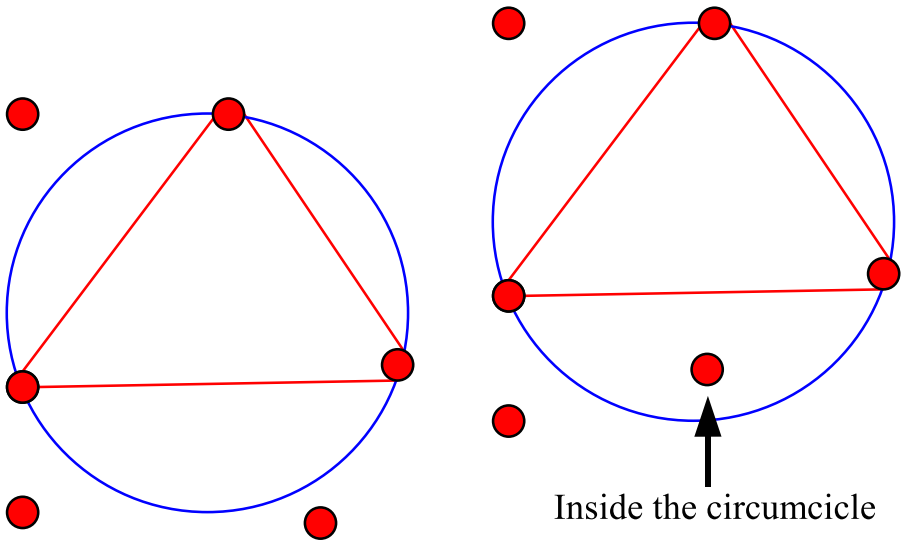
[a] In a non-degenerate planar shape no maximal discs touch the boundary in more than three points.

The above definition is closely related to the definition of the Delaunay triangulations:

[1] A shape represented by points.

Definition 10.5
Given a set of planar points $\mathcal{P}$, a triangulation $T(\mathcal{P})$ is a Delaunay triangulations if it fulfills the Delaunay criterion:
For all $t \in T(\mathcal{P})$ no point $p \in \mathcal{P}$ falls inside the circumcircle of t[a].

[a]Illustrated in Figure 10.5



(a) Fulfills the Delaunay criterium (b) Does not fulfill the Delaunay criterium

Figure 10.5: Illustration of the Delaunay criterium

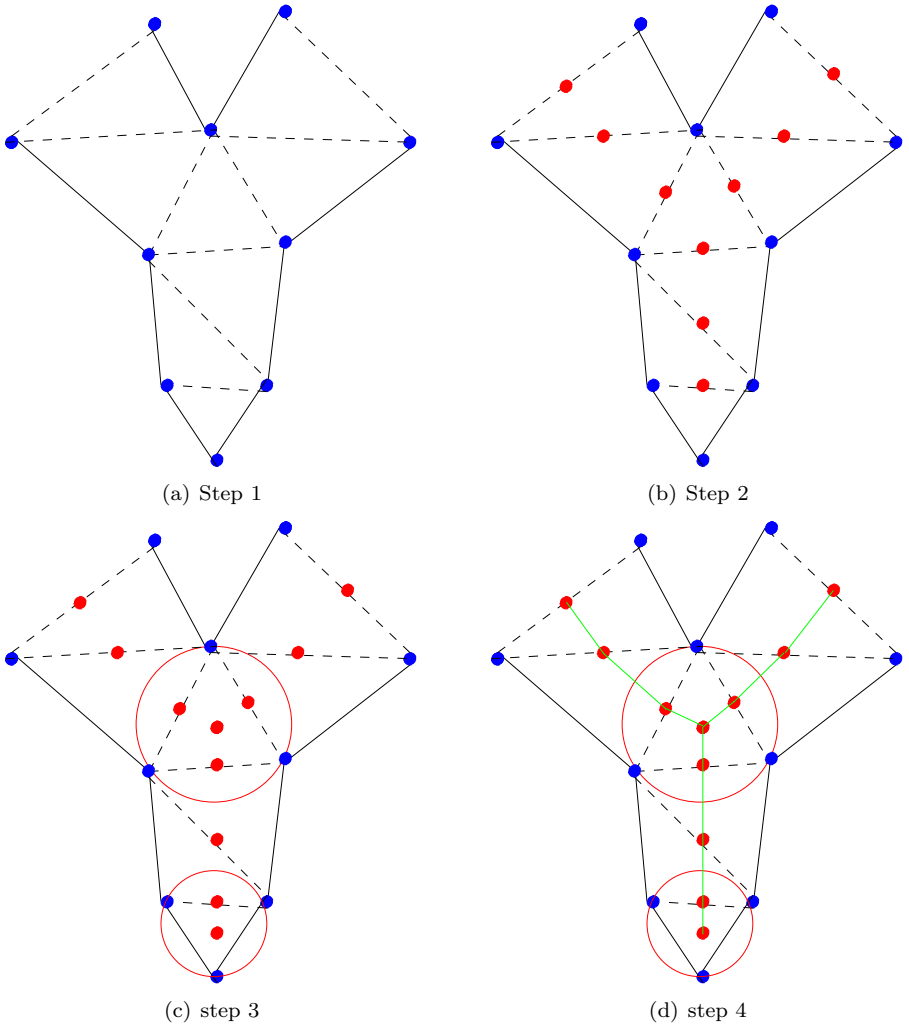
In order to explore the relationship, it is assumed that a shape represented by the planar graph $G = (\mathcal{P}, \mathcal{E})$ has been given. First, it is acknowledged that a maximal disc is a circumcicle of a triangle, where the chords of tangency are the edges in the triangle. Second, there can be no other point $p \in \mathcal{P}$ inside the maximal discs as another point inside the disc implies, that the disc contains a part of one of the edges in $\mathcal{E}$. This is exactly the Delaunay criterium. Hence, in the discrete case, the chords of tangency and the maximal discs can be obtained simply by performing a Delaunay triangulation on the inside of the shape. This brings us to the definition of the constrained Delaunay triangulation (CDT):

Definition 10.6
Given a planar graph $G = (\mathcal{P}, \mathcal{E})$, a triangulation $T(\mathcal{P}, \mathcal{E}) = \mathcal{E}'$ is a constrained Delaunay triangulation if it "fulfills" the Delaunay criterion and the additional condition:
No edge $e' \in \mathcal{E}'$ crosses any edge $e \in \mathcal{E}$. Note that $\mathcal{E} \subseteq \mathcal{E}'$.

"Fulfills" is in quotes as all triangles in a constrained Delaunay triangulation may not be able to fulfill the Dealunay criterium, e.g. the edges in $\mathcal{E}$ might form a triangle which does not fulfill the Delaunay criterium.

The CAT of a discrete planar shape can constructed from a CDT with the following recipe (also illustrated in Figure 10.6):

1. Perform a CDT on the discrete shape.
2. For all non-constrained $\mathcal{E}'\backslash\mathcal{E}$ edges in the CDT place a point in the middle of edge.
3. For all triangles with only one or three non-constrained edges place a point in the center of the circumcicle of the triangle.
4. Connect neighboring points.



(a) Step 1 (b) Step 2

(c) step 3 (d) step 4

Figure 10.6: Illustration of the CAT recipe. The Blue dots are the discrete representation of the shape, the black continuous edges are the constrained edges, the black dashed edges are the non-constrained edges, the red points are points on the chordal axis, the red circles are the circumcicles and the green line is the chordal axis.

10.1.3 Finding the longest path

Finding the longest path in a graph only makes sense, if there a no cycles in the graph (Directed Acyclic graph or just DAG), as the longest path will have an infinity distance if cycles are present. Before presenting the algorithm a simple example helps to understand the basic idea behind the algorithm:

Example 10.1
Imagine, that you have to find the longest road in a non-cyclic road net. This problem can be solved by placing a car at the end of all the blind roads. At the same time the cars start driving with the same speed. When a car reaches a branch, it has to decide whether it should stop and block the road where it came from, or continue. If it can continue down more than one road it stops. When the last two driving cars meet each other the longest road is found.

The algorithm is shown in pseudo code in Algorithm 2. The algorithm takes a DAG represented by a set of nodes with neighborhood information as input and returns the longest path. Since, the nodes contain neighborhood information only the end nodes are in practise supplied as input. The algorithm uses quite a few more or less intuitive types:

- node: A node in a planar graph with a counter and neighborhood information. The counter is equal to the number of neighbors in the beginning.
- array⟨node⟩: An array of nodes.
- path: A sequence of nodes with length information.
- priority_queue⟨path⟩: A priority queue of paths, which returns the shortest path when GET is called.
- stack⟨path, 2⟩: A stack of paths with a capacity of two.

10.2 Estimating the non-deformed axis

As mentioned in Chapter 2, there are always a couple of slices in every CT scan, which have not been deformed by gravity. It is only natural to use these

Algorithm 2 The longest path in a non-cyclic planar graph

```
Require: array⟨node⟩ ends;
 1:
 2: path longestPath, p;
 3:
 4: priority_queue⟨path⟩ subPaths;
 5:
 6: stack⟨path, 2⟩ subs;
 7:
 8: node n;
 9:
10: //Convert all end nodes to single node paths and add them to the queue
11: for i=1:LENGTH(ends) do
12:    ADD(subPaths, APPEND([], [AT(ends, i)]);
13: end for
14:
15: while (NOTEMPTY(subPaths)) do
16:    p ← GET(subPaths);
17:    n ← EXPAND(p);
18:    DECREMENT(n); //Decrese the count by one.
19:    if (POSSIBLEWAYS(n)==1) then
20:       ADD(subPaths, APPEND(p,[n]));
21:    else if (POSSIBLEWAYS(n)==0) then
22:       push(subs, p);
23:    end if
24: end while
25:
26: longestPath ← APPEND(POP(subs), REVERSE(POP(subs)));
27:
28: return longestPath
```

slices to estimate how the deformed slice might have looked, if the pig was not slaughtered. The following denotation will be used in the remainder of the chapter:

DCP The deformed central path.

ECP The estimate of the non-deformed central path.

RCP The central path of the reference slice - any slice which has not been deformed.

A crude solution is to replace the part of the DCP, which has been deformed, by the corresponding part of the RCP. There are a few problems with this approach.

- You need to consider correspondence. Which part of the DCP should replaced with which part of RCP? It is necessary to do some transformation (e.g. scaling) on the RCP to secure a good correspondence.
- By blindly removing parts of DCP and replacing them with parts from the RCP, you risk removing true shape variance.

The correspondence problem will to a large extent disappear, if both the DCP and RCP are used to estimate the part of the DCP, which has to be altered.

Suppose a non-rigid and weightless stick moves with some speed towards a solid rock in an initial space. A simple model of what happens, if it is assumed that the fiction force in any contact point between the stick and the rock is infinite large, and that the internal stick forces can be ignored, is illustrated in Figure 10.7.

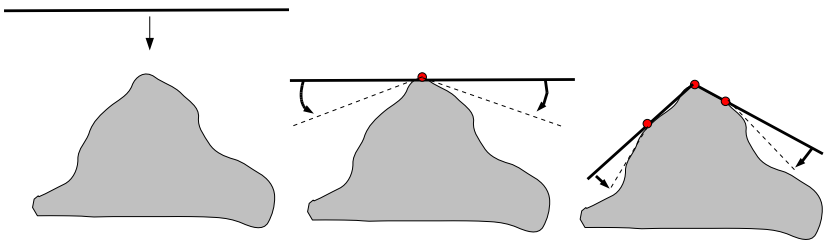


Figure 10.7: Non-rigid stick falling down on a solid rock. It is assumed that the fiction force in a contact point between stick and rock is infinite large.

When the stick hits the rock for the first time the motion changes from being straight to two circular motions. What happens from this point on is rather intuitive. The two interesting things with this example are:

- As the length of the stick does not change it is easy to establish point correspondence on the stick before and after.
- The stick does take the shape of the rock but rather mimics the shape of rock. Thus, the risk of removing actual shape variance is minimized.

Clearly, it is possible to convert this model in to a method for finding the ECP, which does not have the same drawbacks as the prior method.
Before the algorithm can be presented two practical details need to be mentioned.

- It is really not necessary to correct the whole DCP as only the areas away from the ribs are deformed. Hence, it is only necessary to do the rotation motion on one side of the first contact point.
- If the DCP is to be dropped on to the RCP they need to be aligned in the horizontal direction. This is a rather simple task as the reference slice and the deformed slice are almost perfectly aligned as the scan direction is parallel with the spine of the pig.

With these minor details covered it is time to sketch the algorithm:

1. Align the DCP with the RCP horizontally.
2. Drop the DCP onto the RCP. Stop as soon as the central paths touch each other. The point of contact is the first rotation point.
3. Rotate the part of the DCP, which needs to be corrected, until the two central paths touch each other in a new rotation point.

10.3 Building a discrete outline

It is impossible to reconstruct a shape from its central path alone as the simple representation does not carry any information about the thickness of the shape. Consequently, in order to reconstruct the shape of the non-deformed pig back

from the ECP, it is necessary to obtain thickness information from the deformed back.
A straightforward approach is to assume that the thickness measured perpendicular to the central path does not change under a deformation. Using this assumption, a discrete outline of the non-deformed back can be constructed by transferring the thickness of the non-deformed back at a number of points on the DCP to the corresponding points on the ECP.
Naturally, this method alone is rather crude. E.g. the many bends of the skin surface on Figure 10.1 will be transferred to the discrete outline of the non-deformed back. To redeem this, it is necessary to smooth the edge points of the discrete outline. The approach is illustrated in Figure 10.8.

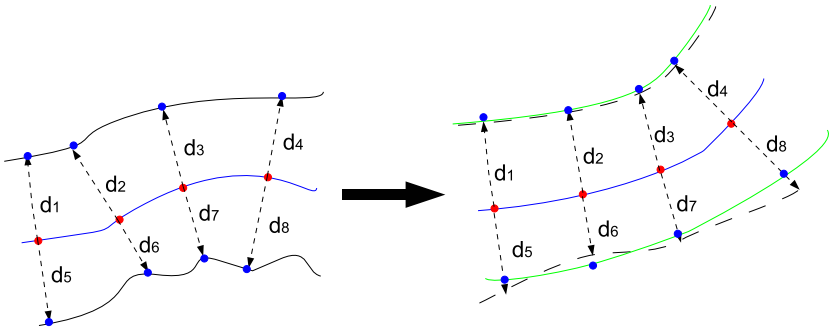


Figure 10.8: Finding corresponding edge points. The red dots are corresponding middle points, the blue dots are corresponding edge points, the dashed lines are interpolated edge splines, and the green lines are smoothed edge splines.

10.4 Stretching a pig step-by-step

With the tools presented earlier in this chapter, it is possible to perform a pig stretching. This section will go through the process the step-by-step, which is exemplified in Figure 10.9:

1. The outline of the deformed back is sampled.

2. The outline of the deformed back is triangulated with a CDT.

3. The skeleton of the deformed back is found by connecting the midpoints of the non-constrained edges and the centers of the non-constrained triangles in the triangulation according to Section 10.1.2.

4. The longest path in the skeleton is located as described in Algorithm 2. The red lines in Figure 10.9(d) a blocked edges, while the green lines are non-blocked edges. Hence, the long green path is the central path.

5. The DCP is smoothed with a cubic spline and re-sampled.

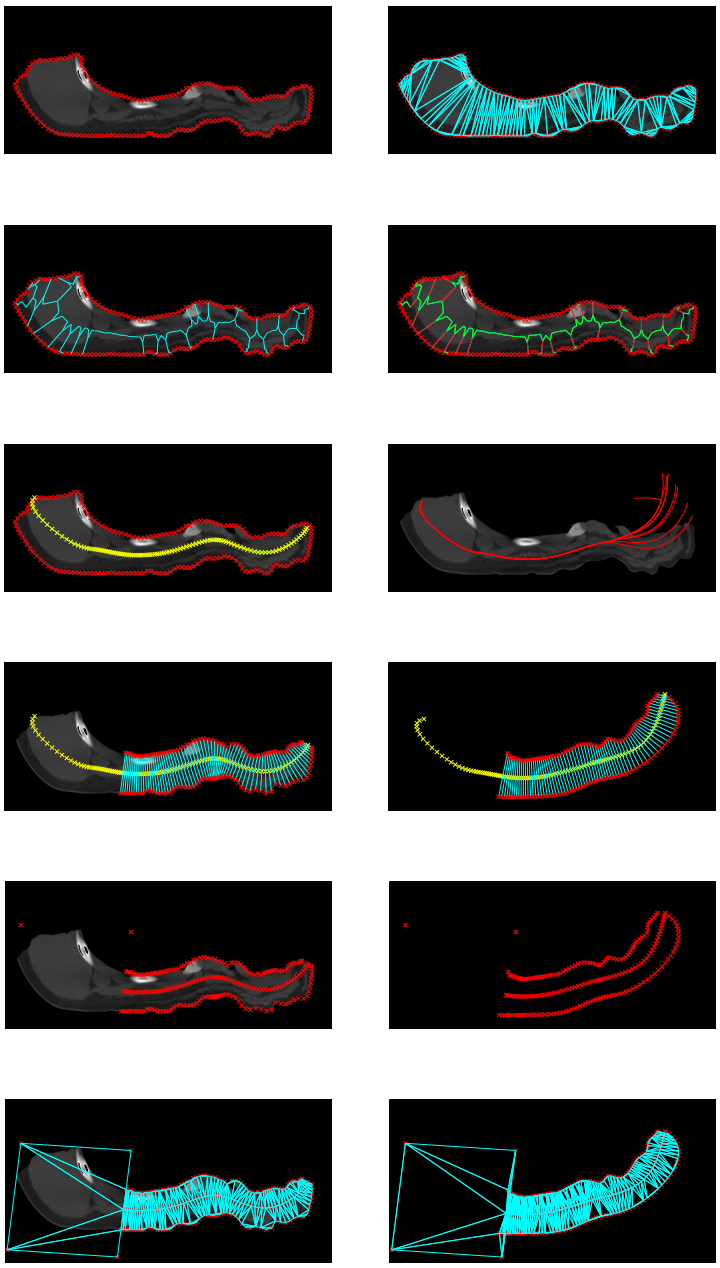
6. The ECP is found according to Section 10.2.

7. The discrete outline of the non-deformed back is constructed as described in Section 10.3. More precisely, the outline of the non-deformed back is only (re)constructed from the first rotation point minus a small distance.

8. Two control point set are build for warping from the discrete outlines and the central paths DCP and ECP. Four additional points are added to the secure that the non-deformed part of the deformed back is warped to the non-deformed back as well.

9. A triangulation is performed on the control set of the deformed back and transferred to the control set of the non-deformed back. At last, the non-deformed back is realized by warping of the non-deformed back.

A partial printout of the result of performing pig stretching on pig 7 is shown in Figure 10.10.



(k) step 9a (l) step 9b

Figure 10.9: Illustration of the nine steps in the pig stretching process.

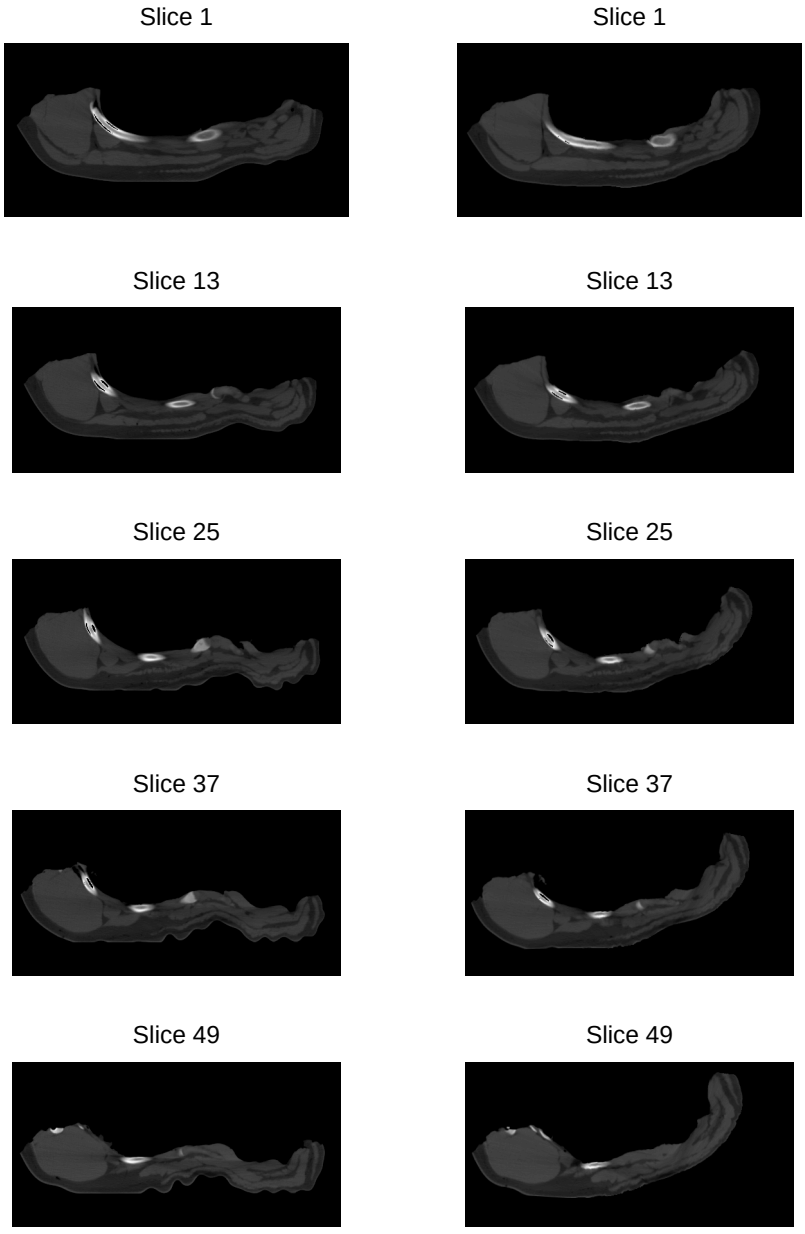


Figure 10.10: The result of a pig stretching on pig 7. Left column contain the before images and the right column contains the after images.

10.5 Discussion

The pig stretching method described in this chapter does not rely on a complex model of the deformation. Consequently, the method is based on a considerable number of assumption which are:

- The deformation of the pig back can be adequately represented by the deformation of the central path of the pig back.
- The central path is the longest path in the skeleton.
- The thickness, measured perpendicular to the central path, does not change significantly under a deformation.
- The warp function can be approximated linear locally.

The many assumptions raise the question whether the estimated deformation is close to actual elastic deformation. Since no ground truth is available it is impossible to evaluate how good the estimation is. Recall, the goal was not to estimate the true elastic deformation perfectly but to remove enough non-anatomical variance for a shape guided segmentation to be applied with success.

Chapter 11

Shape models

11.1 What is a shape?

Before the construction of a shape model can be discussed we need to clarify what a shape is? Vaguely, it can be stated that a shape is the characteristic surface configuration of an object. Special cases of shapes are circles, squares, triangles etc. The following definition by D.J. Kendall [13] will be employed in this thesis:

Definition 11.7
Shape *is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.*

11.2 Representation of shapes

The most common way of representing shapes is with landmarks which are points located at the boundary of the shape or at distinct biological landmarks.

Various research group have applied this approach in many application with good results.
Cootes et al. [5] mention the following *rule of thumb*:

> *Good choices of landmarks are points which can be consistently located from one image to another.*

This implies, that suitable landmarks are:

- points, at the boundary where there is high curvature such as corners.
- points, where boundaries intersect.
- points, at biological landmarks.

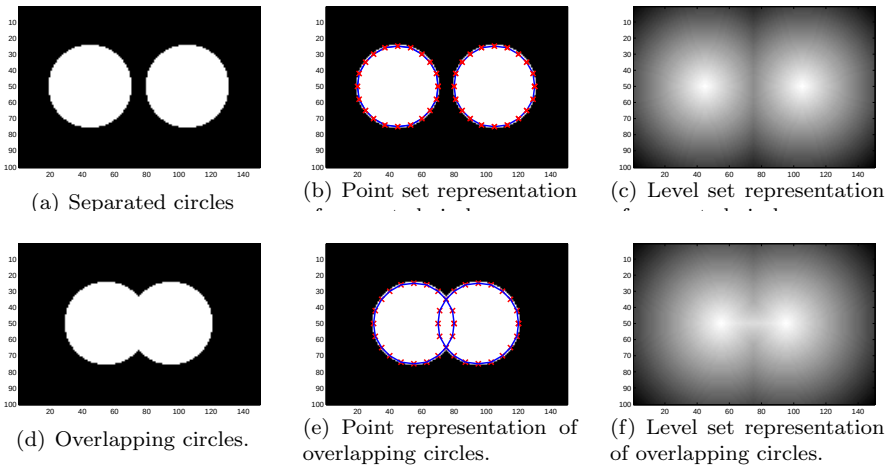
Intermediate points or pseudo landmarks are often added to avoid a too sparse representation of the shape. However, in cases where it is hard to find or identify good landmarks, it is difficult to secure a good correspondence between the shapes of the same class, which consequently might lead to faulty model.
An alternative shape representation is signed distance maps with zeros on the boundary, positive value outside the boundary and negative values inside the boundary[1]. Clearly, this approach does not require the corresponding landmarks. This technique was first presented by S. Osher and J. Sethian [17] to analyze and compute the motion of an open region under er velocity field v. Later, level set functions were introduced in a statistical shape framework by Leventon [15]. An extensive survey on level set methods and their applications is provided in [16].

Another advantage with level set functions are their ability to handle topology changes in the shape, e.g. merging and splitting. The point set cannot handle merging and splitting of the shape. This is illustrated in Figure 11.1.

11.3 Obtaining the outline of an object

Outlining objects in images can easily become a time consuming task as it requires much manual interaction. Manually marking of all pixel or voxels on the border of an object is a painfully and tedious task. Nevertheless, the process can easily speeded up with a few simple tools.

[1] Also denoted level set function (LSF).



(a) Separated circles

(b) Point set representation

(c) Level set representation

(d) Overlapping circles.

(e) Point representation of overlapping circles.

(f) Level set representation of overlapping circles.

Figure 11.1: Region merging and splitting. LSF can handle the topology change while the point set representation fails.

11.3.1 Linear interpolation

A crude, but often effective way of speeding up the outlining, is to use linear interpolation. With this method you simply annotate the object in suitable points along the boundary, and the boundary is then interpolated between the points.

11.3.2 Spline interpolation

Using a linear interpolation approach, when outlining boundaries where there is a high curvature, is not much better than marking all pixel on the boundary. Splines are much more flexible why they do not suffer from this weakness. Hence, instead of linear interpolation the boundary can be interpolated with a spline in the parts with curvature.

11.3.3 Edge search

The two methods for outlining a shape, which have been presented so far, do not utilize any image information. However, if there are strong evidence of how the boundary runs between two border points, it would be stupid not to exploit the

information. A method for finding the edge, which connects two border points, is sketched beneath:

- Place two points on the boundary of the object.
- Subsample the image in a rectangle of size $d \times n$ where d is the ceiling of the Euclidian distance between the two points and n is a predefined constant integer. The rectangle is placed on the image such that the middles of the two sides with length n of the rectangle correspond to the two border points. This step is illustrated in Figure 11.2.
- Apply a canny edge detection filter on the subsampled image.
- The problem is now simplified to finding the cheapest left-right path across a feature image with fixed start and end points. This can easily be solved with generic programming.

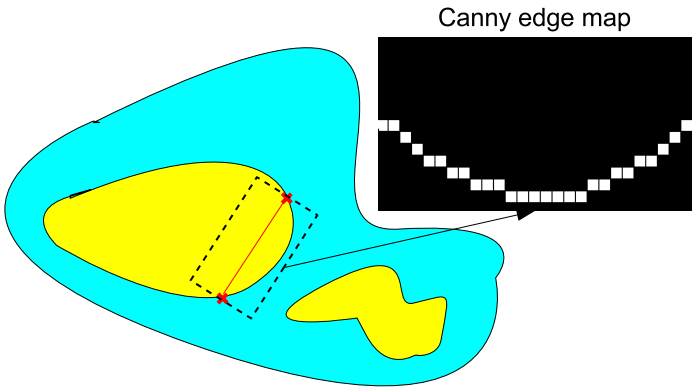


Figure 11.2: Illustration the edge search outlining tool.

11.4 Aligning binary shapes

Before a shape model can be build, it is essential that all variation in the shape due to rotation, scale and translation is eliminated in accordance with Definition 6.5.
Several techniques for the registration and alignment of images have been proposed in literature. However, as shapes can be encoded as binary images[2] the

[2] 0 outside the boundary and 1 inside the boundary

task is greatly simplified. This thesis highlights two method for the alignment of binary shapes; a method which aligns images by equalizing the first and second order moment in the images, and a method, which aligns binary images by minimizing a joint energy functional using a gradient descent scheme. The first method is used to obtain a qualified first guess, while the latter is used to fine tune the alignment. There a two reason for applying the first method before the second:

- As the second method works by minimization of a *joint* energy functional the computation grows significantly with the number of images.
- As the second method employs a gradient descent method to minimize an energy functional, it might get stuck in a local minimum. Hence, an initialization close to the solution decreases the chance of falling in to a local minimum.

11.4.1 Registration by moments

In rigid registration by moments, images are aligned such that the first order[3] and the second order moment[4] are equal for all the images.

The first order moment of an image N-dimensional image I is defined by:

Definition 11.8
The first order moment of an image I is given by

$$\mathbf{m}_1 = \frac{\sum_{s_i \in \mathcal{S}} I(s_i) \cdot \mathbf{v}_i}{\sum_{s_i \in \mathcal{S}} I(s_i)}, \tag{11.1}$$

where $\mathcal{S}$ is the set of all voxels in the image, and $\mathbf{v}_i$ is the Cartesian coordinates of the voxel s_i

Likewise, the second order moments is defined by

[3] Center of gravity
[4] Spatial dispersion matrix

Definition 11.9
The second order moment of an image I is given by

$$\mathbf{M}_2 = \frac{\sum_{s_i \in \mathcal{S}} I(s_i) \cdot (\mathbf{v}_i - \mathbf{m}_1)(\mathbf{v}_i - \mathbf{m}_1)^\top}{\sum_{s_i \in \mathcal{S}} I(s_i)}. \tag{11.2}$$

Armed with these definitions the following lemma can be produced:

Lemma 11.1
Given an image I_a with the first order moment $\mathbf{m}_1^a$ and the second order moment $\mathbf{M}_2^a$. Then the image

$$I_b\{\mathbf{y}\} = I_b\{\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^{-1}(\mathbf{x} - \mathbf{m}_1)\} = I_a\{\mathbf{x}\} \tag{11.3}$$

will have the moments

$$\mathbf{m}_1^b = \mathbf{0} \quad \textit{and} \quad \mathbf{M}_2^b = \mathbf{I}, \tag{11.4}$$

where

$$\mathbf{M}_2^a = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}})^\top. \tag{11.5}$$

This lemma leads to the following theorem:

Theorem 11.1
Given two images I_a and I_b with the moments $\mathbf{m}_1^a$, $\mathbf{m}_1^b$, $\mathbf{M}_2^a$ and $\mathbf{M}_2^b$. The affine transformation $\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{t}$ which gives image I_b the same first and second order moments as image I_a is determined by

$$\mathbf{R} = \mathbf{U}_a\mathbf{\Lambda}_a^{\frac{1}{2}}\mathbf{\Lambda}_b^{-\frac{1}{2}}\mathbf{U}_b^{-1} \tag{11.6}$$

and

$$\mathbf{t} = \mathbf{m}_1^a - \mathbf{R}\mathbf{m}_1^b, \tag{11.7}$$

where the $\mathbf{U}$'s and the $\mathbf{\Lambda}$'s are given in Lemma 11.1.

The proofs of Lemma 11.1 and the Theorem 11.1 are omitted from thesis but they can be found in [10]. If the difference between the eigenvalues is small

it is essential to confirm that the ordering is correct. Furthermore, there are four different solutions to the equalization of the moments, why it is necessary to check that the transformed image has not been flipped around the moment axes. Typical, this is done by checking the diagonal of the rotation matrix. Often, it is reasonable to assume that the shapes should not be rotated more than 90 degrees. Consequently, the diagonal values of the rotation matrix must be positive.

Note, as the alignment process is done by equalization of moments the scaling differs in the dimensions of the image. Often this scaling unwanted why the following theorem arises:

Theorem 11.2
Given two images I_a and I_b with the moments $\mathbf{m}_1^a$, $\mathbf{m}_1^b$, $\mathbf{M}_2^a$ and $\mathbf{M}_2^b$. The affine transformation $\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{t}$, which aligns the axes of first and second order moments of image I_b, with those of image I_a is determined by

$$\mathbf{R} = \mathbf{U}_a \mathbf{U}_b^{-1} \tag{11.8}$$

and

$$\mathbf{t} = \mathbf{m}_1^a - \mathbf{R}\mathbf{m}_1^b, \tag{11.9}$$

where the $\mathbf{U}$'s are given in Lemma 11.1.

11.4.2 Alignment by minimization of joint energy function

The task is to find the set of pose parameters $\{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n, \}$ given the set of N-dimensional binary shape images $\{I^1, I^2, \ldots, I^n\}$ which gives the optional alignment with respect to some joint energy functional. The link between the image I and the transformed image $\tilde{I}$, subject to the pose parameter $\mathbf{p}$, is

$$I\{\mathbf{x}\} = \tilde{I}\{\tilde{\mathbf{x}}\} \tag{11.10}$$

where

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ 1 \end{bmatrix} = T[\mathbf{p}] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \tag{11.11}$$

The looks of the transformation matrix $T[\mathbf{p}]$ depend on the type of transformation which is to be eliminated. E.g. the 2D the Euclidian transformation

matrix[5]

$$T[\mathbf{p}] = \begin{bmatrix} h \cdot \cos(\theta) & h \cdot \sin(\theta) & a \\ -h \cdot \sin(\theta) & h \cdot \cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix}. \tag{11.12}$$

where h is scaling, a and b are translation and θ is rotation. If it is desired to have rotation and scaling around the center of mass of the binary shape the transformation matrix becomes

$$T[\mathbf{p}] = \begin{bmatrix} h \cdot \cos(\theta) & h \cdot \sin(\theta) & a - h \cdot \cos(\theta) \cdot c_1 - h_1 \cdot \sin(\theta) \cdot c_2 \\ -h \cdot \sin(\theta) & h \cdot \cos(\theta) & b - h \cdot \sin(\theta) \cdot c_1 - h_2 \cdot \cos(\theta) \cdot c_2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{11.13}$$

where $[c_1 \ c_2]^\top$ is the center of mass of the binary shape.

The energy function employed in this thesis is the one suggested in [25, 23]. A good criterium for the alignment of images is mutual information. That is, we seek to maximize the mutual information between the images. In the binary image case, this is effectively the same as to maximize the amount of overlap or to minimize difference between the images. So, the strategy is to minimize the energy

$$E_{align} = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left\{ \frac{\int \ldots \int_\Omega (\tilde{I}^i - \tilde{I}^j)^2 dV}{\int \ldots \int_\Omega (\tilde{I}^i + \tilde{I}^j)^2 dV} \right\}, \tag{11.14}$$

where Ω denotes the image domain. The term $\int \ldots \int_\Omega (\tilde{I}^i + \tilde{I}^j)^2 dV$ is added to avoid shrinking.

To minimized the energy function a gradient descent scheme can be applied. The gradient of the energy function (11.14) is

[5] Rigid transformation.

$$\nabla \mathbf{p}^i E_{align} = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left\{ \frac{2 \int \ldots \int_\Omega (\tilde{I}^i - \tilde{I}^j) \nabla \mathbf{p}^i \tilde{I}^i dV}{\int \ldots \int_\Omega (\tilde{I}^i + \tilde{I}^j)^2 dV} \right. \tag{11.15}$$

$$\left. - \frac{\int \ldots \int_\Omega (\tilde{I}^i - \tilde{I}^j)^2 dV \int \ldots \int_\Omega (\tilde{I}^i + \tilde{I}^j) \nabla \mathbf{p}^i \tilde{I}^i dV}{\left(\int \ldots \int_\Omega (\tilde{I}^i + \tilde{I}^j)^2 dV \right)^2} \right\},$$

where $\nabla \mathbf{p}^i \tilde{I}^i$ is the gradient of the transformed image.
By applying the chain rule we obtain

$$\nabla \mathbf{p}^i \tilde{I}^i = \frac{\partial \tilde{I}^i}{\partial \tilde{\mathbf{x}}} \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}^i} \tag{11.16}$$

$$= \begin{bmatrix} \frac{\partial \tilde{I}^i}{\partial \tilde{x}_1} & \frac{\partial \tilde{I}^i}{\partial \tilde{x}_2} & \ldots & \frac{\partial \tilde{I}^i}{\partial \tilde{x}_N} & 0 \end{bmatrix} \frac{\partial T[\mathbf{p}^i]}{\partial \mathbf{p}^i} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ 1 \end{bmatrix}. \tag{11.17}$$

As the alignment problem is under determined the pose of one images need to be fixed.

11.4.3 Multiresolution alignment

As mentioned earlier in the section, a gradient descent scheme has some undesirable properties such as slow convergence and getting stuck in local minima. Registration by moments reduces this problem. Nevertheless, this might not be enough if the objects are complex and large. To further reduce the chance of getting trapped in local minima and to speed up the alignment process, the described method can be adapted into a multiresolution scheme. The simple idea is to subsample the training set to a coarser representation, and then align the lower resolution sets first. For every image in the training set an image pyramid, as shown in Figure 11.3, is constructed where the base image (level 0) is the original image[6]. First the images at the top level are aligned, and the

[6]Since we are working with binary image no smoothing is used between the level such as it would have been done in a gaussian image pyramid.

result is then passed down to next level and used as an initial estimate. When this routine has progressed down to the base level the alignment is finished.

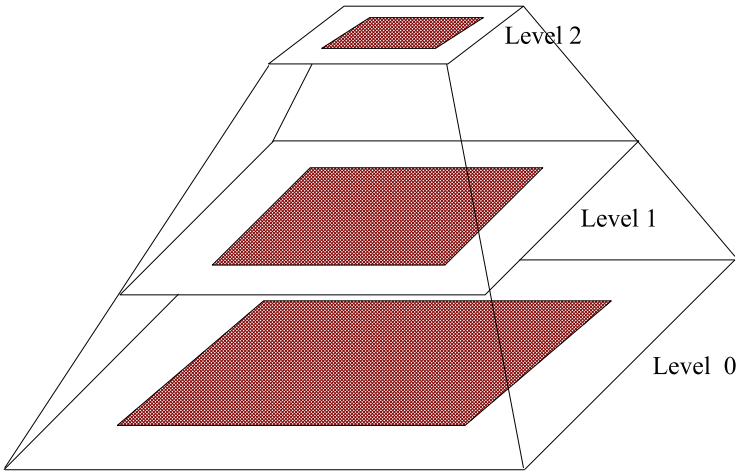


Figure 11.3: Binary image pyramid formed by subsampling.

11.5 Implicit parametric shape model

After the alignment the binary images are converted to level set functions $\{\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_n\}$, where $\mathbf{\Psi}_i \in \mathbb{R}^{M_1 \times M_2 \times \ldots \times M_N}$. From the level set functions, the mean level set function can be computed by

$$\bar{\mathbf{\Phi}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{\Psi}_i. \tag{11.18}$$

To extract the variation of the level set functions, mean offset functions are computed by subtracting the mean level set function:

$$\hat{\mathbf{\Psi}}_i = \mathbf{\Psi}_i - \bar{\mathbf{\Phi}}. \tag{11.19}$$

The mean offset functions are then sampled into lexicographical ordered columns $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n\}$ and arranged in to a $M \times n$[7] matrix $\mathbf{G} = [\boldsymbol{\psi}_1\ \boldsymbol{\psi}_2\ \ldots\ \boldsymbol{\psi}_n]$. The covariance matrix $\frac{1}{n}\mathbf{G}\mathbf{G}^\top$ is decomposed to

$$\frac{1}{n}\mathbf{G}\mathbf{G}^\top = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{11.20}$$

[7] $\boldsymbol{\psi}_i \in \mathbb{R}^M$ where $M = M_1 \cdot M_2 \cdot \ldots \cdot M_N$.

where $\mathbf{U}$ is an $M \times n$ matrix whose columns $\{\tilde{\psi}_i\}$ represent the principal variational modes (VM), and $\mathbf{\Lambda}$ is a $n \times n$ diagonal eigenvalue matrix (sorted so that $\lambda_i > \lambda_{i+1}$). $\tilde{\psi}_i$ is sampled in to the principal variational mode image $\mathbf{\Phi}_i$.

Given some empirical selected k, where $k < n$, new level set functions can be generated from

$$\mathbf{\Phi}[\mathbf{w}] = \bar{\mathbf{\Phi}} + \sum_{i=1}^{k} w_i \mathbf{\Phi}_i, \tag{11.21}$$

where $\mathbf{w} = \{w_1 \ w_2 \ \dots \ w_k\}$ are the weights for the k variational modes. Note that the eigenvalue λ_i is the variance of the weight w_i. The level set function $\mathbf{\Phi}[\mathbf{w}]$ can now be used as an implicit shape model as it is assumed that the zero level set corresponds to the boundary of the shape. Hence, the shape is indirectly varied when the weights are varied. For the remainder of this thesis the shape model will be denoted LSFSM - short for level set function shape model.

11.5.1 Computing the eigenvectors efficiently

It is a well known fact that the computation time of an eigenvector decomposition of a $n \times n$ matrix is $\mathcal{O}(n^3)$.
In most cases, the dimension n of the data vectors is significantly larger than the number of samples m which makes the covariance matrix $\mathbf{\Sigma}_n = \frac{1}{n}\mathbf{G}\mathbf{G}^\top$ highly singular. We can exploit this observation and compute the eigenvectors and eigenvalues from a smaller $m \times m$ matrix.
It can be shown that all m eigenvalues of the matrix $\mathbf{\Sigma}_m = \frac{1}{m}\mathbf{G}^\top\mathbf{G}$ are equal to the non-zero eigenvalues of $\mathbf{\Sigma}_n$. Further the eigenvectors of $\mathbf{\Sigma}_n$ can be computed from

$$\mathbf{\Phi}_n = \mathbf{G}\mathbf{\Phi}_m. \tag{11.22}$$

To assure that the eigenvectors of $\mathbf{\Phi}_n$ have unit length they have to be normalized.

11.5.2 An inconsistent framework?

Unfortunately, the LSFSM has a couple of flaws:

- Linear operations on a set of signed distance maps do not produce a signed

distance map. Hence, the level set functions generated by our model are not true signed distance maps.

- It is not completely intuitive that the sum of squares of the difference between two signed distance maps of two shapes provides a good similarity measure between the shapes.
- Obviously, representing shapes by their signed distance map introduces a large degree of redundancy.

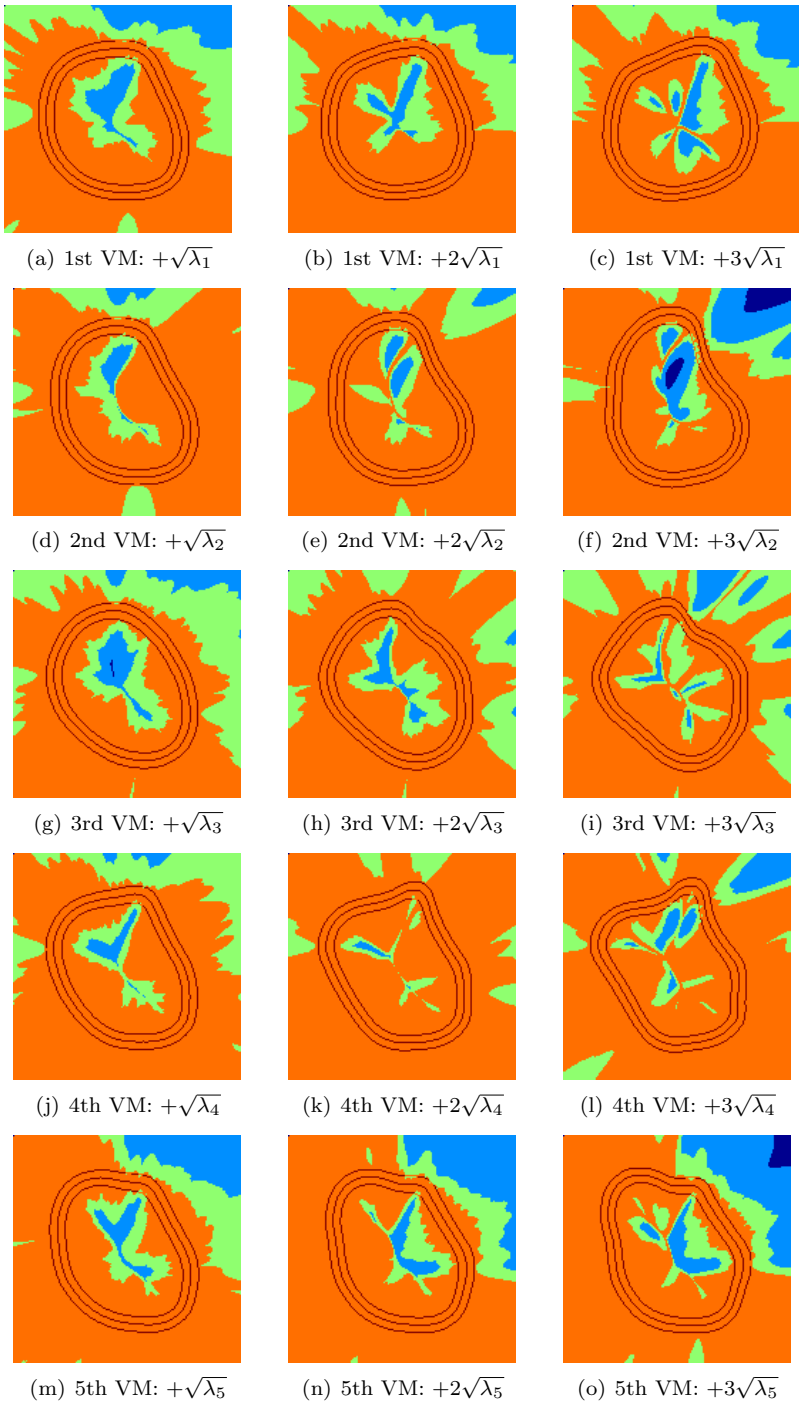
Especially, the first flaw raises a great deal of concern. Thai et al. [25] argue that it is a theoretical issue rather than a practical. Surely, a method should be judged more on the performance in practise rather then the theoretical foundation. Nevertheless, it would be nice to have an idea about how far from true signed distance maps the generated level set functions are, and what can be done to secure that the generated level set functions are as close to signed distance maps as possible.

To test the LSFSM a model of the loin (50) was made from all of the outlines of loin. Information about the model can be found in Appendix E.2.

First, the ability of the generated level sets to approximate true signed distance map were tested with the following recipe.

- Generate level set functions by varying the variational modes 1, 2, 3 standard deviations.
- Estimate the zero level sets of the generated level set functions.
- Compute the true signed distance maps of the estimated zero level sets.
- For all generated level set functions compute the absolute difference map between the level set function and the corresponding true signed distance map.
- Segment the difference map by thresholding, e.g. with the values 1, 2 and 5.

The result is shown in Figure 11.4. In general, it seems that the more the model is varied from the mean map the worse the approximation gets. The majority of the generated level set area approximates the true distance map with an error less than 1.



(a) 1st VM: $+\sqrt{\lambda_1}$ (b) 1st VM: $+2\sqrt{\lambda_1}$ (c) 1st VM: $+3\sqrt{\lambda_1}$

(d) 2nd VM: $+\sqrt{\lambda_2}$ (e) 2nd VM: $+2\sqrt{\lambda_2}$ (f) 2nd VM: $+3\sqrt{\lambda_2}$

(g) 3rd VM: $+\sqrt{\lambda_3}$ (h) 3rd VM: $+2\sqrt{\lambda_3}$ (i) 3rd VM: $+3\sqrt{\lambda_3}$

(j) 4th VM: $+\sqrt{\lambda_4}$ (k) 4th VM: $+2\sqrt{\lambda_4}$ (l) 4th VM: $+3\sqrt{\lambda_4}$

(m) 5th VM: $+\sqrt{\lambda_5}$ (n) 5th VM: $+2\sqrt{\lambda_5}$ (o) 5th VM: $+3\sqrt{\lambda_5}$

Figure 11.4: The thresholded difference maps between 15 generated level set functions and corresponding true signed distance maps. The difference maps have been thresholded with the values 1, 2, 5. The orange, green and light blue areas correspond to differences less than 1, 2 and 5, respectively. The three dark red lines are the level sets -5, 0 and 5 of the true signed distance map.

A process, which has been named shape reinitialization by the author, assures that the generated level set functions are as close to true signed distance map as possible without changing the boundary of the shape significant. The process contains the following step:

1. Compute the zero level set $\mathbf{\Phi}[\mathbf{w}_{org}] = 0$ of the level set function $\mathbf{\Phi}[\mathbf{w}_{org}]$.
2. Compute the true signed distance map $\mathbf{\Psi}_t$ from the zero level set $\mathbf{\Phi}[\mathbf{w}_{org}] = 0$, and rearrange it is to the column vector $\boldsymbol{\psi}_t$.
3. Rearrange the mean map into the column vector $\bar{\boldsymbol{\psi}}$, and sample the principal modes $\mathbf{\Phi}_i$ in to the columns $\tilde{\boldsymbol{\psi}}_i$ and construct the $M \times k$ dimensional matrix $\tilde{\mathbf{U}} = [\tilde{\boldsymbol{\psi}}_1 \ \tilde{\boldsymbol{\psi}}_2 \ \dots \ \tilde{\boldsymbol{\psi}}_k]$.
4. Obtain a new shape parameter estimate $\mathbf{w}_{new}$ by solving the equation[8]

$$\boldsymbol{\psi}_t = \bar{\boldsymbol{\psi}} + \tilde{\mathbf{U}}\mathbf{w}_{new} \Leftrightarrow \mathbf{w}_{new} = \tilde{\mathbf{U}}^\top(\boldsymbol{\psi}_t - \bar{\boldsymbol{\psi}}). \tag{11.23}$$

Obviously, the level set function, which is generated by the parameters $\mathbf{w}_{new}$, is in a least square sense as close to the true distance function as possible. Further, the actual surface configuration of the shape does not change significantly due to the redundancy in a level set function. To support these claims ten shapes were generated randomly and then reinitialized. Two types of reinitialization were applied to the generated shapes - reinitialization by partial fitting and initialization by full fitting. Full fitting is completely similar to the reinitialization sketched above, while partial fitting differs as only the area of the true signed distance map, which lays within an upper and lower level set, is used for the fitting. In this test the tenth and the minus tenth level set were chosen to be the lower and upper limit level set, respectively.
Each shape parameter w_i was chosen randomly between $\pm 3\sqrt{\lambda_i}$ from a uniform distribution. For every trial the true signed distance map was subtracted from the generated, the full fitted and the partial fitted level set function, and the mean square and the absolute error were computed. The result of the test is shown in Table 11.1. In all of the ten trials the outcome was the same, the mean squared error and the absolute error of the full fitted level set functions are smaller than the mean squared and the absolute errors of the generated level set function. These results are confirmed by Figure 11.5 and Figure 11.6, which show the level set functions, the difference maps and the zero level set for the first and second trial, respectively. Under the assumption that the shape parameters w_i are independent and gaussian, we can further conclude that the fitted level set function in general are more "plausible" then the randomly generated as the

[8] $\tilde{\mathbf{U}}^\top = \mathbf{U}^{-1}$, since the columns in $\tilde{\mathbf{U}}$ are orthogonal and have unit length.

measure $0.5\sum_{i=1}^{k}\frac{w_i^2}{\lambda_i}$ is lower for all but one of the random generated level set functions. This is a nice side effect. Unfortunately, shape reinitialization raises as much concern as it removes. If there exist more than one parametrization of the same shape, how do we determine the suitable number of principal components to use? Is 95% of the variance in the set of signed distance maps equal to 95% of the shape variation?

Trial	Level set function	$0.5\sum_{i=1}^{k}\frac{w_i^2}{\lambda_i}$	Mean sq. err.	Mean abs. err.
1	Generated	1757.9	1.0148	2.0798
	Full fit	1879.5	0.51546	0.48055
	Partial fit	2163.6	0.79973	1.2772
2	Generated	2428.5	1.9633	7.9305
	Full fit	2192.8	0.54587	0.55964
	Partial fit	1722.5	0.88269	1.7703
3	Generated	2115.6	1.1046	2.4678
	Full fit	1397.9	0.45189	0.36249
	Partial fit	1359.9	0.46164	0.39814
4	Generated	3017.4	2.1188	14.556
	Full fit	2269.1	0.60891	0.71118
	Partial fit	2067.8	0.83795	1.4334
5	Generated	3785.0	1.3271	3.3838
	Full fit	3420.6	0.59417	0.62987
	Partial fit	2711.5	1.1534	3.6563
6	Generated	4422.4	0.98322	1.9654
	Full fit	4057.6	0.64511	0.78854
	Partial fit	4561.2	1.0035	2.5318
7	Generated	3600.5	1.7673	9.5653
	Full fit	3578.6	0.60421	0.65978
	Partial fit	2531.3	1.0207	3.0722
8	Generated	3590.3	1.6381	6.218
	Full fit	2775.5	0.60471	0.66418
	Partial fit	2731.3	0.75818	1.059
9	Generated	3624.4	2.0022	10.698
	Full fit	2249.8	0.62464	0.7073
	Partial fit	2275.3	0.63823	0.76534
10	Generated	4492.1	2.1913	10.651
	Full fit	4009.6	0.60004	0.68254
	Partial fit	3549.9	0.8038	1.505

Table 11.1: The result of reinitializing ten random generated shapes.

At last to emphasize, that reinitialization does not change the boundary of a shape significantly the zero level sets of the randomly generated level set functions are plotted together with the zero level sets of the fitted level set functions in Figure 11.7.

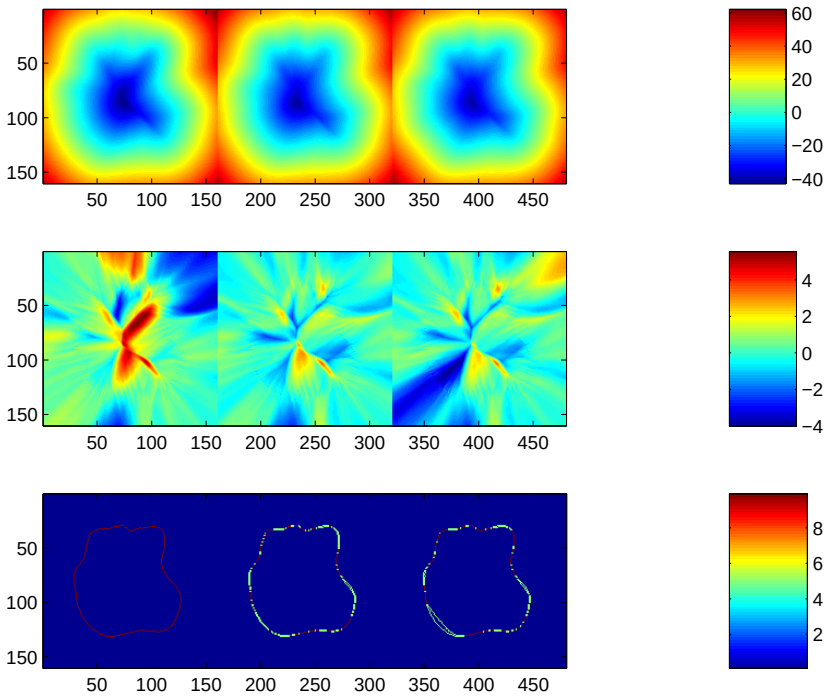


Figure 11.5: The level set functions, the difference maps and the zero level sets from trial 1. The first columns correspond to generated level function, the second column correspond to the full fitted level set function, and the third column correspond to the partial fitted level set function.

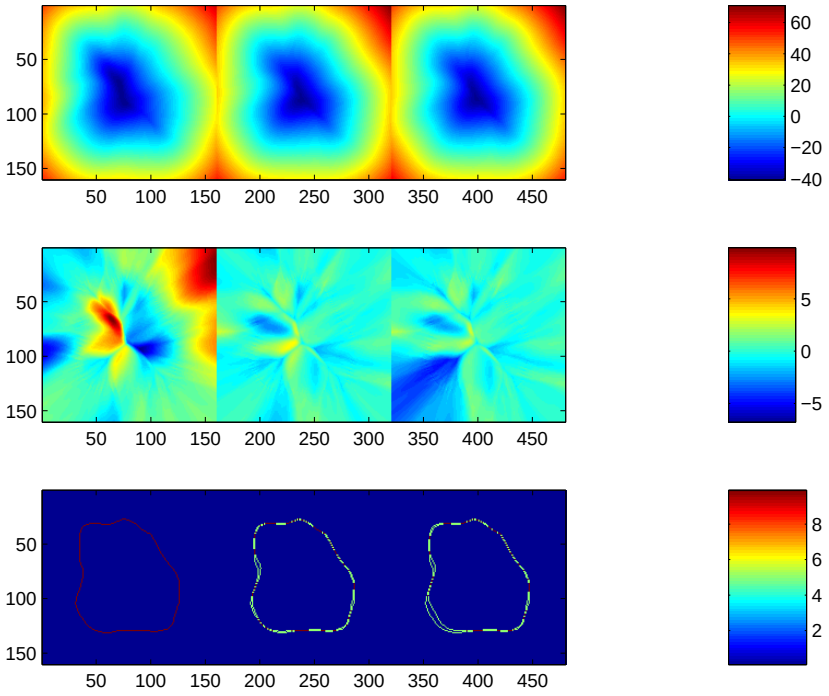
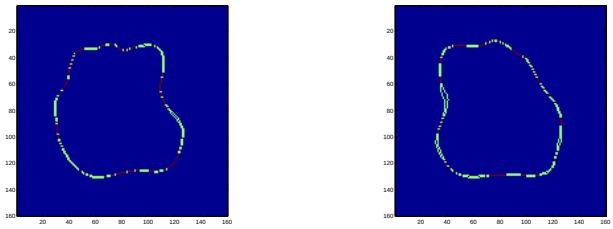
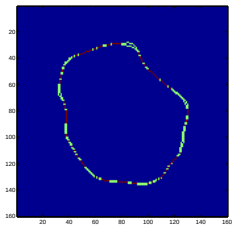


Figure 11.6: The level set functions, the difference maps and the zero level sets from trial 2. The first columns correspond to generated level function, the second column correspond to the full fitted level set function, and the third column correspond to the partial fitted level set function.

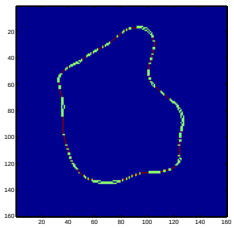


(a) Trial 1

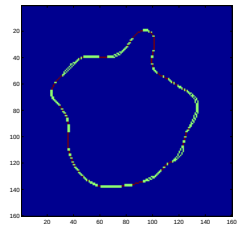
(b) Trial 2



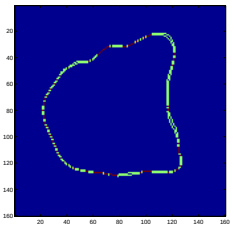
(c) Trial 3



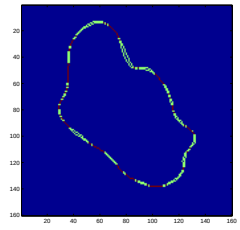
(d) Trial 4



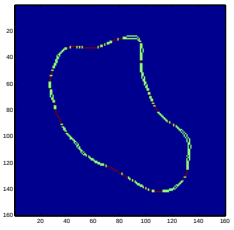
(e) Trial 5



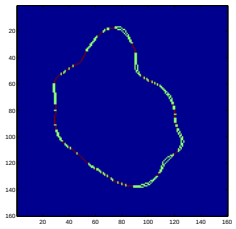
(f) Trial 6



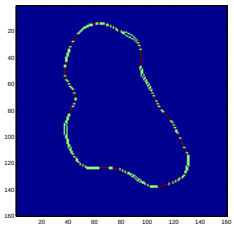
(g) Trial 7



(h) Trial 8



(i) Trial 9



(j) Trial 10

Figure 11.7: The zero level set before and after shape reinitalization. Both level sets are green in non-intersecting parts and dark red in intersecting parts. No significant change in the boundaries of the shapes are present.

By applying PCA on a set of signed distance maps, it is indirectly assumed that the following metric is a good similarity measures between the two shapes Γ and Γ':

$$p_1 = \int \dots \int_{\Omega} (\mathbf{\Psi} - \mathbf{\Psi'})^2 dV, \tag{11.24}$$

where $\mathbf{\Psi}$ and $\mathbf{\Psi'}$ are the signed distance maps of the shapes Γ and Γ', respectively. It can be argued that in many practical cases this metric is rather similar to the squared distance between the boundaries of two shapes:

$$\begin{aligned} p_2 &= \oint_{\overrightarrow{c}'} \mathbf{\Psi}^2 ds + \oint_{\overrightarrow{c}} \mathbf{\Psi'}^2 ds \\ &= \oint_{\overrightarrow{c}'} (\mathbf{\Psi} - \mathbf{\Psi'})^2 ds + \oint_{\overrightarrow{c}} (\mathbf{\Psi} - \mathbf{\Psi'})^2 ds, \end{aligned} \tag{11.25}$$

where $\overrightarrow{c}$ and $\overrightarrow{c}'$ are the boundary curves of Γ and Γ'. Surely, it is naturally to assume that the two metrics are similar as the r level set cannot contain any boundary information, which is not present in the zero level set - BUT it is wrong. The issue is, shape features do not propagate evenly far in a distance map. Figure 11.8 show two U shapes with an equal sized peak. In the left image the peak is placed on the inside arch, and in the right image the peak is placed on the outside arch. Intuitive, the two peaks are evenly important. Nevertheless, the peak in right image will have a larger effect on a distance map than the peak in the left image. This implies, that one must show extreme caution when selecting the size of the signed distance map. An ad hoc approach for determining the optimal size of level set functions is to build a model and then vary all modes with in some predefined intervals (e.g. $\pm 2\sqrt{(\lambda_i)}$) to determine how far to zero level set propagates in the level set functions.

11.5.3 Adding pose to the shape model

Before the PCA is applied the variation in the training set due to pose is removed by the alignment. It is essential incorporate pose back in to the model. The implicit shape model (11.21) becomes

$$\mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \dots, x_n) = \bar{\mathbf{\Phi}}(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) + \sum_{i=1}^{k} w_i \mathbf{\Phi}_i(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n), \tag{11.26}$$

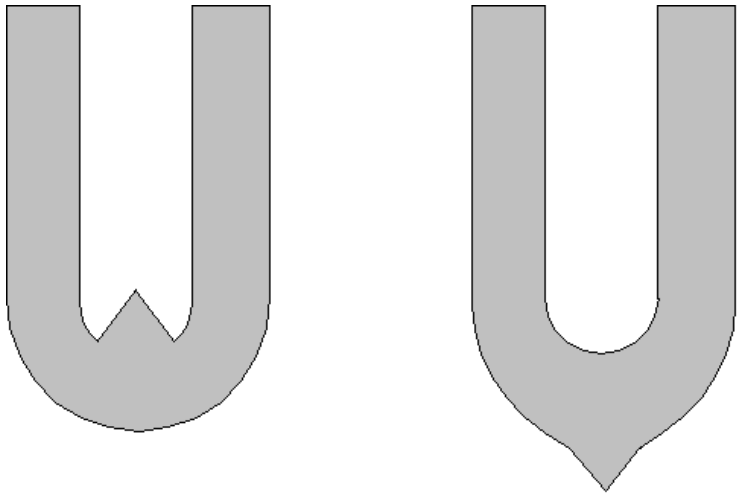


Figure 11.8: Two U shapes with an equal sized peak.

where

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ 1 \end{bmatrix} = T[\mathbf{p}] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \tag{11.27}$$

11.5.4 Finding the zero level set

So far, we have neglected to discuss how the zero level set can be found from a level set function. In some cases, it is necessary to have the zero level set evaluated with subvoxel precision and in other cases voxel precision is good enough. Eg. if only the visual outline of the shape is needed a voxel precision outline is adequate, but if approximations to the integrals in (12.6-12.11, 12.23-12.32) are to be calculates subvoxel precision is necessary to guarantee a good approximation.

Naturally, the best discrete approximation (voxel precision), which can obtained, is the set of zero crossing voxels.
In 2D a pixel is a zero crossing pixel iff there exists a pixel in its four connectivity neighborhood with a higher absolute value and a different sign. More formally,

the pixel s is a zero crossing pixel iff

$$\bigvee_{t \in \mathcal{N}(s)} \mathbf{\Phi}(t) \cdot \mathbf{\Phi}(s) < 0 \wedge abs(\mathbf{\Phi}(s)) < abs(\mathbf{\Phi}(t))$$

is true, where $\mathcal{N}(s)$ is the four-connectivity neighborhood of the pixel s. Figure 11.9 illustrate the principle. Naturally, this can easily the extended to higher dimensions.

							1.6	1.1	1.5					
						0.9	0.4	0.1	0.4	1.2				
					1.2	0.2	-0.6	-0.9	-0.4	0.5				
		1.3	0.9	0.6	0.3	-0.3	-1.3		-0.8	0.2	1.2			
	1.0	0.1	-0.3	-0.4	-0.9	-1.2				-0.3	0.1	0.8		
	0.6	-0.4	-1.4	-1.4							-0.7	0.4	1.4	
	0.6	-0.4	-1.4								-0.8	0.3	1.3	
	0.9	0.1	-0.8	-1.3	-1.3	-0.9	-0.3	-0.4	-0.3	-0.3	0.1	0.8		
		0.7	0.1	-0.3	-0.3	0.1	0.7	0.6	0.7	0.7	1.1			
			0.9	0.7	0.7	1.0								

Figure 11.9: The red line indicates zero level set. The closest pixel to the zero level set is marked gray.

Since the level function $\mathbf{\Phi}$ is a discrete representation, it is impossible to obtain the exact zero level set. Nevertheless, a good estimate with subvoxel precision can rather easily be computed. Given a set $\mathcal{S}$ of zero crossing voxels and the corresponding N-dimensional level set function $\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]$, the closets point on the actual zero level set for all the voxels in $\mathcal{S}$ can estimated by utilizing the fact that the generated level set functions almost are true signed distance maps. Hence, if it is assumed that the value of $\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]$ at the voxel s is the true Euclidian distance from the center of the voxel s to the closets point on zero level set. By moving the value of a zero crossing voxel away from the voxel in the negative normal direction, a better estimate of a point on the zero level set boundary is obtained. An iterative scheme, which computes a set of zero level points from a set of zero crossing voxels, follows:

1. Initialize the point set $\mathcal{P}^0$ to the centers of the zero crossing voxels.
2. Re-estimate the point set $\mathcal{P}^t$:

- For every point $\mathbf{v}_i^t$ in $\mathcal{P}^t$ estimate the normal
  $\mathbf{n}_i^t = \left[\pm \frac{\partial \mathbf{\Phi}[\mathbf{w},\mathbf{p}]}{\partial x_1}\{\mathbf{v_i}\} \quad \pm \frac{\partial \mathbf{\Phi}[\mathbf{w},\mathbf{p}]}{\partial x_2}\{\mathbf{v_i}\} \quad \cdots \quad \pm \frac{\partial \mathbf{\Phi}[\mathbf{w},\mathbf{p}]}{\partial x_N}\{\mathbf{v_i}\} \quad \right]^\top$
- Compute the set $\mathcal{P}^{t+1}$ with the update rule $\mathbf{v}_i^{t+1} = \mathbf{v}_i^t - \mathbf{n}_i^t \cdot \mathbf{\Phi}[\mathbf{w},\mathbf{p}]\{\mathbf{v}_i^t\}$

3. If the points in $\mathcal{P}^t$ are too far from the zero level set go back to 2. A suitable stop criteria is $(\frac{1}{m}\sum_i^m \mathbf{\Phi}[\mathbf{w},\mathbf{p}]\{\mathbf{v}_i^t\})^2 \leq \tau$ where m is the size of the point set, τ is a preselected threshold.

In most cases one iteration with above algorithm will yield a satisfactory solution. The effect of performing one iteration with the above algorithm is shown in Figure 11.10.

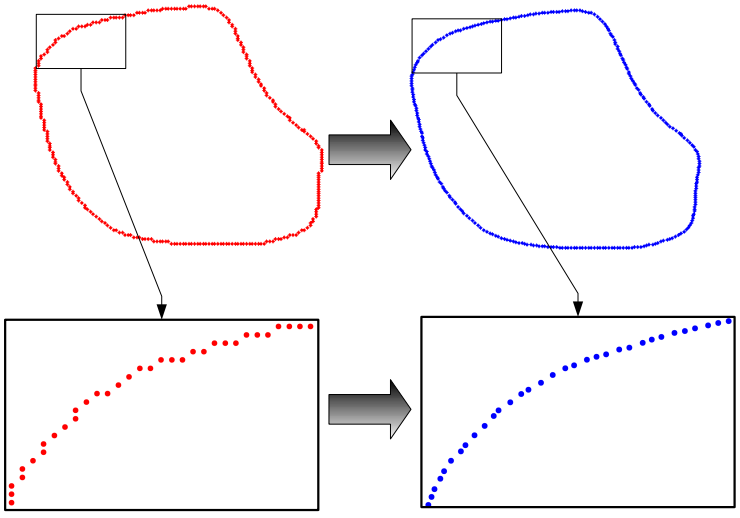


Figure 11.10: Computing the zero level set with subpixel precision.

11.6 Discussion

The LSFSM does not require point correspondence in the training set. This is a very nice property but it comes at a high price. It assumes incorrectly that signed distance maps lay on a linear manifold, and it relies on a rather suspect similarity measure. A method called shape reinitialization tries to redeem the first of the drawbacks but raises as much concern as it removes.
In conclusion, the LSFSM must proof itself in practise.

Chapter 12

Region based segmentation

In contrast to edge based segmentation (EB), region based (RB) utilizes intensity information from the entire inside region of the segmenting curve to guide the evolution of the segmenting curve. Hence, RB segmentation uses regional statistics such as sample mean and variance to drive the segmenting curve towards the boundary of the object, where as EB segmentation only utilizes pixel intensities near the segmenting curve - often gradient information.
This gives RB based segmentation models a couple of advantages:

1. RB segmentation models are more global than EB segmentation model.

2. They are more robust with regards to noise as they in general do not rely on gradient operators.

3. They do not need inflationary terms to drive the curve towards the boundaries, which are common in EB segmentation models.

With confidence on the ability of RB segmentation, we continue with the formulation of the RB segmentation model.

12.1 Coupling the RB segmentation model with a statistical shape model

In [23], the curve $\overrightarrow{C}$, which segments an image I into an inside region and an outside region, is defined by the zero level set of the level set function $\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]$:

$$\overrightarrow{C} = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N : \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) = 0\}. \tag{12.1}$$

The curve $\overrightarrow{C}$ is evolved by varying the shape parameters $\mathbf{w}$ and the pose parameters $\mathbf{p}$. In a segmentation, the evolution of the curve in controlled by minimization of suitable energy function formulated upon suitable region statistics. The inside and the outside region are given by

$$R^u = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N : \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) \leq 0\} \tag{12.2}$$

and

$$R^v = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N : \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) > 0\}. \tag{12.3}$$

As mention in the beginning of the chapter, two common region statistics are mean and variance. In order to guarantee that these measures provide useful information to the evolution of the segmenting curve both the inside and outside region need to be homogeneous. Though, it might be reasonable to assume that the inside region is homogeneous, it is doubtful that the outside region R^v as given in (12.5) is homogeneous in practise. By introducing a region of interest it is possible to overcome the short coming. The region of interest should be selected such that the outside region is (almost) homogenous in the final result. Naturally, it will most likely be necessary to update the region of interest between iterations.
An obvious choice, when dealing with level set functions, is to define the region of interest by some positive level set. The formulation of the inside and outside region is redefined to

$$R^u = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^2 : \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) \leq 0\} \tag{12.4}$$

and

$$\begin{aligned} R^v = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^2 : \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) > 0 \\ \wedge \mathbf{\Phi}[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) < r\}. \end{aligned} \tag{12.5}$$

Hence, the region of interest is $R^u + R^v$. The principal is illustrated in Figure 12.1.

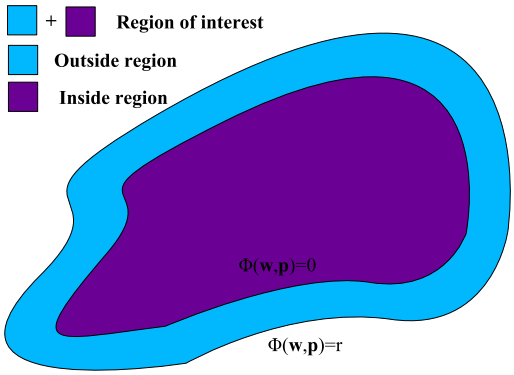


Figure 12.1: Illustration of the inside region, the outside region and the region of interest

12.2 Image statistics

After having coupled the basic RB segmentation model with the LSFSM, it is now possible to introduce the most basic region statistics: *hypervolume*, *intensity sum* and *squared intensity sum*. Given a level set function they can be computed as follows:

$$V_u = \int \dots \int_{R^u+R^v} \mathcal{H}(-\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.6}$$

$$V_v = \int \dots \int_{R^u+R^v} \mathcal{H}(\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.7}$$

$$S_u = \int \dots \int_{R^u+R^v} I\mathcal{H}(-\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.8}$$

$$S_v = \int \dots \int_{R^u+R^v} I\mathcal{H}(\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.9}$$

$$Q_u = \int l, dots \int_{R^u+R^v} I^2\mathcal{H}(-\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.10}$$

$$Q_v = \int \dots \int_{R^u+R^v} I^2\mathcal{H}(\mathbf{\Phi}[\mathbf{w},\mathbf{p}])dV, \tag{12.11}$$

where $\mathcal{H}$ is the Heaviside function given by

$$\mathcal{H}(\mathbf{\Phi}[\mathbf{w},\mathbf{p}]) = \begin{cases} 1, & \text{if } \mathbf{\Phi}[\mathbf{w},\mathbf{p}] \geq 0 \\ 0, & \text{if } \mathbf{\Phi}[\mathbf{w},\mathbf{p}] < 0 \end{cases}. \tag{12.12}$$

Upon these statistics other common region statistics can be formulated:

- The average or mean intensities in the regions R^u and R^v

$$\mu = \frac{S_u}{V_u} \text{ and } \upsilon = \frac{S_v}{V_v}.$$

- The sample variances in the regions R^u and R^v

$$\sigma_u^2 = \frac{Q_u}{V_u} - \mu^2 \text{ and } \sigma_v^2 = \frac{Q_v}{V_v} - \upsilon^2.$$

More complex region statistics, which incorporate knowledge about the distribution of the intensities belonging to the regions, can also be formulated:

$$\begin{aligned}
\hat{h}(I|L = R^u) &= -\frac{1}{V_{R^u}} \int \dots \int_{R^u + R^v} \log(\hat{p}_{R^u}(I)) \mathcal{H}(-\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]), \\
\hat{h}(I|L = R^v) &= -\frac{1}{V_{R^v}} \int \dots \int_{R^u + R^v} \log(\hat{p}_{R^u}(I)) \mathcal{H}(\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]),
\end{aligned}$$

where $\hat{h}(.|.)$ is the estimate of the differential conditional entropy $h(.|.)$, and $\hat{p}_{R^u}$ and $\hat{p}_{R^v}$ are estimates of the probability density functions p_{R^u} and p_{R^v} of I in the inside and outside region, respectively. Note, these statistics are not necessarily effected by the lack of homogeneity in the inside and outside region as $\hat{p}_{R^u}$ and $\hat{p}_{R^v}$ do not require that R^u or R^v are homogeneous.

12.3 Energy functions

Trivial, the energy functions used to guide a segmentation should provide good measures for the fit of the segmentation. Nevertheless, the best choice of energy function highly depends on the problem. If the intensity distributions in the outside and inside region are well separated and if the distributions have approximately the same covariance structure, the difference in mean intensity is a good measure. In more complex scenarios with (different covariance structure, many mixed pixel and noise) the difference in mean measure might not produce adequate results. A more robust measure is the sum of the variances in the inside and outside region. For even more robustness, knowledge about the distribution of the intensities in the outside and inside region can be incorporated. The three types of measures are formulated as energy functions below:

1. *The binary mean model*: ([26])

$$E_{binary} = -\frac{1}{2}(\mu - \upsilon)^2 = -\frac{1}{2}\left(\frac{S_u}{A_u} - \frac{S_v}{A_v}\right)^2. \tag{12.13}$$

2. *The Chan-Vese model*: ([1])

$$\begin{aligned} E_{cv} &= \int_{R^u}(I-\mu)^2 dV + \int_{R^v}(I-\upsilon)^2 dV \\ &\sim -(\mu^2 A_u + \upsilon^2 A_v) = \left(\frac{S_u^2}{A_u} + \frac{S_v^2}{A_v}\right). \end{aligned} \tag{12.14}$$

3. *Entropy model*: ([14])

$$E_{MI} = P_{R_u}\hat{h}(I|L = R^u) + P_{R_v}\hat{h}(I|L = R^v). \tag{12.15}$$

where P_{R^u} and P_{R^v} are the prior probability of pixel values in the inside and outside region, respectively.

12.4 The derivative of the energy functions

As the energy function is minimized by employing a gradient descent scheme, it is necessary to find the derivatives of the energy function:

1. *The binary model*:

$$\nabla \mathbf{w} E_{binary} = (\upsilon - \mu)\left(\frac{\nabla \mathbf{w} S_u - \mu \nabla \mathbf{w} V_u}{V_u} - \frac{\nabla \mathbf{w} S_v - \upsilon \nabla \mathbf{w} V_v}{V_v}\right) \tag{12.16}$$

$$\nabla \mathbf{p} E_{binary} = (\upsilon - \mu)\left(\frac{\nabla \mathbf{p} S_u - \mu \nabla \mathbf{p} A_u}{V_u} - \frac{\nabla \mathbf{p} S_v - \upsilon \nabla \mathbf{p} V_v}{V_v}\right). \tag{12.17}$$

2. *The Chan-Vese model*:

$$\nabla \mathbf{w} E_{cv} = -2(\mu \nabla \mathbf{w} S_u + \upsilon \nabla \mathbf{w} S_v) + (\mu^2 \nabla \mathbf{w} A_u + \upsilon^2 \nabla \mathbf{w} V_v). \tag{12.18}$$

$$\nabla \mathbf{p} E_{cv} = -2(\mu \nabla \mathbf{p} S_u + \upsilon \nabla \mathbf{p} S_v) + (\mu^2 \nabla \mathbf{p} A_u + \upsilon^2 \nabla \mathbf{p} V_v). \tag{12.19}$$

3. *The entropy model*:

$$\nabla \mathbf{w} E_{MI} = P_{R_u} \nabla \mathbf{w} \hat{h}(I|L = R^u) + P_{R_v} \nabla \mathbf{w} \hat{h}(I|L = R^v). \tag{12.20}$$

$$\nabla \mathbf{p} E_{MI} = P_{R_u} \nabla \mathbf{p} \hat{h}(I|L = R^u) + P_{R_v} \nabla \mathbf{p} \hat{h}(I|L = R^v). \tag{12.21}$$

12.5 The derivative of the region statistics

The antiderivative of the Heaviside function is Dirac Delta function

$$\delta(z) = \frac{d}{dz}\mathcal{H}(z). \tag{12.22}$$

With the introduction of the Dirac Delta function it is possible derive the derivatives of the region statistics[1]:

$$\begin{aligned}
\nabla w_i A_u = -\nabla w_i A_v &= \int \dots \int_{R^u+R^v} \frac{\partial}{\partial w_i}\mathcal{H}(-\mathbf{\Phi})dV \\
&= \int \dots \int_{R^u+R^v} \left(\frac{\partial}{\partial w_i} - \mathbf{\Phi}\right)\delta(-\mathbf{\Phi})dV \\
&= -\int \dots \int_{R^u+R^v} \mathbf{\Phi}_i\delta(-\mathbf{\Phi})dV \\
&= -\oint_{\overrightarrow{C}} \mathbf{\Phi}_i ds, && (12.23)\\
\nabla p_i A_u = -\nabla p_i A_v &= -\oint_{\overrightarrow{C}} \nabla p_i \mathbf{\Phi} ds, && (12.24)\\
\nabla w_i S_u = -\nabla w_i S_v &= -\oint_{\overrightarrow{C}} I\mathbf{\Phi}_i ds, && (12.25)\\
\nabla p_i S_u = -\nabla p_i S_v &= -\oint_{\overrightarrow{C}} I\nabla p_i \mathbf{\Phi} ds, && (12.26)\\
\nabla w_i Q_u = -\nabla w_i Q_v &= -\oint_{\overrightarrow{C}} I^2\mathbf{\Phi}_i ds, && (12.27)\\
\nabla p_i Q_u = -\nabla p_i Q_v &= -\oint_{\overrightarrow{C}} I^2\nabla p_i \mathbf{\Phi} ds, && (12.28)\\
\nabla w_i \hat{h}(I|L=R^u) &= \frac{1}{V_u}\oint_{\overrightarrow{C}} \mathbf{\Phi}_i \log(\hat{p}_{R^u}(I))ds, && (12.29)\\
\nabla w_i \hat{h}(I|L=R^v) &= \frac{-1}{V_v}\oint_{\overrightarrow{C}} \mathbf{\Phi}_i \log(\hat{p}_{R^v}(I))ds, && (12.30)\\
\nabla p_i \hat{h}(I|L=R^u) &= \frac{1}{V_u}\oint_{\overrightarrow{C}} \nabla p_i\mathbf{\Phi} \log(\hat{p}_{R^u}(I))ds, && (12.31)\\
\nabla p_i \hat{h}(I|L=R^v) &= \frac{-1}{V_v}\oint_{\overrightarrow{C}} \nabla p_i\mathbf{\Phi} \log(\hat{p}_{R^v}(I))ds, && (12.32)
\end{aligned}$$

[1] For reason of compactness $\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]$ will be denoted $\mathbf{\Phi}$ in the following formulas

where

$$\begin{aligned}\nabla p_i \mathbf{\Phi} &= \nabla p_i \mathbf{\Phi}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N) \\ &= \begin{bmatrix} \frac{\partial \mathbf{\Phi}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N)}{\partial \tilde{x}_1} & \ldots & \frac{\partial \mathbf{\Phi}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N)}{\partial \tilde{x}_N} & 0 \end{bmatrix} \frac{\partial T[\mathbf{p}]}{\partial p_i} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ 1 \end{bmatrix}. \quad (12.33)\end{aligned}$$

The partial derivatives of the 2D Euclidian transformation matrix $T[\mathbf{p}]$ are

$$\frac{\partial T[\mathbf{P}]}{\partial a} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (12.34)$$

$$\frac{\partial T[\mathbf{P}]}{\partial b} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad (12.35)$$

$$\frac{\partial T[\mathbf{P}]}{\partial h} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (12.36)$$

$$\frac{\partial T[\mathbf{P}]}{\partial \theta} = \begin{bmatrix} -h\sin(\theta) & h\cos(\theta) & 0 \\ -h\cos(\theta) & -h\sin(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (12.37)$$

(12.38)

12.6 Parameter optimization

Since the level set function $\mathbf{\Phi}[\mathbf{w}, \mathbf{x}]$ is rather expensive to evaluate and as the number of parameters can be large the most appropriate optimization scheme is gradient descent. Thus, we get the update equations

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \alpha_{\mathbf{w}} \nabla_{\mathbf{w}} E^t \text{ and} \quad (12.39)$$

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \alpha_{\mathbf{p}} \nabla_{\mathbf{p}} E^t, \quad (12.40)$$

where $\alpha_{\mathbf{w}}$ and $\alpha_{\mathbf{p}}$ are step size parameters, and $\mathbf{w}^t$ and $\mathbf{p}^t$ are the estimates of the pose parameters and shape parameters at the tth iteration.

To avoid zig-zagging, it is necessary to adjust the size of step length as the energy minimization progresses. One approach is to half the step length every time a change in direction is observed. A change in direction has occurred when $\nabla E^t \cdot \nabla E^{t-1} < 0$.

Chapter 13

Coupled shape model segmentation

The last chapter dealt with the task of segmenting a single object in an image - foreground and background segmentation. Naturally, this approach can be directly transferred to the segmentation of multiple objects as along as all shape models are initialized manually, and if the objects do not interfere with the segmentation of each other.
This chapter will discuss how the segmentation framework established in the last chapters can be generalized to handle multiple objects. Three very distinct ways of performing multiple object segmentation will be covered in this chapter:

- Multiple muscles in the same shape model.
- Coupling in pose.
- Simultaneous search with multiple shape model.

13.1 Multiple muscles in the same shape model

The incorporation of more than one object in the same shape model does not qualify as a coupled shape model - it is still only one model. Nevertheless,

it does provide an easy alternative to coupling shape models if the intensity distributions inside the objects are the same. There are several advantages with this approach compared to the coupling of the models:

- The relative simple LSFSM framework presented in the last chapter supplies this possibility without any changes.
- If the surface configurations of the objects are closely related, see Figure 13.1 for an example, this approach is properly the easiest way to capture the relationship.
- Other shape models than the LSFSM allow for incorporation of more than one object. However, overlapping and merging of objects are often a very challenging problem or impossible. The implicity of the LSFSM allows for topology change, which again allows for objects to merge or separate.

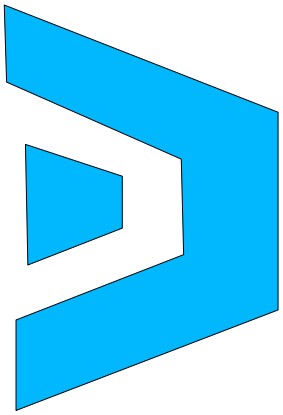


Figure 13.1: The surface configuration of the two objects are closely connected.

There are also disadvantages with having more than one object in the same model:

- By modelling more than one object in to the same model a more constraint and complex model is created. Hence, the task of training the model becomes harder, and the chance of ending up with an over-constraint model grows significant.
- In the LSFSM case, the incorporation of more objects in the same model will likely require a larger evaluation area of the level set functions $\mathbf{\Phi}[\mathbf{w}, \mathbf{p}]$ which is rather costly with regard to computation time. So, a rule of thumb

is not to combine two objects in the same model, if the distance between them are relative large compared to the size of the objects. For further explanation see Figure 13.2.

- In a multi object LSFSM it is not explicit given, which parts of the inside area of the shape belongs to which object.

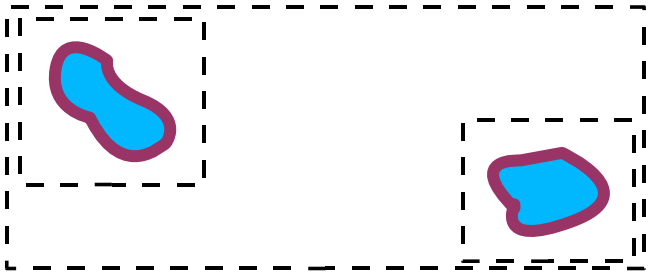


Figure 13.2: The two small boxes indicates the evaluation areas needed if the two objects belong to different models, and the large model indicates the evaluation area needed if they belong to the same model.

13.2 Coupling in pose

The result of a shape guided segmentation is highly depended on the initialization of the shape - the choice of the initial parameter set. For example, if the center of mass of the initial shape is placed too far from the true center of mass of the object, which is to be segmented, the result will in almost every case be of poor quality. Due to this sensitivity proper initialization is a must. It is intuitive as well as an empirical established fact in shape based segmentation that a good initialization of the pose parameters is more important than a good initialization of the shape parameters. In general, the shape parameters are chosen such that the initial shape is equal to the mean shape.
If no knowledge is present about the where-about of the object to be segmented, it is impossible to give an educated estimate of the initial placement of the shape. In such cases the only automated solution is to keep initializing the shape in different position until a satisfactory segmentation is achieve. Naturally, this solution is time consuming, why the ability to narrow the possible initialization area is essential.

In multi-object segmentation, it is in almost every case valid to assume that the objects carry information about placement, size, orientation and perhaps even

surface configuration of each other. The remainder of the section will concentrate on estimating the pose of the models of unsegmented objects based on the pose of the models of segmented objects.

The question is, given a successful segmentation of one object how do we predict the pose of the models of the remaining objects. The most obvious way to model the connection in pose between two objects is to assume that the difference in pose between them can modelled as a stochastic variable with a normal distribution.

Conjecture 13.2
The differences in the pose can be modelled as stochastic variables with a normal distribution.

Let us assume that we have m $\mathbf{\Phi}^i[\mathbf{w}, \mathbf{p}]$, and that we want to model the pose of the first $m-1$ models based on pose of the mth model $\mathbf{\Phi}^m$. Let's denote the mth model the *origo muscle*.
This gives us $m-1$ stochastic variables $\Delta\mathbf{P}_i \in N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. As the pose of the first $m-1$ models also might depend on each other they are grouped into one large stochastic variable $\Delta\mathbf{P} \in N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ given by

$$\Delta\mathbf{P} = \begin{bmatrix} \Delta\mathbf{P}_1 \\ \Delta\mathbf{P}_2 \\ \vdots \\ \Delta\mathbf{P}_{m-1} \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \\ \vdots \\ \boldsymbol{\mu}_{m-1} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{1,2} & \cdots & \boldsymbol{\Sigma}_{1,m} \\ \boldsymbol{\Sigma}_{1,2} & \boldsymbol{\Sigma}_2 & \cdots & \boldsymbol{\Sigma}_{2,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{1,m-1} & \boldsymbol{\Sigma}_{2,m-1} & \cdots & \boldsymbol{\Sigma}_{m-1} \end{bmatrix}.$$

If a successful segmentation is obtained with the $(m-1)$th model, the differences in pose for the remaining models can be re-estimated by partitioning the stochastic variable $\Delta\mathbf{P}$:

$$\begin{aligned} E(\Delta\mathbf{P}_{1:m-2} \quad | \quad & \Delta\mathbf{P}_{m-1} = \Delta\mathbf{p}_{m-1}) = \\ & \boldsymbol{\mu}_{1:m-2} + \boldsymbol{\Sigma}_{1:m-2,m-1}\boldsymbol{\Sigma}^{-1}_{m-1,m-1}(\Delta\mathbf{p}_{m-1} - \boldsymbol{\mu}_{m-1}), \qquad (13.1) \\ D(\Delta\mathbf{P}_{1:m-2} \quad | \quad & \Delta\mathbf{P}_{m-1} = \Delta\mathbf{p}_{m-1}) = \qquad (13.2) \\ & \boldsymbol{\Sigma}_{1:m-2,1:m-2} - \boldsymbol{\Sigma}_{1:m-2,m-1}\boldsymbol{\Sigma}^{-1}_{m-1,1:m-1}\boldsymbol{\Sigma}_{m-1,1:m-2}, \end{aligned}$$

where

$$\Delta\mathbf{P}_{1:m-2} = \begin{bmatrix} \Delta\mathbf{P}_1 \\ \Delta\mathbf{P}_2 \\ \vdots \\ \Delta\mathbf{P}_{m-2} \end{bmatrix}, \quad \boldsymbol{\mu}_{1:m-2} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_{m-2} \end{bmatrix}, \tag{13.3}$$

$$\boldsymbol{\Sigma}_{1:m-2,m-1} = (\boldsymbol{\Sigma}_{m-1,1:m-2})^\top = \begin{bmatrix} \boldsymbol{\Sigma}_{1,m-1} \\ \boldsymbol{\Sigma}_{2,m-1} \\ \vdots \\ \boldsymbol{\Sigma}_{m-2,m-1} \end{bmatrix} \tag{13.4}$$

$$\boldsymbol{\Sigma}_{1:m-2,1:m-2} = \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} & \cdots & \boldsymbol{\Sigma}_{1,m-2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} & \cdots & \boldsymbol{\Sigma}_{2,m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{m-2,1} & \boldsymbol{\Sigma}_{m-2,2} & \cdots & \boldsymbol{\Sigma}_{m-2,m-2} \end{bmatrix} \tag{13.5}$$

Thus

$$E(\mathbf{P}_{1:m-2}|\mathbf{P}_{m-1} = \mathbf{p}_{m-1}) = \mathbf{p}_o + E\left(\Delta\mathbf{P}_{1:m-2}|\Delta\mathbf{P}_{m-1} = (\mathbf{p}_{m-1} - \mathbf{p}_o)\right), \tag{13.6}$$

where $\mathbf{p}_o$ is the pose parameter of the origo model.

Estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be obtained from the training set. All outlined shapes goes through the transformation shown in Figure 13.3.

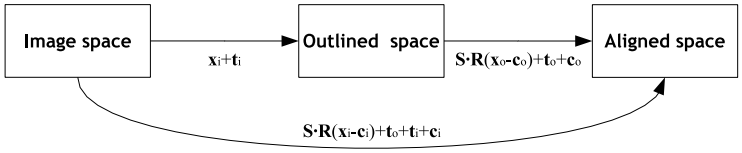


Figure 13.3: Transformations from image space to aligned space.

Let $\mathbf{p}_j^i$ denote the pose parameters which produce the transformation from the ith image to aligned space of object class j. The estimation procedure is sketched beneath:

1. For all $\mathbf{p}_j^i$ calculate the $\tilde{\mathbf{p}}_j^i$ such that $T(\mathbf{p}_j^i)^{-1} = T(\tilde{\mathbf{p}}_j^i)$.
2. Compute $\Delta\tilde{\mathbf{p}}_j^i$ subtracting $\tilde{\mathbf{p}}_o^i$ from $\tilde{\mathbf{p}}_j^i$, where o is the id of the origo muscle.
3. Estimate $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ by calculating sample mean and the dispersion matrix of $[\Delta\tilde{\mathbf{p}}_j^i]$.

4. For all combination of class id's j and k estimate $\mathbf{\Sigma}_{j,k}$ by calculating the sample covariance from all pairs $(\tilde{\mathbf{p}}_j^i, \tilde{\mathbf{p}}_k^i)$.

Figure 13.4 justifies the use of this method. A translation model was build from the aligned training set of the muscles 42, 50 and 90. Clearly, the estimated position of the muscle 90 is improved after the position of 42 is known.

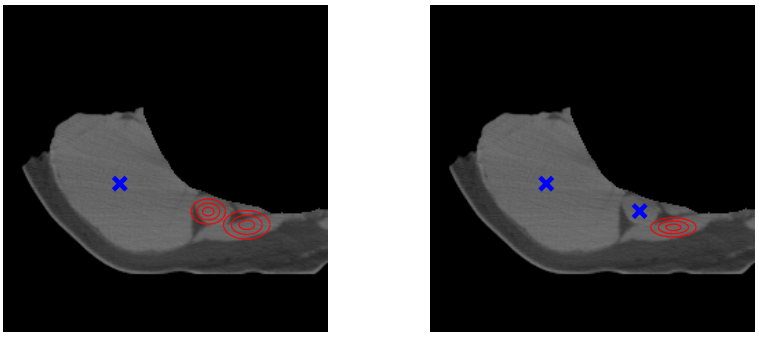


Figure 13.4: Pose estimation from previous segmented objects. The left image show the estimated positions of the muscles 42 and 90 given 50. The right image shows the estimated position of the muscle 90 given 42 and 50. The 3 red lines correspond to 1, 1.5, 2 standard deviations.

13.3 Simultaneous search with multiple shape model

The formulation of the LSFSM segmentation framework presented in the last chapter extends rather easily into a multi shape model segmentation framework.

More formally, we wish to segment m objects given m shape models

$$\begin{aligned}
\mathbf{\Phi}^1[\mathbf{w},\mathbf{p}](x_1,\ldots,x_N) &= \bar{\mathbf{\Phi}}^1(\tilde{x_1},\ldots,\tilde{x_N}) + \sum_{i=1}^{k_1} w_i \mathbf{\Phi}_i^1(\tilde{x_1},\ldots,\tilde{x_N}) && (13.7)\\
\mathbf{\Phi}^2[\mathbf{w},\mathbf{p}](x_1,\ldots,x_N) &= \bar{\mathbf{\Phi}}^2(\tilde{x_1},\ldots,\tilde{x_N}) + \sum_{i=1}^{k_2} w_i \mathbf{\Phi}_i^2(\tilde{x_1},\ldots,\tilde{x_N}) && (13.8)\\
&\vdots\\
\mathbf{\Phi}^m[\mathbf{w},\mathbf{p}](x_1,\ldots,x_N) &= \bar{\mathbf{\Phi}}^m(\tilde{x_1},\ldots,\tilde{x_N}) + \sum_{i=1}^{k_m} w_i \mathbf{\Phi}_i^m(\tilde{x_1},\ldots,\tilde{x_N}). && (13.9)
\end{aligned}$$

Even though, it is a straight forward transferral, the generalization to a multi shape model framework introduces a few new consideration. Primarily, the notion of an inside and outside region is to simple to handle the segmentation of multiple objects with different intensity distributions. For this reason, separate formulation are given for the two cases; same intensity distribution and different intensity distribution. Further, since this is a simultaneous labelling problem the risk or possibility of assigning multiple label to the same voxel arises.

13.3.1 Same intensity distribution

Since, it is assumed that the objects have the same intensity distribution, (12.4) and (12.5) can be generalized to

$$R^u = \{(x_1,x_2,\ldots,x_N) \in \mathbb{R}^N : \bigvee_{i=1}^{m} \mathbf{\Phi}^i[\mathbf{w},\mathbf{p}](x_1,x_2,\ldots,x_N) \leq 0\} \tag{13.10}$$

and

$$\begin{aligned}
R^v = \{(x_1,x_2,\ldots,x_N) \in \mathbb{R}^N : \quad & \left(\bigwedge_{i=1}^{m} \mathbf{\Phi}^i[\mathbf{w},\mathbf{p}](x_1,x_2,\ldots,x_N) > 0\right) \wedge && (13.11)\\
& \left(\bigvee_{i=1}^{m} \mathbf{\Phi}^i[\mathbf{w},\mathbf{p}](x_1,x_2,\ldots,x_N) < r\right)\}
\end{aligned}$$

This formulation does not guarantee unique labelling. Nevertheless, it does actually ease the problem a little bit. To support this claim, we take a closer at the region statistics and their derivatives. The hybervolumes for the inside and outside regions become

$$V_u = \int \ldots \int_{R^u+R^v} 1 - \prod_{i=1}^{m} \mathcal{H}(\mathbf{\Phi}^i) dV \tag{13.12}$$

and

$$V_v = \int \dots \int_{R^u + R^v} \prod_{i=1}^{m} \mathcal{H}(\mathbf{\Phi}^i) dV. \tag{13.13}$$

And the derivatives of hybervolumes are

$$\nabla w_l^k A_v = -\nabla w_l^k A_u = \oint_{\overrightarrow{c_k}} \left(\prod_{\substack{i=1 \\ i \neq k}}^{m} \mathcal{H}(\mathbf{\Phi}^i) \right) \mathbf{\Phi}_l^k ds. \tag{13.14}$$

From (13.14), we acknowledge that the derivatives are only calculated from the parts of the curve $\overrightarrow{c_k}$ which do not lay within another segmenting curve. As a consequence, a first come first serve principal arises which is not necessarily a good thing. Hence, if this formulation is chosen it might be necessary to add a term to the selected energy function, which discourages overlapping. E.g. the *Chan-Vese* energy could be expanded to

$$E_{cv2} = -(\mu^2 A_u + v^2 A_v) = \left(\frac{S_u^2}{A_u} + \frac{S_v^2}{A_v} \right) + \alpha A_{overlap},$$

where $A_{overlap}$ is the overlapping area and α is a constant. This might solve the problem, however it raises several new questions:

- How do we choose a suitable α?
- How sensitive is the final segmentation result to the selection of α?
- Should α be depended on the iteration number?

The easiest way to deal with overlap is to terminate its existence. This can be done by assuming that the intersection between two shapes belongs to the outside region. Hence, the inside region and outside region are reformulated to

$$R^u = \{(x_1, x_2, \dots, x_N) \in \mathbb{R}^N : \bigvee_{i=1}^{m} \Big(\mathbf{\Phi}^i[\mathbf{w}, \mathbf{p}](x_1, x_2, \dots, x_N) \leq 0 \bigwedge_{\substack{j=1 \\ j \neq i}}^{m} \mathbf{\Phi}^j[\mathbf{w}, \mathbf{p}](x_1, x_2, \dots, x_N) > 0 \Big) \} \tag{13.15}$$

and

$$R^v = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N : \quad \left(\left(\bigwedge_{i=1}^{m} \mathbf{\Phi}^i[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) > 0\right) \wedge \left(\bigvee_{i=1}^{m} \mathbf{\Phi}^i[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) < r\right)\right) \bigvee_{i=1}^{m} \Big(\mathbf{\Phi}^i[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) \leq 0 \wedge \bigvee_{\substack{j=1 \\ j \neq i}}^{m} \mathbf{\Phi}^j[\mathbf{w}, \mathbf{p}](x_1, x_2, \ldots, x_N) \leq 0\Big)\} \tag{13.16}$$

.

This formulation yields the hybervolume statistics

$$V_u = \int \ldots \int_{R^u + R^v} \sum_{i=1}^{m} \left(\left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \mathcal{H}(\mathbf{\Phi}^j)\right) \mathcal{H}(-\mathbf{\Phi}^i)\right) dV \tag{13.17}$$

and

$$V_v = \int \ldots \int_{R^u + R^v} 1 - \sum_{i=1}^{m} \left(\left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \mathcal{H}(\mathbf{\Phi}^j)\right) \mathcal{H}(-\mathbf{\Phi}^i)\right) dV \tag{13.18}$$

and the derivatives

$$\nabla w_l^k A_v = -\nabla w_l^k A_u = \quad \oint_{\overrightarrow{c_k}} \left(\prod_{i=1}^{m} \mathcal{H}(\mathbf{\Phi}^i)\right) \mathbf{\Phi}_l^k - \sum_{i=1}^{m} \left(\left(\prod_{\substack{j=1 \\ j \neq k, j \neq i}}^{m} \mathcal{H}(\mathbf{\Phi}^j)\right) \mathcal{H}(-\mathbf{\Phi}^i) \mathbf{\Phi}_l^k\right) ds. \tag{13.19}$$

(13.19) will be equal to (12.23) for a given parametrization of shape as long as the shape does not intersect with another shape. In cases where there is intersection, the part of the curve $\overrightarrow{c_k}$, which is located inside the other shape, will have the opposite effect on the gradient than it would in (12.23). The same applies for the other region statistics. Thus, intersecting shapes will be pushed away from each other with this formulation.

13.3.2 Different intensity distribution

This case is omitted from the thesis as all the objects, which have to be segmented, are muscles - thus they have the same intensity distribution. The interested reader is referred to [25].

Chapter 14

Corpus Callosum: Comparison with AAM

A well visited problem within image analysis is the segmentation of the Corpus Callosum in the human brain. A MR image of the human brain in shown in Figure 14.1 with annotation of the Corpus allosum.

A data set consisting of 62 MR images (similar to Figure 14.1) of the human brain with a 78 point annotation has been borrowed from the Danish Research Centre for Magnetic Resonance at Hvidovre hospital.
This data set has been used to compare performance of the region based segmentation framework with the Chan Vese energy and entropy energy model against the performance of another shape based segmentation method - the Active Appearance Model (AAM). The open source implementation AAM-API [21] by Mikkel B. Stegmann has been used to obtain the segmentation results for the AAM. The AAM-API comes with a console interface **aamc**, which in addition to providing easy access to the functionality of the AAM-API, provides tools such a leave-one-out testing.

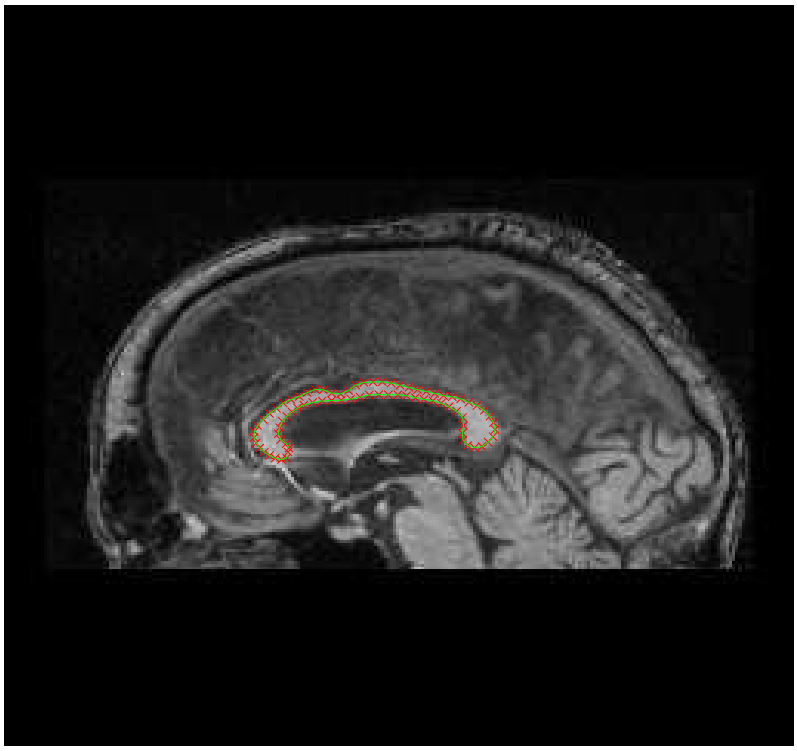


Figure 14.1: MR image of the human brain. The red crosses are annotation.

14.1 Experimental design

14.1.1 Methodology

To evaluate the performance of the segmentation methods the leave-one-out (LOO) cross validation methodology was chosen. That is, the data set is split up into 62 pairs of training and evaluation sets - every image with annotation gets to be the evaluation set exactly one time. The LOO methodology ensures that the models used capture almost the same variance as a model build on the whole data set.

As mentioned, the **aamc** provides a LOO model evaluation facility. For each image and corresponding annotation in the data set the following tasks are

performed:

- The annotation becomes the ground truth.
- A model is build from the data set with the current annotation and image excluded.
- The mean shape of the model is scaled and rotated such that it fits the ground truth as good as possible.
- The scaled and rotated mean shape is initialized at four different positions, and four segmentations are performed. The four initial positions are found by displacing the scaled and rotated mean shape plus and minus 10% of the height and width of the bounding box of the scaled and rotated mean shape from the center of mass of the ground truth - one at a time.
- The results are logged.

To minimize the workload this LLO procedure was adopted.

14.1.2 Measures for comparison

In order to perform an objective comparison of the performance of different segmentation methods, it is essential to assure that measures used for the comparison are objective - e.g. if Chan Vese energy function is used in a region based segmentation scheme, it is not fair to use the sample variance of the intensities in the segmented region as a measure of the quality of the segmentation. Another issue is, that many measures only can be applied if the segmentation is successful. Naturally, the success rate itself is an obvious performance measure. The question is, how do we determine whether a segmentation is successful or not? In practise, we will most likely look at the fit measure or energy function used to guide the segmentation and check, whether it is below or above some threshold. However, it is practically impossible to secure that some chosen threshold in one method is equivalent to the chosen threshold of another method. Therefore, a 75% or above overlap between the ground truth and the segmentation will be the criteria for success for this comparison test.
The author believe that the following measures are objective in the sense that they do not favor any of the two segmentation approaches[1].

[1] Measures followed by a * are only calculated on successful segmentations.

- **Success rate (SR)**: The percentage of success

$$\frac{Success.\ segmentations}{Total\ segmentations}. \tag{14.1}$$

- **Overlap (OL)***: The relative amount of overlap between the segmentation and the ground truth:

$$\frac{\int\int_{\Omega} I_S \wedge I_{GT} dA}{\int\int_{\Omega} I_{GT} dA}, \tag{14.2}$$

  where Ω is the image domain, I_{GT} is the ground truth image and I_S is the segmentation image.

- **Over segmentation (OS)***: The area, which is part of segmented areas and not a part of the ground truth, relative to the ground truth:

$$\frac{\int\int_{\Omega} I_S \backslash I_{GT} dA}{\int\int_{\Omega} I_{GT} dA}. \tag{14.3}$$

- **The mean-squared-error(MSE)***: The mean-squared-error of the distances from the segmenting curve to the ground truth boundary.

- **Randomness of the "run" of the boundaries (RRB)***: The variance of the signed distances from the segmenting curve to the ground truth boundary. A segmentation, where the boundary is slightly inside the boundary of the ground truth, is better than a segmentation, where the boundary crosses the boundary of the ground truth rapidly.

14.1.3 Model deformity

To avoid implausible deformation of a model, hard limits are imposed on the shape parameters. Recall that the variance σ_i^2 of the ith principal component is equal to λ_i. Under the assumption that the model parameters are independent and gaussian distributed, the range $\pm 2\sqrt{\lambda_i}$ contains approximately 95.4% of the distribution of w_i. All parameters in the model will use this range as the hard limit. Obviously, limiting the parameter space to a hyper cube does not guarantee that implausible shapes cannot be generated, e.g. if all the parameters simultaneously takes either of the values $\pm 2\sqrt{\lambda_i}$ a very unlikely shape will be generated.
A more correct approach is to limit the parameter space to an hyper ellipsoid:

$$\sum_{i=1}^{n} \frac{w_i^2}{a_i^2} = 1 \tag{14.4}$$

By replacing a_i with $D_m\sqrt{\lambda_i}$ we get:

$$\sum_{i=1}^{n} \frac{w_i^2}{(D_m\sqrt{\lambda_i})^2} = 1 \Leftrightarrow \sum_{i=1}^{n} \frac{w_i^2}{\lambda_i} = D_m^2 \tag{14.5}$$

Note, D_m is the Mahalanobis distance. The parameters are confined to an ellipsoid by enforcing the constraint $D_m < D_{max}$.
Hard limits has been preferred as they are much easier to enforce than the soft limits given by the ellipsoid.

14.2 Preprocessing the images

The images in the Corpus Callosum have not been calibrated to ensure that a tissue type yield the same intensity distribution in all of the images. Consequently, some of the images appear dark while other appear light. For the entropy based energy function to work the images must be normalized such that a tissue type has approximately the same intensity in all the images.
A small correction scheme was applied to the images which attempts to correct the problem by adding a constant to all of the images such that the mean intensity is the same in all of Corpus Callosums. Surely, this is a rather primitive way of correcting the problem.

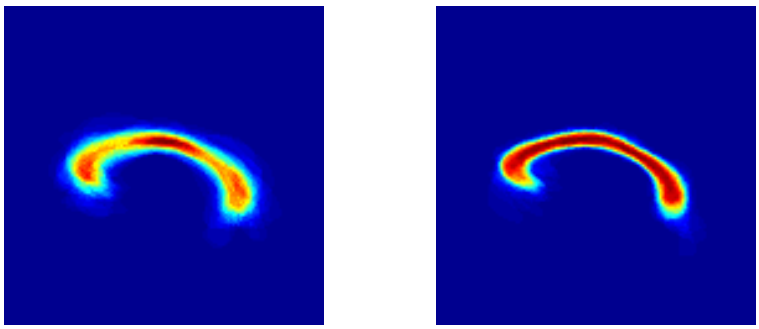
14.3 The shape models and selection of segmentation parameters

14.3.1 LSFSM

The annotations were first converted to binary shapes, and then aligned using the methods described in Section 11.4. The effect of the alignment is shown in Figure 14.2.

All LSFSM models used 10 parameters to explain slightly more than 95% of the variation in the training set.

For the Chan Vese energy function four different regions of interest were used. The regions of interest were defined by the 5 (CV5), 15 (CV15), 25 (CV25) and



(a) Not aligned (b) Aligned

Figure 14.2: The set of the binary shapes of the Corpus Callosum before and after the alignment.

35 (CV35) level set (measured in pixels).

For the entropy energy function the region of the interest was set to the 10 level set. The pdf for the inside and the outside was estimated with a parzen window - consult Section 3.5. The estimated pdf's are shown in Figure 14.3.

14.3.2 AAM

To enhance the robustness of the AAM the models were expanded with five pixels along the point normal of the annotation. That is, an additional 78 points were automatically added to existing annotation by placing an extra point five pixels in the normal direction for every manual annotated point. All AMM models used 35 parameters to explain slightly more than 95% of the variation in the training set.

14.4 Result and Discussion

The result of the test is shown in Table 14.1.

Primarily, we acknowledge that AAM has a significantly higher success rate than the most of LSFSM segmentations. The primary reason for the relative high failure rate for the segmentations with the CV energy function is that

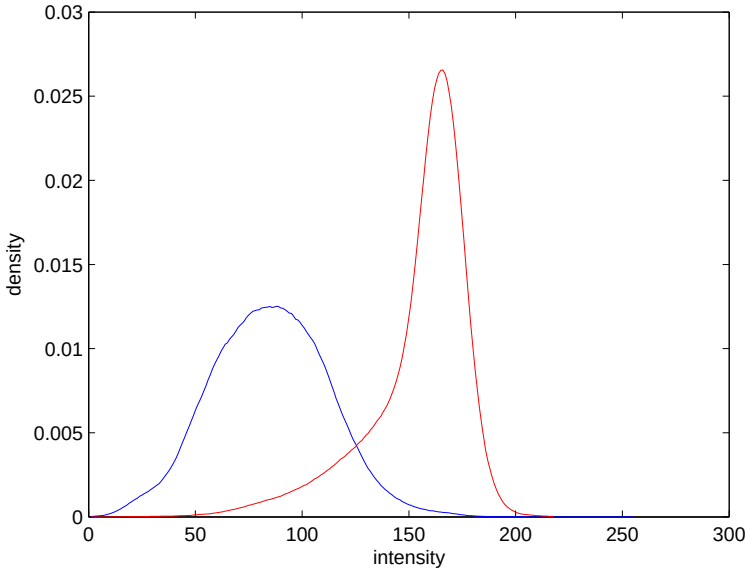


Figure 14.3: The estimated pdf's of the intensity of the inside and outside region. The blue curve correspond to the pdf of the outside region while the red curve correspond to pdf of the inside region.

the CV does not incorporate any information of about the distribution of the intensities in the inside and outside region. If the average intensity of the outside region initially is closer to the average intensity of the Corpus Callosum than the average intensity in the inside region, the energy function will guide the evolution of the curve such that inside region is moved away from the Corpus Callosum. Four examples of this problem are shown in Figure 14.4. As the region of interest grows this problem disappears. CV25 and CV35 do not suffer from the problem.

For the Chan Vese segmentations, the performance measures OS, RRB and MSE change for the worse, the measure OL becomes better and the measure SR remains the same, when the region of interest is expanded from 25 to 35. This implies that expanding the region of interest much further than the 25 level set does not contribute positively to performance. This is good argument for using a region of interest in the first place.

It was the hope of this author that the entropy energy function would produce better results than the Chan Vese energy function as it incorporates information

Measure		CV 5	CV 15	CV 25	CV 35	Entropy	AAM
SR		78.23%	90.73%	93.55%	93.55%	88.71 %	96.37%
OL	Avg.	0.8815	0.8864	0.8899	0.9081	0.8696	0.9148
	Var.	0.0009	0.0009	0.0012	0.0011	0.0030	0.0011
OS	Avg.	0.0940	0.1032	0.1144	0.1310	0.1075	0.1078
	Var.	0.0034	0.0042	0.0060	0.0062	0.0119	0.0047
RRB		0.5748	0.5938	0.6097	0.6345	0.5975	0.5603
MSE		0.5925	0.6168	0.6390	0.6924	0.6150	0.5944

Table 14.1: Results from leave-one-out experiments on Corpus Callosum data set.

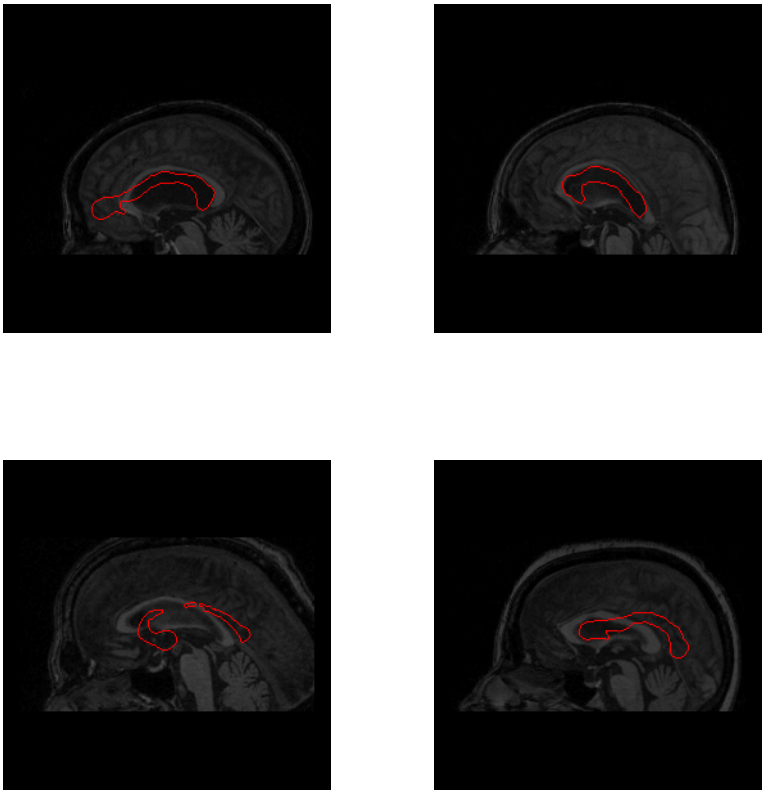


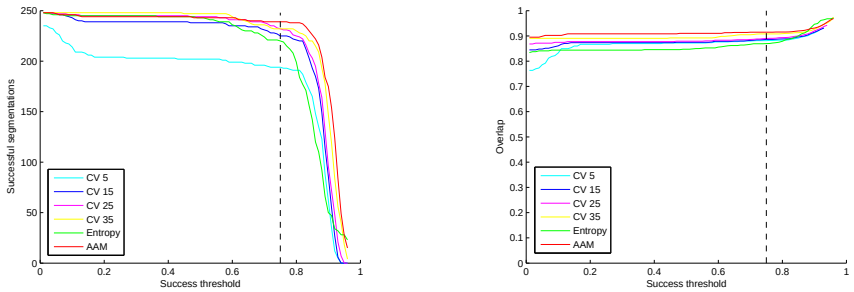
Figure 14.4: Failed Chan Vese segmentation from CV 15.

about the intensity distributions of the inside and the outside region. Nevertheless, it seems that the intensity correction scheme was too simple to guarantee

good results with entropy energy function.

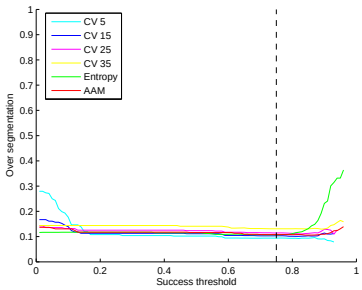
The measures SR, OL, RRB and MSE are in favor of the AAM, while the OS measure in general is in favor of the region based segmentation schemes. Thus, it must be concluded that the AAM performs better on the set of Corpus Collasum images than the region based segmentation methods. However, as the LSFSM does not require point correspondence the CV25 does appear to be good alternative if the training set is relative large.

As mentioned earlier the performance measures are depended on the choice of the success threshold. As a result, the conclusions made from the results shown in Table 14.1 might change if the success threshold is changed. Figure 14.5 shows how the comparison measures change when the success threshold is changed. The plots present the same picture as Table 14.1.

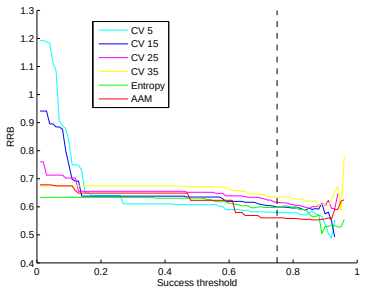


(a) Successful segmentation

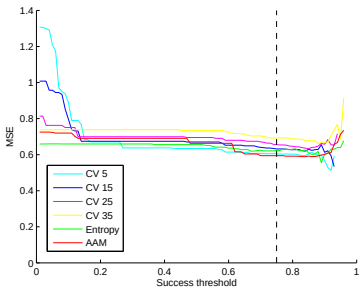
(b) Overlap



(c) Over segmentation



(d) RRB



(e) MSE

Figure 14.5: The comparison measures dependency on the success threshold.

Chapter 15

Sequential shape model segmentation of pig backs

Using shape models to guide the segmentation of a pig back is a rather challenging problem as the muscles in a pig back lay in close proximity to each other. As region based segmentation does not employ gradient operators to evolve the segmenting curve, there is a risk that the segmenting curve will cross the thin fat layers, which divide muscles. As a consequence an automatic sequential LSFSM guided segmentation on a full pig back will undoubtedly fail, and a leave-one-out test is therefore out of the question. However, to demonstrate how the method would work in practise, a sequential model segmentation will be done on the first slice of pig 7. The muscles in this slice are fairly well separated so the risk of ending up with overlapping segmentation areas is relative small. To further minimize the risk of overlapping, the segmented area of a muscle is erased as soon as the segmentation of the muscle is obtained. That is, the intensities of the segmented area is set to the mean intensity of fat.

The segmentation scheme is sketched beneath:

1. The initial pose parameters for the origo model is supplied and a segmentation of the origo muscle is obtained.
2. The initial pose parameters of the next model is estimated from the final

Model	No. of par.	Truncation level
42	12	95%
48	10	98%
50	13	95%
90	9	95%
106	4	95%

Table 15.1: Models generated upon the shapes from pig, 4, 9, 12, 14, 18, 20 and 22. The transformation was constrained to be rigid under the alignment process.

pose parameters of the models, from where segmentation results already have been obtained, and the difference in pose model (13.6). The rotation and scaling parameters are set to their expected value given the final pose parameters of the models, from where segmentation results already have been obtained, while the translation parameters are found with subsampling. The sampling approach is illustrated in Figure 15.1. Basically, the samples are obtained by placing a grid with some specified grid spacing, on the area of the ellipse defined by the dispersion matrix of the translation parameters and some chosen range[1]. The translation parameter sample, which yields the best fit according to the chosen energy function, is chosen. The next model to use is the model, which has the translation parameter ellipse with the smallest area.

3. A segmentation result is obtained, and the segmented area is removed from the image.

4. As long as there are unused models left go to 2.

15.1 Experimental design

The experiment will be done on the first slice of pig 7 using models, which are build upon the outlined muscles of pig 4, 9, 12, 14, 18, 20 and 22. The models in question and corresponding information are listed in Table 15.1.

The Chan Vese energy function is chosen for this experiment, and the shape parameters are limited with $\pm 2\sqrt{\lambda_i}$.

[1]This demonstration will use two standard deviations as the range.

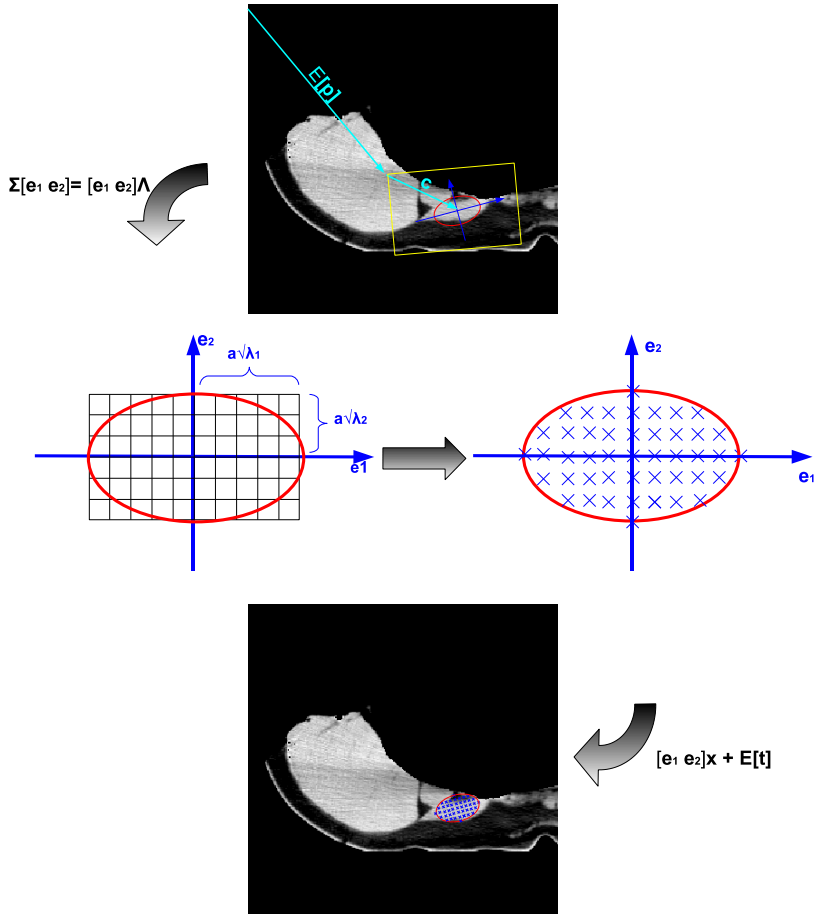


Figure 15.1: Finding the initial translation parameters of a model. The yellow frame marks the placement of the model space in the image space. $E[\mathbf{p}]$ is the expected value of the pose parameters given final pose parameters of models, from where segmentation results already have been obtained, and c is center of mass of the mean shape in the model space. The red ellipse in the image space corresponds to the ellipse of the dispersion matrix given some range $\pm a$. The center of the ellipse has been placed in the center of mass of the mean shape under the transformation specified by expected values of the pose parameters. The ellipse has been placed in the center of mass for purely illustrative purposes.

15.2 Preprocessing

To secure a more homogeneous outside region, the background intensity was set equal to the mean intensity of fat.

15.3 Results and Discussion

Figures 15.2-15.4 show the iterative process of segmenting the first slice in pig 7. Visually, the segmentation of the first slices looks plausible.

To illustrate that sequential segmentation most likely will fail on a whole pig back, the parametrization of the segmentation result of the first slice was passed on to the second slice and a new segmentation result was obtained on the second slice, and so on. After five slices the segmentation fails. Figure 15.5 shows the segmentation images from slice 3 to 6.

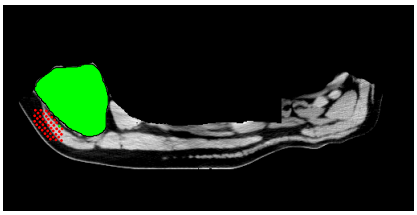
(a) 50: Initial position

(b) 50: After 15 pose iterations.

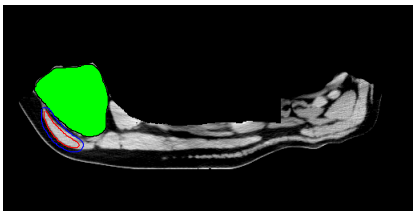
(c) 50: After 15 mixed iterations.

(d) 50: After 30 mixed iterations.

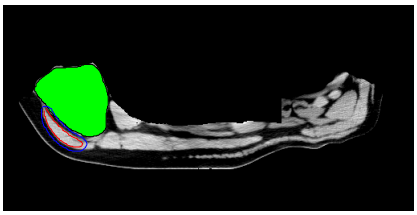
(e) 50: After 45 mixed iterations.



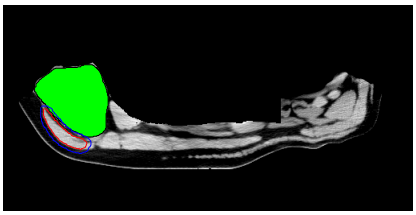
(f) Locating initial position of 106.



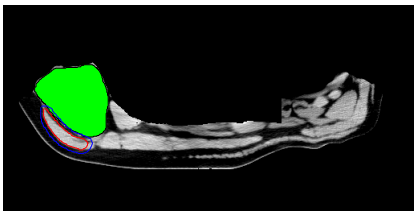
(g) 106: Initial position



(h) 106: After 15 pose iterations.

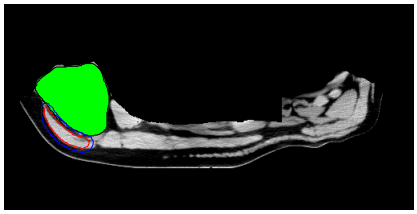


(i) 106: After 15 mixed iterations.

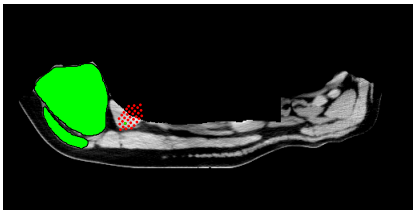


(j) 106: After 30 mixed iterations.

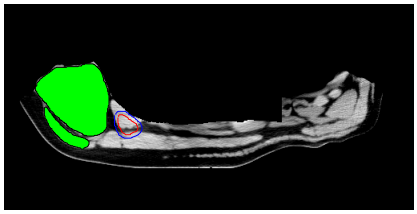
Figure 15.2: Segmentation of the first slice in pig 7. Continues in 15.3.



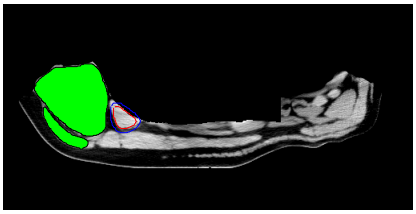
(a) 106: After 45 mixed iterations.



(b) Locating initial position of 42.



(c) 42: Initial position.

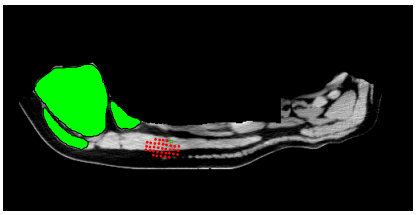


(d) 42: After 15 pose iterations.

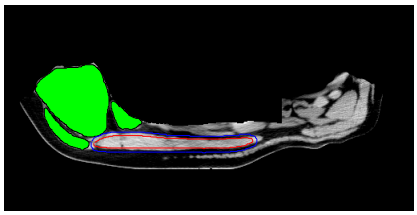
(e) 42: After 15 mixed iterations.

(f) 42: After 30 mixed iterations.

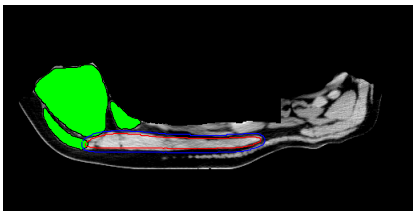
(g) 42: After 45 mixed iterations.



(h) Locating position of 48.

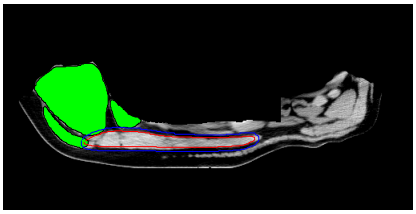


(i) 48: Initial position.

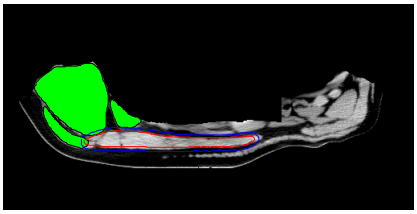


(j) 48:After 15 pose iterations.

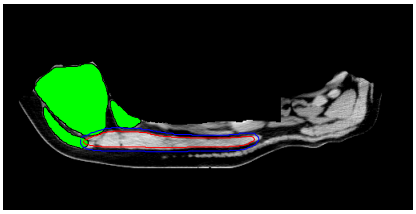
Figure 15.3: Segmentation of the first slice in pig 7. Continues in 15.4



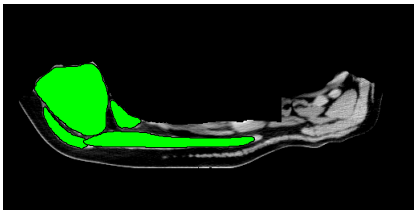
(a) 48: After 15 pose iterations.



(b) 48: After 15 mixed iterations.



(c) 48: After 30 mixed iterations.

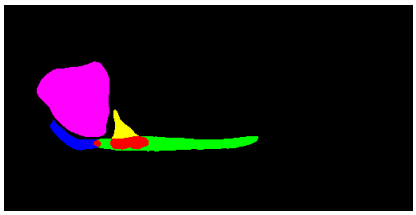


(d) Final result for first slice.

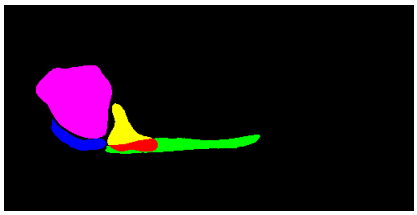
Figure 15.4: Segmentation of the first slice in pig 7.

(a) Slice 3

(b) Slice 4



(c) Slice 5



(d) Slice 6

Figure 15.5: The segmented area of slice 3, 4, 5, 6. Color code: yellow = 42, green = 48, magenta = 50, blue = 106, red = overlap.

Chapter 16

Simultaneous shape model segmentation of pig backs

In the last chapter, it was empirically proven that sequential model segmentation is not possible on a whole pig back. The region based framework given by (13.15) and (13.16) discourages overlap by assuming overlapping hypervolumes belongs to the outside region. Consequently, two shapes are pushed away from each other if they intersect.
This chapter will test this region based framework on the pigs 4, 7, 9, 12, 14, 18, 20 and 22 in a series of leave-one-out experiments.

16.1 Experimental design

16.1.1 Methodology

The eight CT scans with ground truth outlines will be used to evaluate the performance of the coupled model segmentation. As the size of the data set is rather limited the evaluation is done as series of leave-one-out experiments. Information about the models can be found in Table 16.1.

Model	No. of par.	Truncation level
42+90	12-13	95%
48	9-10	98%
50	12-13	95%
106	4-5	95%

Table 16.1: Models generated upon the shapes from pig, 4, 7, 9, 12, 14, 18, 20 and 22. The transformation was constrained to be rigid under the alignment process.

16.1.2 Energy function and model deformity

The Chan Vese energy function is chosen for this experiment, and the shape parameters are limited with $\pm 2\sqrt{\lambda_i}$.

16.1.3 Comparison to ground truth

The following measures are used to access the quality of the segmentation:

- **The overlap** (OL) The overlapping volume between the ground truth image and the segmentation image relative to the volume of the ground truth.

- **Over segmentation** (OS) The volume, which is a part of segmented shape, but not a part of the ground truth relative to the volume of the ground truth.

- **Under segmentation** (US) The volume, which is a part of ground truth, but not a part of the segmented shape relative to the volume of the ground truth:

$$US = 1 - OL. \tag{16.1}$$

16.2 Strategy

16.2.1 Initial input

In the first slice, the mean shapes of the models are placed such that their centers of mas are identical to the ones of the ground truth outlines. Further, the mean shapes are scaled and rotated such that the overlap with the ground truth is maximized.

16.2.2 Porting parameters between slices

Only minor changes occur in the position and the shape of a muscle between two consecutive slices. Hence, the parametrization of a successful segmentation in one slice can be used as a initial guess in the next slice.

16.2.3 Successful segmentation

A segmentation of a muscle is considered successful, if the mean intensity inside the segmenting curve of the muscle deviates with less than 50 (Hounsfield scale) from the entire inside region[1]. As soon as a segmentation has been deemed unsuccessful it will be excluded and can therefore not be used to segment the remaining slices.

16.3 Adaptions to ensure stability

There are a couple of stability issues which make it impossible to complete a successful segmentation of a whole pig back. The reasons for the instabilities are:

- The shape model of muscle 48 fails to follow the significant changes in the size and shape of the muscle 48 between the slices. This causes not only the segmentation of muscle 48 to fail but is also ruins the segmentation of the other muscles. It is the authors opinion that the failure in due to a mixture of the following two things:

[1] The value 50 has been chosen by trial and error.

  - As the muscle 48 changes much in length through the slices and as the shapes have been aligned with respect to the rigid transformation, the change in length is the most significant source of variance in the aligned data set. The first three variational modes of a model build from all of the outlined shapes of muscle 48 are displayed in Figure F.2 - the first variational mode explains more than 70% of the total variation. Therefore, it is often impossible to get a good fit at the end of the shape close to the muscles 50 and 106.
    So, why not use independent scaling in the alignment? The binary shape alignment method is an iterative scheme, and as consequence you cannot have independent scaling without shear. Note, shear can be expressed by $\mathbf{RSR}^{-1}$ where $\mathbf{S}$ is a diagonal matrix and $\mathbf{R}$ is a rotation matrix. Recall from Definition 11.7, that a shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. Hence, according to this definition actual shape variation will be filtered out, if shear is used in the alignment.
  - The gradient descent scheme's unfortunate ability to fall into locale minima[2]. The chance of falling into a local minimum is considerable higher when segmenting long thin structures. As soon as the wide sides of a long thin shape obtain a good fit it becomes a great deal harder to vary the shape without increasing the energy (even if the ends of the shape fit poorly).

  To eliminate the problem the model of muscle 48 is manually removed just before it fails.

- The muscle 90 might change rapidly in size between two consecutive slices (consult Figure 2.7) which causes the segmentations of muscle 42 and muscle 90 to fail. In order to handle this problem, the muscles 42 and 90 are combined in the same shape model.

16.4 Results and discussion

The result of the experiment can be found in Table 16.2, and a 3D visualization of the segmentation of pig 7 can be found in Figure 16.1[3]. At first glance, the result looks rather discouraging as only the segmentations of muscle 50 yield a high degree of overlap. Fortunately, the overall low overlap is mostly due to a

[2]More correctly, the optimization of the parameters of the model in question stops in a suboptimal solution.

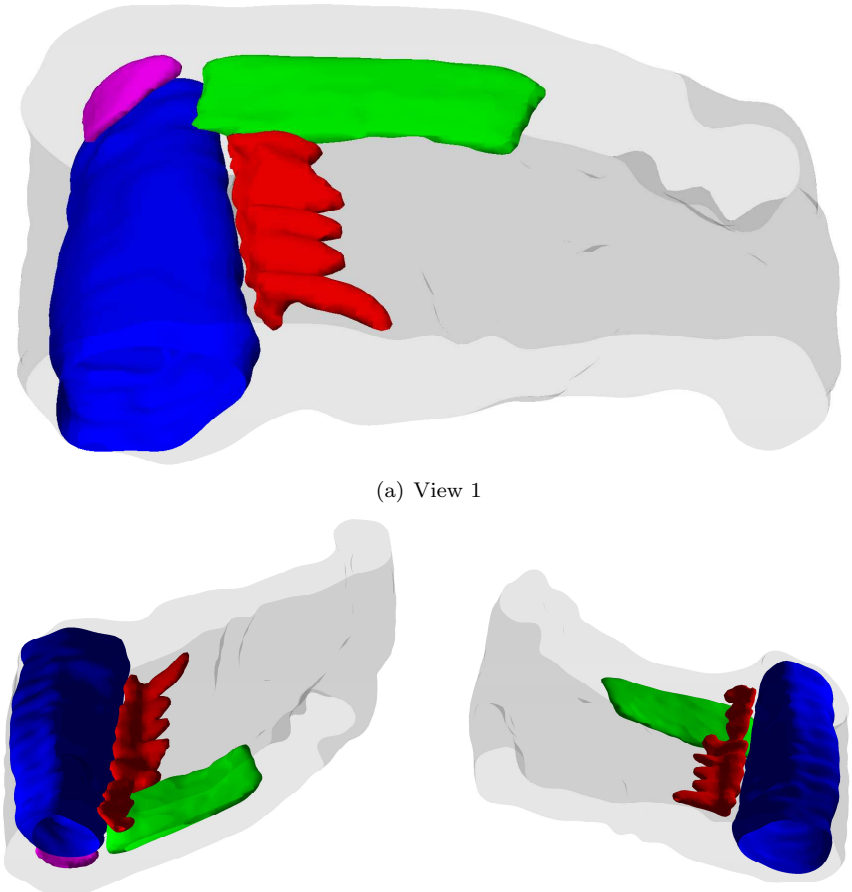
[3]The visualization is constructed from a RBF implicit surfaces interpolation.

Pig ID / muscle ID		42+90	48	50	106
4	Overlap	0.73	0.80	0.94	0.69
	Over segmentation	0.08	0.12	0.01	0.08
	Under segmentation	0.27	0.20	0.06	0.31
7	Overlap	0.63	0.85	0.94	0.78
	Over segmentation	0.15	0.03	0.02	0.04
	Under segmentation	0.37	0.15	0.06	0.22
9	Overlap	0.55	0.64	0.93	0.53
	Over segmentation	0.09	0.03	0.02	0.02
	Under segmentation	0.45	0.36	0.07	0.47
12	Overlap	0.59	0.77	0.95	0.73
	Over segmentation	0.07	0.03	0.02	0.06
	Under segmentation	0.41	0.23	0.05	0.27
14	Overlap	0.59	0.60	0.95	0.73
	Over segmentation	1.00	0.00	0.02	0.08
	Under segmentation	0.41	0.40	0.19	0.28
18	Overlap	0.26	0.74	0.93	0.61
	Over segmentation	0.09	0.02	0.02	0.04
	Under segmentation	0.74	0.26	0.07	0.40
20	Overlap	0.65	0.75	0.95	0.75
	Over segmentation	0.10	0.01	0.03	0.04
	Under segmentation	0.35	0.26	0.05	0.25
22	Overlap	0.64	0.73	0.96	0.72
	Over segmentation	0.10	0.02	0.01	0.03
	Under segmentation	0.36	0.27	0.04	0.28

Table 16.2: Leave-one-out results for simultaneous shape segmentations.

lack of agreement between the Chan Vese energy function and the author on where to place the transition boundary between the meat and fat pixels, and is not due to segmentation failure. The claim is supported by the low degree of over segmentation compared to the degree of under segmentation. The reason, that the segmentation of the loin muscle (50) is less affected by the lack of agreement, is the fact that it has a much higher surface to volume ratio than the other muscles.

Nevertheless, segmentation failures do occur - especially in segmentations with the combined model.
A horrifying example is shown in Figure 16.2, where the combined model of muscle 42 and 90 not only fails but it also ruins the segmentation of muscle 50. This example illustrates the main drawback with the coupled model framework - it is in principle impossible to know which of two intersecting shapes is going



(a) View 1

(b) View 2

(c) View 3

Figure 16.1: 3D visualization of the segmentation of pig 7. Muscle color code: 106=magenta, 50=blue, 48=green, 42+90=red.

to be pushed away. Naturally, in most cases the model with the best fit is going to stay while the other model is going to be pushed away. However, if the model with the best fit has a variational mode which primarily changes the part of the boundary, which intersects, the model with the best fit might be pushed away.

The most common segmentation error is caused by the early removal of the model of muscle 48. In the majority of the eight experiments the combined model of muscle 42 and 90 enters the area of muscle 48 (after muscle 48 has been remove), which explains the higher over segmentation rate of the combined model. This phenomena is visible in Figure 16.1.

At last, it should be noticed from Figure 16.1 that the segmentation method is able to maintain a fine gap between muscle 50 and 42. This insinuates that the discouraging of overlap does work.

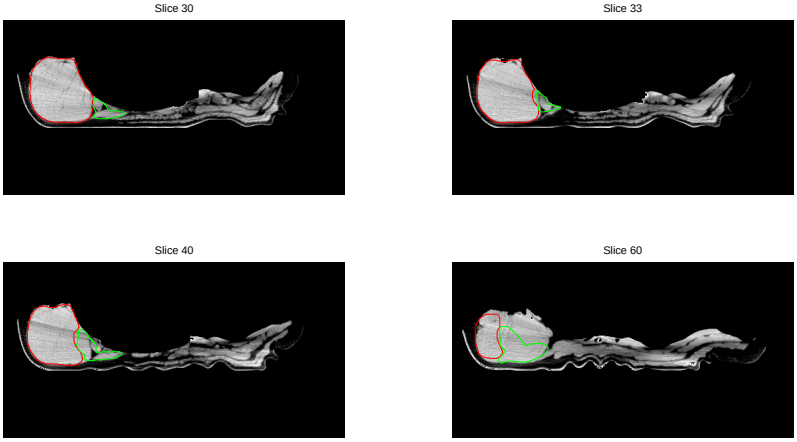


Figure 16.2: Segmentation failure in pig 14

This chapter has clearly demonstrated that the coupled region based segmentation framework is unable to segment a whole pig back successfully in a series of 2D segmentation without some manual intervention. Nevertheless, it is too soon to conclude that the method is unsuitable for the segmentation of pig backs. Surely, many of the issues described in this chapter will most likely disappear in three dimension:

- The problem of muscle 90 disappearing and reappearing between slices and the alignment problem with the shapes of muscle 48 will obviously

vanish as the slices no longer are segmented separately.

- Segmentation failures such as the one observed in pig 14, where the model of muscle 42 pushes the model of muscle 50 away and enters the area of muscle 50, are more unlikely to occur in 3D as the models will be much more constraint.

The method extends to any dimension without an significant growth in computation time[4]. That is, if the number of principal components needed to represent a shape does not grow significant in 3D the segmentation in 3D should take approximately the same time as in a series of 2D segmentations.

[4]The implementation in this thesis project does as well.

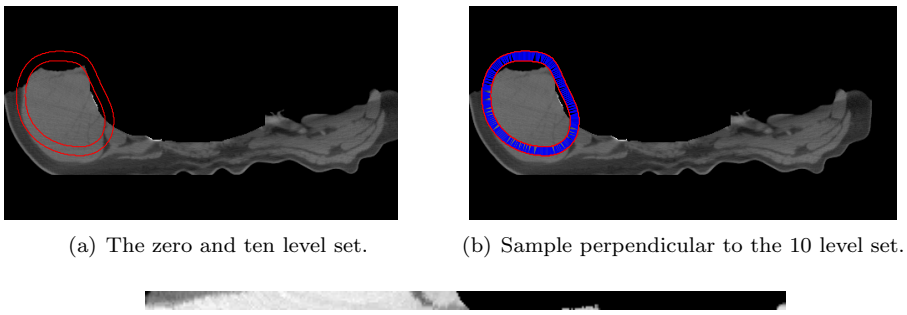
Chapter 17

An intelligent region of interest

The notion of a region of interest was adapted into the region based segmentation framework in this thesis. The incentive for limiting the region of interest from the entire image domain to the r level set was to secure a homogeneous outside region. However, as the pig muscles are in a very close proximity to each other the r level set region of interest approach does not guarantee a homogeneous outside region unless r is selected damaging low. One possible solution is to introduce an intelligent region of interest instead of the r level set.

It is assumed that the region of interest lays between the zero level set and the r level set. The region between the zero and r level set can be subsampled into a rectangular image, where the first row corresponds to the r level set, and the bottom row corresponds to the zero level set. The process is illustrated in Figure 17.1

The basic idea is to determine a *better* region of interest (than that determined by the r level set) based on the region statistics given by the zero level set and r level set and the sample image. That is, we want to find the optimal position of the region of interest in every column in the sample image.
A *better* region of interest should



(a) The zero and ten level set.

(b) Sample perpendicular to the 10 level set.

(c) The result.

Figure 17.1: Subsampling the area between the zero and ten level set. The top row in the (c) corresponds to the ten level set.

- provide a more homogeneous outside region,
- make the inside and outside regions more distinctive.

Distinctiveness is obviously depended on the choice of energy function. For simplicity it will be assumed that either the binary mean function or the Chan Vese energy function are used in a 2D scenario. Furthermore, it will be assumed that the mean intensity in the inside region is higher than the mean intensity in the area between the zero level set and the r level set.

It is possible to determine an optimal intensity threshold between the inside and outside region from the sample means and variances of the two regions[1]. Given such a threshold a simple algorithm for deriving an improved region of interest can be derived.
Let $I_s \in \mathbb{R}^{M_1 \times M_2}$, τ and $\mathcal{T} = \{t_1, \cdots, t_{M_2}\}$ be the sample image, the intensity threshold and the estimated positions of the boundary of the region of interest in the columns of the sample image, respectively. The algorithm is sketched beneath:

1. Initialize all the values of set $\mathcal{T}$ to the number of the bottom row:
   $\mathcal{T} = \underbrace{\{M_2, M_2, \ldots, M_2\}}_{M_1}$
2. For all columns c_i

[1] Consult with Section 3.2.

(a) While $I_s(t_i, c_i) > \tau$ and $t_i > 1$ update t_i with the following update formula $t_i = t_i - 1$.

(b) While $I_s(t_i, c_i) < \tau$ and $t_i > 1$ update t_i with the following update formula $t_i = t_i - 1$.

3. Smooth the values t_i with a spline.

The approach is exemplified in the Figure 17.2.

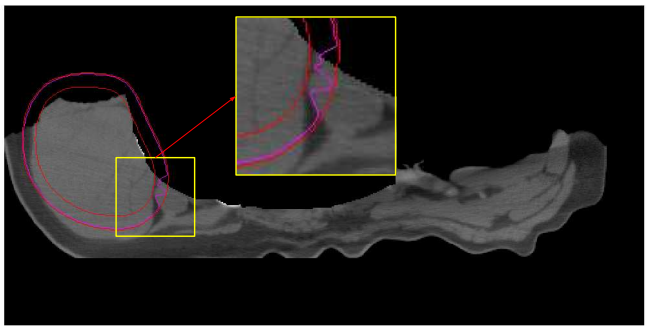


Figure 17.2: Searching for the region of interest without utilization of neighborhood information.

As this method does not incorporate neighborhood information between the columns it has a couple of drawbacks:

- It is not robust with respect to noise.
- Fluctuations in the boundary is likely to occur.

The enhance part of Figure 17.2 illustrate the last drawback with this simple approach.

The problem above can be formulated as a shortest / cheapest path problem and then solved with dynamic programming. That is, a rectangular cost matrix is constructed from the sample image, and the shortest path trough the rectangular cost matrix is assumed equivalent to the boundary of the region of

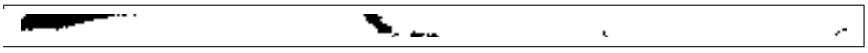
interest. Solutions for finding the shortest path in a cost matric can be found in [22] and [6] and is not covered in the thesis.

The pixel values in column c_i, which belong to the region of interest, are the pixel values $I_s(t, c_i)$ which satisfy $t \geq t_i$. Hence, the value in the cost matrix given by the index (t', c_i) should be calculated from all pixel values $I_s(t, c_i)$ which satisfy $t \geq t'$. As the scope of the energy functions have been narrowed to the binary mean function and Chan Vese energy function, a natural choice of cost matrix is:

$$\mathbf{C}_{mean}(t, c) = \frac{1}{M_1 - i + 1} \sum_{i=t}^{M_1} I_s(i, c) \tag{17.1}$$

The $\mathbf{C}_{mean}$ of the sample image shown in Figure 17.1(c) is shown in Figure 17.3(a). The cost matrix $\mathbf{C}_{mean}$ does not provide a region of interest, which is good enough - the resulting region of interest is shown in Figure 17.4.

(a) The $\mathbf{C}_m$ matrix of the sample image in Figure 17.1(c).



(b) The estimated meat-fat segmentation of the area between the zero and ten level set.

(c) The cost image.

Figure 17.3:

From Figure 17.4, the following problems is discovered:

- The boundary of the region of interest is out of control in columns with pure meat pixel. The uncontrollable behavior will also exist in columns where the bottom pixels are fat pixels.
- As the background value is somewhat lower than the mean intensity of the fat tissue, it pulls the boundary out towards the r level set in one place before it should.

To remove these drawbacks the following rules are incorporated in to the calculation of the cost matrix:

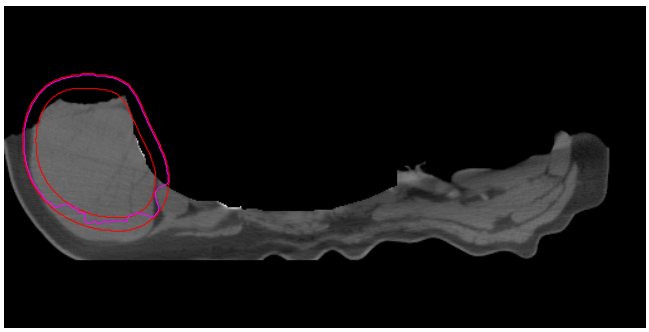


Figure 17.4: Searching for the region of interest using the cost matrix $\mathbf{C}_{mean}$

- All elements in the cost matrix, where the corresponding pixel in the sample image is above the threshold τ, receive the same high value, e.g. the maximum value of the intensities inside the r level set. This basically equalized the importance of all the meat pixel. Thus, removing the uncontrollable behavior in pure meat column.
- Uncontrollable behavior in a fat column can be avoid by changing the sample mean operator slightly. The modified mean operator is $f_\tau(\mathbf{a}) = \frac{1}{n+1}(\tau + \sum_i^n a_i)$. This will push the region of interest slightly towards the r level set in cases where the bottom pixels in a column are fat pixels.
- To eliminate the influence of the background pixels, all elements in the cost matrix, where the corresponding pixel in the sample image is a background pixel, receives the same value as meat pixels.

The following cost matrix emerges:

$$\mathbf{C}_v(t,c) = \begin{cases} \frac{1}{M_1-i+2}(\tau + \sum_{i=t}^{M_1} I_s(i,c)) & \text{if } I_s(i,c) < \tau \text{ and } I_s(i,c) > b \\ \max_{R^u+R^v}(I_s) & \text{otherwise} \end{cases} \tag{17.2}$$

where b is the background value.
Figure 17.3(c) shows the cost matrix $\mathbf{C}_v$ of the sample image shown in Figure 17.1(c). The region of interest obtained from this cost matrix is shown in 17.5.

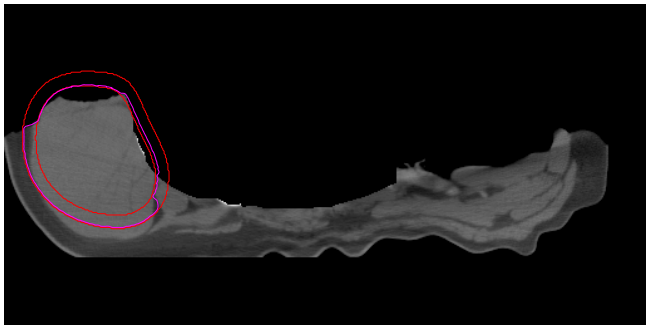


Figure 17.5: Searching for the region of interest using the cost matrix $\mathbf{C}_v$.

The region of interest can easily be expanded in areas with background pixels if it is desirable. The result is shown in Figure 17.6.

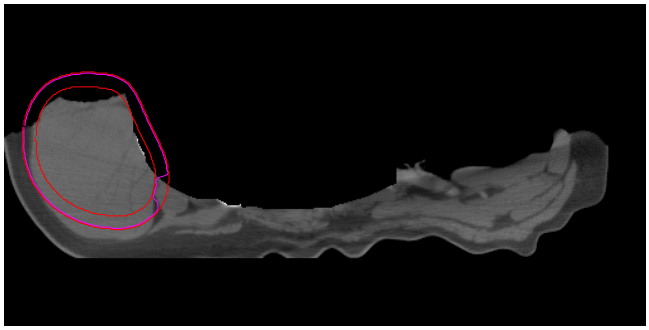
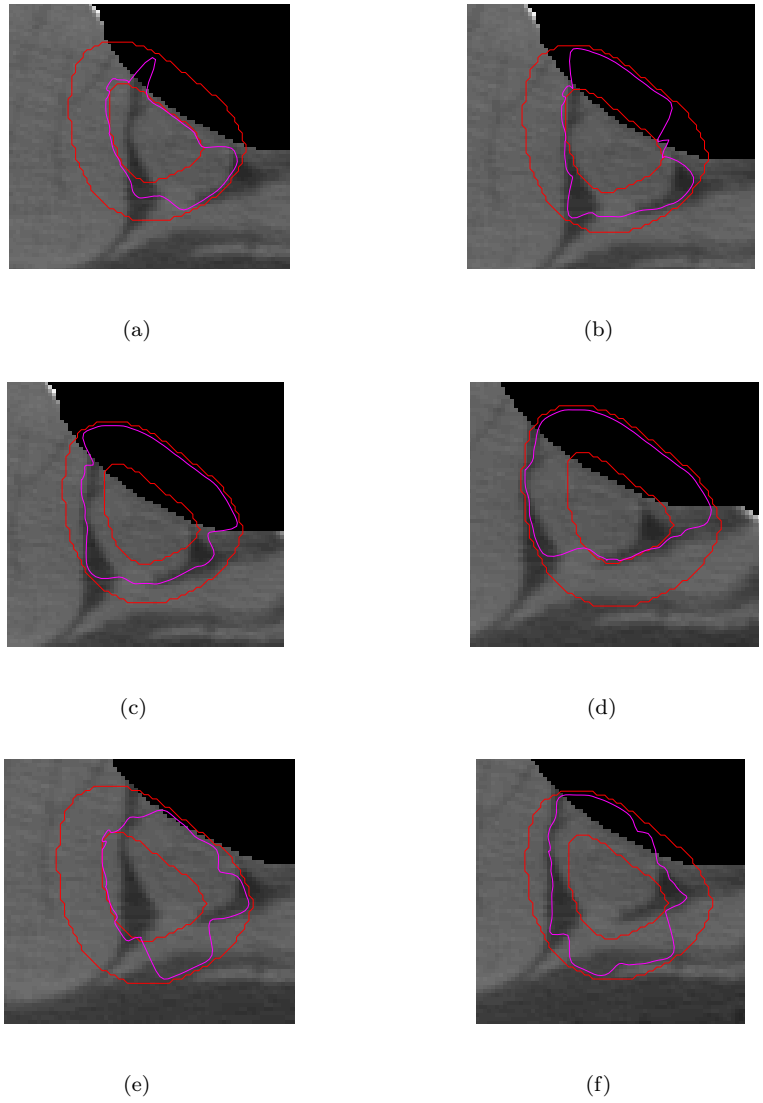


Figure 17.6: Searching for the region of interest using the cost matrix $\mathbf{C}_v$ with background correction.

To illustrate the robustness of this approach the mean shape of muscle 42 was placed in six different positions in the proximity of muscle 42, and the region of interest was calculated from $\mathbf{C}_v$. The resulting regions of interest from the six cases are displayed in Figure 17.7. In every case the located region of interest is *better* than the ten level set.
In the region based segmentation framework presented in Chapter 12 the region of interest is only used to calculate the region statistics. From Figure 17.7, it is seen that the region of interest boundary has a tendency to stay close to the

zero level set in areas, where the zero level set is inside another muscle than 42. This indicates that it might be reasonable to give the region of interest (given by $\mathbf{C}_v$) a direct influence on the evolution of the segmenting curve.



(a) (b)

(c) (d)

(e) (f)

Figure 17.7: The intelligent region of interest of muscle 42 at six different positions.

17.1 Discussion

No testing has been done to confirm that the incorporation of an intelligent region of interest will improve the performance of the region based framework. The reasons, why the intelligent region of interest has not been integrated in the region based segmentation framework, are:

- The implementation of the region based framework is generic with respect to dimension, while the theory described in this chapter only works in two dimension.
- The idea was conceived in the final period of the thesis.

Part III

Discussion

Chapter 18

Future work

This chapter will discuss possible extensions to the work of this thesis.

18.1 Direct 3D segmentation

The most obvious extension to the work done in the thesis is to build 3D models of the muscles in the pig back and then try to segment the pig back directly in 3D. As mentioned, it is the believe of the author, that many of the problems encounted with the simulateneous segmentation in 2D will disappear in 3D. Unfortunately, more tedious outlining is needed in order to get complete outlines of the muscles in the pig back. Surely, the task of aligning the shapes becomes rather complex if the outlines are not complete.
The implementation done in this thesis is generic with respect to dimension, so it should in principle work in any dimension.

3D segmentation with the LSFSM in a region based framework has already been done and described in [25].

18.2 Shape registration using level sets

In this thesis, shape registration is applied prior to the PCA to remove the variance from the training which is due to rigid transformation (rotation, scaling and translation). The binary shape alignment approach aligns a set of shapes by maximizing the overlap among them. Clearly, this does not guarantee that the variance in the set of signed distance maps due to rotation, scaling and translation is complete removed.

A more correct approach would be to minimize the sum of squared differences between the level set functions. More formally, given a set of level set functions $\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \ldots, \boldsymbol{\Psi}_n\}$ we wish to minimize the following energy with respect rotation, scaling and translation:

$$E_{align} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \int \ldots \int_{\Omega} (\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_j)^2 dV \tag{18.1}$$

where Ω denotes the image domain. Such an approach for two shapes is described in [18]. The approach is somewhat more complicated than the binary shape alignment approach as signed distance maps are not invariant with respect to scaling.

18.3 Different shape metric

As mention there are several weaknesses with the LSFSM. To summarize these weaknesses are:

- Linear operations on a set of signed distance maps do not produce a signed distance map. Hence, the level set functions generated by our model are not true signed distance maps.
- Obviously, representing shapes by their signed distance map introduces a large degree of redundancy.
- It is not completely intuitive that the sum of squares of the difference between two signed distance maps of two shapes provides a good similarity measure between the shapes.

In recent work by Charpiat et al.[2] another shape metric for measuring the similarity between shapes is proposed. The shape metric in question is a differential approximation to the well known Hausdorff distance ρ_H. The Hausdorff distance between two shapes is the maximum distance between any point on one of the shape boundaries to the closest point on the boundary of the other shape. Thus, the Hausdorff distance between the two shapes Γ and Γ' can be written as:

$$\rho_H(\Gamma,\Gamma') = \lim_{\alpha,\beta,\gamma\to+\infty} \left\langle \langle\langle d(\cdot,\cdot)\rangle_{\Gamma'}^{\varphi_\alpha}\rangle_{\Gamma}^{p_\beta}, \langle\langle d(\cdot,\cdot)\rangle_{\Gamma}^{\varphi_\alpha}\rangle_{\Gamma'}^{p_\beta} \right\rangle^{p_\gamma} \tag{18.2}$$

where $d(\cdot,\cdot)$ is the Euclidian distance between two points, φ is any continuous strictly monotonically descreasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$, $\langle f\rangle_\Gamma^\varphi$ is the φ-average of f along the boundary of Γ, $\langle\cdots\rangle^\varphi$ is the discrete φ-average,$\varphi_\alpha(x) = \varphi(x)^\alpha$ and finally $p_\alpha(x) = x^\alpha$
As a result $\tilde{\rho}_H$ is an approximation to the Hausdorff distance for positive values of α, β and γ:

$$\tilde{\rho}_H(\Gamma,\Gamma') = \left\langle \langle\langle d(\cdot,\cdot)\rangle_{\Gamma'}^{\varphi_\alpha}\rangle_{\Gamma}^{p_\beta}, \langle\langle d(\cdot,\cdot)\rangle_{\Gamma}^{\varphi_\alpha}\rangle_{\Gamma'}^{p_\beta} \right\rangle^{p_\gamma} \tag{18.3}$$

In [2], an empirical mean shape $\bar{\Gamma}$ of n shapes is obtained by minimizing the following energy function:

$$\Gamma \to \mu(\Gamma,\Gamma_1,\Gamma_2,\ldots,\Gamma_n) = \frac{1}{n}\sum_{i=1}^{n} E^2(\Gamma,\Gamma_i) \tag{18.4}$$

where the energy E is a regularized version of the approximated Hausdorff distance. More precisely, the energy E is defined by:

$$E(\Gamma,\Gamma') = \left\langle \tilde{p}_H(\Gamma,\Gamma'), \langle|\Gamma|,|\Gamma'|\rangle^{\tilde{\Psi}} \right\rangle \tag{18.5}$$

where $|\Gamma|$ is the length of the curve of Γ.

The gradient of $E(\Gamma,\Gamma_i)$ defines a normal velocity field β_i such that the infinite small deformation $\Gamma - \beta_i \mathbf{n} d\tau$ decreases $E(\Gamma,\Gamma')$. A symmetric $n\times n$ matrix $\hat{\Lambda}$, similar to the covariance matrix, is defined by $\hat{\Lambda}_{ij} =< \beta_i - \hat{\beta}, \beta_j - \hat{\beta} >$ where $\hat{\beta}$ is the mean velocity field $\hat{\beta} = \frac{1}{N}\sum_{i=0}^{n}\beta_i$. The modes of variation (normal deformation flows) are obtained from k n-dimensional eigenvectors $\mathbf{u}_j$ of $\hat{\Lambda}_i$ and

the k corresponding eigenvalues λ_j:

$$v_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} u_{ij}(\beta_i - \hat{\beta}) \tag{18.6}$$

The j mode of variation can be viewed by solving the PDE:

$$\Gamma_t = \pm v_j \mathbf{n} \tag{18.7}$$

with the initial condition $\Gamma(0, .) = \hat{\Gamma}(.)$.

In contrast to the LSFSM this shape model does not make the incorrect assumption that the distance maps lay on a linear manifold. In conclusion, the adoption of this model would remove two of three drawback mentioned above.

18.4 Predicting the elastic deformation

The pig stretching method described in Chapter 10 has spawned a new project inside the VSH study which aims to predict the deformation of a cutting before it is cut out. Naturally, this will be a valuable tool for product planning both with regards to quality and profit.
Establishing 3D correspondence between the scanned carcasses and the scanned cuttings is essential for building a cut prediction model. One obvious approach is to use some kind of intensity based registration. However, a more interesting approach (in light of this thesis) is to derive the correspondence by applying segmentation using 3D shape models of anatomical structures.

Chapter 19

Discussion

19.1 Summary of the main contributions

The primary of objectives of this thesis were to

- develop a method, which can locate the 18cm back in a CT scan,
- find the dissecting lines which divide the 18cm back in to the loin part and the three tail parts,
- and identify a suitable method for the meat-fat segmentation of the 18cm back as well as possible problems such as the existence of artifacts in the images.

These requirements have all been fulfilled. Furthermore, it is the authors opinion, that the developed or chosen methods for solving these problems are *simple* in a conceptual sense. Thus, they perform robustly.

The extraction of the 18cm back problem was reduced to one of triviality as soon at the problem of locating the ribs in the back was solved. The rib removal

problem was treated as a *simple* data fitting problem; (i) finding data points, (ii) pruning outliers and (iii) fitting a function. Though, the method relies on the selection of several pruning parameters, parameter tuning is not an issue.

The location of the dissecting lines relies on the *simple* yet intelligent observation that the meat percentages of the scan lines drops drastically at the end of the loin.

Linear discriminant analysis is a relative *simple* classification approach, which most likely will perform adequately when it comes to determination of the meat-fat percentages. Both linear and quadratic discriminant analysis were applied with reasonable success despite the existence of inhomogeneities.

More advanced segmentation methods, which incorporate shape priors, are interesting alternatives to the pure intensity based classification as they can provide the volumes of specific structures in addition to the mean-fat percentage. Furthermore, such methods are less fragile to intensity inhomogeneities than the pure intensity based classification. An implicit shape model was constructed by converting a set of shapes to a set of signed distance maps and then performing a PCA on the set of signed distance maps. Consequently, the shape model does not rely on the point correspondence, which is a major advantage compared to the more common point model. Simple outlining is preferred to annotation as:

- It requires less manual labor.
- Locating corresponding landmarks on muscles consistently is an issue as muscles are soft structures with low curvature.
- 3D annotation is a substantial harder task than 2D annotation - especially if it is impossible to establish correspondence between the slices in the data set.

The implicit shape model was integrated into a region based framework, which was extended to incorporate a region of interest. The two reasons for incorporating a region of interest are:

- It secures a more homogeneous outside region.
- It requires a substantial less amount of computation time.

The basic region based segmentation approach was compared against the Active Appearance Model in a series of leave-one-out experiments on a set of Corpus Callosum images. The region based framework with the Chan Vese energy function and a region of interest defined by the 25 level set was almost able to match the performance of the AAM. Despite the better performance of the AAM, it is the opinion of the author, that the LSFSM is the preferred shape model to use for the segmentation of pig backs as the AAM relies on landmarks. If the final goal is to be able to segments a pig carcass with shape models or just to have a virtual statistic model of a pig carcass, the annotation task becomes infeasible.

Segmenting a pig back with shape models is a complex problem as the muscles in a pig back lay in close proximity to each other. The basic region based framework was reformulated to handle multiple models simultaneously and to discourage overlap between the models. A series of leave-one-out experiments proved that the coupled model framework is unable to segment a whole pig back in a series of 2D segmentations due to a couple of stability problems. These stability issues will however disappear in 3D.

19.2 Conclusion

On a yearly basis, approximately 25,000,000 pigs enter and leave the slaughter lines of the Danish slaughterhouses, and a modern slaughterhouse processes around 7,000 pigs every day or approximately one pig every eight second. These facts place high requirements on a vision system with regard to robustness and speed, if it is to be installed in a slaughter line.

Presently, CT acquisition devices are not able to achieve acquisition times which can match the demands. For example the average acquisition time of the CT scans used in the thesis was approximately half an hour. Due to the growing evolution within non-invasive acquisition devices, it is not unrealistic to believe that CT or another technique with similar quality can accomplish a satisfactory acquisition time within the next decade.

Fear of radiation among the consumers might also prove to be a big obstacle, no matter how irrational the fear might be. The Danish veterinary and food administration has already approved the use of CT in the slaughter lines. Hence, there are no health risk involved with using CT for carcass classification.

The vision software must also uphold the requirements. The *simple* methods presented in this thesis require an insignificant amount of computation time on a standard personal computer why time constraints are no obstacle. With regard to robustness, it the believe of the author that the methods are robust enough to be installed in practise. Nevertheless, as only a relative small data set has been available to the thesis a large scale robustness test is needed.

It is the conclusion of this thesis that it is possible to perform a valid quality estimation of an 18cm from a CT scan of the pork middle. At least, it is not a lack of image analysis tools which prevents this vision project from being carried out in practise.

Bibliography

[1] Tony Chan and Luminita A. Vese. Active contours without edges. *IEEE transaction on image processing*, 10, 2001.

[2] Guillaume Charpiat, Olivier Faugeras, and Renaud Keriven. Approximations of shape metrics and application to shape warping and emperical statistics. Technical report, Mai 2005.

[3] Lars Bager Christensen, Bo L. Jespersen, and Claus Borggaard. Spectral impact from reconstruction of ct tomograms. *Vision Days*, 2005.

[4] Knut Conradsen. *Introduktion til statistik*, volume 2. IMM.

[5] Tim F. Cootes and Camillo J. Taylor. Statistical models of appearance for computer vision. Technical report, 2001.

[6] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Company, 2000.

[7] DMRI. Home page of the danish meat research institute, 2005. http://www.DMRI.dk.

[8] EUPIGCLASS. Home page of the eupigclass, 2005. http://www.eupigclass.net.

[9] C.A. Glasbey and K.V. Mardia. A review of image warping methods. *Journal of Applied Statistics*, 25:155–171, 1998.

[10] Claus Gramkow. *Registration of 2D and 3D Medical Images*. PhD thesis, IMM, Technical University of Denmark, 1996.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, pages 144–148. Springer, 2003.

[12] Luis Ibanez, Will Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide*. Insight Software Consortium, 2 edition, 2005.

[13] David G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. Wiley, 1999.

[14] Junmo Kim, John W. Fisher III, Anthony Yezzi, Jr. Mujdat Cetin, and Alan S. Willsky. Nonparamtric methods for image segmentation using information theory and curve evolution. *ICIP*, 2002.

[15] Michael E. Leventon, W. Eric L. Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. *IEEE conference on computer vision ans pattern recognization*, 2000.

[16] Stanley Osher. *Geometric level set methods*. Springer, 2003.

[17] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. *Journal of Computational Physics*, 1988.

[18] Nikos Paragios and Mikhael Rousson. Shape analysis towards model-based segmentation. In *Geometric level set methods*. Springer, 2003.

[19] Lakshman Prasad. Mmorphological analysis of shapes. Technical report, 1997.

[20] L. E. Romans. *Introduction to Computed Tomography*. Williams & Wilkins, 1995.

[21] Mikkel B. Stegmann. Active appearance models: Theory, extensions and cases, 2000.

[22] Changming Sun and Stefano Pallottino. Circular shortest path in images. *Pattern Recognition*, 2002.

[23] Andy Tsai, Anthony Yezzi, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric Grimson, and Allan Willsky. A shape-based approach to the segmentation of medical imagery using levet set. *IEEE transactions on Medical imaging*, 22, 2003.

[24] Pamela J. Vermeer. *MAT to Boundary Representation Conversion*. PhD thesis, Department of Computer Sciences, Purdue University, 1994.

[25] Andy Tsai William Wells, Clare Tempany, W. Eric Grimson, and Allan Willsky. Coupled multi-shape model and mutual information for medical image segmentation. *Information Processing in Medical Imaging*, 2003.

[26] Anthony Yezzi, Andy Tsai, and Alan S. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. *International Conference on Computer Vision*, 2, 1999.

[27] Gary Yngve and Greg Turk. Robust creation of implicit surfaces from polygonal meshes. *IEEE transactions on visualization and computer graphics*, 8, 2002.

Appendix A

The VSH outlined muscle database

A.1 Database description

This database consists of 8 CT scan of pig backs scanned by Danish Meat Research Institute. Each scanned pig back has been saved as a series of dicom files. Information such as the resolution, the field of view and so on can be extracted from the dicom header. The IDs of the pigs are 4, 7, 9, 12, 14, 18, 20 and 22. A number of slices have been cropped from each scan such that all the scans consist of exactly 60 slices. In all of these slice the following muscles have been outlined:

42 Iliocostalis.

48 Latissimus Dorsi.

50 Longissimus (Loin).

89 Rectus Femoris.

90 Rectus Thoracis.

106 Trapezius.

IDs of the muscles have been taken for the online pig atlas at http://www.porcine.unl.edu. A total of 1701 outlined shapes can be found in the database. The outlined shapes have been converted to binary shape images and saved as BMP images.

A.2 Specification

A.2.1 Library structure

The library structure of the database is shown Figure A.1. The root directory contains two directories - the scan directory named `images` and the shape directory named `shapes`. In both the `shapes` and the `images` directory there exist several directories with names identical the IDs of the pig. Naturally, the directory with the name 4 in the `images` directory contains the dicom files from the CT scan of pig 4, and the directory with the name 4 in the `shapes` directory contains the outline shape files of pig 4.

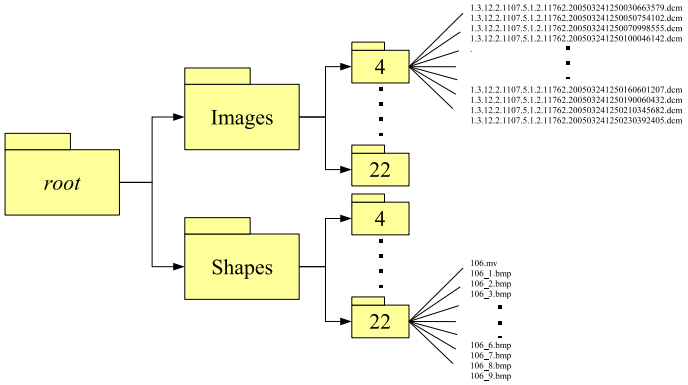


Figure A.1: Database library structure

A.2.2 Image formats and naming

There exist to image formats in the database; the dicom format in which the actual scans are stored, and the BMP format in which the outlined shapes are stored. The BMP images are named `XX_YY.BMP` where `XX` is the ID of the outlined muscle and `YY` is the slice number.

A.2.3 The `.mv` file

The move files `.mv` is used to store information about the position of the outlined shapes in their corresponding slice images and is placed in the same directory as the outlined shape images. The line format of the file is shown beneath:

```
(#PIG_ID, #SLICE_NUMBER) #X_TRANS #Y_TRANS
```

where `(#PIG_ID` is the ID of the pig, `#SLICE_NUMBER` is the number of the slice, `#X_TRANS` and `#X_TRANS` specify the translation from the slice image space to the outlined shape image space.

A.3 Term of Use

The database can only be used with a prior approval from DMRI. Contact: Lars Bager Christensen from DMRI.

Appendix B

Implementations in ITK

Most of the implementation has been done in c++ within the Insight Tool Kit (ITK), which is a framework for medical imaging.

B.1 A way too brief introduction to ITK

A detailed description of ITK is beyond the scope of this report simply because ITK is to far too complex and large to be described in less than a few hundred pages. That said, a small description of ITK is needed in order to understand why the author has spend a rather considerable amount of time learning ITK and implementing the theory in ITK. For more information about ITK consult the 800+ page software guide to ITK [12].

In addition to image tools and numeric packages, ITK also provide a general programming tool and structure such as memory management by reference counting, pipelining between filters and suitable base classes. These features combined with the fact that ITK utilizes all of C++'s generic capabilities (templates, inheritance etc.) makes this highly a recommendable framework for most image analysis problem.

Further ITK is 100% platform independent and can easily be installed on any linux or windows platform using CMake which is a cross-platform, open-source make system. ITK utilizes a number of numerical and image analysis libraries such as VNL (numeric library) and GDCM (Gnu Dicom libraries). However, as these libraries are open source everything comes in one installation packages, which can be downloaded from http://itk.org.

B.1.1 Filter and data object

The most essential concept in ITK is the Data Processing Pipeline(DPP). The DPP operates with two types of objects *data objects*, which represent data, and filters, which operates on data objects. Data objects are either images or meshes, and all image and mesh classes are either derived from the `itk::Image` class or the `itk::Mesh` class. A filter reads data objects from the input pipe, performs some operation on the data objects, and outputs the resulting data object to the output pipe, whenever the `Update()` function is called. As a result filters can be connected in a pipeline. A simple pipeline is illustrated in Figure B.1. If the `Update()` function is called on a filter a in a pipeline the `Update()` will propagate backwards through the pipeline such that all filters before filter a are updated before filter a is updated. Further, the pipeline system is smart such that a filter is not updated unless the input has been changed.

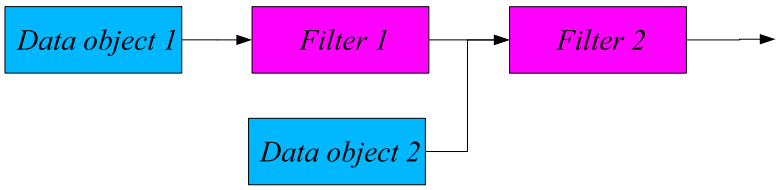


Figure B.1: Pipeline in ITK.

B.2 Additional libraries

One additional library has been integrated into the ITK framework by the author. Library in question is the TRIANGLE library by Jonathan Shewchuk, which can perform 2D CDT. The functionality of the TRIANGLE library has been wrapped by the filter `ConstrainedDelaunayTriangulationFilter`.

B.3 Own implementation

In this section, the most important classes, implemented doing the course of this project, will be listed together with a small description,

B.3.1 Basic objects

The following basic objects have been implemented:

ExtendedPCAShapeSignedDistanceFunction The implementation of the LSFSM. Functionality includes pose transformation, evaluation of the zero level set and point evaluation. Generic with respect to dimension.

DependedParameterController The implementation of the difference pose model.

***CostFunction** Implementation of the cost functions described in Chapter 12 and Chapter 13.

ConstrainedGradientDescentOptimizer A simple gradient descent optimizer with optional upper and lower parameter limits.

RadialBasisApproximationFunction Implementation of the radial basis approximation from Chapter 3.4 with pruning functionality.

B.3.2 Filters

The following filters have been implemented:

RibRemovalImageFilter This filter removes the rib from a 3D image of a pig back.

Find18cmBackImageFilter This filter finds the 18cm back from a 3D image of pig back.

Segment18cmBackImageFilter This filter finds the dissection lines of an 18cm back and segments the 18cm back by thresholding.

ChordalAxisTransformationFilter This filter finds the chordal axis of an object in a 2D image. Includes longest path functionality.

PigStretchingFilter This filter perform pig stretching on a 3D image of a pig back according to Chapter 10.

SingleShapePriorRegionBasedSegmentationImageFilter This filter segments a single object with a shape model using a specified cost function.

MultiShapePriorRegionBasedSegmentationImageFilter This filter segments multiple objects sequentially with multiple shape models using a specified cost function in a series of N dimensional images. The filter uses the `DependedParameterController` to initialize the pose of the shapes.

CoupledShapePriorRegionBasedSegmentationImageFilter This filter segments multiple objects simultaneously with multiple shape models using a specified cost function in a series of N dimensional images. The filter uses the `DependedParameterController` to initialize the pose of the shapes.

Note, the three segmentation filters are generic with respect to dimension, so they should in principal work with on any dimensional image. The source code can be found on the enclosed cd under the directory `source c++`.

Appendix C

Implementations in Matlab

The model building facility used to construct the models in this thesis has been implemented in Matlab. The model building facility includes the following toolboxes:

The outline toolbox The toolbox contains the functionality needed for outlining objects in images.

The shape registration toolbox The toolbox contains implementation of the two shape registration methods used in this thesis - Rigid registration by moments and binary shape alignment.

The model builder toolbox In addition to actual statistical model builder this toolbox also contains functionality for combining two or more shape classes, cropping and padding binary shape images, and building a pose model.

The source code can be found on the enclosed cd under the directory `source matlab`.

Appendix D

File formats

No less than a total of five different file formats are used to store information and models. The formats with extension and functionality description are shown beneath:

.mv The move file format. The file is placed in the shape directory of a CT scan, and specifies the translation needed to get from a slice image in the CT scan to the space of the outlined shape.

.trf The transformation file. The files contains a list of all the shapes used to build the model. Each line in the file contains the pig id and slice id of a shape, and the parameters of the reverse transformation of the Euclidian transformation, which has been applied to the shape in order to get it from the slice image space to the aligned image space.

.inf The model information file. This file contains essential information about a shape model such as the number of principal modes, the truncation level, the map size etc.

.dsm The model file.

.dpp The depended pose parameter file. The file contains the estimated mean and covariance matrix for "difference in pose" model.

Examples of the non-binary files can be found in Chapter E.

Appendix E

Examples of file formats

E.1 .mv file:

```
(7, 1): -9.86 -35.37
(7, 2): -17.62 -36.55
(7, 3): -19.86 -39.73
(7, 4): -15.33 -38.77
(7, 5): -12.75 -38.43
(7, 6): -13.42 -41.26
(7, 7): -14.93 -40.54
(7, 8): -13.05 -39.40
(7, 9): -14.65 -41.29
      ...
      ...
      ...
(7, 58): -15.55 -68.04
(7, 59): -17.73 -67.78
(7, 60): -18.35 -70.30
```

E.2 .inf file:

```
############################################################
Distance Map Shape Model File

Date of creation           : 22-Sep-2005

Format                     : 0.1

Model description          : 50

Map size                   : 160x160

Number of shapes           : 481

Truncation level           : 0.95

Number of parameters       : 14

PC variation               :
        1        :        34.87%        (34.87%)
        2        :        18.71%        (53.58%)
        3        :        13.40%        (66.98%)
        4        :        10.28%        (77.26%)
        5        :         6.94%        (84.20%)
        6        :         2.33%        (86.53%)
        7        :         1.95%        (88.47%)
        8        :         1.61%        (90.08%)
        9        :         1.14%        (91.22%)
        10        :         0.91%        (92.13%)
        11        :         0.85%        (92.98%)
        12        :         0.70%        (93.68%)
        13        :         0.63%        (94.31%)
        14        :         0.52%        (94.83%)

############################################################
```

E.3 .trf file

```
(4 , 1): -0.14 1.14 -2.28 41.79
(4 , 2): -0.36 1.09 5.80 39.08
(4 , 3): -0.47 1.05 5.33 49.83
(4 , 4): -0.36 1.07 3.19 50.23
(4 , 5): -0.18 1.09 -1.57 51.09
(4 , 6): 0.14 1.03 -0.65 48.64
(4 , 7): -0.25 1.07 3.97 44.57
(4 , 8): -0.20 1.07 0.46 50.71
(4 , 9): -0.30 1.05 -0.04 55.86
(4 , 10): -0.25 1.05 -3.30 53.85
          ...
          ...
          ...
(22 , 58): -0.49 1.04 23.82 44.39
(22 , 59): -0.42 1.03 27.90 45.15
(22 , 60): -0.48 1.03 63.15 18.81
```

E.4 .dpp file

```
Origo muscle: 50

Order: 106 -> 42 -> 48 -> 90 -> combined_42_89_50

Mean:
-0.27 0.01 0.01 -21.86 53.32 0.00 -0.02 0.10 0.10 105.30  ...

Covariance matrix:
0.08 -0.01 -0.01 0.78 1.02 0.00 0.04 -0.00 -0.00 0.19 0.16  ...

```

Appendix F

Models

F.1 42

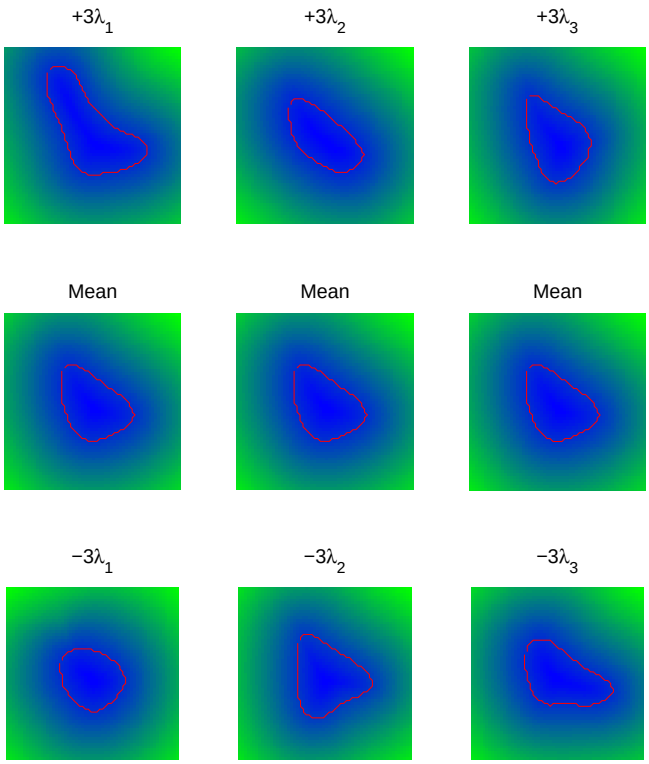


Figure F.1: The first three principal components

F.2 48

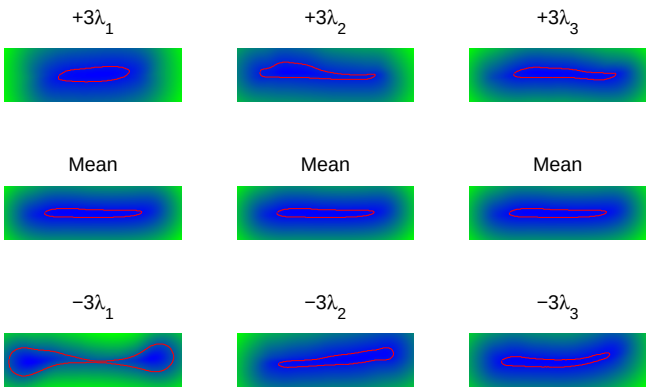


Figure F.2: The first three principal components

F.3 50

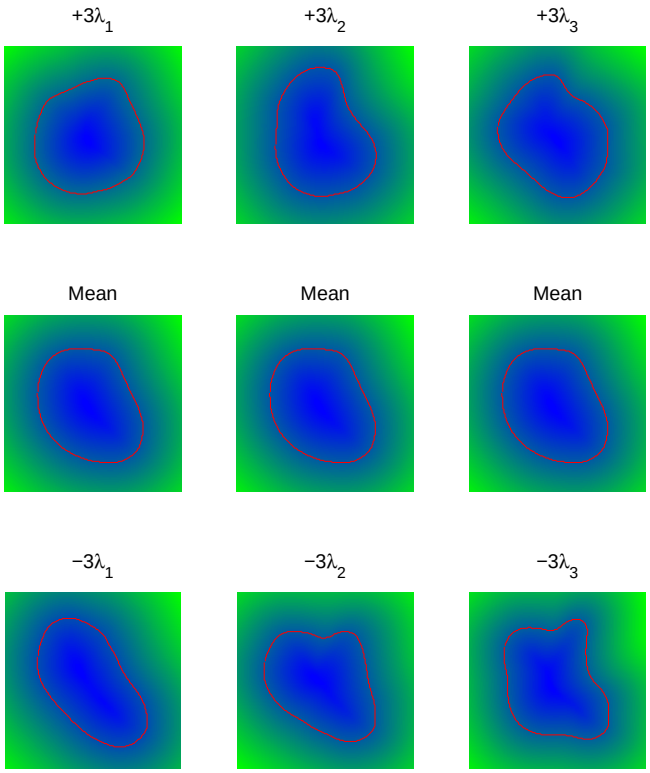


Figure F.3: The first three principal components

F.4 90

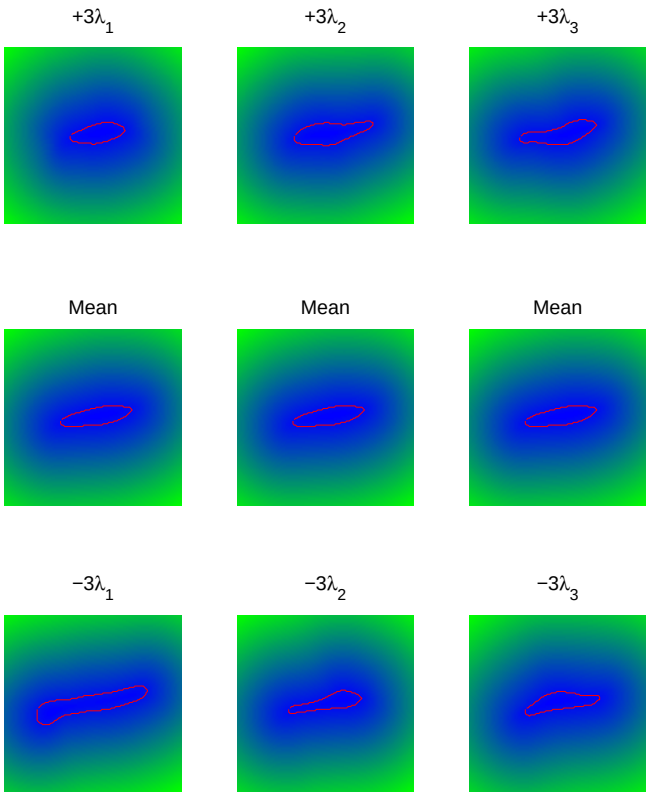


Figure F.4: The first three principal components

F.5 106

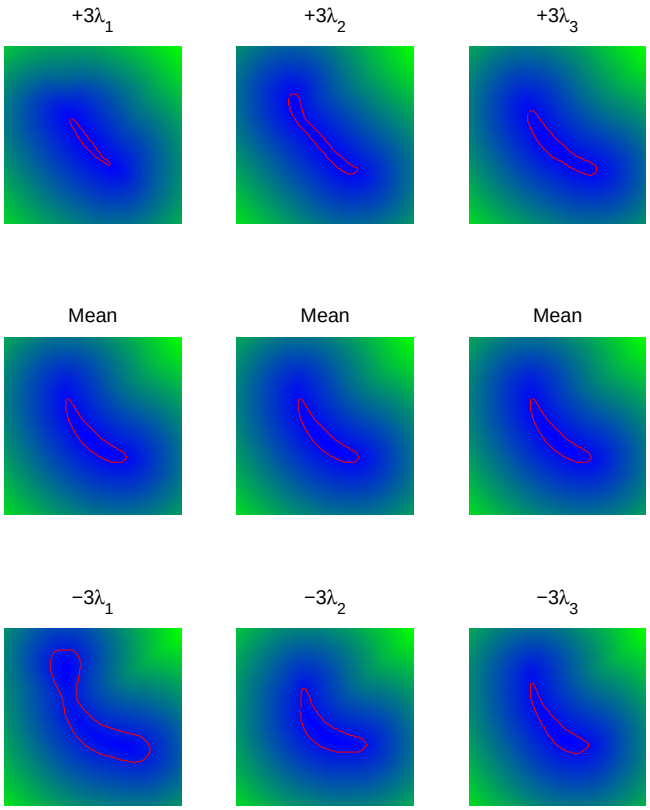


Figure F.5: The first three principal components