Automatic Tissue Segmentation of Volumetric CT Data of the Pelvic Region

Julius Jeuthe
Master of Science Thesis in Biomedical Engineering

Automatic Tissue Segmentation of Volumetric CT Data of the Pelvic Region

Julius Jeuthe

LiTH-IMT/BIT30-A-EX–15/524–SE

Supervisor: Alexandr Malusek, Maria Magnusson, Mats Andersson
Examiner: Hans Knutsson

Division of Medical Informatics
Department of Biomedical Engineering
Linköping University
SE-581 85 Linköping, Sweden

Copyright © 2017 Julius Jeuthe
Abstract

Automatic segmentation of human organs allows more accurate calculation of organ doses in radiation treatment planning, as it adds prior information about the material composition of imaged tissues. For instance, the separation of tissues into bone, adipose tissue and remaining soft tissues allows to use tabulated material compositions of those tissues. This approximation is not perfect because of variability of tissue composition among patients, but is still better than no approximation at all. Another use for automated tissue segmentation is in model based iterative reconstruction algorithms. An example of such an algorithm is DIRA, which is developed at the Medical Radiation Physics and the Center for Medical Imaging Science and Visualization (CMIV) at Linköping University. DIRA uses dual-energy computed tomography (DECT) data to decompose patient tissues into two or three base components. So far DIRA has used the MK2014 algorithm which segments human pelvis into bones, adipose tissue, gluteus maximus muscles and the prostate. One problem was that MK2014 was limited to 2D and it was not very robust.

Aim:
The aim of this thesis work was to extend the MK2014 to 3D as well as to improve it. The task was structured to the following activities: selection of suitable segmentation algorithms, evaluation of their results and combining of those to an automated segmentation algorithm. Of special interest was image registration using the Morphon.

Methods:
Several different algorithms were tested. For instance: Otsu’s method followed by threshold segmentation; histogram matching followed by threshold segmentation, region growing and hole-filling; affine phase-based registration and the Morphon. The best-performing algorithms were combined into the newly developed JJ2016.

Results:
For the segmentation of adipose tissue and the bones in the eight investigated data sets, the JJ2016 algorithm gave better results than the MK2014. The better results of the JJ2016 were achieved by: (i) a new segmentation algorithm for adipose tissue which was not affected by the amount of air surrounding the patient and segmented smaller regions of adipose tissue and (ii) a new filling algorithm for connecting segments of compact bone. The JJ2016 algorithm also estimates a likely position for the prostate and the rectum by combining linear and non-linear phase-based registration for atlas based segmentation. The estimated position (center point) was in most cases close to the true position of the organs. Several deficiencies of the MK2014 algorithm were removed but the improved version (MK2014v2) did not perform as well as the JJ2016.

Conclusions:
JJ2016 performed well for all data sets. The JJ2016 algorithm is usable for the intended application, but is (without further improvements) too slow for interactive usage. Additionally, a validation of the algorithm for clinical use should be performed on a larger number of data sets, covering the variability of patients in shape and size.

Nyckelord
Image segmentation, Computed tomography, Pelvic region, Threshold segmentation, Region growing, Histogram matching, Atlas segmentation, Morphon
Abstract

Automatic segmentation of human organs allows more accurate calculation of organ doses in radiation treatment planning, as it adds prior information about the material composition of imaged tissues. For instance the separation of tissues into bone, adipose tissue and remaining soft tissues allows to use tabulated material compositions of those tissues. This approximation is not perfect because of variability of tissue composition among patients, but is still better than no approximation at all. Another use for automated tissue segmentation is in model based iterative reconstruction algorithms. An example of such an algorithm is DIRA, which is developed at the Medical Radiation Physics and the Center for Medical Imaging Science and Visualization (CMIV) at Linköpings University. DIRA uses dual-energy computed tomography (DECT) data to decompose patient tissues into two or three base components. So far DIRA has used the MK2014 algorithm which segments human pelvis into bones, adipose tissue, gluteus maximus muscles and the prostate. One problem was that MK2014 was limited to 2D and it was not very robust.

Aim: The aim of this thesis work was to extend the MK2014 to 3D as well as to improve it. The task was structured to the following activities: selection of suitable segmentation algorithms, evaluation of their results and combining of those to an automated segmentation algorithm. Of special interest was image registration using the Morphon.

Methods: Several different algorithms were tested. For instance: Otsu’s method followed by threshold segmentation; histogram matching followed by threshold segmentation, region growing and hole-filling; affine phase-based registration and the Morphon. The best-performing algorithms were combined into the newly developed JJ2016.

Results: For the segmentation of adipose tissue and the bones in the eight investigated data sets, the JJ2016 algorithm gave better results than the MK2014. The better results of the JJ2016 were achieved by: (i) a new segmentation algorithm for adipose tissue which was not affected by the amount of air surrounding the patient and segmented smaller regions of adipose tissue and (ii) a new filling algorithm for connecting segments of compact bone. The JJ2016 algorithm also estimates a likely position for the prostate and the rectum by combining linear and non-linear phase-based registration for atlas based segmentation. The estimated position (center point) was in most cases close to the true position of the organs. Several deficiencies of the MK2014 algorithm were removed but the improved version (MK2014v2) did not perform as well as the JJ2016.

Conclusions: JJ2016 performed well for all data sets. The JJ2016 algorithm is usable for the intended application, but is (without further improvements) too slow for interactive usage. Additionally, a validation of the algorithm for clinical use should be performed on a larger number of data sets, covering the variability of patients in shape and size.
Acknowledgments

First, I would like to express my sincere gratitude to my supervisors Alexandr Malusek and Maria Magnusson for their continuous support, guidance and patience during the writing of this thesis work. I am especially grateful to Alexandr Malusek, for assisting me with setting up the structure of the report and later helping me with the corrections.

Secondly, I want to thank Anders Eklund and my supervisor Mats Andersson for their invaluable advice on atlas segmentation and Michael Sandborg for providing the data sets for my thesis work. I would also like to thank Hans Knutsson, my examiner, for his useful advice and comments during the thesis work and my opponent, Daniel Sandsveden, for his final comments on the report.

Additional, I would like to thank the staff at the department of Medical Radiation Physics at Linköpings University Hospital for a great and interesting time.

Lastly I want to thank my friends and my family for supporting me throughout my studies and my writing of this thesis work.

Linköping, January 2017
Julius Jeuthe
# Contents

<table>
<thead>
<tr>
<th>Notation</th>
<th>xiii</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Computed tomography</td>
<td>2</td>
</tr>
<tr>
<td>1.2 DIRA</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Image segmentation</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Aim</td>
<td>6</td>
</tr>
<tr>
<td>1.5 Thesis Outline</td>
<td>6</td>
</tr>
<tr>
<td><strong>2 Theory</strong></td>
<td>7</td>
</tr>
<tr>
<td>2.1 Threshold segmentation</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Otsu’s method</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Seeded Region Growing</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Neighborhood</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Morphological operations</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Histogram equalization</td>
<td>10</td>
</tr>
<tr>
<td>2.5 Histogram matching</td>
<td>10</td>
</tr>
<tr>
<td>2.6 Atlas Segmentation</td>
<td>11</td>
</tr>
<tr>
<td>2.6.1 Linear image registration</td>
<td>12</td>
</tr>
<tr>
<td>2.6.2 Non-linear registration</td>
<td>20</td>
</tr>
<tr>
<td>2.7 Dice Similarity Coefficient</td>
<td>22</td>
</tr>
<tr>
<td><strong>3 Data sets</strong></td>
<td>23</td>
</tr>
<tr>
<td>3.1 Data</td>
<td>23</td>
</tr>
<tr>
<td>3.2 CT images</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Reference Volume</td>
<td>24</td>
</tr>
<tr>
<td><strong>4 Study on segmentation of adipose tissue</strong></td>
<td>27</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>27</td>
</tr>
<tr>
<td>4.2 Background</td>
<td>29</td>
</tr>
<tr>
<td>4.2.1 MK2014v1</td>
<td>29</td>
</tr>
<tr>
<td>4.2.2 MK2014v2</td>
<td>33</td>
</tr>
<tr>
<td>4.3 Methods</td>
<td>35</td>
</tr>
</tbody>
</table>
5 Study on segmentation of bones
  5.1 Introduction ........................................... 47
  5.2 Background ............................................. 47
     5.2.1 MK2014v1 ........................................ 49
     5.2.2 MK2014v2 ........................................ 49
  5.3 Methods ................................................. 50
     5.3.1 B2: Extension of MK2014v2 to 3D .................. 51
     5.3.2 B3 algorithm ...................................... 52
     5.3.3 Hole filling in 3D .................................. 53
  5.4 Results and discussion .................................. 53
     5.4.1 Comparison of B2 and MK2014v2 .................. 53
     5.4.2 Comparison between B2 and B3 .................... 57
  5.5 Conclusion .............................................. 62
  5.6 Future work ............................................. 62

6 Study on atlas registration
  6.1 Introduction ........................................... 63
  6.2 Background ............................................. 63
     6.2.1 MK2014v1 ........................................ 64
     6.2.2 Deficiencies of MK2014v1 ......................... 64
     6.2.3 MK2014v2 ........................................ 65
  6.3 Methods ................................................. 66
     6.3.1 Z range ........................................... 67
     6.3.2 Preprocessing step ............................... 67
     6.3.3 RL algorithm ..................................... 69
     6.3.4 RN algorithm ................................... 71
     6.3.5 RLN algorithm .................................. 71
  6.4 Data ..................................................... 71
     6.4.1 Atlas ............................................. 71
     6.4.2 Quadrature filters ............................... 72
  6.5 Results and discussion .................................. 72
     6.5.1 MK2014v2 ........................................ 72
     6.5.2 Linear registration ............................... 73
     6.5.3 Bone-to-bone registration ......................... 74
     6.5.4 Scales in RLN algorithm ......................... 82
     6.5.5 Hole-volume registrations ......................... 83
  6.6 Conclusions ............................................ 84
  6.7 Future work ............................................ 85
## The JJ2016 Segmentation Algorithm

7 The JJ2016 segmentation algorithm 87

7.1 Description of the algorithm 87

7.1.1 Transfer of segmented prostate and rectum 88

7.1.2 Resolving of tissue conflicts 88

7.1.3 Adipose tissue: remove partial volume effect 89

7.2 Results and discussion 89

7.3 Conclusion 91

7.4 Future work 92

### Appendix

A Appendix A 95

B Appendix B 99

Bibliography 109
## Notation

### Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{z}$</td>
<td>Complex conjugate</td>
</tr>
<tr>
<td>$a \ast b$</td>
<td>Convolve a by b</td>
</tr>
<tr>
<td>$A^{-1}$</td>
<td>Inverse of a matrix or function</td>
</tr>
<tr>
<td>$a^{(t)}$</td>
<td>Variable $a$ in the iteration $t$</td>
</tr>
<tr>
<td>$a \leftarrow a + b$</td>
<td>Assign the values of $a + b$ to the variable $a$</td>
</tr>
</tbody>
</table>

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF</td>
<td>Cumulative distribution function</td>
</tr>
<tr>
<td>CMIV</td>
<td>Center for Medical Imaging and Visualization</td>
</tr>
<tr>
<td>CT</td>
<td>Computed tomography</td>
</tr>
<tr>
<td>DECT</td>
<td>Dual-energy computed tomography</td>
</tr>
<tr>
<td>FBP</td>
<td>Filtered backprojection</td>
</tr>
<tr>
<td>HM</td>
<td>Histogram matching</td>
</tr>
<tr>
<td>HU</td>
<td>Hounsfield Units</td>
</tr>
<tr>
<td>IU</td>
<td>Image unit</td>
</tr>
<tr>
<td>LMM</td>
<td>Local maximum mask</td>
</tr>
<tr>
<td>MIP</td>
<td>Maximum intensity projections</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnet resonance imaging</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>PET</td>
<td>Positron emission tomography</td>
</tr>
<tr>
<td>RG</td>
<td>Region growing</td>
</tr>
<tr>
<td>RTP</td>
<td>Radiation treatment plan</td>
</tr>
<tr>
<td>TS</td>
<td>Threshold segmentation</td>
</tr>
</tbody>
</table>
Cancer is a group of diseases in which abnormal cells have lost their ability to regulate their growth and start to grow uncontrolled with the potential to spread to other parts of the body [Cancer, 2016].

Depending on the speed of growth and the location of the tumor cancer may lead to a significant reduction of a person’s lifespan and deterioration of the person’s health. The worldwide second most common type of cancer for men is prostate cancer. One common treatment method for prostate cancer is brachytherapy [Prostate cancer, 2016]. In brachytherapy sources of radiation are inserted in the cancerous tissue in order to kill the cancer cells or slow down their growth [Radiation therapy, 2016].

In order to effectively target the cancerous cells without exposing the healthy surrounding tissue to a high dose of radiation, an oncologist sets up a radiation treatment plan (RTP). The first step in setting up the RTP is to locate the tumor. This is done using computed tomography (CT) or other imaging techniques like magnetic resonance imaging (MRI), positron emission tomography (PET) and ultrasound. A radiologist delineates the cancerous areas and important organs like the prostate, rectum and bladder in the image data by manual segmentation. Those are thereafter used by the treatment planing software in order to calculate the optimal positioning of the radioactive sources.

Most hospitals still use the TG-43 formalism introduced in 1995, which became a worldwide standard for the calculation of radiation dose in branchytherapy. In this formalism the patient dose is approximated by superpositioning precalculated single-source dose-distributions in water on the positions of the radioactive sources, taking the geometry and the composition of different radiation sources into account. A large drawback of the formalism is that it does not take the in-
homogeneities of the patient’s body into account; spatial dose distributions are derived for a homogeneous water phantom. This can result in large inaccuracies, mainly for radiation sources with low energy (< 50 keV), when the tissue composition of the patient or geometry differs significantly from that of the water phantom [Verhaegen and Beaulieu, 2013, Landry, 2014].

During recent years, the branchytherapy community has put a lot of effort into the development of new algorithms that take the tissue composition of the patient into account in order to create patient-specific treatment plans. One project that aims at improving the characterization of the patient’s tissue is DIRA. DIRA is an iterative CT reconstruction algorithm developed at the Medical Radiation Physics and the Center for Medical Imaging Science and Visualization (CMIV) at Linköpings University, which uses dual-energy computed tomography (DECT) data to decompose patient tissues into two or three base components [Malusek et al., 2014]. A more detailed description can be found in the next chapters.

1.1 Computed tomography

X-ray computed axial transmission tomography, often simply called computed tomography, is a visualization technique for creating cross-sectional images of a patient. The patient is placed on a table of the CT scanner. During the CT scan, the table is slowly moved through a large ring consisting of a rotating x-ray source and a detector array. When the emitted x-rays pass through the patient they are attenuated by the different tissues in the body. On the opposite side of the x-ray source the attenuated x-rays are measured by the detector array. From these measured values can thereafter projections of the body be obtained, see Figure 1.1.

![Figure 1.1: Illustration of a CT scanner. The x-ray source and the detector array rotate around the patient in the same time as the patient is move through the scanner by the CT table (see arrows). A reconstruction algorithm is thereafter used to obtain CT images from the measured projections.](image-url)
A cross-sectional image is created from the projections by using a CT reconstruction algorithm. The most well known reconstruction algorithm is the filtered backprojection (FBP). This algorithm is currently being replaced with iterative reconstruction algorithms as they typically result in better image quality [Dendy and Heaton, 2011]. The principle of dual-energy CT is the same as of single-energy CT, the difference is that scans are obtained at two different tube voltages instead of just one.

The attenuation of the x-ray beam depends on the thickness, the density and the material composition of the imaged object as well as the energy of the photons. The object’s attenuation can be represented by the linear attenuation coefficient, $\mu$ [Dendy and Heaton, 2011]. In order to reduce the dependence of the attenuation coefficient on the photon energy, a CT image is represented using Hounsfield values (CT numbers) defined as

$$H = 1000 \frac{\mu_{\text{obj}} - \mu_{\text{w}}}{\mu_{\text{w}}} ,$$

where $\mu_{\text{obj}}$ and $\mu_{\text{w}}$ are the linear attenuation coefficients of the object and water, respectively [Kalender, 2011]. Typical Hounsfield values of air, water, fat and bone can be found in table 1.1 [Dendy and Heaton, 2011].

**Table 1.1: Typical CT numbers for selected tissues.**

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Range (Hounsfield Units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air</td>
<td>$-1000$</td>
</tr>
<tr>
<td>Lung</td>
<td>$-200$ to $-500$</td>
</tr>
<tr>
<td>Fat</td>
<td>$-200$ to $-50$</td>
</tr>
<tr>
<td>Water</td>
<td>$0$</td>
</tr>
<tr>
<td>Muscle</td>
<td>$+25$ to $+40$</td>
</tr>
<tr>
<td>Bone</td>
<td>$+200$ to $+1000$</td>
</tr>
</tbody>
</table>

### 1.2 DIRA

DIRA is a model-based iterative reconstruction algorithm in DECT that decomposes tissues into base components. Each voxel is classified into a set of predefined tissue types, whose material composition is determined by calculating mass fractions of doublet or triplet base materials. Examples of material doublets and triplets are [bone, bone marrow] and [lipid, protein, water], respectively. The algorithm can use any number of doublets and triplets. A flowchart of DIRA can be seen in Figure 1.2.
The main aim of DIRA is to improve radiation therapy dose planning by creating a volume showing elemental composition of the patient. Additionally, DIRA improves the image quality by removing beam hardening artifacts. DIRA is a fully automatic algorithm that after initialization does not need any further user interaction. Each iteration starts by reconstructing the measured data by filtered backprojection. After the reconstruction, the slices are segmented into different tissues followed by classification of the voxels. In this step the mass fractions are updated. Before starting with the next iteration the measured polyenergetic projections are updated by a correction term. This correction term is calculated as the difference between the simulated monoenergetic and polyenergetic projections of the classified volume. By this the polyenergetic projection are iteratively converted into monoenergetic projections. More detailed descriptions of DIRA are found in [Magnusson et al., 2011] and [Malusek et al., 2014].

The source code for DIRA is under a MIT license and can be downloaded from GitHub using the following link: https://github.com/AlexandrMalusek/cmiv-dira.

1.3 Image segmentation

The objective of medical image segmentation is to select certain tissues or structures in a medical image. The segmentation can be done manually, semi-automatically or fully automatically, see Figure 1.3 for an example.
When the segmentation is done manually a person (often a radiologist or some other professional) selects the areas of interest by using anatomical knowledge as well as the content in the images. This is a very time consuming task and results differ between persons making the segmentation.

The most common approach is to use a semi-automated segmentation where an operator sets the initial conditions, e.g. the starting parameters. This allows the operator to include his knowledge (experience). The main advantage of this approach is the high control over the segmentation while the speed of the segmentation is significantly improved compared to a purely manual segmentation.

An example of a fully automated segmentation algorithm is the MK2014 which was developed to segment bones, adipose tissue, the gluteus maximus muscles and the prostate in CT images of the pelvic region. The MK2014 was developed by Martin Kadell for the tissue segmentation in DIRA.
The original MK2014 will from now on be referred to as MK2014 version 1 (briefly MK2014v1). An improved version of the algorithm (MK2014v2) was developed for the conference paper [Kardell et al., 2016].

1.4 Aim

The MK2014v1 algorithm was developed by Martin Kardell with the aim to combine different segmentation methods into an automatic tissue segmentation algorithm for DIRA. The aim of this thesis work is to extend the MK2014v1 algorithm to 3D as well as to improve its segmentation results. The main objectives of the thesis are:

- Extend the MK2014v1 algorithm to 3D. The extended algorithm should not require any user interaction during the segmentation.
- Improve the extended MK2014v1 algorithm by using methods that perform well in 3D. This may be needed since adding a third dimension will make it necessary for the algorithm to handle new challenges.

1.5 Thesis Outline

This thesis consists of seven chapters. Chapter 2 provides the theoretical background of the algorithms used in this thesis work. Chapter 3 describes data sets used for testing. The main part of the thesis are Chapters 4, 5 and 6. These are structured as three separate studies (investigations) with their own introduction, background section, methods section, result and discussion section and conclusion. Chapter 4 compares four algorithms for the segmentation of adipose tissue, Chapter 5 compares three algorithms for the segmentation of bones and Chapter 6 compares three algorithms for atlas-segmentation.

The best-performing algorithms from the three studies are combined to the JJ2016 algorithm in Chapter 7. This chapter shows the results from the JJ2016 and contains the final discussion and conclusion of the thesis.
2.1 Threshold segmentation

Threshold segmentation is a pixel-based segmentation technique where each pixel is tested if it has a higher or lower intensity than a threshold value $t$. This results in a binary image where pixels fulfilling the condition are set to one and remaining pixels are set to zero [Gonzalez and Woods, 2002]. Figure 2.1 shows the segmentation of the compact bone; all voxels greater than 149 Hounsfield units (HU) are regarded as compact bone.

\[ \begin{align*}
\text{Figure 2.1: Segmentation of the compact bone by threshold segmentation.} \\
\text{(a) Slice of data set 1 (see Section 3).} \\
\text{(b) The resulting binary mask for a threshold of 150 HU.}
\end{align*} \]
2.2 Otsu’s method

Otsu’s method is an algorithm to automate the selection of a threshold. The basic Otsu’s method splits the histogram of pixel intensities into two classes. The assumption made by Otsu’s method is that the intensity values of the classes come from two distributions, one for each of the classes, and that these distributions are clearly separable by finding the threshold between them. A histogram of a CT volume can be seen in Figure 2.2.

![Histogram of a CT volume](image)

*Figure 2.2: A histogram of a CT volume of the pelvic region. The peak at -1000 HU corresponds to the air in the CT volume. The tissues of the patient are located above -300 HU in the histogram. The distributions with the centers at -105 HU and 26 HU correspond respectively to the adipose tissue and a combination of muscular tissue, organs and bones.*

In order to find the threshold that separates the distributions, Otsu’s method iterates through all possible thresholds, forming the within class variance for each of them. The threshold with the lowest within class variance is considered the optimal threshold. The within class variance is formed as

\[
\sigma_W^2 = W_1 \cdot \sigma_1^2 + W_2 \cdot \sigma_2^2, \tag{2.1}
\]

\[
W_1 = \frac{n_1}{N}, \quad W_2 = \frac{n_2}{N}, \tag{2.2}
\]

where \(\sigma_1\) and \(\sigma_2\) are the standard deviations of the classes, \(n_1\) and \(n_2\) are the numbers of pixels of the classes, and \(N\) is the total number of pixels. [Otsu, 1975]

2.3 Seeded Region Growing

Region growing is a region based image segmentation method that is build on the assumption that pixels in a region have similar characteristics. The idea is to grow a set of start points, also called seed points, by including all surrounding pixels that fulfill a predefined condition. If the condition is fulfilled, the pixel is included into the seed region and acts as a seed point to its neighbors. The region
growing stops when all neighboring pixels fulfilling the condition are included in the seed region. An example of the condition is that a pixel needs to be in a certain intensity range [Gonzalez and Woods, 2002]. The method can easily be extended from 2D to 3D.

### 2.3.1 Neighborhood

There are different kinds of neighborhoods that can be used when performing region growing. The most common ones for 2D and 3D are shown in Figures 2.3 and 2.4.

![4 and 8 connectivity](image1.png)

*Figure 2.3: Illustration of 4 and 8 connectivity. The center point is shown in darker gray than the neighboring pixels.*

![6, 18 and 26 connectivity](image2.png)

*Figure 2.4: Illustration of 6, 18 and 26 connectivity. The center point is shown in darker gray than the neighboring pixels.*

### 2.3.2 Morphological operations

The binary images and volumes produced by the segmentation algorithms can be modified by using morphological operations. The two most basic operations are dilation and erosion, which result in the expansion and in the shrinking of the binary regions, respectively. How the region is expanded or shrunk is defined by a structure element, which in itself is a binary image or volume.

Often are dilation and erosion combined into the operations opening and closing. Opening is the erosion of a binary image or volume followed by the same amount of dilation. The result is the removal of small binary regions and the connections between such regions.
Closing is the dilation of a binary image followed by the same amount of erosion and is typically performed in order to remove small holes in binary regions. Those operations are described in detail in [Haralick et al., 1987] and in [Gonzalez and Woods, 2002].

### 2.4 Histogram equalization

Histogram equalization is an image enhancement technique in which the gray values (intensities) of an input image are transformed so that the resulting intensities are equally distributed over the intensity range, i.e. the image intensity probability density function (PDF) is uniform. Since the gray values of an image are discrete, it is often not possible to redistribute the gray values in such way that the histogram gets perfectly uniform; the new gray values of the image can however span the entire intensity range and the image contrast is increased for the gray values appearing most frequently in the image.

In histogram equalization, a gray value of the input image, $r_k$, is transformed to the gray value, $s_k$, of the output image as

$$s_k = T(r) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, \quad k = 0, 1, 2, ..., L-1,$$

where $T(r)$ is the transformation, $p_r(r_j)$ is the PDF of $r$, $n_j$ is the number of pixels with the gray value $r_j$, $n$ is the total number of pixels and $L-1$ is maximum amount of gray levels in the gray range of the input image [Gonzalez and Woods, 2002].

### 2.5 Histogram matching

In histogram matching, also called histogram specification the intensities of an image are changed so that the cumulative distribution function (CDF) matches a user specified CDF. This enables the user to customize the contrast of the image.

Assume that the gray values of the regarded images can be seen as continuous random variables with continuous PDFs. Also, assume that all the transfer functions and their inverse used in this thought experiment are single valued. For this special case, a histogram equalization of an image always results in the same PDF, $p_s(s)$, independent of initial PDF of the image. This means that when two different images with the exact same image content, but with different PDFs are histogram equalized, they would get the same gray values. Let transfer function $T(r)$ be the histogram equalization of the continuous gray values $r$ to the continuous gray values $s$,

$$s = T(r) = \int_{0}^{r} p_r(w)dw$$

(2.4)
A second histogram equalization could be used in order to transform the gray values \( z \) with the user specified PDF, \( p_z(z) \), to the same gray values \( s \),

\[
s = G(z) = \int_0^z p_z(w) \, dw. \tag{2.5}
\]

Since both \( T(r) \) and \( G(r) \) result in the gray values \( s \), it is possible to transform the gray values \( r \) to the gray values \( z \),

\[
z = G^{-1}(T(r)), \tag{2.6}
\]

under the condition that \( G(z) \) is invertible. The result is a histogram matching of the gray values \( r \) to the gray values \( z \).

The same histogram matching can also be performed for discrete gray values, using (2.3),

\[
s_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i), \quad k = 0, 1, 2, \ldots, L - 1, \tag{2.7}
\]

and

\[
z_k = G^{-1}(T(r_k)), \quad k = 0, 1, 2, \ldots, L - 1. \tag{2.8}
\]

which then, however, only results in an approximation of the specified histogram \( p_z(z) \) [Gonzalez and Woods, 2002].

### 2.6 Atlas Segmentation

Atlas segmentation is a segmentation technique where a manually segmented or artificially created template image or template volume, a so called atlas, is fitted onto the reference data (i.e. a CT image or CT volume) in order to classify it. Thereby prior knowledge and knowledge about the anatomy can be included into the segmentation, which allows atlas segmentation to segment organs that are challenging to segment with conventional image segmentation techniques.

The transformation of the atlas to the reference volume is done by an image technique called image registration. Different registration algorithms are classified by the type of transformation model they use. Two commonly used models are the linear registration, where the same linear transformation is applied to the entire volume, and non-linear registration, which transforms the atlas locally, resulting in a unique transformation for each pixel/voxel [Eklund et al., 2010a].

In the following section is the concept of linear registration described using the image intensity. In later sections, the image intensity is changed to the local phase of quadrature filter responses, which makes the registration more robust. The sections describing the linear registration are based on [Eklund et al., 2010b], [Forsberg et al., 2014] and [Svensson et al., 2008].

Thereafter follows a description of the Morphon, which is a non-linear registra-
tion. The section describing the Morphon is based on the following three articles: [Forsberg et al., 2011], [Knutsson and Andersson, 2005] and [Forsberg et al., 2014].

2.6.1 Linear image registration

Linear image registration is an imaging technique that fits a target volume to a reference volume by a linear transformation. The implementation used in this thesis is based on the discretized optical flow equation [Eklund et al., 2014],

\[ \nabla I^T \mathbf{v} - \Delta I = 0, \] (2.9)

where \( \Delta I \) is the difference in intensities between the target volume \( I_{tar} \) and the reference volume \( I_{ref} \) and \( \nabla I \) is the image gradient. The motion vector \( \mathbf{v} \) describes the wanted transformation for fitting the target volume to the reference volume. The original optical flow equation [Horn and Schunck, 1981] is valid in the continuous space, where there is an unambiguous transformation between the target and the reference volume. In this case the gradient is an estimate of the movement between the volumes. This is, however, far from true in the discrete case, where the two volumes may differ significantly from each other in shape and size. The relation between the volumes can be approximated when both volumes are downsampled to a coarse scale.

The aperture problem is, however, preventing the displacement field from being calculated directly from (2.9). The aperture problem states that, for a point laying on a line only the motion component perpendicular to the line can be estimated, since the component parallel to the line may result in multiple solutions. Thereby is \( \mathbf{v} \) unknown in one or several directions for many of the voxels. One way to solve the equation system is to reformulate the optical flow equation as a least square minimization problem over the motion vector \( \mathbf{v} \) and to minimize the error for all voxels in the volumes at once

\[ \epsilon^2 = \sum_j (\nabla I(x_j)^T \mathbf{v} - \Delta I(x_j))^2, \] (2.10)

where \( x_j \) is the coordinate of the voxel \( j \). Solving (2.10) for \( \mathbf{v} \) would lead to the best fit of the target volume to the reference volume by a translation. In order to allow other transformations than translation, \( \mathbf{v} \) is separated into a base matrix \( \mathbf{B} \), describing the allowed transformations, and a parameter vector \( \mathbf{p} \), describing the size of the transformation. Below, \( \mathbf{v} \) is separated into a base matrix and parameter vector for translation (2.11), affine transformation, (2.12), and rigid transformation (2.13).

\[ \mathbf{v} = \mathbf{Bp} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \mathbf{p} \] (2.11)
\[ \mathbf{v}(\mathbf{x}) = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \\ p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x & y & z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & x & y & z & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & x & y & z \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{12} \end{bmatrix}, \]

(2.12)

\[ \mathbf{v}(\mathbf{x}) = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} 0 & -p_4 & p_5 \\ p_4 & 0 & -p_6 \\ -p_5 & p_6 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -y & z & 0 \\ 0 & 1 & 0 & x & 0 & -z \\ 0 & 0 & 1 & 0 & -x & y \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_6 \end{bmatrix}, \]

(2.13)

The parameter vectors for the affine transformation contains 3 parameters describing the translation and 9 parameters describing the amount of scaling, rotation and skewing. The displacement of a voxel due to scaling, rotation and skewing is given by the product between the parameter vector and the voxel coordinates (see base matrix in (2.12)) [Hemmendorff et al., 2002]. For the base matrix of the rigid transformation the parameters \( p_4 \) to \( p_6 \) describe a simple model of a rotation. This model is valid as long as the following conditions are fulfilled: (i) the rotation angle is small and (ii) the rotation is only large around one of the euclidean angles [Larsson, 2010]. Inserting \( \mathbf{v} = \mathbf{B}(\mathbf{x}_j)\mathbf{p} \) into (2.10) gives the following equation,

\[ \varepsilon^2 = \sum_j (\nabla I(\mathbf{x}_j)^T \mathbf{B}(\mathbf{x}_j)\mathbf{p} - \Delta I(\mathbf{x}_j))^2. \]

(2.14)

\( \varepsilon^2 \) can be minimized by setting the first order derivative to zero and solving it for \( \mathbf{p} \),

\[ \frac{\partial \varepsilon^2}{\partial \mathbf{p}} = 2 \sum_j \mathbf{B}^T(\mathbf{x}_j) \nabla I(\mathbf{x}_j)(\nabla I^T(\mathbf{x}_j)\mathbf{B}(\mathbf{x}_j)\mathbf{p} - \Delta I(\mathbf{x}_j)) = 0, \]

(2.15)

\[ \sum_j \mathbf{B}^T(\mathbf{x}_j) \nabla I(\mathbf{x}_j) \nabla I^T(\mathbf{x}_j)\mathbf{B}(\mathbf{x}_j) \mathbf{p} = \sum_j \mathbf{B}^T(\mathbf{x}_j) \nabla I(\mathbf{x}_j)\Delta I(\mathbf{x}_j), \]

(2.16)

which gives the solution,
\[ \mathbf{p} = \mathbf{A}^{-1} \mathbf{h}. \]  \hspace{1cm} (2.17)

To improve the transformation described by \( \mathbf{p} \) and thereby make \( \varepsilon \) even smaller, the algorithm is often run over several iterations, where the calculated \( \mathbf{p} \) is added to the parameter vector describing the total movement \( \mathbf{p}_{tot} \) in each iteration \( t \), see (2.18).

\[ \mathbf{p}_{tot}^{(t)} = \mathbf{p}_{tot}^{(t-1)} + \mathbf{p}^{(t)}, \quad t = 1, 2, \ldots \]  \hspace{1cm} (2.18)

**Local phase**

The robustness of the registration algorithm can be increased by using the local phase of quadrature filters instead of using the image intensity directly. The advantages of the local phase are that the local phase varies more smoothly than the image intensity and that it is invariant to changes in intensity as long as the image structures (lines and edges in the data) are the same in both volumes.

A quadrature filter is a complex valued filter, where the real part of the filter is a line detector, see Figure 2.5a, and the imaginary part of the filter is an edge detector, see Figure 2.5b, in the spatial domain. Note that a quadrature filter is designed to detect lines and edges in one direction and that several filters, each with a different filter direction, are needed to be able to detect lines and edges with different orientations. Figure 2.5 shows one such filter. The concept of the local phase of the quadrature filters is illustrated in Figure 2.6.

*Figure 2.5:* (a) A quadrature filter \( f \) consists of a line detector \( \text{real}(f) \) and (b) an edge detector \( \text{imag}(f) \) in the spatial domain. (c) Image of the quadrature filter in the Fourier domain, \( F \).
Figure 2.6: The concept of the local phase. If the filter response of the quadrature filter only has a real part the detected image structure corresponds to a line and when it only has an imaginary component to an edge. Since the phase only gives the type of structure the amplitude of the filter response is needed in order to determine the relevance of the structure. (Image source: [Eklund et al., 2014])

In this thesis log-normal quadrature filters are used. The log-normal quadrature filters $F_k(u)$ are in the Fourier domain a combination between a radial function $R(||u||)$ and a directional function $D(u)$, where $u$ is the frequency in the Fourier domain. The quadrature function is formulated as

$$F_k(u) = R(||u||)D_k(u), \quad (2.19)$$

$$R(||u||) = e^{C \ln^2 \left( \frac{||u||}{u_0} \right)}, \quad C = \frac{-4}{B_0^2 \ln(2)}, \quad (2.20)$$

where $B_0$ is the relative bandwidth in octaves and $u_0$ is the central frequency. The directional function is only dependent on the image frequency as well as the direction $\mathbf{n}_k$ of filter $k$. Moreover

$$D_k(u) = \begin{cases} (u^T\mathbf{n}_k)^2, & u^T\mathbf{n}_k > 0, \\ 0, & otherwise. \end{cases} \quad (2.21)$$

The filter response is

$$q_k = f_k * I. \quad (2.22)$$
The magnitude, $A_k$, of the filter response, $A_k = |q_k|$, gives the phase invariant signal intensity for the found structure while the phase, $\psi_k = \arg(q_k)$, of the filter response describes whether the structure is more similar a line or an edge, i.e.

$$q_k = A_k \cdot (\cos(\psi_k) + i \sin(\psi_k)).$$

A more detailed description on the local phase of quadrature filters is found in [Granlund and Knutsson, 1995].

**Linear registration using local phase**

A few changes need to be made in the linear registration algorithm in order to change the algorithm from using the image intensity to use the local phase. The first step is to calculate the filter responses for the target volume and the reference volume.

$$q_{\text{tar},k} = I_{\text{tar}} \ast f_k$$
$$q_{\text{ref},k} = I_{\text{ref}} \ast f_k$$

Three different quadrature filters $f_k$ are applied, $k = 1, 2, 3$, for finding the edges and lines in $x$, $y$ and $z$ direction. The image intensity in the optical flow equation can simply be changed to the local phase,

$$\nabla \psi^T v - \Delta \psi = 0,$$

where $\Delta \psi$ is the difference in local phase between the volumes and $\nabla \psi$ is the gradient of the local phase. Those are expressed as

$$\Delta \psi_k = \psi_{\text{tar},k} - \psi_{\text{ref},k} = \arg(q_{\text{ref},k} q_{\text{tar},k}^*),$$

$$\nabla \psi = \begin{bmatrix} \nabla_x \psi \\ \nabla_y \psi \\ \nabla_z \psi \end{bmatrix} = \begin{bmatrix} \arg(q_{\text{ref},x+} q_{\text{ref},x-}^* + q_{\text{tar},x+} q_{\text{tar},x-}^*) \\ \arg(q_{\text{ref},y+} q_{\text{ref},y-}^* + q_{\text{tar},y+} q_{\text{tar},y-}^*) \\ \arg(q_{\text{ref},z+} q_{\text{ref},z-}^* + q_{\text{tar},z+} q_{\text{tar},z-}^*) \end{bmatrix},$$

where

$$q_{x+} = q(x + 1, y, z), \quad q_{x-} = q(x - 1, y, z),$$
$$q_{y+} = q(x, y + 1, z), \quad q_{y-} = q(x, y - 1, z),$$
$$q_{z+} = q(x, y, z + 1), \quad q_{z-} = q(x, y, z - 1),$$

(2.29)
2.6 Atlas Segmentation

and where the symbol $\ast$ is used for expressing the complex conjugate. Since the local phase is independent of the image intensity all image structures in the volumes get the same importance. To differentiate between filter responses from structures with a high amplitude (strong visible line or edge), with low amplitude and image noise, a certainty measurement $c$ is introduced,

$$c = \sqrt{|q_{tar} \cdot q_{ref}|} \cdot \cos\left(\frac{\Delta \phi}{2}\right),$$

(2.30)

where $\sqrt{|q_{tar} \cdot q_{ref}|}$ ensures that the filter response of the structure has a high amplitude in both the target and the reference volume. The second part, $\cos\left(\frac{\Delta \phi}{2}\right)^2$, compares the phase of the two volumes, where two opposing structure, i.e. a dark line and a bright line, give a small result near zero and two similar structures a result near one. By calculating $c$ for each of the filters, $c$ can be used as a weight between the different filter responses.

Including the local phase and the certainty measurement $c$ in (2.14) gives the following least square problem,

$$\varepsilon^2 = \sum_k \sum_j c(x_j, k)(\nabla \phi_k(x_j)^T B(x_j)p - \Delta \phi(x_j, k))^2.$$  

(2.31)

Solving (2.31) in the same manner as (2.14) results in

$$\sum_k \sum_j c(x_j, k)B^T(x_j) \nabla \phi(x_j, k) \nabla \phi^T(x_j, k)B(x_j)p = \sum_k \sum_j c(x_j, k)B^T(x_j) \nabla \phi(x_j, k)\Delta \phi(x_j, k)$$

$$\underbrace{A}_{h} \underbrace{p}_{h} = \sum_k \sum_j c(x_j, k)B^T(x_j) \nabla \phi(x_j, k)\Delta \phi(x_j, k)$$

(2.32)

Consequently

$$p = A^{-1}h.$$  

(2.17)

**Different scales**

When the algorithm needs to find a larger transformation the algorithm is normally run over several scales, starting on a coarse scale where the target and reference volume are strongly downsampled. Finding a good initial solution on the coarse scale has two advantages: (i) the much lower number of voxels makes the iterations of the algorithm much faster on the coarse scales than on the finer scales and (ii) that a translation on the coarse scales correspond to a much larger translation on the finer scales resulting in a significant speed up of the algorithm.

In this thesis a downsampling of factor 2 is used. This means that the volumes
on scale 0, which is the original scale, were downsampled by $2^s$, where $s$ is the scale of the image. An illustration of the downsampling of a CT image is shown in Figure 2.7.

\[\text{Figure 2.7: A Gaussian scale pyramid illustrating the downsampling of one CT image from the resolution } 512 \times 512 \text{ on scale 0 (the original scale) to } 16 \times 16 \text{ on scale 5. The reference and the target volume were also downsampled in z-direction.}\]

To get the correct translation when going from a coarser scale to a finer scale the first three parameters need to be multiplied by the factor of the downsampling (here a factor of 2). Before the downsampling of the volumes, the volumes need to be smoothed in order to avoid aliasing effects. Here this is done by a Gaussian filter [Chan et al., 2001].

The pseudo code in Listing 2.1 shows how a linear registration can be performed on different scales.
Listing 2.1: Pseudo code for running the linear registration for different scales.

```plaintext
% Downsample the volumes to the coarsest scale
downs_tar = Downsample(tar_vol, coarsest_scale)
small_ref = Downsample(ref_vol, coarsest_scale)
x = NewCoordinateSystem(coarsest_scale)

small_tar = downs_tar

for scale = coarsest_scale to original_scale
    for iter = 1 to max_iteration(scale)
        % Calculation of the parameter vector p
        p = CalculateParameterVector(small_tar, small_ref)
p_tot = p_tot + p

        % If it is the last iteration on a scale
        if (iter == max_iteration(scale)) && (scales != 0)
            % Rescaling of the translation
            for k = 1 to 3
                p_tot(k) = 2*p_tot(k)
            end

        % Calculation of the displacement field and applying it to the downsampled target volume
        x = newCoordinateSystem(scale - 1)
v = B(x)*p_tot

        small_ref = Downsample(ref_vol, scales)
downs_tar = Downsample(tar_vol, scales)
small_tar = ApplyDisplacement(downs_tar, v)
    else
        % If it is not the last iteration, apply the transformation
        v = B(x)*p_tot
        small_tar = ApplyDisplacement(downs_tar, v)
    end
end
```

2.6.2 Non-linear registration

The Morphon is a non-linear registration algorithm that is based on the phase-based optical flow. The main difference to the linear registration, described in Section 2.6.1, is that the Morphon calculates the displacement for each voxel instead of one global displacement for the entire volume. The following equation system is solved for each voxel,

$$\varepsilon^2 = \sum_{k=1}^{6} \left[ c_k \hat{T}_{LP} (\Delta \phi_k \hat{n}_k - d_i) \right]^2,$$

(2.33)

where $\hat{T}_{LP}$ is defined in (2.35) and (2.36) and $\hat{n}_k$, $k = 1, 2, \ldots, 6$, are the directions of six quadrature filters $f_k$. The minimization of the error is done over the displacement field $d_i$. The certainty for the quadrature filters $c_k$ and local phase difference $\Delta \phi_k$ are the same as for the linear registration.

The local structure tensor represents the local orientation of the image structures (edges and lines) in the volumes. In 3D the structure tensor can be defined as

$$T = \sum_{k=1}^{6} |q_k| M_k = \sum_{k=1}^{6} |q_k| \left( \frac{5}{4} \hat{n}_k \hat{n}_k^T - \frac{1}{4} I \right),$$

(2.34)

where $M_k$ is a constant tensor associated with the quadrature filter $f_k$ and $I$ is the identity matrix [Granlund and Knutsson, 1995]. One characteristic of the local orientation is that it changes much slower than the voxel intensities. When estimating the local orientation, irregularities and noise in the data can result in that two adjacent voxels get orientation estimates that points in very different directions. To get a more smooth varying orientation estimate, the estimate is often smoothed by normalized averaging. In difference to normal filtering, normalized averaging takes the certainty of the voxel values into account, which means that voxel values with low certainty are suppressed. The normalized averaging of the structure tensor is performed as

$$\hat{T}_{LP} = \frac{(||T|| * g)}{||T|| * g},$$

(2.35)

where $g$ is a Gaussian low pass filter and $||T||$ is used as the certainty of $T$. Additionally, the structure tensor is normalized to only represent the direction of the structure before it is used for the minimization of the error $\varepsilon$,

$$\hat{T}_{LP} = \frac{T_{LP}}{||T_{LP}||}.$$

(2.36)

As for the linear registration, the least square equation of the Morphon is solved
by calculating its derivative and setting it equal to zero.

\[
\frac{\partial \varepsilon^2}{\partial \mathbf{d}_i} = -\sum_{k=1}^{6} 2 c_k^2 \hat{T}_{\text{LP}}^T \hat{T}_{\text{LP}} (\Delta \varphi_k \mathbf{n}_k - \mathbf{d}_i) = 0 \quad (2.37)
\]

To make the local displacement more stable, the left and right side of the equation system are smoothed by a Gaussian filter \( g \). Solving Equation (2.38) for \( \mathbf{d}_i \) leads to:

\[
\mathbf{d}_i = \left( g \ast \sum_{k=1}^{6} c_k^2 \hat{T}_{\text{LP}}^T \hat{T}_{\text{LP}} \right)^{-1} \left( g \ast \sum_{k=1}^{6} c_k^2 \hat{T}_{\text{LP}}^T \hat{T}_{\text{LP}} \Delta \varphi_k \mathbf{n}_k = \mathbf{A}^{-1} \mathbf{b} \right) \quad (2.38)
\]

The iterative certainty \( c_i \) of the voxel is given by the diagonal components of the 3 x 3 matrix \( \mathbf{A} \):

\[
c_i = \text{tr}(\mathbf{A}) \quad (2.39)
\]

Both \( \mathbf{d}_i \) and \( c_i \) are calculated anew in each iteration. The displacement field from each iteration is accumulated to the accumulated displacement field \( \mathbf{d}_a \). The accumulation is done by weighting between the current \( \mathbf{d}_a \) and the new displacement \( \mathbf{d}_a + \mathbf{d}_i \) using the accumulated certainty \( c_a \) and the certainty for the current iteration \( c_i \) as weights,

\[
\mathbf{d}_a \leftarrow \frac{c_a \mathbf{d}_a + c_i (\mathbf{d}_a + \mathbf{d}_i)}{c_a + c_i} \quad (2.40)
\]

Similarly \( c_a \) is given by weighting between \( c_a \) and \( c_i \) where both certainties are set to their own certainty estimates,

\[
c_a \leftarrow \frac{c_a^2 + c_i^2}{c_a + c_i} \quad (2.41)
\]

All steps described so far have been about the calculation of the displacement of each voxel separately, where it is possible that two neighboring voxels get very different displacements. To get a similar displacement for a neighborhood, \( \mathbf{d}_a \) is regularized using normalized averaging. This is done by weighting the voxels with \( c_a \) before spreading the displacement field of a voxel to its neighbors using a Gaussian filter \( g \),
The last step of each iteration is to apply the smoothed displacement field to the target volume. The deformed target volume is in turn used for the calculation of the filter responses of the target volume in the next iteration.

The Morphon is normally run over several scales, starting on a coarse scale with a high regularization. The regularization is typically reduced on the finer scales in order to make the deformation more local. Knutsson and Andersson [2005] recommended to increase the certainty $c_i$ in (2.41) when going from a coarser to a finer scale. This gives more priority to the finer scales where the fitting is adapted locally. Therefore $c_i$ is multiplied with the factor $2^{-p}$, where $p$ is the number of downsampling performed on the volumes,

$$c_a \leftarrow \frac{c_a^2 + (2^{-p}c_i)^2}{c_a + 2^{-p}c_i}. \quad (2.43)$$

### 2.7 Dice Similarity Coefficient

A commonly used evaluation metric for the evaluation of segmentation is the Dice Similarity Coefficient (DSC). It is a validation metric used to measure the spatial overlap between two binary images or volumes and results in a measurement that range between 0 (no overlap) and 1 (perfect overlap). The DSC is calculated as

$$DSC = \frac{2|A \cap B|}{|A \cap B + A \cup B|}, \quad (2.44)$$

where $A$ and $B$ are segmented volumes. The absolute value is used to symbolize the number of voxels in a volume [Dice, 1945].
3.1 Data

In this thesis work, 8 data sets consisting of stacked CT images covering approximately the same part of the pelvic region were used. Each image size was $512 \times 512$ pixels. The images were collected by a SOMATOM Definition AS+ scanner from Siemens Healthcare or by a LightSpeed Ultra scanner from GE medical systems. Data sets 7 and 8 were reconstructed by FBP and the remaining ones using iterative reconstruction, see table 3.1. The data sets are illustrated in Figures 3.1 and 3.2. The CT volumes were resampled to isotropic volumes for the sagittal images and maximum intensity projections (MIP).
Table 3.1: The age, number of reconstructed slices, voxel size, tube voltage \((U)\), reconstructed field of view (FOV), reconstruction diameter (RD), CT scanner manufacturer and image reconstruction method for each of the eight examined subjects. NA stands for not available.

<table>
<thead>
<tr>
<th>i</th>
<th>Age (years)</th>
<th>Slices</th>
<th>Voxel size ((\text{mm}^3))</th>
<th>(U) (kV)</th>
<th>FOV (mm)</th>
<th>RD (mm)</th>
<th>Manu.</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82</td>
<td>80</td>
<td>(0.76 \times 0.76 \times 2.5)</td>
<td>120</td>
<td>500</td>
<td>389</td>
<td>GE</td>
<td>Iterative</td>
</tr>
<tr>
<td>2</td>
<td>69</td>
<td>50</td>
<td>(0.89 \times 0.89 \times 3)</td>
<td>120</td>
<td>500</td>
<td>456</td>
<td>Siemens</td>
<td>Iterative</td>
</tr>
<tr>
<td>3</td>
<td>66</td>
<td>200</td>
<td>(0.70 \times 0.70 \times 1)</td>
<td>120</td>
<td>500</td>
<td>358</td>
<td>Siemens</td>
<td>Iterative</td>
</tr>
<tr>
<td>4</td>
<td>58</td>
<td>70</td>
<td>(0.79 \times 0.79 \times 2.5)</td>
<td>120</td>
<td>500</td>
<td>402</td>
<td>GE</td>
<td>Iterative</td>
</tr>
<tr>
<td>5</td>
<td>62</td>
<td>200</td>
<td>(0.87 \times 0.87 \times 1)</td>
<td>140</td>
<td>500</td>
<td>447</td>
<td>Siemens</td>
<td>Iterative</td>
</tr>
<tr>
<td>6</td>
<td>61</td>
<td>200</td>
<td>(0.88 \times 0.88 \times 1)</td>
<td>120</td>
<td>500</td>
<td>452</td>
<td>Siemens</td>
<td>Iterative</td>
</tr>
<tr>
<td>7</td>
<td>NA</td>
<td>80</td>
<td>(0.72 \times 0.72 \times 2.5)</td>
<td>120</td>
<td>500</td>
<td>370</td>
<td>GE</td>
<td>FBP</td>
</tr>
<tr>
<td>8</td>
<td>NA</td>
<td>80</td>
<td>(0.76 \times 0.76 \times 2.5)</td>
<td>120</td>
<td>500</td>
<td>387</td>
<td>GE</td>
<td>FBP</td>
</tr>
</tbody>
</table>

Data sets 5 and 7 differed from the other data sets. In data set 5, the patient had one hand on the pelvis. This is normally avoided since it results in exposing the hand to unnecessary radiation. Normal situations where the arms are kept at the side or the stomach are when the patient has a fracture in the bones of the shoulder or is subject to multiple trauma, for instance after a traffic accident. In data set 7, a contrast agent was injected into the rectum. It increased the attenuation in the rectum, which appears brighter in the images.

3.2 CT images

The CT data sets were scanned at Linköpings University Hospital and were thereafter anonymized to not include the patient information. They were provided as DICOM images together with DICOM directory files which contained information about the order of the images. In the DICOM format, the intensity values of the CT images are rescaled by a linear transformation so that they can be represented as unsigned integers. In the data files, the addition of 1024 was used. This gave air the value of 24 image units (IU) instead of the CT number -1000 and water the value of 1024 IU instead of the CT number 0. These transformed values were used for all algorithms in this thesis work.

3.3 Reference Volume

The reference volume for the histogram matching was created by rescaling the intensities from the range \([850, 1250]\) IU to the range \([0, 1250]\). The CT slices in Figures 3.1 and 3.2 are shown for the range \([850, 1250]\).
Figure 3.1: Data sets 1-4. Left: Axial CT slices of the prostate region. Middle: Sagittal slices. Right: MIP in the Anterior-Posterior direction.
Figure 3.2: The same as Figure 3.1 for data sets 5-8.
4 Study on segmentation of adipose tissue

4.1 Introduction

Adipose tissue is loose connective tissue with a high concentration of fat cells. It is primarily located beneath the skin (subcutaneous fat), between the organs (visceral fat), in small deposits between the muscles (intramuscular fat) and around the heart (cardiovascular fat), see Figure 1.3. Two types of adipose tissue are found in mammals, namely white adipose tissue and brown adipose tissue. The white adipose tissue is mainly associated with insulating the body against cold and heat, forming protective pads around the inner organs and serving as energy deposits for storing triglycerides (fats). It also secretes several hormones and signal proteins used in the metabolism and in the endocrine system [Adipose tissue, 2016, Trayhurn and Wood, 2004, Tortora and Derrickson, 2011]. Brown adipose tissue’s main function is to convert glucose and fatty acids into heat which plays an important role in keeping infants warm. The amount of brown adipose tissue decreases with age, but remains in smaller quantities throughout the lifespan of a human [Brown adipose tissue, 2016, Graja and Schulz, 2014].

The chemical composition of adipose tissue and several other tissues found in the pelvic region is shown in Table 4.1. Compared to the other tissues it has a higher content of carbon and a lower content of oxygen which makes it clearly visible in a CT image. Its lower mass density and electron density compared to the other tissues must be considered in radiation treatment planning.
Table 4.1: The elemental composition, mass density, $\rho$, and electron density, $n_0$, adult #2 of adipose tissue, adult skeletal muscle, skeleton-cortical bone adult, adult skeleton-spongiosa, adult skeleton-cartilage, adult skeleton-red marrow and adult skeleton-yellow marrow taken from ICRU [1992].

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Elemental compositions (percentage by mass)</th>
<th>$\rho$ (kg m$^{-3}$)</th>
<th>$n_0 \times 10^{26}$ (m$^{-3}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>adipose tissue</td>
<td>11.4 H, 59.8 C, 0.7 N, 27.8 O, 0.1 Na, 0.1 S, 0.1 Cl</td>
<td>950</td>
<td>3180</td>
</tr>
<tr>
<td>muscle</td>
<td>10.2 H, 14.3 C, 3.4 N, 71.0 O, 0.1 Na, 0.2 P, 0.3 S, 0.1 Cl, 0.4 K</td>
<td>1050</td>
<td>3480</td>
</tr>
<tr>
<td>cortical bone</td>
<td>3.4 H, 15.5 C, 4.2 N, 43.5 O, 0.1 Na, 0.2 Mg, 10.3 P, 0.3 S, 22.5 Ca</td>
<td>1920</td>
<td>5950</td>
</tr>
<tr>
<td>spongiosa</td>
<td>8.5 H, 40.4 C, 2.8 N, 36.7 O, 0.1 Na, 0.1 Mg, 3.4 P, 0.2 S, 0.2 Cl, 0.1 K, 7.4 Ca, 0.1 Fe</td>
<td>1180</td>
<td>3840</td>
</tr>
<tr>
<td>cartilage</td>
<td>9.6 H, 9.9 C, 2.2 N, 74.4 O, 0.5 Na, 2.2 P, 0.9 S, 0.3 Cl</td>
<td>1100</td>
<td>3620</td>
</tr>
<tr>
<td>red marrow</td>
<td>10.5 H, 41.4 C, 3.4 N, 43.9 O, 0.1 P, 0.2 S, 0.2 Cl, 0.2 K, 0.1 Fe</td>
<td>1030</td>
<td>3420</td>
</tr>
<tr>
<td>yellow marrow</td>
<td>11.5 H, 64.4 C, 0.7 N, 23.1 O, 0.1 N, 0.1 S, 0.1 Cl</td>
<td>980</td>
<td>3280</td>
</tr>
</tbody>
</table>

The accumulation of visceral adipose tissue (adipose tissue located inside the abdominal cavity surrounding the organs) has shown to increase the risk of getting diabetes, different kinds of cancers and Alzheimer disease [Obesity, 2016]. This has made automatic segmentation and separation of visceral adipose tissue and subcutaneous adipose tissue (the adipose tissue beneath the skin) to a subject undergoing intense study in the recent years.

A simple way to segment the adipose tissue is by threshold segmentation. This can either be done by applying preset upper and lower thresholds or a dynamic range derived from the image. The algorithm by Nemoto et al. [2014] uses fixed thresholds in combination with morphological operations to segment the muscular tissue, the bone, the visceral and subcutaneous fat and the air. In [Yoshizumi et al., 1999] the thresholds are calculated from the mean and the standard deviation of an operator-initialised interest region.

Makrogiannis et al. [2013] segment the adipose tissue using fuzzy C-means clustering and separate it into subcutaneous and visceral adipose tissue by active contours. In some areas the segmented visceral adipose tissue may contain food residues. Those are excluded from the visceral adipose tissue by a support vector machine trained to separate those by texture and local shape.

In the MK2014v1 algorithm, the CT image intensity is first transformed using histogram matching in order to reduce the effect of for instance tube voltage on the
image appearance. The adipose tissue is thereafter segmented using threshold segmentation. The subcutaneous and visceral adipose tissue are separated by region growing. One drawback of the MK2014v1 is the large dependence of the CT image on the reference image during the histogram matching step. This causes unwanted results for some images where parts of the air and the blankets covering the patient are included incorrectly into the segmentation or where some areas of adipose tissue are not detected. The other drawback of MK2014v1 is that it works with 2D images only.

The aim of this study is to compare the results of MK2014v2 with three proposed 3D algorithms for the segmentation of adipose tissue. The evaluation of the quality of segmentation was done by visual assessment.

4.2 Background

The following sections give a brief overview over the MK2014v1 algorithm, its problems and how some of those were addressed in the MK2014v2 algorithm.

4.2.1 MK2014v1

The MK2014v1 consist of two parts: (i) a preprocessing step which is the same for the segmentation of the bones, the adipose tissue, the gluteus maximus muscles and the prostate, and (ii) algorithms for the segmentation of each of the four tissues. The preprocessing step consists of the following steps:

1. The image intensities are rescaled from IU to the range [0, 255].
2. Histogram matching is performed to transform the intensity range of the CT image to the range of the reference image. It makes the image independent of the settings of the CT scanner.
3. The patient’s body is extracted from the CT image by threshold segmentation followed by morphological opening on the resulting mask. The mask is multiplied with the CT image to remove everything but the patient’s body.

The adipose tissue is segmented as follows:

1. An initial segmentation of the adipose tissue is done by threshold segmentation of the CT image using the range [6, 75]. Note that the threshold values depend on the reference image used by the MK2014v1 algorithm. Holes in the resulting binary mask are filled by morphological operations.
2. The largest region is selected from the binary mask for the generation of seed points. Those are created by eroding the region to points. One of the seeds is selected and used in the region growing.
3. The final segmentation is given by region growing. A neighbor of the seed point is added to the seed region if the intensity value is in a predefined
acceptance range. The acceptance range is defined as the interval
\[
m_r - v_{\text{marg}}, m_r + v_{\text{marg}},
\]
where \(m_r\) is the mean of pixel intensity values in the seed region and \(v_{\text{marg}}\) a user specified margin. If a pixel is added to the seed region, the mean value of the region is updated and the neighbors of the added pixels are tested.

4. Small holes in the final segmentation are removed by morphological closing and hole filling.

A flowchart of the MK2014v1 algorithm is shown in Figure 4.1.

For most data sets the MK2014v1 gives good results. The exception is, however, when the histogram matching does not correctly adjust the intensities of tissues in the CT image to the intensities of these tissues in the reference image. In this case, the whole segmentation becomes unpredictable. Other consequences of this deficiency are mentioned below.

In MK2014v1 the histogram matching is always performed on the entire image, which includes the patient’s body, CT table, blankets and the surrounding air. An excessive amount of the surrounding air may lead to an improper adjustment of image intensities: (i) the intensity of the blankets gets in the range of the adipose

**Figure 4.1: Flowchart of the MK2014v1 algorithm.**
tissue (see Figure 4.2a), and (ii) the intensities of the air and the adipose tissue increase and become similar. As a consequence the region growing may leak into the blanket. This decreases the mean value of the grown region and thus prevents the region growing to include the remaining adipose tissue, see Figure 4.2b.

**Figure 4.2:** When the body was small relative to the surrounding air, the MK2014v1 algorithm lead to improper segmentation of the adipose tissue as the image intensities were wrongly adjusted by the histogram matching. (a) Threshold segmentation followed by selection of the largest region resulted in incorrect inclusion of the blanket in the adipose tissue. (b) The resulting segmentation included the air in the blanket. Note: Black pixels are pixels outside the field of view (FOV), they were set to -2000 in the DICOM images. Dark gray pixels correspond to air, i.e. $\approx 24$ IU (-1000 HU).

Too small amount of the surrounding air, on the other hand, may decrease the intensity of adipose tissue. Some pixels may fall into the range of air for threshold segmentation, see Figure 4.3a. The low intensity difference between adipose tissue and air causes the region growing to leak into the surrounding air, see Figure 4.3b. As in the case with excessive amount of air, the leaking region reduces the mean intensity and this prevents its growing into some parts of the adipose tissue.
Figure 4.3: When the body was large relative to the surrounding air, the MK2014v1 algorithm undesirably lowered the intensity of the adipose tissue; some pixels even got zero intensity. (a) Threshold segmentation excluded adipose pixels with low intensity. (b) Region growing leaked out to the surrounding air because of the small difference in intensity between the adipose tissue and the air.

The original aim of the MK2014 algorithm was to segment the subcutaneous fat only. It was achieved by selecting the largest region obtained by threshold segmentation. The implementation can, however, include some visceral fat into the region of subcutaneous fat when those two regions are connected, see Figure 4.4a. This makes the algorithm unstable since a small number of pixels in the connection between the two regions may decide whether a large area of visceral fat is added or not. Region growing may also be effected by image artifacts, for instance streaks, see Figure 4.5. In Figure 4.4b a small region in the lower left corner was not included because the streaks stopped the process. A small field of view may also disconnect the upper and the lower parts of the subcutaneous fat region. The MK2014v1 algorithm then incorrectly omits the smaller region, see Figure 4.3b.
4.2 Background

Figure 4.4: Segmentation of adipose tissue by (a) threshold segmentation and (b) region growing. Both methods incorrectly included the visceral fat into the subcutaneous fat region. The region growing omitted a small area in the lower left part of the subcutaneous fat owing to streaks artifacts.

Figure 4.5: Streaks in the lower left part of the image in Figure 4.4 after threshold segmentation.

4.2.2 MK2014v2

In the MK2014v2 algorithm these problems are addressed by the following changes:

- The CT image is preprocessed before the segmentation in order to set all intensities below 500 IU (-524 HU) to 0 IU. This removes the blankets and noise in the air from the CT image.

- The pixels surrounding the patient’s body are set to 0 IU by threshold segmentation of the CT image by a low threshold (here 7 IU) followed by masking the CT image by the biggest binary object from the binary image from the threshold segmentation. To increase the difference in intensity between the voxels of the patient’s body and those surrounding it, the voxels outside the body are set to -100 IU.
• Instead of only using the binary region with the most pixels from the initial segmentation as in MK2014v1, all regions (beside very small ones) are used for finding seed points. Regions with fewer than 200 pixels are removed to avoid the region growing from leaking into the surrounding tissues.

• The seeds are generated by eroding the regions in the initial segmentation result to points. The image is divided into 16 rectangular blocks (128 pixels times 128 pixels). Two seeds are selected in each block. Each seed is grown separately and added to the final segmentation result.

• The final segmentation result is post processed using morphological closing with a disc shaped structure element of radius 7. Holes smaller than 1000 pixels are filled.

A flowchart for the MK2014v2 algorithm can be seen in Figure 4.6.

Figure 4.6: Flowchart of the MK2014v2 algorithm.
4.3 Methods

This study compared the results from four algorithms for the segmentation of adipose tissue: (i) the MK2014v2 (2D), (ii) Otsu’s method combined with threshold segmentation (3D), (iii) the MK2014v2 algorithm without region growing (3D), and (iv) the MK2014v2 algorithm with region growing (3D); the algorithms are named $A_1$, $A_2$, $A_3$ and $A_4$, see Table 4.2.

**Table 4.2:** The four evaluated adipose tissue segmentation algorithms. The methods have been abbreviated as follows: threshold segmentation (TS), histogram matching (HM) and region growing (RG).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>MK2014v2</td>
</tr>
<tr>
<td>$A_2$</td>
<td>Otsu’s method and TS</td>
</tr>
<tr>
<td>$A_3$</td>
<td>HM and TS</td>
</tr>
<tr>
<td>$A_4$</td>
<td>HM, TS and RG</td>
</tr>
</tbody>
</table>

The four algorithms were compared using the 8 data sets described in Section 3. The 2D-based $A_1$ algorithm was applied on only two images of each data set. Those were selected so that they covered a similar region as the reference and atlas images included in the MK2014. The other three algorithms were applied on the entire data sets. Data set 1 was enhanced according to Section 3.3 and used as reference volume for the histogram matching. All results were evaluated by visual assessment only since the ground truth was missing.

Details on the algorithms are described below. In Figure 4.7 the methodology of the algorithms are illustrated by a flowchart.
4.3.1 Preprocessing

In the preprocessing step all voxels outside the patient’s body are set to 0 IU. The first step consists of creating a binary volume which only contains the voxels of the patient. This is done by threshold segmentation of the CT volume by a threshold value of 600 IU (-424 HU) followed by the selection of the biggest binary object in the resulting binary volume. Holes in the binary volume are filled before the binary volume is used as a mask for removing all voxels outside the patient’s body. The air inside the patient’s body is given by the filled areas in the binary volume.

4.3.2 Histogram matching without air

The 3D histogram matching is implemented according to Section 2.5. Instead of performing the histogram matching for the entire volumes, only the voxels corresponding to the patient’s body are included in the calculation of the histograms. This makes the histogram matching more robust against variations in patient’s body size relative to the amount of air in the CT volume.

4.3.3 Description of the $A_2$, $A_3$ and $A_4$ algorithm

This section describes the main steps of the $A_2$, $A_3$ and $A_4$ algorithms. The $A_1$ algorithm has already been described in Section 4.2.2.

$A_2$ algorithm: Otsu’s method and TS

After the image preprocessing (Section 4.3.1), the $A_2$ algorithm uses the Otsu’s method to find the upper threshold for the threshold segmentation of the adipose
4.4 Results and discussion

Otsu’s method is only applied to voxels within the intensity range \([824 \text{ IU}, 1124 \text{ IU}]\) (i.e., \([-200 \text{ HU}, 100 \text{ HU}]\)). This range assures that the histogram used for Otsu’s method only includes the two peaks corresponding to the adipose tissue and the muscular tissue, organs and bones. The implementation of Otsu’s method was done according to Section 2.2.

The adipose tissue is segmented by using threshold segmentation between a fixed lower threshold of 624 IU (-400 HU) and the upper threshold value given by Otsu’s method.

**A_3 algorithm: HM and TS**

After the image preprocessing (Section 4.3.1), the \(A_3\) algorithm uses histogram matching (Section 4.3.2). To make the histogram matching independent of the body size of the patient, only the voxels of the patient’s body are used for the calculation of the look-up table. The final segmentation of the algorithm is given by threshold segmentation of the enhanced volume using two predefined thresholds.

**A_4 algorithm: HM, TS and RG**

The \(A_4\) algorithm uses the segmentation result from the \(A_3\) algorithm to generate seed points for the region growing. The seed points are created by eroding the threshold segmentation result to points, which results in a mask containing the seed points. In contrast to MK2014 version 1 and 2, none of the seed points are removed before the growing step. The region growing is performed for all seed points at the same time, growing the seeds using 26 connectivity (see Section 2.3.1). A voxel is included into the seed region and considered a seed point if its has an intensity value inside the acceptance range.

4.4 Results and discussion

The changes introduced in the MK2014v2 made the segmentation more stable in comparison to the original algorithm, see Figure 4.8. By giving the air the value of -100 IU neither the threshold segmentation nor the region growing segmented the air as adipose tissue. The blankets were successfully removed from the CT image in the preprocessing step.
Figure 4.8: Segmentation results of the MK2014v2. (a) Neither the threshold segmentation nor (b) the region growing included the air in the adipose tissue.

Much of the adipose tissue was not found by the region growing because of the limited number of seed points. One easy way to improve the result would be to increase the number of blocks for the seed selection as well as the number of seeds that are selected in each block. This would however also increase the processing time significantly, since the algorithm in its current state is growing each seed at a time.

A better way to improve the segmentation result would be to use all pixels that are eroded to points as seed points. Growing all of them at once would make the algorithm much faster since each pixel only would be segmented once and more stable since the mean is calculated for a larger number of pixels.

One problem that still remains is the dependency between the size of the patient and amount of air when performing the histogram matching, see Figure 4.9.
4.4 Results and discussion

Figure 4.9: Segmented slice of data set 7 by the MK2014v2. The patient in the CT image covers more of the image than the patient in the reference image. (a) This results in that parts of the adipose tissue get an intensity value outside of the predefined range and are therefore not identified as adipose tissue by the threshold segmentation. (b) The same happens when performing the region growing, with the difference that the not detected adipose tissue is filled in by the morphological operations in the post-processing step. In this result parts of the intramuscular adipose tissue and vascular tissue is missing, due to the limited number of seed points.

4.4.1 Adipose segmentation in 3D

The dependency between the proportion of the number of voxels corresponding to the patient and those surrounding the patient is removed in the $A_3$ and $A_4$ algorithms by excluding the air from the histogram matching. This approach improved the stability of the histogram matching and lead to good results for the threshold segmentation, see Figure 4.10.
Figure 4.10: Segmentation for a slice of data set 7. Both (a) the threshold segmentation of the $A_3$ and (b) the region growing of $A_4$ were able to segment all larger regions of the adipose tissue.

The $A_4$ has, unlike the MK2014v2, no predefined maximal number of seed points, see Figure 4.11. This has two advantages: (i) It is possible to place seed points at smaller patches of adipose tissue. Leakage at the smaller areas of adipose tissue is avoided by growing all seed points at the same time and calculating one shared mean value for all seed regions. (ii) In the Matlab implementation the neighbors are found using dilation. Dilating all seed points at the same time prevents the voxels from being segmented multiple times, reducing the computational time significantly. A higher number of seed points results in that less iterations are needed to segment all voxels fulfilling the criterion of the region growing algorithm.
4.4 Results and discussion

Figure 4.11: Comparison of segmentation results from MK2014v2 and $A_4$. (a) The MK2014v2 uses much fewer seed points; many regions of the adipose tissue are not found as a result. (b) The $A_4$ algorithm uses more seed points; the result is better.

Segmenting the adipose tissue by threshold segmentation gave about the same result as region growing; the computational load was lower. To illustrate the similarity between the results, the dice similarity coefficient between those was calculated for all 8 data sets, see Table 4.3.

Table 4.3: The dice coefficients comparing $A_3$ with $A_4$ and $A_2$ with $A_3$ for the 8 data sets.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_3$ &amp; $A_4$</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>$A_2$ &amp; $A_3$</td>
<td>0.90</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.89</td>
<td>0.97</td>
<td>0.91</td>
<td>0.92</td>
</tr>
</tbody>
</table>

The largest difference between the segmentation results was found for data set 2, where the results matched to 97%. The main difference between the results was that region growing included more voxels at the borders of the seed regions than the threshold segmentation. The threshold segmentation was able to find small regions of adipose tissue not found by the region growing, see Figure 4.12. The threshold segmentation also included voxels effected by the partial volume effect, for instance the skin-air and muscular tissue-air borders, which where about 1 voxel wide. The partial volume effect occurs when a voxel contains more than one tissue. Then the voxel gets an image intensity that corresponds to the weighted average of all tissues in the voxel.
Figure 4.12: Segmentation of the adipose tissue in data set 2. Both the threshold segmentation by the $A_3$ algorithm and the region growing by the $A_4$ algorithm result in very similar results. Voxels segmented by both algorithms are displayed in yellow, voxels only segmented by the $A_3$ algorithm in red and voxels only segmented by the $A_4$ algorithm in blue.

The difference in results between $A_3$ and $A_4$ was quite small. Thus region growing additional to threshold segmentation did not improve the segmentation results enough to make up for the additional computation time.

MK2014 thresh and Otsu

The results from the $A_2$ and the $A_3$ algorithms were mainly affected by the setting of the upper threshold for the threshold segmentation in $A_3$. Since this threshold did not differ much from the one determined by the Otsu’s method, $A_2$ and $A_3$ provided similar results, see Table 4.3; for most data sets the threshold by Otsu’s method was only slightly higher, see Figure 4.13.

For data set 7 the $A_3$ algorithm resulted in a higher upper threshold. The reason is that the patient had less adipose tissue than the patient in the reference volume and that some of the adipose tissue was excluded by the limited field of view, see Figure 4.14.
Figure 4.13: (a) A CT slice from data set 5. (b) Segmentation by the \(A_2\) and \(A_3\) algorithm. Both algorithms are segmenting the regions that clearly can be recognized as adipose tissue (yellow). Additionally the \(A_2\) algorithm segments voxels with slightly higher intensity (blue). The thresholds from the \(A_3\) algorithm could be adjusted to segment the same regions by increasing the predefined threshold. (c) and (d) show the same results but zoomed in.
Figure 4.14: (a) A CT slice from data set 7. (b) Both algorithms segment the adipose tissue as well as adipose tissue mixed with other tissues (yellow). Since the patient in the CT volume has less adipose tissue than the patient in the reference volume, the \( A_3 \) algorithm regards some muscular tissue as adipose tissue (red). (c) and (d) are zoomed in versions of (a) and (b).
A too small field of view could easily be avoided by the operator when performing the CT examination. This would, however, not handle the problem with different amount of adipose tissue between different individuals. One way would be to use different reference volumes for patients with different body sizes. This would work if separate reference volume-specific thresholds would be set for each of the reference volumes.

The $A_2$ algorithm was not influenced by the amount of voxels of the different tissues, as long as the peaks of the muscular and adipose tissue in the histogram of the CT volume were well defined. The drawback of the $A_2$ algorithm is that the operator cannot fine tune the threshold values in an easy way, which can be done for the $A_3$ algorithm.

## 4.5 Conclusions

All three 3D algorithms gave better segmentation results for the adipose tissue than the MK2014v2. The $A_4$ algorithm, which is a 3D version of the adipose tissue segmentation of the MK2014v2, was improved by excluding the air from the histogram matching and by not limiting the number of seed points. Those improvements increased the robustness of the algorithm and allowed it to segment smaller patches of adipose tissue. A disadvantage of the $A_4$ algorithm was that performing region growing after threshold segmentation did not notably improve the segmentation result while it notably increased the computation time.

The $A_2$ and $A_3$ algorithms gave very similar results to the $A_4$ algorithm, but they were much less computationally heavy. No final conclusion could be made to which of the two algorithms provided the best results. When using the $A_3$ algorithm the operator should be aware that the histogram matching is affected by proportions of the different tissues in the body. Therefore it would be necessary to have more than one reference volume to get good results for patients with different body shapes. The advantage of $A_3$ algorithm over the $A_2$ algorithm is that it is very easy for the operator to fine tune the threshold values.
5

Study on segmentation of bones

5.1 Introduction

All bones consist of an outer shell of cortical bone, also referred to as compact bone, that allows the bones to withstand physical stress and functions as a protection to the tissues inside [Compact bone, 2016]. Beneath the layer of compact bone, the less compact and more flexible trabecular bone (spongy bone, cancellous bone) is found. The trabecular bone is lighter than the compact bone and has microscopic cavities, which reduce the overall weight of the bone and give it a spongy appearance. The trabecular bone consists of a complex latticework, which is arranged along the stress lines in order to transfer the pressure applied to the bone and thereby prevent the bone from breaking [Bone morphology, 2016, Tortora and Derrickson, 2011]. Inside the trabecular bone the red and the yellow bone marrow is found. The red bone marrow is responsible for the production of the red blood cells, the white blood cells and the platelets found in the blood. Until the age of seven the entire bone marrow is red. Thereafter the red bone marrow is gradually transformed into yellow bone marrow. Typical regions where red bone marrow is found in an adult are the hip bones, the ribs, the skull, the vertebrae and the ends of the long bones. The remaining regions in the trabecular bones are filled by the yellow bone marrow, which is storing adipose tissue, giving it a yellow color [Bone marrow, 2016, Caracappa et al., 2009, Tortora and Derrickson, 2011].

The segmentation of bones was studied by a number of authors in the last decades, but the problem has not been fully solved yet. One reason is the large variety in bones, which ranges from the dense cortex of cortical bone to the much less dense spongy bone it envelops. Additional factors that complicate the task are: (i) weak or diffuse boundaries of the bone as a result of the partial volume effect;
(ii) brightening of tissue between bones close to each other as a result of the low-pass effect of the CT system; (iii) holes in the cortical bone for the blood vessels [Sebastian et al., 1998]. During the last years several new automated algorithms for the segmentation of the pelvic and femur bones have been developed. Most of these algorithms are either combining several simple methods in order to overcome the drawbacks of the individual algorithms or include prior-knowledge by applying a model of the segmented organ.

Typical methods that use a model to segment the bones are active shape models [Wu et al., 2012], statistical shape models [Yokota et al., 2009] and atlas segmentation [Pettersson et al., 2006, 2008]. In active and statistical shape models, the model is created from a training set, which describes the allowed deformations of the model. After the initialization the model is iteratively adapted to the image content. Crucial for both methods is a large training set to allow an accurate segmentation of the bone.

Two algorithms for automated bone segmentation in the pelvic region that do not use a trained model were published by Vasilache and Najarian [2008] and Krčah et al. [2011]. The algorithm by Vasilache and Najarian [2008] segments the compact bone by a seeded region growing algorithm in 2D. The region growing algorithm handles regions with weak contrast between the bone and the other tissues by using a criterion that demands that a pixel needs to have a high image intensity relative to the other pixels in its neighborhood in order to be classified as bone. Krčah et al. [2011] segmented the femur bone by first enhancing the boarders of the data by a boundary enhancement filter before the bone was segmented using graph-cut.

The MK2014 handles the low contrast by transforming the values of the CT image to predefined ranges using histogram matching followed by the segmentation of the bone using threshold segmentation and region growing. Omitted bone tissue is included by a hole filling step that merges bone parts that are close to each other.

This study gives a short overview over the updates done to the bone segmentation of the MK2014v1 algorithm and how those improved the results of the algorithm. The main goal of this study is to develop and evaluate two 3D algorithms for the automated segmentation of bone. The first algorithm is based on the bone segmentation of the MK2014 and the second algorithm is inspired by the segmentation algorithm of Vasilache and Najarian [2008]. The segmentation results of both algorithms are compared to those of the MK2014v2 algorithm. Segmentation of bone using atlas segmentation is part of the segmentation of rectum and prostate in Chapter 6 and is therefore not included in this study.
5.2 Background

This section gives a brief overview over the bone segmentation of the MK2014v1 and how the segmentation is improved in the MK2014v2.

5.2.1 MK2014v1

The bone segmentation of the MK2014v1 can be divided into 3 parts. The first part consists of the histogram matching and the extraction of the patient’s body. Both procedures have been described in Sections 4.3.1 and 4.3.2. The second part consists of the segmentation of the bone. This part begins with threshold segmentation the image-intensity-transformed CT image. Holes and noise are removed from the resulting binary mask using morphological operations. A complementary segmentation of the bone is done by region growing, see Section 4.2.1. The seed points are created by labeling the bones in the threshold segmentation result and eroding the labeled bones to points. The growing is done for two seed points for each bone. The results from the threshold segmentation and the region growing algorithms are thereafter combined to one binary mask. The third part of the MK2014 consists of closing the holes in the compact bone to fill the interior of the bones. The holes in the compact bone are filled by shrinking the binary mask to a morphological skeleton and connecting separate parts of the skeleton by performing morphological closing. The final segmentation result is given by combining the morphological skeleton with the segmentation results of the threshold segmentation and region growing followed by filling the interior of the bones. A flowchart of the algorithm can be found in Figure 4.1.

5.2.2 MK2014v2

Crucial to get good segmentation results is that the histogram matching transforms the intensity values of the CT image to the predefined ranges for the different tissues. For this to be possible, the reference volume needs to contain about the same amount of bone as the CT image. When the reference image contains more bone tissue, other tissues may incorrectly be classified as bone tissue by the threshold segmentation. For instance a seed point can be placed in the muscular tissue, see Figure 5.1a. This deficiency is handled in the MK2014v2 by performing a second threshold segmentation using a higher threshold, c.f. Figures 5.1a and 5.1b.
Figure 5.1: (a) Segmentation result by the MK2014v1 algorithm. A seed point placed outside the bone tissue resulted in the surrounding tissue being segmented as bone. (b) The MK2014v2 algorithm prevents seed points from being placed outside the bones by performing a second threshold segmentation for the generation of the seed points with a higher threshold value. The seeds are marked by a blue ‘*’. The seed that resulted in the segmentation of the surrounding tissue has been encircled in blue.

5.3 Methods

Three algorithms for segmentation of bone in the pelvic region were compared, see Table 5.1. The later two algorithms are described in Section 5.3.1 to Section 5.3.2. A flowchart of the algorithms is displayed in Figure 5.2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>MK2014v2</td>
</tr>
<tr>
<td>$B_2$</td>
<td>Extension of MK2014v2 to 3D</td>
</tr>
<tr>
<td>$B_3$</td>
<td>Bone segmentation algorithm inspired by [Vasilache and Najarian, 2008]</td>
</tr>
</tbody>
</table>

The algorithms were compared for the eight data sets described in Chapter 3. The $B_2$ algorithm and the $B_3$ algorithm were used for the segmentation of the entire data set. Since the $B_1$ algorithm is a 2D segmentation algorithm and needs a reference image with similar anatomy as the slice segmented, two slices similar to the reference image were selected for each data set. The segmentation result were compared to the same slices in the segmentation result for the $B_2$ and the $B_3$ algorithm. Data set 1 was used to create the reference volume. The evaluation was done by visual assessment.
5.3 Methods

5.3.1 $B_2$: Extension of MK2014v2 to 3D

The first step of the algorithm is to extract the patient’s body and transform intensity values of the CT volume using histogram matching, see Section 4.3.1 and Section 4.3.2 for more details. Two different threshold segmentations are performed: (i) The first threshold segmentation selects bones. Corresponding threshold is set to include most of the compact bone but exclude all other tissues. In this thesis a threshold value of 950 IU is used. (ii) The second threshold segmentation selects a seed region for the region growing. Corresponding threshold is set higher than the first threshold, $t_{RG} = 1100$ IU, in order to reduce the risk of the region growing leaking into tissues surrounding the bones. The growing of the seed region is done by including all adjacent voxels (26 connectivity) that have intensity values inside the acceptance range, see Section 4.2.1. The final segmentation is given by combining the results from the region growing and the threshold segmentation.

*Figure 5.2: Flowchart of the three bone segmentation algorithms.*
5.3.2 $B_3$ algorithm

The $B_3$ algorithm is a region growing algorithm inspired by [Vasilache and Najarian, 2008]. The algorithm begins by extracting the patient’s body from the CT volume and by transforming its intensity values using histogram matching. A seed volume for the region growing algorithm is created by threshold segmentation of the transformed CT volume using a threshold of 900 IU. This threshold is high enough to only include the compact bone, but at the same time it gives a good initial solution for the region growing.

The region growing step uses two criteria which are calculated in advance to speed up the growing in Matlab.

The first criterion states that a voxel needs a higher intensity than the mean intensity in its neighborhood in order to be added to the seed region. A mask (binary volume) containing voxels fulfilling this criterion is created as follows: (i) A copy of the CT volume is averaged by an $n \times n \times n$ box filter ($n = 9$ in this study). (ii) A mask is created, where voxels that have a higher intensity value in the original CT volume than in the averaged CT volume are set to 1 IU and the remaining voxels to 0 IU. This mask (binary volume) will from here on be called local maximum mask, abbreviated LMM. This mask contains an accurate binary representation of the compact bone but also many structures from tissues with lower intensity level, see Figure 5.3.

![Figure 5.3](image_url)

*Figure 5.3: (a) A CT slice of data set 2. (b) The LMM contains all structures that have a higher intensity than its surrounding. In regions where the compact bone has much higher intensity than its surrounding tissue the bone is clearly separated from the other tissues.*

The second criterion states that a voxel needs an intensity above 800 IU to be included in the seed region. A binary volume is created using threshold segmentation, containing the voxels fulfilling the criterion.
5.3.3 Hole filling in 3D

Compact bone

The hole filling algorithm used for the $B_2$ and the $B_3$ algorithms is based on the LMM. The first step is to dilate the segmentation result $l$ times ($l = 5$ here) using a $3 \times 3 \times 3$ kernel. The dilation operation is restricted by the LMM ensuring that the bone mask could not grow uncontrolled in all directions.

Another mask is created by applying closing to the compact bone. The results are combined using element-wise multiplication. Small blobs are removed from the resulting mask before the mask is combined with the segmentation result.

This results in the closing of most holes in the compact bone, but it may also lead to leaking of the bone into other tissues in some regions. To remove the leakage, the algorithm iterates over all filled regions and tests if the filled region is connected to the bone in at least two different parts. Only connections fulfilling this criterion are used to close the holes in the compact bone. The result of this operation is illustrated in Figure 5.9c.

Bone marrow

The interior of the closed compact bone is filled slice by slice. Filling each slice separately makes it possible to fill the bones even when there are holes left in other slices. Slices that cannot be filled by the morphological filling are filled by performing morphological closing in the z-direction (closing between the slices). Small blobs are removed from the mask before the mask is combined with the original segmentation result. The result of this operation is illustrated in Figure 5.9d.

5.4 Results and discussion

This discussion has been divided in a section were the $B_2$ algorithm and its predecessor MK2014v2 are compared and a section were the $B_2$ algorithm is compared to the $B_3$ algorithm.

5.4.1 Comparison of $B_2$ and MK2014v2

The comparison of the two algorithms was done in two steps. First the algorithms were compared without the hole-filling, only comparing how well they were able to segment the bone by the image content. Thereafter a comparison using the hole-filling was done.

When the algorithms were compared without hole-filling, the MK2014v2 gave a slightly better result in areas where the bone was very thin and had a low intensity compared to the rest of the bone, see Figure 5.4. This was a consequence of the following: The reference image of the MK2014v2 contained more bone than the CT images segmented by the algorithm, and thus intensity of pixels of the
bone was increased by histogram matching. These increased intensities fell in the predefined range for bone.

![Images](image1.jpg) ![Images](image2.jpg)

Figure 5.4: Segmentation of the left head of the femur bone. (i) The MK2014v2 was able to segment most of the compact bone. (ii) The segmentation result by the B\textsubscript{2} algorithm at the other hand contained several regions where the compact bone was not detected.

The downside was however that other tissues could incorrectly be classified as bone, when the amount of bone in the reference image was much higher than in the CT image, see Figure 5.5.
5.4 Results and discussion

Figure 5.5: Segmentation of the head of femur bone. (i) The MK2014v2 misclassified muscular tissue as bone when there was more bone in the reference image than in the CT image. (ii) Most compact bone was correctly segmented by the $B_2$ algorithm.

The $B_2$ algorithm classified less tissues incorrectly as bone than the MK2014v2, see Figure 5.5, but it missed regions of compact bone for some data sets. The segmentation results then contained larger holes, see Figure 5.4. When the hole filling was added to the segmentation results, those holes were successful closed.

Both filling algorithms struggled with that bone parts that did not belong together could be incorrectly merged by the algorithms. This happened, however, in a much larger extend for the MK2014v2 than for the $B_2$ algorithm, which used the image content to determine which bone parts should be merged, see Figure 5.6.
Figure 5.6: Comparison between hole filling algorithms for data set 4. (a) Here the MK2014v2 found all bone in the image. (b) When the filling algorithm of the MK2014v2 was used to close the compact bone, tissues surrounding the bone were incorrectly classified as bone. (c) The $B_2$ algorithm initially gave a worse result than the MK2014v2 for this data set. (d) When, however, the filling of the compact bone was applied, all compact bone was detected.
5.4 Results and discussion

5.4.2 Comparison between $B_2$ and $B_3$

Both $B_2$ and $B_3$ detected most of the compact bone in the 8 data sets. In data sets 2 and 7 some of the compact bone was not detected by the algorithms as a result of the histogram matching giving those regions a too low image intensity. For data set 2, this was caused by the patient having a higher amount of compact bone relative to its body size than the other data sets.

Data set 7 at the other hand, was scanned using an intravenous contrast agent, giving the rectum the same image intensity as bone. This resulted in some bone regions not getting a value inside the predefined range for bone by the histogram matching. The result was that the $B_2$ and the $B_3$ algorithm thereby missed parts of the bone compact bone, see Figure 5.7.

![Figure 5.7](image)

**Figure 5.7:** (a) The high intensity of the contrast agent in the rectum in data set 7 resulted in the histogram matching giving the compact bone a intensity similar to that of muscular tissue. This made (b) the $B_2$ and (c) the $B_3$ algorithm segmented parts of the rectum as bone instead of segmenting all the compact bone tissue.

A drawback of the $B_3$ algorithm was that the LMM prevented bone regions surrounded by bone with higher image intensities from being included in the bone segmentation, see Figure 5.8. In the $B_2$ algorithm those regions were correctly classified as bone.
Study on segmentation of bones

Figure 5.8: Segmentation of the pelvic bone and the femur head in data set 4. The local maximum mask prevented the $B_3$ algorithm to identify all of the bone tissue.

Adding the filling algorithm to the $B_2$ algorithm resulted in that most of the smaller holes in the compact bone were closed, making it possible to fill the bone marrow, see Figure 5.9. The exceptions were data set 2 and 7 where entire parts of the bone were missed in the segmentation result.
Figure 5.9: Segmentation of the bone in data set 1 by the $B_2$ algorithm. (a) CT slice of the segmented volume. Some parts of the bone had image intensities in the same range as the muscular tissue. This made those hard to find by the segmentation algorithm. (b) When the filling algorithm was applied to the result from the $B_2$ algorithm (green) those regions were successfully included in the segmentation result (red). (c) The first part of the filling algorithm connected parts of the compact bone and (d) the second part filled the interior of the bones.
One problem was that the bone filling algorithm could connect the head of the femur bone with the pelvic bone which resulted in the incorrect classification of the cortege between those as bone tissue, see Figure 5.10. This connection was only included in the LMM when the mask was created from the histogram matched data. Since only 8 data sets were available for this study, it cannot be excluded that no similar connections could occur when creating the local maximum mask from the original CT data.

**Figure 5.10:** Closing of holes in the bone segmentation in data set 6. (a) In one of the slides a connection between the bones is created when filling holes in the compact bone. (b) This results in filling the gape between the head of the femur bone (left bone) and the pelvic bone (right bone). The segmentation result of the $B_2$ algorithm is shown in green and the filled regions are shown in red.

It was, however, more common that the cortege between the two bones was filled by the morphological closing in z-direction than the cortege being filled as a result of the compact bones being connected and thereafter filled. The morphological closing was introduced in order to close gaps between filled slices and holes in the compact bone, which resulted in that also the gape with the cortege was filled. Improving the closing of holes in the compact bone would make the morphological closing in z-direction unnecessarily, since the interior could be filled in 3D.

The $B_3$ algorithm had problems with small holes in the LMM, which prevented the inside of the bone to be filled, see Figure 5.11. Since both the $B_3$ algorithm and hole filling algorithm were based on the LMM neither algorithm was able to fill those holes.
Figure 5.11: Images of the femur bone. Image (a) is an image of the CT data. (b) Segmentation of the CT data by the $B_3$ algorithm followed by hole filling. (c) Some regions could not be filled due to holes in the LMM. (d) When the same data was segmented by the $B_2$ algorithm the bone was successfully filled.
In general the $B_2$ algorithm gave slightly better segmentation results than the $B_3$ algorithm and has potential to get even better if the hole filling algorithm is improved.

5.5 Conclusion

In this study the results from two 3D segmentation algorithms for bone were evaluated and were compared to the bone segmentation result of the MK2014v2. Both 3D algorithms gave good segmentation results for most of the tested data sets and classified less of the tissue surrounding the bones incorrectly as bone tissue than the MK2014v2. Of the two 3D algorithms, the $B_2$ algorithm was considered giving slightly better results than the $B_3$ algorithm, since it was able to fill more of the holes in the compact bone as well as the interior of the bones. The filling of the bone could, however, not be solved completely, since large regions of unclassified compact bone still could lead to the algorithm not being able to fill the inside of the bone.

5.6 Future work

A problem that remains in the bone segmentation algorithm is to close the holes in the compact bone. For this the general direction of the bone could be used in combination with information about the image content. A simple first step would be to apply a directional dilation in the direction of the bone instead of the combination of closing and dilation (see Section 5.3.3). This could be combined with a criterion similar to the local maximum mask to include information from the data set and thereby restrict the dilation.

An alternative approach would be to merge the segments of compact bones by applying a model describing the shape of the bones. Such a model would, however, need to be deformable enough to adjust for variability in bone anatomy as well as fractures in the bones, but rigid enough only to segment the bone tissue.
6.1 Introduction

Prostate cancer is the second most common cancer disease worldwide directly after skin cancer [Prostate cancer, 2016]. In Sweden 10 452 cases were registered in National Program of Cancer Registries (NPCR) in 2014 [Regionalt cancercentrum, 2015].

One way to treat the cancer is by radiotherapy, in which the cancer cells are irradiated by radiation from an external radiation device or by inserting radioactive sources into the tumor. In both methods exact knowledge of the position of the prostate, rectum and bladder is necessary. Typically a radiologist manually segments the images which are then used for the treatment planning. This may be a very time consuming task. An automated segmentation algorithm for prostate, rectum and bladder is highly desirable.

In the last years a number of automated algorithms have been developed. Common methods for the segmentation of the prostate and the rectum are deformable models [Costa et al., 2007, Shao et al., 2015], shape models [Martinez et al.] and atlas based methods. A method often used for the registration in atlas based segmentation is Demons algorithm, see for instance [Dowling et al., 2011, Cheung and Krishnan, 2012] for MRI image and [Acosta et al., 2011, Rodriguez-Vila et al., 2012] for CT images.

Another registration algorithm for non-linear registration is the Morphon. In Rodriguez-Vila et al., the registration of Demons algorithm was compared to the registration of the Morphon for the registertion of a template volume to 5 data sets containing prostate. Their conclusion was that the Morphon resulted in a smoother displacement field and thereby preserved the structures of the template
volume better than Demons algorithm. They also showed that the Morphon, in
difference to Demons algorithm, is able to register a template volume (without
contrast agent) to a CT volume containing a contrast agent, since the Morphon is
independent of the intensity levels of the data.

Pettersson et al. [2006] showed that the Morphon is a promising method for seg-
mentation of the femur and the pelvic bone in CT data. In their case the atlas
contained the cortical bone only and thus their results may not fully apply to this
work, which uses a differently designed atlas.

The aim of the third study was to evaluate the performance of three atlas based
segmentation algorithms and to estimate the position of the prostate and the rec-
tum.

6.2 Background

6.2.1 MK2014v1

After the segmentation of the bone and the adipose tissue, the gluteus maximus
muscle and the prostate are segmented by atlas segmentation followed by active
contours without edges. The atlas segmentation is done by phase based affine
registration using two different quadrature filters for the x and y direction. A
detailed description of affine registration can be found in Section 2.6.1.

The initial registration is done between the outline of the patient’s body in the
atlas and the CT image as follows. A binary mask containing the patient’s body
is created by threshold segmentation, which thereafter is downsampled in order
to reduce the processing time of the registration. After the registration, the dis-
placement field is rescaled to the original image size and applied on the atlas.

The transformed atlas gives the initial position of the gluteus maximus muscle.
Active contours are applied on the muscles on the left and right side separately
using the built in Matlab function activecontour.

For the segmentation of the prostate a second registration is done, where the
bones of the registered atlas are registered to those of the CT image. The result
is compared with the result of initial registration and the transformation giving
higher DICE coefficient for bones is selected. Active contours are used to get the
final segmentation of the prostate.

6.2.2 Deficiencies of MK2014v1

In most cases the affine registration results in a registration which gives a good
initial position for the active contours, see Figure 6.1.
6.2 Background

Figure 6.1: Atlas registration by the MK2014v1. (a) The atlas in its initial position overlayed as colored contours on the CT image. (b) Registration of the outer border of the body and (c) registration of the bones of the patient. Both results give a good initial position for the prostate.

The affine registration fails when the distance between the borders of the patient and the atlas is larger than the reach of the quadrature filter. For instance, Figure 6.2 shows such a situation. The affine registration of the outer border leads to an overstretching of the frontal part of the atlas, see Figure 6.2b. The subsequent registration of the bones then completely fails, see Figure 6.2c.

Figure 6.2: (a) The initial position of the atlas is overlayed on the CT image. (b) The registration gives a good fit on the lower border (the back of the patient) but results in a stretching of the atlas. (c) The bad initial solution from the registration of the border leads to a bad registration of the bones.

The deformable model (Figure 4.1) may fail (lead to a leakage) when the initial contour is overlapping more than one tissue, for instance parts of the prostate and the bone.

6.2.3 MK2014v2

Several changes were made in the MK2014v1 to mitigate the deficiencies described in Section 6.2.2. The updated version (MK2014v2) works as follows: The registration of the borders of the body uses four points on the outline of the patient’s body; the linear registration is not used. Those points are the pixels with
the largest and the lowest x coordinate inside the patient and similarly for the y coordinate. It is assumed that the patient and the atlas are rotated (positioned) similarly.

For the registration of the bones, the affine registration is performed on different scales (Section 2.6.1).

### 6.3 Methods

First, (i) linear registration in 3D, (ii) non-linear registration by the Morphon and (iii) linear registration followed by non-rigid registration by the Morphon were implemented in Matlab, according to 2.6. Second, those algorithms were evaluated by comparing their results for bone-to-bone and whole-volume registrations, see Table 6.1. The former registers the bones of the atlas to bones of the CT data and the resulting displacement field is used for the placement of the prostate and the rectum. The latter registers all tissues of the atlas volume directly to those of the CT data. Also, the improved performance of the MK2014v2 compared to MK2014v1 was demonstrated.

<table>
<thead>
<tr>
<th>Alg</th>
<th>Description</th>
<th>bone to bone</th>
<th>all to all</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_L$</td>
<td>Linear registration (3D)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$R_N$</td>
<td>Non-linear registration (3D)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>$R_{LN}$</td>
<td>Linear and non-linear registration (3D)</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The evaluation of the bone-to-bone and whole-volume registration were done by comparing the results for the 8 data sets described in Chapter 3 using visual comparison. The results for the bone-to-bone registration were evaluated according to how closely the prostate and rectum were placed to their true positions. For the whole-volume registration the positions of all the organs were compared. The main focus of the study was on the bone-to-bone registration.
6.3 Methods

6.3.1 Z range

Each data set is defined by voxel ranges in x, y and z directions. All data sets in Chapter 3 (source volumes) and the atlas were cropped to contain the pelvic region by selecting the corresponding z-ranges. The source volumes functions here as reference volumes as described in Section 2.6. For the $R_L$ and the $R_{LN}$ algorithms, the z-range of the atlas volume was equally extended on both sides to allow the linear registration to transform the atlas without getting regions with missing information, see Section 6.5.2.

6.3.2 Preprocessing step

A preprocessing step was used for the atlas and source volumes before applying the $R_L$, $R_N$ and $R_{LN}$ algorithms. The preprocessing step consisted of: (i) extraction of the body by setting all voxels outside the body to zero, (ii) resampling the volumes from non-isotropic to isotropic voxels and (iii) creating volumes consisting of bones only. More detailed information follows.

Extraction of the body

The patient’s body in the source volume was extracted using the preprocessing algorithm described in Section 4.3.1. The body in the atlas volume was extracted by selecting appropriate tissues.
Resampling of volumes

Both the linear and non-linear registration are based on the phase of a filter response. Before the filter responses of the two volumes were calculated, the volumes were resampled, to isotropic voxels, so that available quadrature filters could be used. The new voxel size of the two volumes was set to the pixel size of the source volume.

The resampling was done in two steps. The first step was to calculate the new size of the volumes. The new size $S_k'$ of the volume (in voxels) was calculated for each direction $k, k = \{x, y, z\}$, according to

$$S_k' = \frac{v_k'}{v_k} \cdot S_k,$$

(6.1)

where $v_k'$ and $v_k$ are the voxel sizes in the new and the old volumes respectively, $S_k$ is the size of the old volume. The second step was to calculate the voxel values in the new volume using linear interpolation. The voxel coordinate in the old volume, $c_k$, is related to the voxel coordinate in the new volume, $c_k'$, as

$$c_k = \frac{v_k}{v_k'} \cdot c_k'.$$

(6.2)

The atlas was resampled using nearest neighbor interpolation, which has the advantage of preserving the concept of labels (categorical variables).

Creation of the bone volumes

For the bone-to-bone registration two volumes containing the bone in the source volume and the bone in the atlas volume were created. The former was created by applying histogram matching followed by subtracting 700 from the intensity value in each voxel. Voxels that got negative intensity values were set to zero. The latter was created by setting voxels with bone labels to one and the remaining voxels to zero.
6.3 Methods

Figure 6.4: Image (a) shows a CT slice taken from data set 2. (b) The bone volume includes the bone and some of the muscular tissue.

6.3.3 $R_L$ algorithm

The $R_L$ algorithm consist of: resizing of the volumes to the same voxel array dimensions by zero-padding, defining a weight for the certainty function $c$, the phase-based affine registration and the cropping of the deformed atlas. These steps are described in more detail below.

Resizing of volumes

The first step in the $R_L$ algorithm was to zero-pad the atlas and the source volumes, i.e. adding voxels filled with zeros until both arrays had the same dimensions. No zero-padding was done in the z-direction if the atlas had more elements in the z-direction than the source volume. This was done to keep the volumes smaller when using an atlas with a large z-range. The higher amount of elements in z-direction was instead handled inside the linear registration.

Weight of certainty function

The second step was to define a weight function for the certainty $c$. The weight function was used to decrease the certainty for the voxels near the border of the source volume, where the filter responses of the quadrature filters were effected by the abrupt change in voxel values at the edge of the volume.

The used weight function $h$ was defined as

$$h(x, y, z) = h_x(x)h_y(y)h_z(z),$$

(6.3)

where $h_x$, $h_y$ and $h_z$ are x, y and z direction specific 1D weight functions.

In the discretized case the weight functions had a constant central part in the range $[b_{x,2} \ b_{x,3}]$ for the x direction and similarly for the y and z directions. The
slopes (see Figure 6.5) were products of normalized logarithmic and Gaussian functions. So, for the x direction the weight function was defined as

\[
h(x) = \begin{cases} 
  g(x)f(x) & 1 \leq x \leq b_x,2 \\
  1 & (b_x,2 + 1) \leq x \leq (b_x,4 - b_x,2) \\
  g(b_x,4 - x + 1)f(b_x,4 - x + 1) & (b_x,4 - b_x,2 + 1) \leq x \leq b_x,4 
\end{cases}
\]

where

\[
f(i_x) = \frac{\ln(i_x)}{\ln(b_x,2)}, \quad i_x = 1, 2, \ldots, b_x,2 \tag{6.4}\]

and

\[
g(i_x) = e^{-\frac{(i_x-b_x,2)^2}{2c^2}}, \quad i_x = 1, 2, \ldots, b_x,2. \tag{6.5}\]

In Equation (6.5) \( c = \frac{1}{3} b_x,2 \). The boundary indexies were defined as follows for z-direction: \( b_z,1 = 1, b_z,4 \) was the dimension of the array, \( b_z,2 = \text{floor}((b_z,1 - b_z,1)/4) \) and \( b_z,3 = b_z,4 - b_z,2 \); x direction: \( b_x,1 = 1, b_x,4 \) was the dimension of the array, \( b_x,2 = 20 \) and \( b_x,3 = b_x,4 - b_x,2 \); y direction: similarly as x direction.

\[\text{Figure 6.5: A one dimensional illustration of the one-dimensional weight function } h_z \text{ for data set 2.}\]

The linear registration

The larger number of voxels in the z-direction of the atlas volume than the source volume was handled by cropping the atlas volume to the same dimensionality as the source volume before calculating the parameter vector \( p \) in each iteration. The parameter vectors \( p \) and \( p_{\text{tot}} \) were thereafter calculated according to Section 2.6.1. From the parameter vector \( p_{\text{tot}} \) a displacement for the entire atlas was
calculated and the atlas was transformed accordingly. Those steps were repeated in each iteration.

**Cropping of atlas**

The resulting displacement field from the linear registration was applied to the atlas. Thereafter the atlas was cropped to the same voxel array dimensionality as the source volume.

### 6.3.4 $R_N$ algorithm

The $R_N$ algorithm based on the non-linear registration by the Morphon, see Chapter 2.6.2, was used for both the bone-to-bone and the whole-body registration.

For the bone-to-bone registration, the bone volume of the atlas, bone volume of the source volume and the atlas volume were zero-padded. Thereafter the bone volume of the atlas was registered to the bone volume of the source volume. The resulting displacement field was applied on the zero-padded atlas.

For the whole-body registration, the atlas and the source volume were zero-padded. The atlas was thereafter registered to the source volume.

### 6.3.5 $R_{LN}$ algorithm

The $R_{LN}$ algorithm is a combination of the $R_L$ and the $R_N$ algorithm. The algorithm began by applying the $R_L$ algorithm to get an initial transformation for the atlas. This was done by registering the bones of the atlas to the bones of the source volume.

After the atlas was deformed and cropped by the $R_L$ algorithm, the non-linear registration was applied in order to get the final position of the prostate and rectum. For the bone-to-bone registration, the non-linear registration was done between the bone volumes. The resulting displacement field was used to deform the atlas. For the whole-body registration, the non-linear registration was performed using all the tissues in the atlas and the source volume.

### 6.4 Data

#### 6.4.1 Atlas

For the atlas segmentation the VISHUM atlas from the Virtual Human database of the Helmholtz Zentrum München [Zankl et al., 2002] was used. The atlas covers the area from the head to the upper legs and contains 137 organs. Each of the 250 images is $512 \times 512$ and has a voxel size of $0.91 \times 0.94 \times 5.0$ mm.
6.4.2 Quadrature filters

The quadrature filters used for the linear registration were taken from Broccoli, a software for analysis of fMRI data [Eklund et al., 2014]. The quadrature filters and the constant tensor $M_k$ used for the non-linear registration were taken from Daniel Forsberg’s image registration toolbox (http://www.imt.liu.se/mi/Tools/).

6.5 Results and discussion

6.5.1 MK2014v2

The MK2014v2 gave good results for the registration of the patient outline for all slices selected from the 8 data sets. For the segmentation of prostate a linear registration defined the position of the bones of the transformed atlas. These positions overlapped well with bones in the CT image. Differences stemmed from the fact that the linear registration was not able to deform the shape of the bone of the atlas.

Figure 6.7 shows, that the MK2014v2 algorithm gives good registration results for the patient’s outline and bones even when the initial placement of the atlas is far from true position. The CT image is the same one as in Figure 6.2.
6.5 Results and discussion

Figure 6.7: (a) Image of the initial position of the atlas (colored lines) relative to the CT image. (b) In the segmentation of muscle, translating and scaling the atlas by selecting 4 reference points at the border of the patient gave a good result for the outline. (c) In the segmentation of prostate, the registration of the atlas bones to the bones in the CT image gave a reasonable good result.

6.5.2 Linear registration

Figure 6.8 shows the position of the bones of the atlas and data set 2 before and after the linear registration. A visual comparison of the projections shows that the registration worked well. This observation was further confirmed by visual inspection of individual slices. Similar results were observed for the other data sets.

To verify that the linear registration gave good results when the z-range of the atlas was inaccurately selected, the z-range of the atlas was extended on one of the sides. The results were fine as long as the z-range was not increased by more than 50%. A larger increase led to an incorrect fitting for some of the data sets.
Figure 6.8: Projections (radiological paths) of the bone perpendicular to the (a) axial, (b) coronal and (c) sagittal planes before the registration. The atlas and the CT image are plotted in yellow and white respectively. (d-f) are the same projections but after the linear registration.

In general the linear registration in 3D gave better results than in 2D. Obviously, the better performance was caused by having more data and the additional degree of freedom in 3D.

6.5.3 Bone-to-bone registration

$R_L$ gave a good global fitting of the atlas bones. Best results were achieved for the data set 2 where most of the bones were placed correctly, see Figure 6.9. Since the anatomy is different for each person, a perfect match between the bones of the transformed atlas and the source volume cannot be achieved.
6.5 Results and discussion

Figure 6.9: The images (a), (b) and (c) show three different slices from the registration result from \( R_L \) of data set 2.

For data sets 1 and 8 the \( R_L \) algorithm correctly registered the shaft of the femur bone, the pubis and ischium; the latter two are parts of the pelvic bone. However this led to an incorrect scaling of ilium and a wrongly positioning of the tail bone outside the body of the source volume.

Figure 6.10: For data set 1 the \( R_L \) algorithm resulted in an upscaling of pelvic bone of the atlas. (a-b) The pelvic bone is enlarged in dorsal direction resulting in the tail bone being placed outside the body. A good fit was achieved for the inner edge of the pelvic bone and (c) for the femur bone.

The \( R_N \) algorithm gave good registration result for data sets 2, 4, 5, 6 and 8, see Figure 6.11 for the data set 2. In those data sets, the bones of the patient were about the same size as the bones in the atlas volume.
For data sets 1, 3 and 7 the anatomy of the patient (the voxel size was the same) made the bones in the source volume much larger than the bones in the atlas, see Figure 3.1. Scaling up the bone in the atlas to the correct size proved to be a big challenge, since the initial scaling and fitting were done when the source volume and the atlas were strongly downsampled. Edges (borders) of the pelvic bone in the downsampled source volume strongly depended on the thickness of the compact bone and the amount of bone marrow in the source volume.

For example, low contrast edges of the pelvic bone combined with high contrast edges of the femur head bone in data set 1,3 and 7 led the algorithm to register the atlas to the head of the femur bone instead of the pelvic bone, see Figure 6.12. This resulted in a larger distance between the left and right pelvic bone of the transformed atlas (overstretched atlas in the lateral direction), see Figure 6.12c.
Figure 6.12: Intermediate results from $R_N$ on data set 3. (a) and (b) are images of the bones in the source volume and in the atlas, respectively, on scale 4. The concentration of trabecular bone (spongy bone) in the head of the femur bone makes the femur bone appear much brighter than the pelvic bone. Since the pelvic bone of the atlas is homogeneous the partial volume effect caused by the downsampling only appears on the edges of the bones. The high contrast edge of the head of the femur bone made the non-linear registration to incorrectly deform the pelvic bone in the atlas to the femur bone of the source volume only. (c) The registration result on scale 3; the bones of the atlas and the source volume are shown in yellow and white, respectively.

The parameters of the $R_N$ are displayed in Table 6.2.

In the $R_{LN}$ algorithm the result from the linear registration was used as an initial solution for the non-linear registration. This gave good results for all data sets with only a few incorrect fittings for some of the data sets. For example Figure 6.13 shows the good registration result for data set 2.
Table 6.2: The number of iterations, the filter sizes $FS_i$ and $FS_a$ and the standard deviations $STD_i$ and $STD_a$ for each of the scales of the Morphon in the $RN$ algorithm. $FS_i$ and $STD_i$ are the parameters of the Gaussian filter used for the regularization of the left and right side of (2.38) and $FS_a$ and $STD_a$ are the parameters of the Gaussian filter used for the regularization of the accumulated displacement field.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>scale</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interations</td>
<td></td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>$FS_i$</td>
<td></td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>$STD_i$</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>$FS_a$</td>
<td></td>
<td>9</td>
<td>13</td>
<td>25</td>
<td>25</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>$STD_a$</td>
<td></td>
<td>0.7</td>
<td>1.0</td>
<td>1.5</td>
<td>2.5</td>
<td>2.5</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Figure 6.13: Registration of the bone of the atlas to the bone in data set 2 by the $RLN$ algorithm. The upper row shows the registration results (yellow) plotted on the CT data. $RLN$ was able to get a good fit for all of the bone in data set 2. The lower row shows a grid visualizing the deformation field. The grid has been plotted over the deformed bone of the atlas volume.
The $R_{LN}$ algorithm gave a better fit between the bone in the atlas and the source volume than the other two algorithms, see Figure 6.13. The biggest improvement was seen for the femoral head, the great trochanter and the ilium.

The disadvantage of the $R_{LN}$ algorithm was that the linear registration could lead to a bad local fit which was very hard to correct by the non-linear registration without the displacement field getting folded over. The best example for this was the tail bone in data set 1 and 8 that was placed far from its "true" position by the linear registration. A subsequent non-linear registration with a low regularization resulted in a good fit for the tail bone but the displacement field was getting “folded over” (the originally parallel grid lines cross each other in the deformed image), see Figure 6.14(a,b,d,e). Regions that were folded over in data set 1 were the area anterior to the tail bone and some parts of cartilage between the head of the femur bone and the pelvic bone.

![Figure 6.14:](image-url) Comparison of the results of the $R_{LN}$ when applying a low and a high regularization on the displacement field. (a) CT image showing the pelvic bone and the head of the femur bone of data set 1. (d) The initial deformation of the bone of the atlas is given by the linear registration (yellow). (b) The bones of the atlas were successfully registered to the bones of the source volume by the non-linear registration when a relatively low regularization was used. This resulted, however, in the displacement field being folded. (e) A uniform grid being deformed by the displacement field. (f) A higher regularization removed the folded areas in the displacement field. (c) The higher regularization led to a worse fit of the bone of the atlas to the bone of the source volume.
For data set 8, the displacement field got folded over at the cartilage. No folding occurred at the tail bone as a consequence of the $R_{LN}$ algorithm not being able to register the tail bone of the atlas to the tail bone in the source volume. A higher regularization was needed in order to avoid the displacement field from getting folded over. The higher regularization resulted in smoothed out displacement in areas with a strong local deformations. This in turn led to the loss of some deformations, see Figure 6.14(d,f).

Since the displacement field only was folded over for data set 1 and 8 when selecting the lower regularization, the lower regularization (Table 6.3) was selected for the non-linear registration in $R_{LN}$. The reasoning was that the large local displacement of the tail bone should be handled by the registration algorithm doing the initial registration instead of the non-rigid registration.

In an attempt to get a better initial registration, the affine registration was replaced with a rigid registration followed by a registration performing translation and scaling. This combination gave however a worse initial registration than the affine registration, since the rotation and scaling was performed after one another instead of simultaneously.

**Table 6.3:** The number of iterations, the filter size $FS_x$ and the standard deviation $STD_x$, measured in voxels, for each of the scales of the non-linear registration in the $R_{LN}$. $FS_x$ and $STD_x$ are the hsize and sigma parameters, respectively, of the Matlabs Gaussian filter fspecial. The subscripts $i$ and $a$ are used for iterative and accumulative displacement fields in equations (2.38) and (2.42), respectively. Scale 0 is the original scale and scale 5 is the scale where the volume is downsampled the strongest.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>$FS_i$</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>$STD_i$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>$FS_a$</td>
<td>9</td>
<td>13</td>
<td>25</td>
<td>25</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>$STD_a$</td>
<td>0.7</td>
<td>1.0</td>
<td>1.5</td>
<td>2.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

3D visualization of the bone

Figure 6.15 shows the bones of the undeformed atlas volume, the deformed atlas volume by the $R_{LN}$ algorithm and the bone segmentation result of the $B_2$ algorithm in Chapter 5. The large distance between the slices of the atlas volume gave the upsampled bone atlas surface a lego-like appearance, see Figure 6.15a. Registering the atlas volume to the source volume by the $R_{LN}$ algorithm smoothed the surface. In order to get rid of the steps entirely, more iterations and a lower regularization on the finer scales would be needed, see Figure 6.15b. This would however increase the risk of the displacement field getting folded over.
Figure 6.15: (a) 3D visualization of the undeformed bone of the atlas volume. Resampling the bone of the atlas by nearest neighbor interpolation resulted in lego-like appearance of its surface. (b) The bone of the atlas deformed by $R_{LN}$ in order to fit it to data set 2. The $R_{LN}$ algorithm smoothed the bone surface partly. (c) The segmentation of data set 2 by the $B_2$ algorithm gave a much smoother result. All bone volumes were smoothed by a Gaussian kernel of size 5 and a standard deviation of 1.4 using Matlab function smooth3 before visualizing them as isosurfaces.

Apply deformation to Rectum and Prostate

Figure 6.16 shows the estimated position of the rectum and the prostate by the $R_L$, $R_N$ and $R_{LN}$ algorithms for data set 4 and 6. All three algorithms gave a similar placement of the two organs. None of the algorithms performed clearly better than the other two. $R_N$ and $R_{LN}$ better positioned the organs for some data sets, but they also deformed them more than the $R_L$ algorithm, see Appendix A.
Figure 6.16: The segmentation results of the bone, the rectum and prostate for data set 4 (upper row) and data set 6 (lower row). The results were obtained using the \(R_L(a,d)\), the \(R_N(b,e)\) and the \(R_{LN}(c,f)\) algorithm.

One drawback of the approach was that it did not take into account that the relative position of the rectum and the prostate may differ between different individuals. It was also not taken into account if the rectum was full of bowel gas or not. Thus, the results should be seen more as guesses of the organ positions than a segmentation of these. The estimated positions coincided with the actual (‘true’) positions of the organs in most cases, but the shapes of the organs differed from the true shapes.

In order to get a better segmentation of the two organs, it would be necessary to include the information about the soft tissue in the registration. This is however difficult because of for instance: (i) the low contrast between the prostate and the surrounding tissues and (ii) the missing CT texture of the prostate in the atlas. The use of several atlases would take into account the difference in anatomy between patients.

6.5.4 Scales in \(R_{LN}\) algorithm

Figure 6.17 shows a comparison between the segmentation results of the \(R_{LN}\) algorithm when running the non-linear registration using all the scales and when only using the coarse scales. Both cases resulted in a similar placement for the prostate and the rectum. Limiting the non-linear registration to the coarse scales had the benefit of the displacement field remaining smoother and in the compu-
tation time being reduced significantly.

![Image](image_url)

**Figure 6.17**: Results from the $R_{LN}$ algorithm for data set 1 when using (a) all scales and when using (b) scales 5 to 3 of the non-linear registration. Using all the scales gave a slightly better positioning of the prostate (blue) and rectum (red) than when only using the coarse scales. (c) When the non-linear registration was performed for all scales the resulting displacement field was strongly folded over. (d) A much smoother displacement field was obtained when the non-linear registration only was performed for the coarse scales.

### 6.5.5 Hole-volume registrations

Of interest was whatever a better segmentation of the prostate and the rectum could be achieved by including the soft tissue into the registration, i.e. by not using the bone only. The results showed that the Morphon was not robust enough to fit the structures of the soft tissue in the atlas to the corresponding structures in the CT data in all cases, see Figure 6.18. Common misclassifications were:

(i) The border of the muscular tissue was registered to the border of the patient.
(ii) The bones were not deformed enough to fit the bones in the source volume.
(iii) The upscaled prostate fitted the bladder, since the algorithm could not identify the low contrast boundary between the bladder and the prostate. In general the problem was caused by the fact that the Morphon algorithm tries to fit edges. There are however many edges in a CT image and the algorithm does not have any information to which structures those belong. Moreover some of those edges may not be present in the atlas, since they originate from inner structures of tissues or
Figure 6.18: Registration of all tissues in the atlas to data set 2 by $R_N$ (upper row) and $R_{LN}$ (lower row). (a) CT slice of data set 2 (d) the undeformed atlas volume. (b,d) Deformed atlas volume by $R_N$ and $R_{LN}$. (c) The source volume and the outlines from the tissues in the deformed atlas. The outer border of the patient is colored green, the muscular tissue blue, the bones red and the rectum and the prostate cyan.

the atlas omits details not important for the application for which it was created. The used atlas was designed for calculations of absorbed dose distributions from ionizing radiation exposures.

Some of the incorrect fittings may be avoided by further fine tuning the parameters of the non-linear registration. It is however unlikely that fine tuning would make the two algorithms sufficiently robust.

6.6 Conclusions

In this study linear registration, non-rigid registration and linear registration followed by non-linear registration were used for the segmentation of the prostate and the rectum. The three algorithms were examined for bone-to-bone registration and whole-body registration. For bone-to-bone registration, the $R_{LN}$ algorithm gave a fairly good localization of the prostate and the rectum and performed slightly better than the other two algorithms. The placement of the two organs did not depend much on the scales (all or coarse) of the non-rigid registration. All scales, however, led to more strongly folded displacement fields than in
case of coarse scales. Also the computational time was longer.

The whole-body registration did not work well, since the atlas was not suitable for the approach used by the Morphon algorithm; edges in the source volume and the atlas did not fully match and the algorithm could not adapt to this situation in a robust way.

6.7 Future work

Of interest is whether the registration of the atlas to the source volume can be improved by using an atlas more similar to the human anatomy for the whole-body registration. The used atlas included many edges, like the edge around the prostate, that are barely visible in the source volume.

The weight function in the $R_I$ algorithm could be designed to prioritize certain regions inside the voxel array. Right now the weight function is constant inside the volume sufficiently far from the boarders. Those regions would dominate the registration process.

In this thesis, the linear registration fits the bones of the atlas volume to the bones of the source volume. Another way to perform the affine registration would be to fit the bones of the source volume to the bones of the atlas volume. The found deformation could thereafter be applied to the atlas by inverting the found parameter vector. This may give a better registration result for some of the data sets.

One way to make the displacement field of the Morphon smoother would be to replace the normalized averaging of the accumulative displacement field $d_a$ with adaptive anisotropic regularization of the iterative displacement field $d_i$. In adaptive anisotropic regularization the shape of the Gaussian kernel is defined by the structures in the image data. This results in that regions with few image structures are regularized by a large isotropic filter, regions with one dominant structure orientation are regularized along the structure and areas with multiple structure orientation are regularized by a small isotropic kernel Forsberg et al. [2010]. Another option would be to regularize the final displacement field by a second regularization. Johansson et al. [2012] proposed a post processing step where the applied regularization is controlled by a cost function. This cost function is used to force the partial derivatives of the displacement field to be small in the same time as it only allows changes in the displacement field that not alter the placement of the image structures significantly.
The JJ2016 segmentation algorithm

The most promising algorithms from the previous studies were combined with several additional methods into one automatic segmentation algorithm called the JJ2016. The algorithms that gave the best results in each of the studies are: (i) the $A_2$ for the segmentation of adipose tissue, (ii) the $B_2$ for the segmentation of bone and (iii) the $R_{LN}$ for the estimation of the position of the prostate and the rectum. The non-linear registration of the $R_{LN}$ is only used with the coarse scales (scales 5 to 3) in order to get a smooth displacement (reduce the amount of folding). The additional methods are: (i) the segmentation of air (Section 4.3.1), (ii) transfer of the atlas rectum and prostate into the original data set, (iii) resolving of tissue conflicts for voxels containing a mixture of air and adipose tissue.

### 7.1 Description of the algorithm

A flowchart of the JJ2016 algorithm can be seen in Figure 7.1
The inputs of the JJ2016 algorithm are: (i) the CT data set that is to be segmented, (ii) a reference volume of a CT data set for the histogram matching and (iii) an atlas volume for the atlas-segmentation. The output of the algorithm is a list of binary volumes defining the location of adipose tissue, bones, rectum, prostate and air inside the body.

7.1.1 Transfer of segmented prostate and rectum

The $R_{LN}$ algorithm transforms the voxels of the atlas and the CT volume to isotropic voxels. After the localization of the rectum and prostate, the voxels are rescaled to voxel sizes of the original CT volume.

7.1.2 Resolving of tissue conflicts

Since each tissue is segmented separately, a voxel can be assigned to more than one tissue. This situation is called a tissue conflict here. For instance voxels containing bone marrow may be segmented as both adipose tissue and bone tissue.
To ensure that every voxel is classified to one tissue type only, the tissues are given numbers (priorities) used for tissue conflict resolution, see Table 7.1. A voxel with a tissue conflict is set to the tissue with the highest priority.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Tissue</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Bone</td>
</tr>
<tr>
<td>4</td>
<td>Air</td>
</tr>
<tr>
<td>3</td>
<td>Prostate</td>
</tr>
<tr>
<td>2</td>
<td>Rectum</td>
</tr>
<tr>
<td>1</td>
<td>Adipose tissue</td>
</tr>
</tbody>
</table>

### 7.1.3 Adipose tissue: remove partial volume effect

A post processing step was introduced for the adipose tissue segmentation in order to remove voxels that were incorrectly classified as adipose tissue due to the partial volume effect. This is done by excluding all voxels that are neighbors to voxels segmented as air from the segmentation result of the adipose tissue.

### 7.2 Results and discussion

The results of the JJ2016 algorithm for data set 2 can be seen in Figures 7.2 and 7.3. The quality of the segmentation of individual tissues is described in previous chapters. The segmentation of bone and adipose tissue were considered quite accurate by a radiologist, while a better segmentation of rectum and prostate was deemed necessary to be of medical value.

More segmentation results can be found in Appendix-B, which contains 20 segmented transverse slices for each of the data sets.
Figure 7.2: The segmentation result by the JJ2016 for data set 2 shown in (a) the transverse (horizontal) plane, (b) the coronal (frontal) plane and (c) the sagittal (median) plane. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.

Figure 7.3: 3D visualization of the results by the JJ2016 for data set 2. The segmentation results of the bones and adipose tissue are shown in gray respective yellow and the estimated positions of the rectum and the prostate in red respective blue.
The average running time of the algorithm for the eight data sets was 13 minutes and 47 seconds. The most time consuming part was the removal of the incorrect connections between the bone segments in the bone filling algorithm, see Table 7.2. Each connection was tested separately for removal using time consuming for-loops in Matlab.

<table>
<thead>
<tr>
<th></th>
<th>PH</th>
<th>Adipose</th>
<th>Bone</th>
<th>Filling</th>
<th>Atlas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.80%</td>
<td>1.18%</td>
<td>16.59%</td>
<td>60.79%</td>
<td>19.64%</td>
</tr>
</tbody>
</table>

The presented results are valid for the 8 data sets only. It is reasonable to expect that the algorithm will work for other data sets too, but it may fail for data sets containing abnormalities. To validate the algorithm for a clinical environment it needs to be evaluated on a larger number of data sets reflecting the variability in patient sizes and shapes. One things that should be investigated is how the JJ2016 performs for different scan parameters. In this study most data sets were scanned using the tube voltage of 120 kV.

For DIRA it is not necessary that the segmentation algorithm is independent of the tube voltage, since the scans will be done according to one or several protocols designed for DIRA. In this case histogram matching can be skipped since the CT numbers will be known accurately.

### 7.3 Conclusion

The aim of this thesis work was to extend the MK2014v1 algorithm to 3D as well as to improve its segmentation results. Several different algorithms were tested. The best-performing algorithms were combined into the newly developed JJ2016 algorithm, namely: (i) Otsu’s method and threshold segmentation for the segmentation of adipose tissue, (ii) threshold segmentation, region growing and hole filling for the segmentation of bone and (iii) linear registration followed by the non-rigid registration with the Morphon for the segmentation of the rectum and the prostate.

For all the considered data sets the JJ2016 algorithm performed better than the MK2014v2 for the segmentation of adipose tissue and bones. Unlike the adipose tissue segmentation of the MK2014, the adipose tissue segmentation of the JJ2016 algorithm was not affected by the amount of air surrounding the patient. This resulted in a more robust result. Additionally the algorithm segmented smaller regions of adipose tissue. For the segmentation of the bones, the largest improvement was achieved by including the local maximum mask in the hole filling algorithm for the compact bone. Then less unwanted tissues were incorrectly classified as bones.

The JJ2016 estimates a likely position for the prostate and the rectum, by using
the prior knowledge contained in the atlas volume. The combination of linear and non-linear registration gave a good estimation of their position (the center point) in most of the cases. Their size, however, differed from the true size in most cases, since only their position with respect to the bones was taken into account.

The JJ2016 algorithm performs accurately enough for the intended use by DIRA. However, further optimization to reduce computational time is needed since it is too slow for interactive use in its current state. Additionally, a larger number of data sets as well as a validation through clinical trials is necessary in order to cover a wider variety of physical differences.

### 7.4 Future work

Possible enhancements of the JJ2016 algorithm have already been mentioned in Sections 5.6 and 6.7. The next step of the JJ2016 would be to extend the segmentation algorithm to use data of a dual energy CT instead of a single-energy CT. This can be seen as a natural step since DIRA also works with dual energy.
Appendix A

Segmentation results for the bones, rectum and prostate by $R_L$, $R_N$ and $R_{LN}$. The results of the algorithms are discussed in Section 6.5.3.

Figure A.1: Results from $R_L$ (left column), $R_N$ (middle column) and $R_{LN}$ (right column) for data set 1 and 2.
Figure A.2: Results from $R_L$ (left column), $R_N$ (middle column) and $R_{LN}$ (right column) for data set 3, 4 and 5. The first row shows data set 3, the second row data set 4 and the last two data set 4. The second CT image of data set 2 was selected to show that this approach cannot handle bowel gas in the rectum.
Figure A.3: Results from $R_L$ (left column), $R_N$ (middle column) and $R_{LN}$ (right column) for data set 6 (first row), data set 7 (second and third row) and data set 8 (last row).
This appendix shows the segmentation results of the JJ2016. The JJ2016 algorithm as well as its results are described in Chapter 7.
Figure B.1: Data set 1 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.2: Data set 2 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.3: Data set 3 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.4: Data set 4 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.5: Data set 5 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.6: Data set 6 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.7: Data set 7 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.
Figure B.8: Data set 8 segmented by the JJ2016 algorithm. The segmented tissues are colorized as follows: adipose tissue in green, bone in yellow, prostate in blue, rectum in red and air inside the body purple.


M.R. Cheung and K. Krishnan. Using manual prostate contours to enhance


Futoshi Yokota, Toshiyuki Okada, Masaki Takao, Nobuhiko Sugano, Yukio Tada, and Yoshinobu Sato. Automated segmentation of the femur and pelvis from 3D CT data of diseased hip using hierarchical statistical shape model of joint


Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Julius Jeuthe