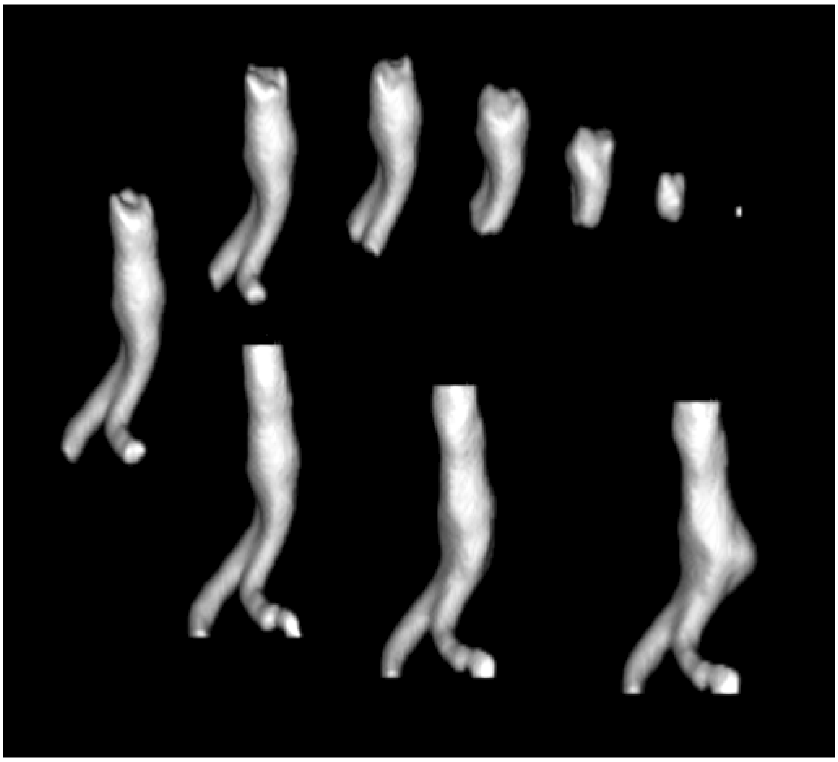
Automatic segmentation and registration of CT and US images of abdominal aortic aneurysm using ITK



Bjørn Hanch Sollie

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology,
Mathematics and Electrical Engineering

DIPLOMA THESIS

FACULTY OF INFORMATION TECHNOLOGY, MATHEMATICS AND ELECTRICAL ENGINEERING

NTNU

Candidate:	Stud.Techn. Bjørn Hanch Sollie
Discipline:	Mathematics
Date started:	February 26, 2002
Date due:	July 23, 2002
New date due:	August 20, 2002
Title:	Automatic segmentation and registration of CT and US images of abdominal aortic aneurysm using ITK

Thesis formulation: The thesis is about automatic segmentation and registration of abdominal aortic aneurysm (AAA) as seen in postoperative CT images, using the Insight segmentation and registration toolkit (ITK). The focus should be set primarily on the segmentation. With this background, the thesis is expected to contain:

- Theoretical background for the employed segmentation algorithm.
- Theoretical background for the employed registration algorithm.
- Segmentation of AAA as seen in postoperative CT images.
- Registration of postoperative CT and US images of AAA.
- Evaluation of the usefulness of ITK for solving these problems.

The diploma thesis is to be carried out at the Department of mathematical sciences under the supervision of Ketil Bø and Harald Hanche-Olsen, in cooperation with Sintef Unimed under supervision of Frank Lindseth and Jon Harald Kaspersen.

Trond Digernes
Chair
Dept. of Mathematical Sciences

Harald Hanche-Olsen
Associate Professor
Dept. of Mathematical Sciences

Preface

This diploma thesis was written at the Faculty of Information Technology, Mathematics and Electrical Engineering for the dept. of Mathematical Sciences at the Norwegian University of Science and Technology (NTNU). The thesis was done in cooperation with Sintef Unimed, which is part of Sintef, The Foundation for Scientific and Industrial Research. The Sintef group is the largest independent research organization in Scandinavia.

The author of this thesis is Bjørn Hanch Sollie. Supervisor from the dept. of Computer and Information Science was Associate Professor Ketil Bø. Supervisor from the dept. of Mathematical Sciences was Associate Professor Harald Hanche-Olsen. External supervisors at Sintef Unimed were Frank Lindseth and Jon Harald Kaspersen.

I started the work on this thesis with no prior knowledge of image processing or medicine in general or medical imaging in particular. This work has been very exploratory in the sense that none of my supervisors or I had any prior experience with the in-development medical imaging tool used, the Insight Segmentation and Registration Toolkit (ITK). The ability to learn and use known and unknown elements from the sciences of mathematics, programming, medicine and image processing, has been essential. Additionally, achieving a firm grasp of both the basics and some more complex components of medical imaging from the bottom up, has also required a lot of learning. In hindsight I can safely say that it has been both a fun and rewarding experience.

Lastly, the way I got involved with this work is a matter of funny and bizarre coincidence, so much as to be worth mentioning here: When the time came to start the work on my diploma thesis I still hadn't managed to decide what I wanted to work with, despite a considerable search. Time continued to pass by, and by late fall last year the issue of finding an appealing topic was starting to become a matter of worrisome inconvenience. This was underlined by the fact that all I really knew about what I wanted to do was that it preferably include some practical and immediately useful work. Then, one evening, when I was flipping through the channels on my TV, and this very matter was occupying my mind, there was one program on that caught my attention. It was a report about Sintef Unimed and their recent innovations and work in the field of medical imaging. Here was the potential for something to do which was both practical, interesting and useful and which could possibly even carry the reward of being fun. I jotted down a few names, as they appeared in the interviews on the TV screen. The next day I contacted the people at Sintef Unimed with the hopes of appointing a meeting on the matter, the eventual result of which is described on the next one hundred or so pages.

Trondheim, August 20, 2002

Bjørn Hanch Sollie

Abstract

The goal of this project was to perform automatic segmentation and registration of the inner and outer aortic wall in abdominal aortic aneurysm as seen in post-operative CT and US images. These tasks were performed by using the existing framework provided by the Insight segmentation and registration toolkit (ITK), a new in-development software toolkit for performing segmentation and registration. An evaluation is to be provided of the current usefulness of ITK to perform the cited tasks.

The methods explored for segmentation include use of the watershed algorithm, fuzzy connectedness and level sets, while for registration, the use of mutual information was investigated.

The achieved results are mixed. A scheme to perform segmentation of the inner and outer aortic walls with minimal user intervention has been presented. The segmentation is performed automatically after manual selection of only four initial values. The spatial extent of the segmented structure includes a region from below the renal arteries to the top of the iliac arteries, below the aortic bifurcation point in the lower abdomen. Searches indicate that no such scheme has previously been presented. The use of watershed and fuzzy connectedness algorithms respectively are also discussed. Using using mutual information to automatically register CT and US images, with the use of two different image alignment optimizers, did not produce satisfactory results.

The achieved results of the segmentation indicate that ITK is a medical imaging tool with great potential. The achieved results of the registration indicate that it is a bit too early to make full use of the toolkit in clinical applications. The current limitations of the ITK framework are thought to have been met for both of our specific problems, and thus the goals of the project were achieved.

Acknowledgements

Special thanks to the friendly staff at Sintef Unimed for offering me this work.

Special thanks to Frank Lindseth, Jon Harald Kaspersen, Ketil Bø and Harald Hanche-Olsen for all the help and assistance they have provided.

Thanks to Luis Ibanez, Joshua Cates and the other members of the insight-users mailing list for patiently and thoroughly answering my questions on several occasions.

Thanks to Bjørn Olstad for help and advice prior to starting this work.

Thanks to Sven Loncaric and Marko Subasic for sharing with me some of the details of their work presented in [LONCA-01].

Thanks to Marleen de Bruijne for providing me with an article ([BRUIN-02]) about the most recent work by her research team at the Image Sciences Institute at the University Medical Center in Utrecht in the Netherlands.

Thanks to Toril Nagelhus Hernes for reading and providing feedback on my report.

Thanks to everyone I forgot to mention.

Contents

List of Figures

1 Introduction

In this chapter we present a brief introduction and explanation for some important terms and concepts essential to understand and to solve the tasks at hand. Starting with our motivations and definition of the problem, we continue by explaining the basics of computer tomography (CT) and ultrasound (US) imaging, abdominal aortic aneurysm (AAA), image segmentation and image registration.

1.1 Motivation

The interest for segmentation and registration of medical images has greatly increased over the past decades. Our knowledge of the causes and treatment of medical conditions has increased by orders of magnitude. With the advances in computer technology in general, and processing power and image acquisition techniques in particular, the amount of research in the field of medical imaging has grown dramatically.

The introduction of x-ray computer tomography (CT) 25 years ago revolutionized medical imaging. CT provided the first clear cross sectional images of the human body with substantial contrast between different types of soft tissues. Since then, medical imaging has increasingly become a more important tool in all stages of patient treatment. Today, surgeons and radiologists commonly use complex visualization software to plan, simulate and monitor complicated surgery.

Image segmentation denotes the process of subdividing an image into its constituent parts or objects [RCEGW-93], while image registration denotes the process of bringing the involved pictures into spatial alignment [VIERG-97]. An imaging modality refers to a specific way of acquiring images, such as CT or ultrasound (US) for example. Multimodal registration refers to registration of images acquired through different image acquisition techniques (such as CT and US).

Accurate and reproducible segmentation and registration schemes are becoming more important in view of the rapid increase in the use of three-dimensional imaging modalities. An accurate segmentation allows for accurate quantitative and morphological analysis and is indispensable for proper visualization and interpretation of images, for preoperative planning and for postoperative assessment. Manual segmentation and registration of, especially three-dimensional, images are time-consuming and hence expensive tasks. Manual segmentation is also subjective and thus cannot be reproduced, and often a high level of expertise is required.

Minimally invasive endovascular surgery on the abdominal aorta and postoperative assessment after such surgery are areas in which the use of automation and multimodal imaging is becoming a promising, realistic and viable possibility. The abdominal aorta is a delicate and crucial part of the human body, and developing new and effective procedures to reduce the risks of treatment is therefore essential. This is also true in the treatment of abdominal aortic aneurysm, a disease which will commonly lead to serious impairment or death, if left untreated.

CT is the primary tool for patient followup assessment today. By employing ultrasound equipment instead, patients may be spared from going through up to several CT sessions, thereby reducing health risks from x-ray radiation considerably, as ultrasound equipment is non-radiating. In this context, development of an automatic segmentation and registration scheme for CT and US has the potential to contribute to both safer and better treatment.

Software development is generally both time-consuming and costly. With the development of ITK, the Insight segmentation and registration toolkit, the medical community will receive a new tool, freely available to anyone, specifically designed for segmentation and registration of images in medical settings. ITK has the potential to make development of software for medical imaging applications easier, faster and more cost-effective. Its potential makes it well worth for the medical imaging communities to make use of the new software and assess how well it performs for the specific need of each community; in this case segmentation and registration of abdominal aortic aneurysm as seen in CT and US images.

1.2 Problem definition

1.2.1 Background

The tools used to implement the software are the C++ programming language and the Insight segmentation and registration toolkit (ITK). As the central problems are segmentation and registration, and not developing ITK, the focus has been kept on using the ITK package as-is. Extra functionality was only implemented when strictly necessary, or when the time-cost of adding needed features was not critical.

Although focusing strictly on either segmentation or registration would have been possible, there was agreement between all supervisors and the candidate from the beginning that the study include both a segmentation problem and a registration problem. This choice was made to test and evaluate both of the two main branches of functionality in ITK. The segmentation problem is presented in the most detail.

1.2.2 Segmentation of the CT images

We seek to extract the structures of the inner and outer aortic walls in postoperative CT images of patients with abdominal aortic aneurysm. The extracted structures should contain all parts of the abnormally dilated aortic tissue and the inner and outer aortic wall in the height of the surgically inserted stent graft. Thus, the segmented structures will represent the aorta from below the renal arteries to the top of the iliac arteries, including the aortic bifurcation point. See figures 1.1 and 1.2 for illustrations.

1.2.3 Registration of the CT and US images

We seek to register the postoperative CT images with postoperative followup US images using rigid registration techniques only. Although the CT and US images are initially registered using a marker and positioning system, this initial registration contains inaccuracies, and the objective is to improve it.

1.2.4 Evaluation of ITK

A conclusive evaluation of the current usefulness of ITK for solving the segmentation and registration problems described is provided.

1.3 Abdominal Aortic Aneurysm (AAA)

1.3.1 Introduction to AAA

Abdominal aortic aneurysm (AAA) denotes the disease in which the infrarenal abdominal aorta tends to increase in size, either slowly or suddenly, resulting from weakened arterial walls. Aneurysms may occur in any blood vessel in the body, but the most common place in the abdomen is on the aorta between the renal arteries and the aortic bifurcation point in the lower abdomen. An illustration of this can be seen in figure 1.1.

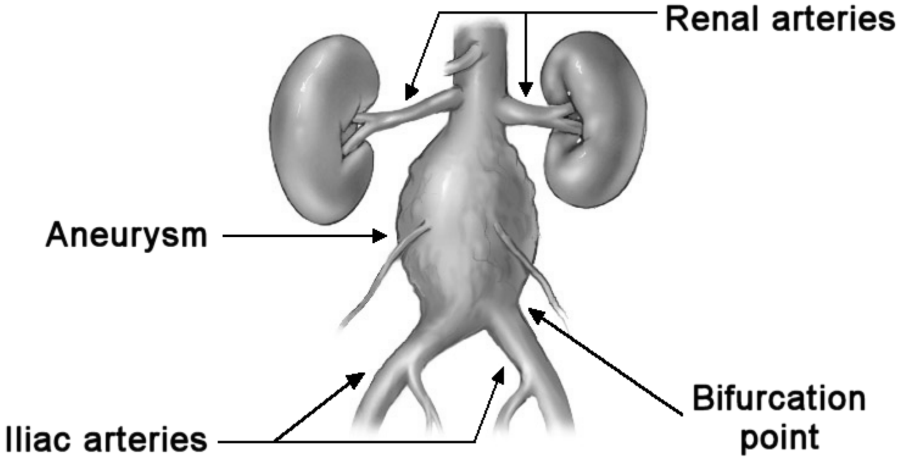


Figure 1.1: Abdominal aortic aneurysm.

An AAA is usually diagnosed when an increase of more than 50 % of the aortic diameter is detected relative to a normal healthy diameter [RAVHO-98], or when the diameter is bigger than 50-55mm and increasing. Once present, AAAs may continue to enlarge and, if left untreated, become increasingly susceptible to rupture, usually resulting in lethal hemorrhage [MAGEE-00]. AAA is the 13th major cause of death in the United States [BELKI-94], and occurs in up to seven percent of people of age 60 and older.

1.3.2 Repair surgery

Worldwide, approximately 100,000 interventions for AAA repair are performed each year, of which around 30 % are endovascular [ECALL-97]. An AAA expanding at a faster rate than 5mm over a period of six months is perceived to be at a high risk for imminent rupture, usually prompting surgical repair [BROWN-92]. During the endovascular repair surgery, a synthetic stent graft is positioned inside the aortic lumen to correct the blood flow and to reduce stress on the aortic walls. See figure 1.2 for an illustration.

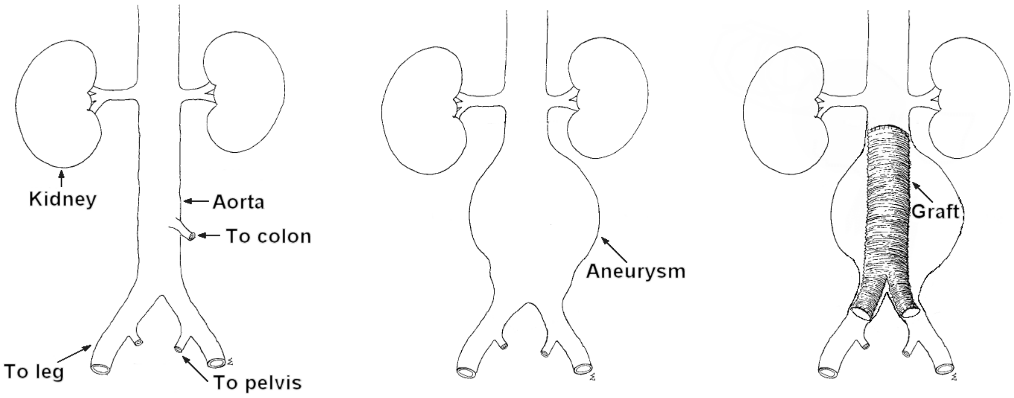


Figure 1.2: Treatment of abdominal aortic aneurysm. To the left, a healthy aorta. In the middle, a diseased aorta prior to surgery. To the right, an aorta after endovascular surgery, repaired with a stent graft.

Progressing aneurysmal disease after surgery and damage to or fatigue of the graft material may result in leakage, curling, twisting and migration of the graft. Complications of this nature may eventually result in rupture or occlusion [BRUIN-01]. As a consequence of this, careful and frequent patient followup is required. A patient is imaged every three to twelve months, depending on the state of the aneurysm.

After surgery, the volume in the aneurysm between the graft and the aortic wall is usually filled with thrombus. In the remainder of this text, the outer aortic wall will, for the sake of simplicity, often be referred to as the thrombus region, or just the thrombus. Also, the inner aortic wall, which includes the stent graft and the region inside the aorta with unobstructed blood flow, will regularly be referred to as the lumen region or just the lumen, unless otherwise noted.

The surgically inserted stent graft is made up of a woven polyester tube (usually gore-tex) covered by a tubular metal mesh (usually stainless steel). An example of what such a graft may look like can be seen in figure 1.3.

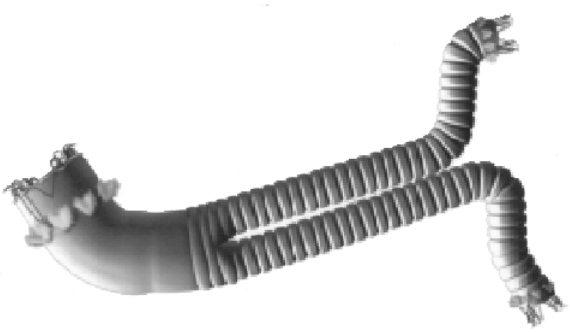


Figure 1.3: A stent graft for surgical repair of abdominal aortic aneurysm.

1.3.3 Detection and condition assessment

Ultrasound is the imaging modality most frequently used to determine if a patient has an abdominal aortic aneurysm [BLANK-00]. The most widely used method for further AAA planning and condition assessment is computer tomography (CT). Intravenous injection of contrast during CT image acquisition provides good enhancement of the abdominal aorta.

The followup examination procedure usually includes some form of aneurysm delimitation. As of today, this procedure is most commonly performed with some degree of manual intervention. As previously mentioned, the problems with this is that performing this task manually is time consuming, thus expensive, and it is subject to different radiologists producing different results.

To reduce analysis time, reduce variability and to increase reproducibility, automatic segmentation of the abdominal aorta and the aneurysm would be of great value. Unfortunately, CT images of AAA are difficult to segment, because the outer aortic boundary is often obscured by surrounding tissue of similar density. There are also lot of other structures close in proximity to the aortic wall, which will frequently reduce the visibility of edges.

The radius of the aneurysm may also vary greatly over a short distance, and variations in size and shape may be large between patients as well as in one patient over time. This can make the boundary difficult to detect even when surrounding structures are absent. Lumen and thrombus texture and grayvalue can vary with the presence of calcifications, graft metal, intravenous contrast and differences between individual CT scanners.

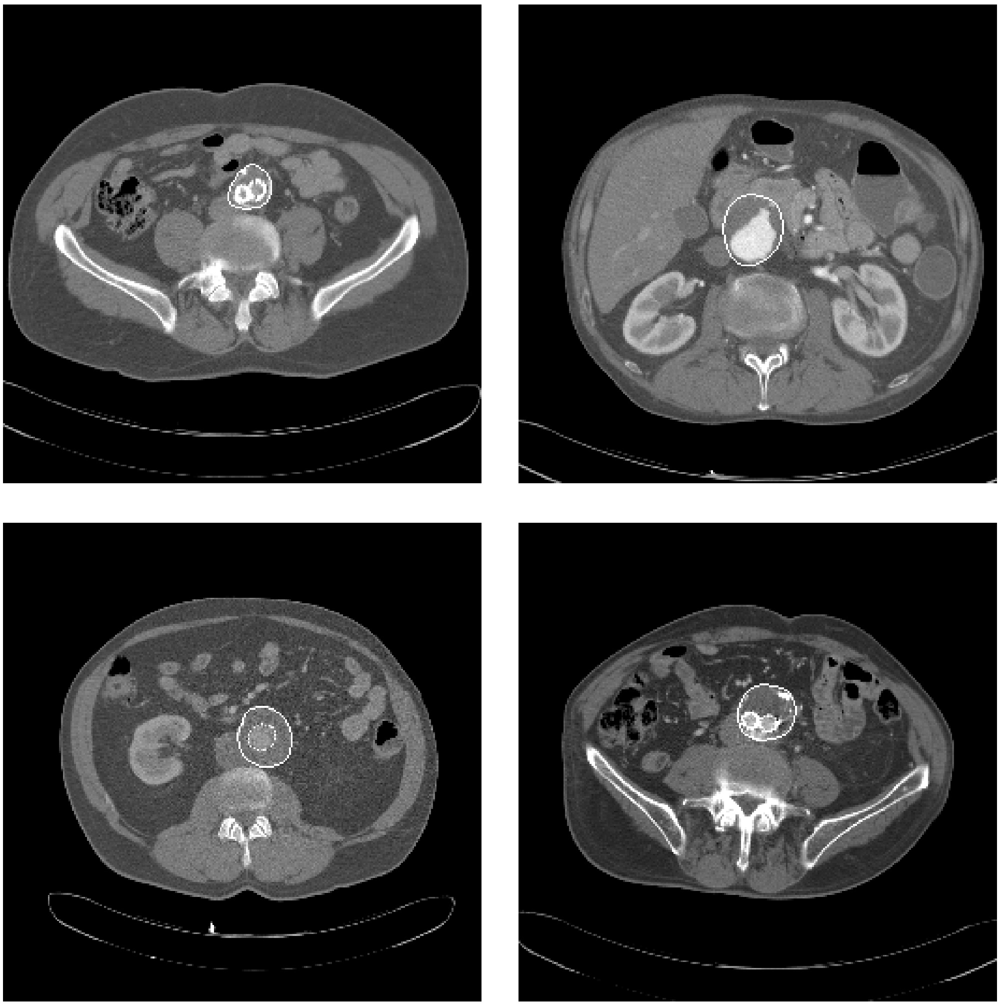


Figure 1.4: Examples of abdominal aortic aneurysms. The outer aortic wall (thrombus region) has been manually delimited by a solid white line.

1.4 Computer tomography (CT) imaging

Computer tomography (CT), also referred to as computer assisted tomography (CAT), is a method of obtaining image data from different angles of different parts of the body using x-rays. With the help of a computer, this information is processed to create a cross sectional view of body tissues and organs.

CT imaging is a powerful imaging tool because it can show several types of tissues and materials, fluids, bone, blood vessels and internal organs with great clarity compared to most other imaging techniques. For this reason, CT is one of the best tools today for studying the abdomen. Using specialized equipment and expertise to create and interpret CT scans of the body, radiologists can more easily diagnose problems such as cancers, infectious diseases, cardiovascular disease and, in our case, abdominal aortic aneurysm.

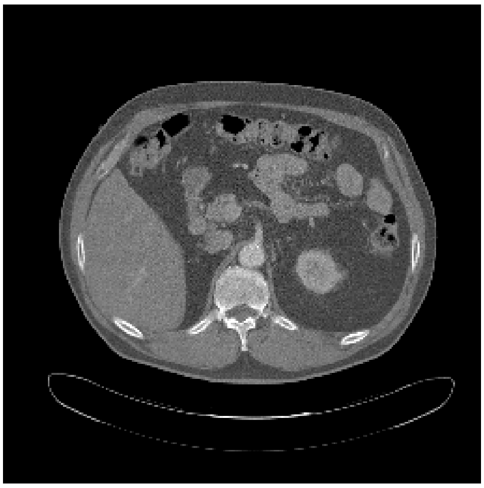


Figure 1.5: Example of a CT image.

CT imaging works by passing small controlled amounts of x-ray radiation through the body [RSNAW-02]. Different materials and tissues inside the body absorb variable amounts of radiation, and the differences in the level of radiation emerging on the other side is recorded by an array of detectors, which measure the x-ray profile. This is in contrast to conventional x-ray radiology, where the x-rays passing through the imaged object are instead captured on a special film.

A rotating gantry inside the CT scanner has an x-ray tube mounted on one side and an arc-shaped detector on the opposite side. An x-ray beam is emitted in a fan-shape as the x-ray tube and detector rotates around the patient. Each time the tube and detector makes one full rotation, the image of a thin section is acquired.

During each rotation, the detector records about 1,000 profiles of the expanded x-ray beam. Each profile is then reconstructed by a dedicated computer into a two dimensional cross-sectional image, or slice, of the section that was scanned.

When this is done multiple times in succession, while moving the patient's body a small distance relative to the frame for each time, the result is a set of multiple images which may be assembled to give a detailed three-dimensional view of the interior of the patient's body.

Advantages:

- CT examinations are fast and simple and can quickly reveal internal injuries and bleeding.
- CT imaging has been shown to be a cost-effective tool for a wide range of clinical problems.
- CT imaging offers detailed views of many different kinds of tissues.
- CT imaging is painless, noninvasive and accurate.
- Through use of CT scanning, it is possible to identify both normal and abnormal structures. This makes it a useful tool for guiding radiotherapy, needle biopsies and other minimally invasive procedures. In many cases this can eliminate the need for invasive surgery.

Disadvantages:

- CT involves exposure to radiation in the form of x-rays. The typical radiation dose from a CT exam is equivalent to the natural background radiation received over a year's time. Special care must be taken during x-ray examinations and the patient's abdomen and pelvis should normally be shielded by a lead apron. In Norway alone, it is estimated that 40-50 patients develop fatal cancer every year, due to exposure to x-rays from CT scanners [TNRPA-02].
- CT exams are generally not recommended for pregnant women.

Limitations:

- Very fine details in soft tissue cannot always be seen with CT imaging. In some situations, soft tissues may be obscured by bone structures. In these cases, magnetic resonance (MR) imaging may be preferable.
- Using CT imaging as a means of guidance during patient surgery is inconvenient, as the patient will have to be moved in and out of the CT scanner each time an updated image is needed.

1.5 Ultrasound (US) imaging

Ultrasound (US) imaging, also referred to as sonography, is a method of obtaining images of the inside of the body through the use of high frequency sound waves. Ultrasound imaging is based on the same principles involved in the sonar used for navigation by ships at sea. As a controlled sound bounces against an object, the echoing waves can be used to identify how far away the object is, how big it is, and how uniform it is.

In preparation for the procedure, the skin of the area to be examined is exposed and coated with a special gel. This gel serves to ensure that there is no air between the ultrasound transducer and the skin during the time of image acquisition, thus reducing noise and providing a clearer picture.

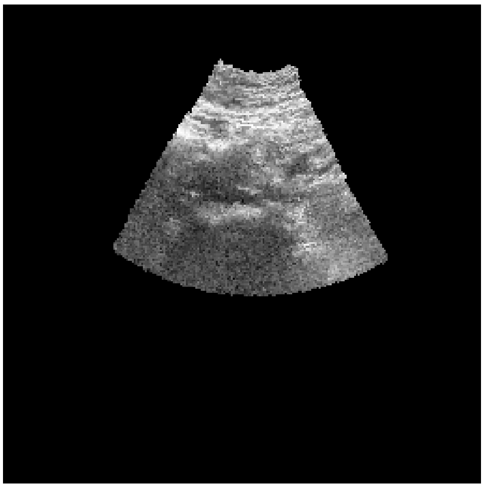


Figure 1.6: Example of an ultrasound image.

An ultrasound transducer functions as both a loudspeaker (to create the sounds) and a microphone (to record them) [RSNAW-02]. When the transducer is pressed against the skin, it directs a stream of inaudible, high-frequency sound waves into the body. As the sound waves echo from the tissues and structures inside the body, the microphone in the transducer records small changes in the direction, intensity, frequency and wavelength in the reflected sound [SHOLM-98]. These signature waves are measured by a computer, which converts them into a real-time moving picture. Still frames of the moving picture may be captured to produce a series of images, or slices. Figure 1.6 shows an example. By moving the transducer along the skin, while at the same time measuring its physical position, it is possible to create a three-dimensional view of the inside of the patient's body.

Advantages:

- Unlike CT, ultrasound does not use x-rays or any other kinds of potentially harmful radiation.
- Ultrasound equipment can produce moving images in real-time.
- Ultrasound has been used for abdominal examinations for about 40 years, and for standard diagnostic ultrasound there are no known risks or harmful effects to humans.
- Ultrasound is a cost-effective means of image acquisition in medicine.

Disadvantages:

- The patient has to undergo a slightly more intrusive session than is the case with a CT session, including the removal of clothes and application of the gel.
- The quality of the recorded images is dependent on the operator's skill of handling the equipment.

Limitations:

- Ultrasound imaging produces images that are far inferior in quality to CT. Proper identification of structures and regions in the finalized ultrasound images generally requires personnel with expertise and training to do so.

1.6 Image segmentation

1.6.1 General image segmentation

Image segmentation denotes the process of subdividing an image into its constituent parts or objects [RCEGW-93]. The amount of subdivision performed is dependent of the problem, so that the segmentation should stop when the structures of interest have been isolated.

In general, autonomous segmentation is one of the most difficult tasks in image processing [RCEGW-93]. This step in the process determines the eventual success or failure of the image analysis. In fact, effective segmentation rarely fails to lead to a successful solution. For this reason, considerable care should be taken to improve the probability of getting a segmentation output of high quality.

1.6.2 Image segmentation in medicine

In medical sciences, image segmentation allows us to do volume measurements, generate 3D models for visualizing complex structures, see the placement of structures in relation to each other, and to perform better preoperative planning, interoperative guidance and postoperative control.

The objective of segmentation of medical images is generally to find regions which represent single anatomical structures. Segmentation is a crucial step in building systems for the further analysis of an image.

The availability of regions which represent single anatomical structures makes tasks such as interactive visualization and automatic measurement of clinical parameters directly feasible. In addition, segmented images can be further processed with computers to perform higher-level tasks such as shape analysis and comparison, recognition and other kinds of decision-making.

Unfortunately, automatic segmentation of medical images is a very difficult task. This is due to noise, masking of structures, individual variations in biological shape, tissue inhomogeneity and more. Completely automated methods that are fool-proof and that have been demonstrated to work correctly routinely in trials involving a large number of patient studies do not seem to have been constructed yet [UDUPA-00].

1.7 Image registration

1.7.1 General image registration

Image registration denotes the process of bringing the involved pictures into spatial alignment [VIERG-97]. In other words, image registration denotes the process of matching two images so that corresponding coordinate points in the two images correspond to the same physical region of the scene being imaged. This is done by calculating an optimal transformation matrix between the two images.

To do this, one image is selected as the fixed image and the other as the moving image. Following the definitions of the terms, as they are used in ITK, the fixed image is then moved relative to the moving image, guided by an optimization function which measures how well some predefined features of the images correspond to each other. Thus, the method generally works by minimizing an error function or maximizing a suitable quality function.

The registration process basically takes pixels from the fixed image (or voxels, which are the equivalents of pixels in three dimensions) and map their spatial location through a transform into the geometric space of the moving image. This means that the moving image should be the image of greater resolution and extent, as the time to compute the optimal transformation will be shorter.

Once an acceptable registration has been calculated, image data may be transformed (or resampled) into the coordinate system of the other image, or combined with the other image.

1.7.2 Image registration in medicine

In medical imaging registration is necessary primarily in four different situations [UDUPA-00]. In the first case, images are acquired for the same body region from different modalities, for example CT and US. By combining images from different modalities, registration can help improve the visual accuracy of the imaged region. In the second case, images are acquired for the same body region using the same modality at different points in time. The distance in time may be small for studying the motion or displacement of a structure inside the body, or bigger for studying the growth or change of a structure. In the third case, in certain interventional procedures, information derived from acquired image data is used to provide navigational aid for the devices used in the procedure. In these situations it becomes necessary to register the body region, and the scene. In the fourth and final case, images acquired for a given body region are matched to a computerized model of the same body region. This is often helpful for studying statistical variations in structures in a population, as well as in scene segmentation.

With the increasing use of imaging in medicine, automated registration of images has become a very important field of research. A wide range of registration techniques has been developed for many different types of applications and data. Given the diversity of the data, it is unlikely that a single registration scheme will work satisfactorily for all different applications.

2 Previous work

In this chapter, a brief overview of previous work regarding segmentation and registration of AAA images is presented. The methods regarding works about segmentation include vessel axis and border estimation approaches, neural network approaches, active shape models (ASMs), watershed approaches, region growing approaches and level set-based approaches. The studied works about registration reported the use of intensity and gradient information and mutual information. The chapter is concluded with a summary of our findings.

2.1 Segmentation of abdominal aortic aneurysm

2.1.1 Background

Compared to the number of people affected by AAA, there has been relatively little effort and funding for research to explore and develop new methods of treatment for the disease [TILSO-02]. While a lot of lot of different approaches have been researched in the area of segmenting vessels in general, [JENSE-01], relatively few works deal with segmentation of AAA. Out of the works that deal with AAA, several deal with segmenting the inner aortic wall or the stent graft only, while the much more difficult problem of segmenting the outer aortic wall and the aneurysm is only relatively scarcely covered, in comparison. We wish to find out what approaches have been attempted for segmentation of AAA in the past, and implement a scheme based on the current framework for segmentation in ITK.

2.1.2 Vessel axis extraction and border estimation approaches

A method for automated central vessel axis extraction and border estimate is presented in [OWINK-00]. In [VERDO-96], a method to determine the lumen boundary is established through dynamic programming using slices reformatted to be perpendicular to that axis.

According to [BRUIN-01], these methods work best in cases where the patient has received a graft with radiopaque markers sewn onto the outside of the graft, which produce artifacts in the image, signaling the position of the graft. Also, the strategy as presented suffers from the drawback of being unable to satisfactorily handle bifurcated vessels.

2.1.3 Neural network approaches

The method outlined in [SMADA-95] uses a neural network to learn thresholds for multilevel thresholding and a constraint-satisfaction neural network to smooth the boundaries of labeled segments. After segmentation, a small number of images are edited manually, before a connectivity procedure automatically selects corresponding segments from other sections by comparing adjacent voxels within, and across, sections for label identity.

The results suggest that automated segmentation followed by manual editing is a promising approach to segmentation of CT images of AAA. The biggest problem with this approach with regard to our motivations however, is that ITK at present has no tools for neural network segmentation at all.

2.1.4 Active shape model (ASM) approaches

[BRUIN-01] presents a method for segmentation the outer aortic wall of abdominal aortic aneurysms, based on active shape models (ASMs), as put forward by Cootes and Taylor in [TAYLO-95], [TAYLO-00] and [TAYLO-01]. Active shape models combine statistical knowledge of object shape and shape variation with local appearance models near object contours. A model generated from grayvalue profiles in training images is used to fit the shape model to the image. Subsequent fitting in sequential slices is performed, using the contour obtained in one slice to initialize the contour in the adjacent slice. Two significant modifications with respect to the conventional ASM approach are reported. The first involves the correlation with grayvalue profiles of adjacent slices, rather than grayvalue profiles obtained from the training set. The second involves the extension of the scheme with a penalty function for inclusion of low-intensity tissue and a refinement step to locally adjust the position of the landmark points to points with maximum gradient. The results are reported to outperform the conventional ASM significantly with these extensions. Further improvements and results to this approach are presented in [BRUIN-02], again confirming its promising potential.

Although accurate and robust, the slice-by-slice scheme outlined contains no details on how to handle bifurcated vessels, as the method described is specifically devised to segment the outer wall of the aneurysm region only. Also, the method requires extensive manual initialization, and may require some user intervention underway.

2.1.5 Watershed-based approaches

No previous work has been found on using the watershed algorithm for segmentation of abdominal aortic aneurysm. However, the algorithm is well implemented in ITK, and the supervisors at Sintef Unimed considered this as an interesting approach with good potential.

2.1.6 Region growing approaches

Region growing algorithms build on the principle of allowing a number of seed points to grow into a region in the image as long as the addition of new points to the region doesn't violate defined constraints. [POHLE-00] outlines a fully automatic region growing algorithm that learns its homogeneity criterion automatically from characteristics of the region to be segmented. The method is based on a model that describes homogeneity and simple shape properties of the region. Parameters of the homogeneity criterion are estimated from sample locations in the region. These locations are selected sequentially in a random walk starting at the seed point, and the homogeneity criterion is updated continuously. The methods were tested by segmenting the inner aortic wall in abdominal aortic aneurysms, among other structures, in CT and MR images.

The method is reported to be robust and produce reliable results, as long as the assumptions the model makes about homogeneity and region characteristics hold. As ITK encompasses the required tools for this type of segmentation (fuzzy connectedness), this method seems like a suitable approach, at least for the segmentation of the lumen.

2.1.7 Level set-based approaches

In [MAGEE-00] a level set based method for the segmentation of complex anatomical structures from CT images is reported. The level set method is based on the work by J. A. Sethian described in [JASET-99]. The method is concluded to have much promise in the area of 3D arterial segmentation if the application is not time-critical. The only cited disadvantage to the level set method is the computational cost involved.

[LONCA-01] also presents a technique for segmentation of AAA from CT images using level sets, additionally incorporating narrow banding. The inner aortic border is initially segmented using 3D level sets, while 2D level sets are used to segment the outer wall, using the output of the initial segmentation as a zero level set. The stopping criterion is based on curve expansion speed designed to keep the boundary from growing into surrounding tissue. Their experiments with this scheme are cited to have shown good results.

The strengths of the level set method lies in its generality as it is able to handle image data of different dimensionality equally well and handles topological changes satisfactorily. It encompasses mechanisms to handle regions with lacking boundary information, and it has been demonstrated to be readily able to handle bifurcated vessels. This looks like a promising approach for segmenting both the lumen and the thrombus.

2.2 Registration of CT and US images

2.2.1 Background

Although much work has been done in the area of multimodal image registration, much less work has been conducted on the specific problem of registering CT and US images. The work reported in [MAINT-98] also seems to confirm this. Based on our findings, and the available selection of registration methods in ITK, the most appropriate method will be chosen.

2.2.2 Gradient and intensity information approaches

A technique to rigidly register intraoperative three-dimensional ultrasound images with preoperative MR images is demonstrated in [ROCHE-01]. Images are automatically registered by maximization of a similarity measure which generalizes the correlation ratio, which involves incorporating multivariate information from the MR data, both intensity and gradient. In addition, the similarity measure is built on an intensity-based distance measure, which makes it possible to handle a variety of US artifacts.

The registration errors reported are of the order of the MR image resolution at worst. The method looks very promising, but unfortunately ITK doesn't yet have the required tools for performing this type of registration.

2.2.3 Mutual information approaches

In [FMAES-97], a method for registering multimodal images is reported. The method presented applies mutual information, or relative entropy, to measure the statistical dependence or information redundancy between the image intensities of corresponding voxels in images, which is assumed to be maximal if the images are geometrically aligned. The method is validated for rigid body registration of computed tomography (CT), magnetic resonance (MR), and photon emission tomography (PET) images. In [UNSER-00] mutual information is used with a multiresolution optimizer to achieve a registration accuracy of about a tenth of a pixel under very noisy conditions using normal photographs. [GRIMS-00] reports the use of a mutual information-based registration algorithm which establishes the proper alignment via a stochastic gradient ascent strategy. Their primary achievement is improved execution time of the algorithm.

The results indicate that sub-voxel accuracy can be achieved completely automatically and without any prior segmentation, feature extraction, or other preprocessing steps. Although little work is reported on the use of mutual information to register CT and US images, ITK has most of the tools for doing this type of registration implemented. Thus, the potential for registering CT and US images using this technique remains unknown, but promising.

2.3 Summary

Studies of previous work indicate that segmentation of AAA as seen in CT images and registration of CT and US images are problems that have received relatively little attention in the past. Much of the work devoted to create automatic segmentation schemes of AAA from CT images has been mostly exploratory and experimental, and there has been relatively little focus on developing functional end products. A common denominator for much of the earlier work regarding our particular segmentation problem is that the focus is set on a considerably smaller problem than the one we're interested in. Relatively few works deal with all the issues of segmenting the inner, and especially the outer, wall of the aorta from below the renal arteries to the iliac arteries, including the bifurcation point. Instead, the focus often remains on one of the following two problems:

- Segmentation of the outer aortic wall in healthy patients only, avoiding the very difficult problem of thrombus segmentation and vascular structures with irregular anatomy.
- Segmentation of only the thrombus region in AAA patients, avoiding the problems associated with the segmentation of bifurcated vessels and vascular structures with a more complex topology.

One of the reasons for the tendency towards opting to focus on only one of the problems appears to be that schemes appropriate for vessel extraction lack the properties required for segmenting structures where edge information is scarce and where lack of graylevel information makes it hard to distinguish between relevant and irrelevant regions. On the other hand, the deformable model methods commonly used for segmentation of the dilated parts of the aorta, cannot easily deal with topologically complex structures, such as bifurcated vessels. The presented schemes, which are often very capable of handling a limited problem, often have weaknesses when used at the bigger problem we are looking at.

After extensive studies, it becomes clear that segmentation of AAA as seen in CT images is a complex and difficult task, and a scheme to perform such segmentation automatically with minimal initialization does not seem to have been devised.

Little work on registration of CT and US images of AAA were found, and automatic schemes to perform such registration automatically do not seem to have been reported.

3 Materials and methods

In this chapter, we first present a general problem solving strategy and some general solution criteria for the segmentation and registration schemes to be devised. We then present an overview of ITK and have a brief look at the tools it encompasses for segmentation and registration of medical images. Last in this chapter, a more detailed background is presented on the theory of the level set method for segmentation and the mutual information method for registration. A less elaborate background on the watershed and fuzzy connectedness segmentation algorithms is also provided.

3.1 Problem solving strategy

3.1.1 Segmentation

The basic strategy, as discussed and agreed on with the supervisors at Sintef Unimed, was to first find out which algorithms could be effective for solving the problem. The least complex schemes would be tested first, and in the case of a scheme producing unsatisfactory results, the scheme would be abandoned and a more advanced scheme would be introduced to replace it.

Based on the literature and previous work studied, the task of segmenting a structure is usually divided into the following three general steps:

- Preprocessing: Enhancement of the desired structure.
- Extraction: Separation of the structure from the rest of the image.
- Postprocessing: Improvement of the extracted structure.

Preprocessing is necessary to reduce noise, enhance the relevant structures and reduce the possibility of irrelevant image features from interfering with the later analysis. The purpose of this step is to increase the chances for success when we segment the image later on. When segmenting AAA images, preprocessing will typically include applying filters for noise reduction, smoothing and contrast enhancement, for example.

Postprocessing is necessary to improve the quality and topology of the extracted structure and generally to obtain a final shape which makes sense with regards to what we know about the actual anatomy of the structure. The purpose of this step is to increase the accuracy and correctness of the extracted structure. In our problem, this typically includes applying filters, such as median filters, to reduce sharp corners and edges and improve topology.

The studies of previous work makes clear that segmenting the inner and outer aortic walls are quite different, and difficult, problems. It was therefore decided that the aortic structure would be segmented in two separate steps:

1. Segmentation of the inner aortic wall including the stent graft (lumen). This is the least difficult part to segment as the difference in gray level to the surrounding tissues is generally good due to the injected blood contrast. This step may also serve as a good indicator of the robustness of the algorithm. An algorithm producing an unsatisfactory result in this step is unlikely to perform better when applied to segment the outer aortic wall later on. The desired result of this step is an solid region outline of the lumen that may be used for initialization or some other way of general guidance or help for performing the next step.

2. Segmentation of the outer aortic wall including the dilated parts of the aorta (thrombus). This is by far the most difficult of the two steps, as contrast to the surrounding tissues may be very poor, and edge information is much weaker or may even be missing completely. One way of making this step easier to accomplish is to find a way to use the more easily obtainable lumen segmentation from the previous step for guidance. The desired output is a solid region outline of the outer aortic wall and the thrombus.

Since the first part of the segmentation is the least complicated to perform, it was assumed that this step would also be the easiest to implement with a minimal amount of manual initialization. It was therefore presumed to be a good starting point. Once a segmentation of the lumen has been achieved, the acquired structure would serve as a stepping stone for performing the second step, as it provides us with significant knowledge about the location of the outer aortic wall. Using the information we obtain about the structure in the first step has potential for reducing much of the need for manual initialization that would otherwise have been required for performing the second step.

After the initial studies of previous work, and following the recommendations of the supervisors, it was decided that the watershed, fuzzy connectedness and level set algorithms for segmentation were the techniques offering the greatest prospects of success. These algorithms were subsequently tested in the aforementioned order.

3.1.2 Registration

The basic strategy was discussed and agreed on in advance with supervisors at Sintef Unimed. The postoperative CT image should be registered with the postoperative US image. The desired output of this step is the optimal transformation matrix that aligns the two volumes in the best possible way. Care must be taken when choosing a metric and a metric for the registration method as the CT and US images have quite different properties and qualities:

- The 3D CT images are relatively noise-free, and the abdomen is imaged in full cross sections. The image contains a relatively high amount of detail. In addition to edges, it also contains regional information in the form of varying graylevels. In the images of patients who have been injected with contrast, the abdominal aorta can be seen roughly as the shape of an inverted “Y”, of relatively high intensity, stretching through most of the image from top to bottom.
- The 3D US images are generally extremely noisy and contain a much smaller region; only the part of the abdomen containing the aneurysm is contained in these images (the bifurcation is generally not included). The ultrasound image also contains edge information for the most part, and it is much more difficult to distinguish between different structures. In US images, the abdominal aorta is considerably harder to discern, and a fuzzy, roundish partial edge is often the only indicator of its presence.

In cooperation with the supervisors, it was decided that the mutual information metric should be used for registration. Maximization of mutual information is a very general and powerful registration criterion, because no assumptions are made regarding the intensities of the images, and no limiting constraints are imposed on the image content of the modalities involved. In theory, this makes it very useful for registering images with very different properties, which is the case with CT and US. The mutual information functionality was also the most complete part of the registration framework in ITK, at the time of this work.

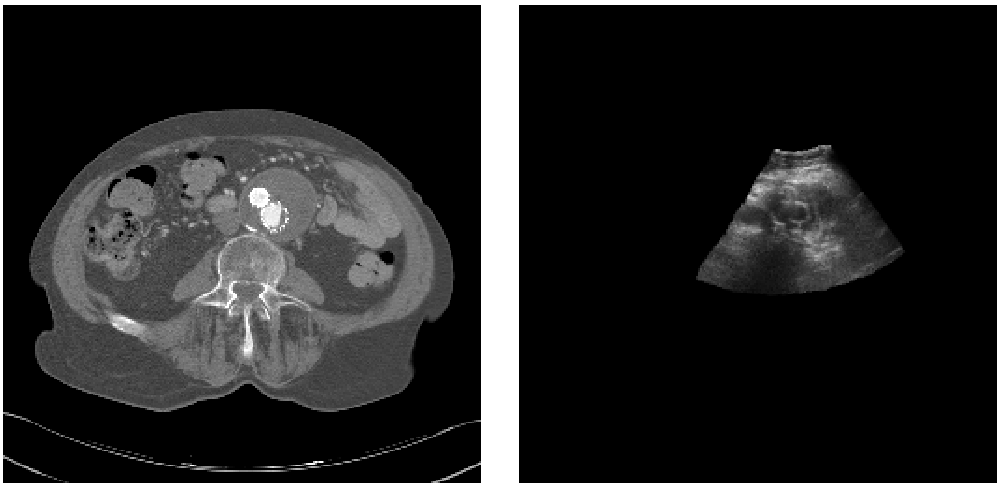


Figure 3.1: CT and US images, registered with a marker and positioning system, displaying a similar anatomical region of the abdomen.

3.2 Solution criteria

A segmentation and registration scheme with the following properties is desired:

- High degree of automatization. Any manual intervention should preferably be performed in an initialization step before starting the procedure.
- High degree of extensibility. Additions, improvements and refinements to the scheme should be easy to implement.

3.3 Visualization

As previously stated, the problem focus has been set mainly on segmentation and registration and not on visualization. The only visualization performed is that which has been strictly necessary to document, evaluate and to get a better view of the finished output. The 2D cross sectional images in this report were produced by an application, written by the author, to convert 3D image volumes to a set of 2D image slices for fast and easy viewing. The rendered 3D views of the segmented image data were created with Dynamic Imager. Dynamic Imager, a program developed by Ceetron ASA, is an easy-to-use visualization tool developed in accordance with the ISO 12087 standard.

3.4 ITK

3.4.1 Introduction to ITK

ITK is an abbreviation for the National Library of Medicine Insight Segmentation and Registration Toolkit. ITK is an open-source software system for performing segmentation and registration of data in two, three and more dimensions. The toolkit is implemented in generic (templated) C++ and is intended to be as cross-platform as possible [ITKSR-02]. The system is currently under active development and today runs under the Microsoft Windows and Linux operating systems, while efforts to port it to MacOS are underway. Additionally, an automated wrapping process exists to generates interfaces between C++ and interpreted programming languages such as Tcl, Java and Python.

ITK was developed by six principal organizations: three academic (University of North Carolina at Chapel Hill, University of Utah and University of Pennsylvania) and three commercial (GE Corporate Research & Development, Kitware and Insightful) [ITKSR-02]. Several other smaller team members and individual users also contribute actively.

ITK has been developed to support the Visible Human Project [VHPRO-02] and to be a repository of fundamental algorithms for image segmentation and registration, saving medical imaging communities from reinventing the wheel over and over again. The system is intended to serve to establish a foundation for future research, as well as providing a platform for advanced product development and conventions for future work.

The idea behind the open-source license of ITK is to open up for the possibility for developers from around the world to freely contribute to the software's further extension and development. Creating a self-sustaining community of both users and developers is cited as a main objective by the ITK development team.

The following is a summary of important points regarding the philosophy behind the toolkit.

Design:

- ITK provides algorithms for performing segmentation and registration.
- The focus is primarily on medical applications.
- ITK provides data representation in a general form for images with arbitrary dimension.
- Multi-threaded shared memory parallel processing is supported.

Architecture:

- ITK is organized around an object oriented data flow architecture. Data is represented using data objects (e.g. images). These data objects are processed by process objects (filters).
- Data objects and process objects are connected together into pipelines.
- Pipelines can process the data in pieces according to a user-specified memory-limit set on the pipeline.

Implementation:

- ITK is implemented using templated C++.
- ITK is cross-platform (Linux, Unix and Windows).
- Binding to interpreted languages such as Tcl, Python and Java is supported.
- Memory management is handled automatically through the use of so-called “smart pointers”.

ITK does not provide any tools for visualization and does not provide any graphical user interface (GUI). Also, the toolkit provides only a minimal framework for handling of files and file format. Both of these are intended to be provided by other tools.

3.4.2 Overview of the segmentation filters

ITK contains the following three different types of image segmentation filters:

1. Intensity-based segmentation filters use the intensity values of the pixels to segment an image. Usually, spatial contiguity is not considered in intensity-based segmentation filters. These segmentation filters are often used to detect structure boundaries. The following submodules exist:
   - Pixel classification filters
     - Supervised classification filters
     - Unsupervised classification filters
   - Watershed-based segmentation filters
2. Region-based segmentation filters segment an image based on similarity of intensity values between spatially adjacent pixels. These filters are often used to detect object regions. There are the following submodules:
   - Fuzzy connectedness-based segmentation filters
   - Region growing filters
   - Markov random field-based filters
3. Model-based segmentation filters segment an image by starting with a model and then updating the model based on image features. The updates are typically constrained by a priori knowledge about the models. The following submodules exist:
   - Mesh-based segmentation filters
   - Level set-based segmentation filters

As mentioned earlier, the architecture of ITK makes it possible to create hybrid filters by combining the various intensity-, region-, or model-based filters.

3.4.3 Overview of the registration filters

Registration methods in ITK are implemented by combining basic components, allowing for great flexibility. When creating a registration filter, the following components are used (as defined in ITK):

- Fixed image: This is an image that will be transformed into the coordinate system of the moving image.

- Moving image: This is the image into which we map the fixed image.
- Transform: A mapping that associates a point in the fixed image space with a point in the moving image space.
- Interpolator: A technique used to interpolate intensity values when images are resampled through the transform.
- Metric: A measure of how well the fixed image matches the moving image after transformation.
- Optimizer: A method used to find the transform parameters that optimize the metric.

A registration method is defined by selecting specific implementations of each one of the listed basic components.

The registration tools in ITK are organized in the following manner:

- Components of registration methods
    - Metrics
    - Optimizers
- Image registration methods
    - Rigid registration methods
    - Affine registration methods
    - Deformable registration methods
- Model to image registration methods
    - Pointset to image registration methods

3.4.4 Documentation

As ITK is still in an early stage of development, the only documentation for the toolkit is that which can be found online, on the ITK website ([ITKSR-02]). The documentation is provided in the form of a brief description of the API (Application Program Interface) and a suite of example- and test-programs. As of today there are no printed books yet to be found, documenting the functionality of the toolkit. Consequently, frequent and elaborate reading of source code and implementation of trial and error schemes are often necessary to acquire the needed understanding to make meaningful use of the software. The members of the insightusers mailing list, and the mailing list archive, are also very helpful and valuable sources of information.

3.4.5 Getting started

ITK comes in the form of a set of libraries. Pre-compiled versions of the libraries are not provided, so a source code archive had to be downloaded from the ITK website, configured and compiled. With ITK being in a state of relative infancy, and without any substantial documentation, getting comfortable with the API and the ITK programming style was a very challenging task. When the work on this project started, no working code was provided with ITK for either reading or writing the image files. Also, as ITK doesn't contain any kind of functionality for visualization, some tools to facilitate viewing of the results had to be implemented as well. To implement the needed functionality to handle the input and output of images and files necessitate a thorough understanding of the way ITK handles data and data processing. The needed utilities to get raw image data in and out of ITK were implemented along with tools to convert 3D volumetric images into 2D slices, needed in order to make viewing with a simple 2D viewer possible.

3.4.6 Using ITK

Segmentation and registration schemes are implemented in ITK by combining filters as illustrated in figure 3.2.

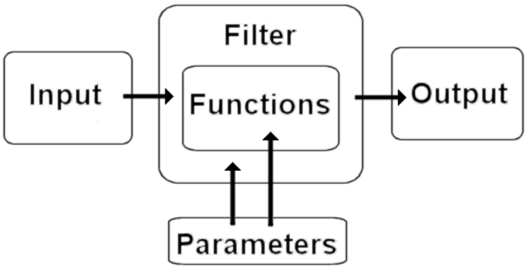


Figure 3.2: Illustration of general filter operation in ITK. The input and output are usually images, but may be other types of data objects as well. The input is usually the output of another filter. Similarly, the output is usually passed on to another filter. Connecting filters sequentially, the output data of one filter serves as the input data for the next.

The development of this project is for the most part based on the intermittent versions of ITK released after the first beta, the latter of which became available in late February 2002. Although it was initially seen as desirable to stay with the initial beta version of ITK to ensure full compatibility between all parts of the software written for this project, the increasing requirements for more complex functionality as development went on meant that intermediate development versions of the toolkit had to be used instead. Although a beta release is usually associated with a feature-locked version of the software, with the primary focus set on fixing errors, ITK remains in very active development.

3.5 Segmentation algorithms

3.5.1 Overview

As mentioned before, three segmentation schemes using three different algorithms were implemented: watershed, fuzzy connectedness and level sets. Each of these algorithms belong to each of the three different main groups of segmentation algorithms, described in section 3.4.2. This way, we also get a means of evaluating which one of the three groups of segmentation algorithms is best suited for solving the problem: intensity-based, region-based or model-based segmentation filters.

Since level sets are the centerpiece of the implemented segmentation scheme, the theory of this method is presented in more extensive detail than the other two, along with some clues to why this might be a good choice for segmenting AAA in CT images.

3.5.2 Watershed

The watershed segmentation algorithm is a grayscale-based algorithm. Watershed segmentation gets its name from the manner in which the algorithm segments regions into catchment basins. If a function f is a continuous height function defined over an image domain, then a catchment basin is defined as the set of points whose paths of steepest descent terminate at the same local minimum of f. In other words, the catchment basins should theoretically correspond to the homogeneous graylevel regions of the image.

An appropriate choice of height function is made, dependent on the application, and the basic watershed algorithm operates independently of that choice. For intensity-based image data, a calculation of gradient magnitude is often used.

The watershed algorithm proceeds in several steps. First, an initial classification of all points into catchment basin regions is done by tracing each point down its path of steepest descent to local minima. Next, neighboring regions and the boundaries between them are analyzed according to some saliency measure, such as minimum boundary height, to produce a tree of merges among adjacent regions. These merges occur at different maximum saliency values. The set of all possible merges up to a specified saliency “flood level” is often referred to as a “merge tree”.

Metaphorically, the flood level is a value that reflects the amount of water that is rained into the catchment basins. As the flood level rises, boundaries between adjacent segments will merge. The minimum value of the flood level is zero and the maximum value is the difference between the highest and lowest values in the input image.

Once the segmentation is done to produce the merge tree, it is easy to produce a hierarchy of labeled images. The complexity of the watershed algorithm is in the computation of the merge tree. Once that tree has been created, the initial segmented image can be relabeled to reflect any maximum saliency value found in the tree by identifying a subset of segment merges from the tree.

3.5.3 Fuzzy connectedness

The fuzzy connectedness algorithm is a region-based method. In the case of a grayscale image, the algorithm performs segmentation via thresholding of a fuzzy connectedness scene. A seed point, or a set of seed points, is first specified within the region of interest. A construct named the fuzzy affinity is then computed between neighboring pixels such as to reflect their similarity and assign a probability that these pixels belong to the same object. This way, a “path” is created between pixels in the form of a list of pixels that connect them. The strength of a particular path is defined as the weakest affinity between the neighbor pixels that form the path. The fuzzy connectedness between two pixels is defined as the strongest path strength between these two pixels.

The segmentation based on fuzzy connectedness assumes that the fuzzy connectedness between any two pixels from a single structure is significantly higher than those for pixels belonging to different structures. The strength depends on how close pixels are located spatially and in terms of intensity and intensity-based properties. After the fuzzy connectedness scene is first computed, a threshold is applied to the fuzzy scene, and a binary segmented object may be extracted.

3.5.4 Level set methods

3.5.4.1 Introduction to level sets

Level set algorithms are model-based algorithms. They were introduced by S. Osher and J. A. Sethian in 1988, and are techniques created to follow the evolution of N-dimensional curves (interfaces), by observing their curvature. Level sets are designed to handle problems in which the evolving interfaces can develop sharp corners and cusps, change topology and become very complex.

Most existing shape modeling techniques require that the topology of the object be known before the shape recovery can commence. However, it is not always possible to specify the topology of an object prior to its recovery. One important concern is topological change resulting from tracking the evolution of curve or surface boundaries through time. During their evolution, interfaces may change connectivity and split, thereby undergoing a topological transformation which is often very difficult to follow using traditional approaches. In the level set approach, the convergence to the final result may be relatively independent of the initial shape, and branches, splits and merges can develop without problems as the front moves. Generally, the method may be applied even where no a priori assumptions about the object's topology are made.

Also, the evolution of curves and surfaces is a defining component of many physical phenomena. For example, surface tension in a soap bubble and freezing rates at the edge of a snowflake both depend on the curvature at a point [JASET-96]. The level set methods have been widely used in the fields of fluid mechanics and material sciences for some time [MALLA-95] and have in recent years been applied in image processing for segmentation problems [JASET-99].

3.5.4.2 Level sets vs. deformable models

Deformable model techniques generally attempt to follow boundaries by placing a set of discrete marker points on the evolving front and then changing the position of these markers to correspond to the front as it moves. The discrete markers are updated in time using a set of finite difference approximations to the equations of motion. However, there are several problems with this approach [JASET-02], the most important of which are the following:

- Deformable models have a tendency to become unstable as the curvature increases around a cusp. An entropy condition must then be observed to produce a correct solution [JASET-99].

- Deformable models suffer from the tendency of small errors in the position to produce large errors in the determination of the curvature [BRUIN-01]. This may be solved by a frequent redistribution of the markers, using a small time step, but has the drawback of altering the motion of the curve in a nonobvious way.
- Deformable models have great difficulties coping with topological changes. When an interface is deforming, and its topology changes significantly (split or merge, for example), tracking the possible changes is an extremely complex task which is incredibly hard to perform [SAPIR-01].

Due to these problems associated with the traditional deformable model-methods, we instead choose to look more closely at level sets.

3.5.4.3 Evolving interfaces

Let us consider interfaces evolving in time. Let $\Gamma(p,t) : S^{N-1} \times [0,T) \rightarrow \mathbb{R}^N$ denote a family of closed interfaces, where t parametrizes the family (time) and p parametrizes the interface (space). We will assume that this family of interfaces obeys the following partial differential equation:

$$\frac{\partial \Gamma(p,t)}{\partial t} = \alpha(p,t)\vec{\mathcal{T}}(p,t) + \beta(p,t)\vec{\mathcal{N}}(p,t), \tag{3.1}$$

where $\Gamma(p,0)$ is the initial condition. Here $\vec{\mathcal{N}}$ stands for the outward unit normal and $\vec{\mathcal{T}}$ is a unit tangent vector. This is the most general form and means that the interface Γ is moving with α velocity in the tangential direction $\vec{\mathcal{T}}$ and β velocity in the normal direction $\vec{\mathcal{N}}$.

We choose to parametrize Γ because it is a more general approach to the problem, as there is no guarantee that an evolving front can always be expressed as the graph of a function in a fixed coordinate system. Parametrizing the interface relieves us of this concern.

3.5.4.4 Finding a level set representation

The central idea of level sets is that, rather than follow the movement of Γ itself, we instead add one extra dimension to the problem. At first glance, it might seem counterintuitive to add an extra dimension, since more dimensions usually means more work and a more complex model. However, the extra dimension will turn out to be a very powerful addition because, rather than track discrete points of Γ

around, which can collide and stretch apart, we will be able to track the front by adjusting the height of a whole function instead.

Using this approach, the evolving front Γ can be represented by a level set function Ψ in one higher dimension. In the case of the parametrized interface $\Gamma(p,t)$ in $(N-1)$D space, this involves using a function $\Psi(\mathbf{x},t)$ in a fixed coordinate system in ND space, with $\mathbf{x} = [x_1, x_2, ..., x_N]$. Note that we reintroduce a coordinate system when we're working with the function Ψ.

Specifically, if we let Γ be represented by

$$\begin{aligned} \frac{\partial \Gamma(p,t)}{\partial t} &= \alpha(p,t)\vec{\mathcal{T}}(p,t) + \beta(p,t)\vec{\mathcal{N}}(p,t) \\ &= \alpha\vec{\mathcal{T}} + \beta\vec{\mathcal{N}} \\ &= \vec{\mathcal{V}}, \end{aligned} \tag{3.2}$$

using $\Gamma(p,0)$ as the initial condition, we may represent Γ as the level set of an embedding function, a level set function, Ψ. By this we mean that Ψ is defined as a mapping $\Psi(\mathbf{x},t) : \mathbb{R}^N \times [0,T) \rightarrow \mathbb{R}^{N-1}$ such that

$$\begin{aligned} \Lambda_c(t) &= \{\mathbf{x} \in \mathbb{R}^N : \Psi(\mathbf{x},t) = c\} \\ &= \{\Gamma(p,t) : p \in S^{N-1}\}, \end{aligned} \tag{3.3}$$

where $c \in \mathbb{R}$ is a given constant.

The initial interface $\Gamma(p,0)$ can be represented as the zero level set of the higher dimensional function $\Psi(\mathbf{x},0)$. We can express $\Psi(\mathbf{x},0)$ by considering the signed distance function $d(\mathbf{x})$ from a point in $(N-1)$D space to the curve $\Gamma(\mathbf{x},0)$, negative in the interior and positive in the exterior of $\Gamma(\mathbf{x},0)$, so that

$$\Psi(\mathbf{x},0) = d(\mathbf{x}) + c. \tag{3.4}$$

We now have to find the evolution of Ψ so that

$$\Gamma(p,t) \in \Lambda_c(t), \tag{3.5}$$

that is, the evolution of the original curve Γ coincides with the evolution of the level sets of Ψ.

A consequence of equation 3.3 is that:

$$\Psi(\Gamma(p,t),t) = c. \tag{3.6}$$

By differentiating with respect to t, we get

$$\frac{\partial}{\partial t}\left(\Psi(\Gamma(p,t),t)\right) = \frac{\partial}{\partial t}(c)$$

$$\nabla\Psi(\Gamma(p,t),t)\cdot\frac{\partial\Gamma(p,t)}{\partial t}+\frac{\partial\Psi(\Gamma(p,t),t)}{\partial t}=0. \tag{3.7}$$

We observe that there is no need to specify the level set value c, as it is eliminated from the equation when the derivative is taken.

The unit normal vector can be expressed by the directional derivative as

$$\frac{\nabla\Psi}{\|\nabla\Psi\|}=\vec{\mathcal{N}}, \tag{3.8}$$

where $\vec{\mathcal{N}}$ is the normal to the level set Λ_c.

Combining equations 3.2, 3.7 and 3.8, we get:

$$\begin{aligned}0&=\nabla\Psi\cdot\vec{\mathcal{V}}+\frac{\partial\Psi}{\partial t}\\&=\nabla\Psi\cdot(\alpha\vec{\mathcal{T}}+\beta\vec{\mathcal{N}})+\frac{\partial\Psi}{\partial t}\\&=\nabla\Psi\cdot(\alpha\vec{\mathcal{T}}+\beta\left(\frac{\nabla\Psi}{\|\nabla\Psi\|}\right))+\frac{\partial\Psi}{\partial t}\\&=0+\beta\|\nabla\Psi\|+\frac{\partial\Psi}{\partial t}\\&=\beta\|\nabla\Psi\|+\frac{\partial\Psi}{\partial t}.\end{aligned}$$

We observe that due to the dot product, the tangential component has no effect on the evolution of the front, so that observing the evolution of

$$\frac{\partial\Gamma(p,t)}{\partial t}=\beta(p,t)\vec{\mathcal{N}}(p,t),$$

is equivalent to observing

$$\frac{\partial\Gamma(p,t)}{\partial t}=\alpha(p,t)\vec{\mathcal{T}}(p,t)+\beta(p,t)\vec{\mathcal{N}}(p,t).$$

Rearranging and using the more customary notation $F=\beta$ for the speed function, we get the basic level set equation:

$$\frac{\partial\Psi}{\partial t}+F\|\nabla\Psi\|=0. \tag{3.9}$$

We refer to this as a Hamilton–Jacobi type equation, since for certain forms of the speed function F, we obtain the standard Hamilton–Jacobi equation. The evolving function $\Psi(\mathbf{x},t)$ will always remain a function as long as F is smooth [MALLA-95].

However, the level set Λ_c and hence the propagating interface Γ, may change topology considerably as the function Ψ evolves.

To summarize, level set methods exchange a general geometric moving coordinate representation for a fixed coordinate perspective where each point adjusts its value to measure the distance to the evolving interface.

A further appealing aspect of the level set approach is that the basic concept is the same regardless of the dimensionality of the problem. First, we embed the evolving surface in a function in one higher dimension. Then, we adjust this higher dimensional function corresponding to motion of the interface, and compute the zero level set to find the position of the propagating interface. All together, the trick of embedding the front in a higher dimensional function is well worth the added cost.

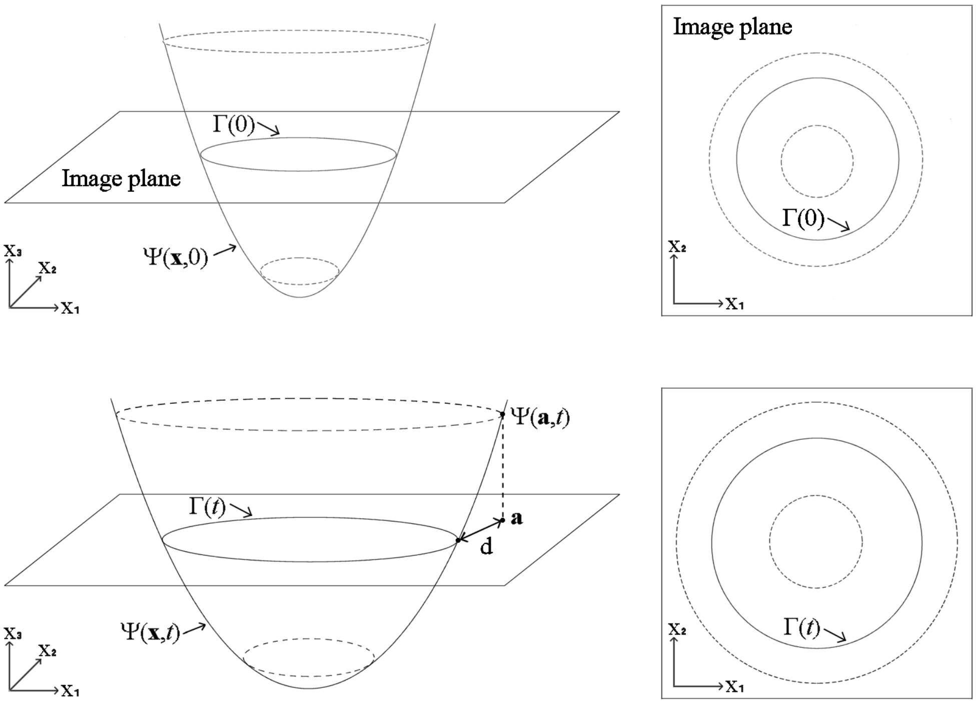


Figure 3.3: Level set formulation of the equations of motion. The upper two images show the curve Γ and the surface $\Psi(\mathbf{x})$ at $t = 0$. The lower two images show the curve Γ and the corresponding surface $\Psi(\mathbf{x})$ at time t.

3.5.4.5 Selecting a speed function

In image segmentation applications, the speed function in equation 3.9 is typically composed so that the shape of the evolving front is influenced by the following three components ([MAGEE-00]):

- A constant advection component, independent of the geometry of the front.
- A component depending on the curvature of the front.
- A component containing information about the image (such as edges).

To do this, we use a speed function on the following form, as proposed by [MALLA-95]:

$$F = P(F_0 + F_1(\kappa)). \tag{3.10}$$

F_0 is the constant advection component, representing a uniform speed, independent of the geometry of the evolving front. This is reminiscent of the inflation force used in some deformable model techniques.

F_1, being dependent on the curvature κ, acts like a diffusion component, and depends on the intrinsic geometric properties of the front. This component has the effect of smoothing out regions of high curvature, and metaphorically it can be thought of as the viscosity of a fluid. In fluid mechanics, viscosity is a measure of the relationship between the shear stress exerted on a fluid and the fluid's rate of deformation [GERHA-92]. Loosely speaking, viscosity measures the ability of a fluid to damp sharp transitions and mute sudden changes.

P is the image dependent component, usually in the form of an edge potential function, derived from the gradient map of the input image. In order to segment images, we need to make the speed function F in equation 3.9 dependent on the image we wish to segment. This dependency condition should preferably cause an initial level set interface in the image to stop at the boundary of the desired object.

As proposed by [MALLA-95] we formulate the speed function as follows:

$$F = P(I)(1 - \epsilon\kappa). \tag{3.11}$$

Here, $0 < \epsilon < 1$ is a constant, I is the image intensity and κ is the curvature, obtained from divergence of the gradient of the normal vector to the front:

$$\kappa = \nabla \frac{\nabla\Psi}{\|\nabla\Psi\|}. \tag{3.12}$$

The basic speed equation component $1 - \epsilon\kappa$ is fundamental for image analysis purposes [SAPIR-01]. The advection term $F_0 = 1$, provides us with a means of growing a region from an initial front, the viscosity term $F_1 = \epsilon\kappa$ provides us with a way of controlling the front of the region as it expands, and the potential function $P = P(I)$, provides the image information required to halt the evolving front at the desired structure boundaries.

3.5.4.6 Selecting a potential function

The edge potential function, $P(I)$, where I is still the image intensity, takes the original image as input and gives an edge representation of the original image as output. Thus, the edge potential image is in essence an edge map created from the initial image I we want to segment.

The potential map is usually defined in the way that seems most appropriate for the current application. Indeed, when applying the level set algorithm for segmentation purposes in image processing, much of the preprocessing step is about finding the right potential function to produce the best potential image for the level set algorithm. Two examples of functions commonly used for creating a potential image, $P(I)$, are:

$$P(I) = e^{-|\nabla G * I|} \quad \text{and} \quad P(I) = \frac{1}{1 + |\nabla G * I|}, \tag{3.13}$$

where ∇G is the derivative of the gaussian operator. Depending on the amount of preprocessing needed, this function can be chosen to be arbitrarily simple or arbitrarily complex.

For our application, none of functions in 3.13 were used, as more extensive preprocessing turned out to be necessary in order to achieve a satisfactory segmentation. As $P(I)$ is basically a function that contains the whole preprocessing step it is not always easy or feasible to provide a simple expression for it, and this is the case here. However, the details of the preprocessing are explained later on in chapter 4.

Thus, our final level set equation for segmentation is given by

$$\frac{\partial \Psi}{\partial t} + P(I)(1 - \epsilon\kappa)\|\nabla\Psi\| = 0 \tag{3.14}$$

3.5.4.7 Improving the performance

In a numerical implementation, the spatial parameters ((x_1, x_2) in the 2D case; in the 3D case (x_1, x_2, x_3)) are discretized into a grid of points. Each pixel (voxel) in this grid, or distance map, holds a value for the level set function and updates its value as the surface moves, using neighborhood values to determine the necessary partial derivatives in the level set equation.

One of the biggest problems of implementing equation 3.9 is that it is rather slow. When solving equation 3.9 for a 2D curve evolution, the complexity is $O(N^2)$, and similarly $O(N^3)$ for a 3D surface, where N is again the size of the grid. Because of this, level set algorithms commonly use the following improvements:

- Fast marching: In the case of fronts that move forward under speeds where the speed function F never changes sign, a considerably more time-efficient method may be preferred over the general level set method.
- Narrow banding: Rather than update the level set function everywhere, the work is confined to a thin region on both sides of the evolving front.

3.5.4.8 Fast marching

The fast marching method makes the simplifying assumption that the speed function F never changes sign, so that the front is always moving forward or backward.

The position of the expanding front is then characterized by computing the arrival time $T(\mathbf{x})$ as it crosses each point $\mathbf{x}$. For this reason, T is often referred to as the time-of-arrival function. Since F never changes sign, $T(\mathbf{x})$ is indeed a function, and the curve crosses each planar point no more than once. ∇T is orthogonal to the level sets of T, and its magnitude is inversely proportional to the speed:

$$|\nabla T|F = 1, \quad T = 0 \quad \text{on} \quad \Gamma. \tag{3.15}$$

Here, Γ is the initial position of the interface.

In other words, the solution is found by solving a boundary value problem. This is again a Hamilton–Jacobi equation, which if the speed F depends only on position, reduces to what is known as the Eikonal equation. The main idea is to exploit a fast heapsort technique to systematically locate the proper grid point to update, so that backtracking over previously evaluated grid points is never necessary. The resulting technique traverses a grid of N total points in $N\log(N)$ steps to obtain the evolving time position of the front as it propagates through the grid, thus the name for the algorithm.

3.5.4.9 Narrow banding

Execution time can be improved by using the narrow band technique. The basic idea behind narrow banding is to operate only on a surrounding band around the region of the level set being tracked. This will reduce the complexity to $O(kN)$ in the 2D case and $O(kN^2)$ in the 3D case, where k is the width of the narrow band. Using this technique, equation 3.9 is solved only inside the narrow band, and this band is updated each time the interface approaches the border of the band, or after a certain number of iterations.

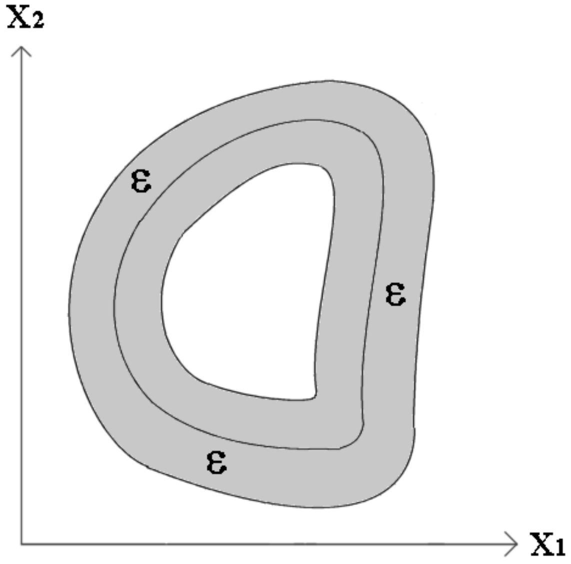


Figure 3.4: A narrow band of width ϵ around the level set.

When the level set function used is a distance function, a narrow band width of ϵ around the zero level set, is given by those points on the level set function with an absolute value less than or equal to ϵ. This way, all we need to do is to maintain the level set function as a signed distance function and update only those points where the level set function is less than a certain ϵ. This simplifies the computation of the narrow band greatly.

3.5.4.10 Benefits of using level sets in image processing

A skilled eye can pick out the desired boundaries from a noisy image, even those delimited by slight changes in image intensity. Drawing outlines manually on each image slice is both extremely time-consuming and inexact. To create a piece of software to ignore noise and at the same time avoid introducing non-existent features has the potential to solve these problems, but is hard to implement. In the level set approach, the imaginary front is allowed to propagate from an initial position. The trick is to adjust the speed F to detect the edge of the shape:

- When the interface passes over places where the image gradient is small, we assume that we are not near a boundary, and we let the curve expand quickly.
- When the curve passes over places where the image gradient is large, we suspect we are near the boundary, and the expansion is slowed down.
- In addition, a little surface tension (in the form of motion by curvature) is included to slightly constrain the expanding contours.

3.6 Registration algorithms

3.6.1 Overview

The registration method tested for registering CT and US images is mutual information, a method based on a formulation of a general measure of content similarity between the images to be processed. Due to the differences of the nature of the images to be registered, and considering the other alternatives available in ITK at the time of this work, it was strongly suggested by the supervisors that this method be used.

3.6.2 Mutual information

3.6.2.1 Introduction to mutual information

One of the biggest challenges when registering images of different modalities, is to find a metric able to cope with similar structures in images with very different characteristics. Rather than require the moving image to be given by a function of the fixed image, as is the case with many registration methods, one generalization is to just require the moving image to be predictable from the fixed image. In statistics, predictability is closely related to the concept of entropy. If a random variable is predictable, it has low entropy, while if it is unpredictable, it has high entropy. By formulating a connection based on entropy, many of the drawbacks of relying on strict connection through a function can be eliminated.

3.6.2.2 Entropy

The entropy H of a random variable x is defined as:

$$H(x) = -\int p(x)\ln(p(x))dx. \tag{3.16}$$

The joint entropy of two random variables x and y is given by

$$H(x,y) = -\int p(x,y)\ln p(x,y)dxdy. \tag{3.17}$$

Log likelihood and entropy are closely related, and using [COVER-91] it can be shown that under certain conditions, the conditional log likelihood of the moving image is a multiple of the conditional entropy of the moving image given the fixed image:

$$\log p(m(T(x) \mid f(x),T)) = -NH(m(T(x)) \mid f(x),T), \tag{3.18}$$

where N is the number of fixed image points.

Mutual information is a measure of the statistical independence between two random variables, or the amount of information one variable contains about the other. Mutual information I is defined in terms of entropy in the following way:

$$I(f(x),m(T(x))) = H(f(x)) + H(m(T(x))) - H(f(x),m(T(x))). \tag{3.19}$$

Here H is the entropy function of the random variable x, f is the fixed image, m is the moving image and T is a transform, allocating points from the fixed image into the moving image.

Mutual information has three components. The first term on the right is the entropy in the fixed image. This does not depend on T. The second term is the entropy of the part of the fixed image the moving image is transformed into. The third term is the negative joint entropy of the fixed image and the moving image.

3.6.2.3 Finding a transformation estimator

We now wish to find an estimate of the transform $\hat{T}$ that aligns the moving image $m(x)$ and fixed image $f(x)$ by maximizing their mutual information over the transform T. The random variable x is defined over the coordinate space of the fixed image.

In other words, we want to find the transformation estimator $\hat{T}$ such that

$$\hat{T} = \arg\max_{T} I(f(x),m(T(x))). \tag{3.20}$$

The described entropies are defined by integrals over the probability densities associated with the fixed image f and the moving image m. When we look at images we will generally not have access to the densities. To compensate, we use a differentiable estimate of the entropy of a random variable calculated from samples instead. It is possible to express the entropy of a random variable y as the expected value of the negative logarithm of the probability density:

$$H(y) = -E_y(\ln p(y)). \tag{3.21}$$

To estimate the entropies from samples, the first step is to approximate the underlying probability density $p(y)$ through a superposition of gaussian densities centered on the elements of a sample S_1 drawn from y:

$$p(y) \approx \frac{1}{N_{S_1}} \sum_{x_j \in S_1} G_\sigma(x - x_j), \tag{3.22}$$

where

$$G_\sigma(y) = \frac{1}{2\pi^{n/2}} \frac{1}{|\sigma|^{1/2}} e^{(-\frac{1}{2} y^T \sigma^{-1} y)}. \tag{3.23}$$

This is known as the Parzen window method of estimating density. Using the gaussian density in the Parzen density estimate simplifies some of the following analysis, but any differentiable function can be used instead.

The next step is to approximate the statistical expectation with the sample mean over another sample S_2, drawn from y:

$$E_y(u(y)) \approx \frac{1}{N_{S_2}} \sum_{y_i \in S_2} u(y_i). \tag{3.24}$$

An approximation for the entropy of a random variable y may now be expressed as

$$H(y) \approx -\frac{1}{N_{S_2}} \sum_{y_i \in S_2} \ln \frac{1}{N_{S_1}} \sum_{y_j \in S_1} G_\sigma(y_i - y_j). \tag{3.25}$$

The density of y may be a function of a set of parameters T. We can now find the maxima of mutual information. By calculating the derivative of the entropy with respect to T, we get the following expression, after simplifying:

$$\frac{d}{dT} H(y(T)) \approx \frac{1}{N_{S_2}} \sum_{y_i \in S_2} \sum_{y_j \in S_1} W_y(y_i, y_j)(y_i - y_j)^T \sigma^{-1} \frac{d}{dT}(y_i - y_j), \tag{3.26}$$

where

$$W_x(x_i, x_j) = \frac{G_\sigma(x_i - x_j)}{\sum_{x_k \in S_1} G_\sigma(x_i - x_k)}. \tag{3.27}$$

Here, $W_y(y_i, y_j)$ is a weighing factor with values between 0 and 1. If y_i is significantly closer to y_j than it is to any other element of S_1, it will approach 1. It will approach 0 if some other element of S_1 is significantly closer to y_i. Distance is interpreted with respect to the squared Mahalanobis distance, given by:

$$D_\sigma(y) = y^T \sigma^{-1} y. \tag{3.28}$$

In other words, $W_y(y_i, y_j)$ is an indicator of the degree of match between y_i and y_j.

3.6.2.4 Stochastic maximization of the mutual information

The entropy approximation in equation 3.25 may now be used to evaluate the mutual information of the fixed image and the moving image in equation 3.19. To obtain a maximum of the mutual information, we first calculate an approximation to its derivative:

$$\frac{d}{dT} I(f(x), m(T(x))) = \frac{d}{dT} H(m(T(x))) - \frac{d}{dT} H(f(x), m(T(x))). \tag{3.29}$$

We will assume that the covariance matrices of the component densities used in the approximation scheme for the joint density, are block diagonal. That is:

$$\sigma_{fm}^{-1} = \text{diag}(\sigma_{ff}^{-1}, \sigma_{mm}^{-1}). \tag{3.30}$$

Using this assumption and equation 3.26, we find an estimate for the derivative of the mutual information:

$$\begin{aligned} \widehat{\frac{dI}{dT}} = \frac{1}{N_{S_2}} \sum_{x_i \in S_2} \sum_{x_j \in S_1} & (m_i - m_j)^T [W_m(m_i, m_j)\sigma_m^{-1} \\ & - W_{fm}(w_i, w_j)\sigma_{mm}^{-1}] \frac{d}{dT}(m_i - m_j). \end{aligned} \tag{3.31}$$

The weighting factors are defined according to equation 3.27:

$$W_m(m_i, m_j) = \frac{G_{\sigma_m}(m_i - m_j)}{\sum_{x_k \in S_1} G_{\sigma_m}(m_i - m_k)} \tag{3.32}$$

and

$$W_{fm}(w_i, w_j) = \frac{G_{\sigma_{fm}}(w_i - w_j)}{\sum_{x_k \in S_1} G_{\sigma_{fm}}(w_i - w_k)}, \tag{3.33}$$

using the following notation:

$$f_i = f(x_i), m_i = m(T(x_i)), \quad \text{and} \quad w_i = [f_i, m_i]^T. \tag{3.34}$$

The notation is similar for indices j and k.

If we are to increase the mutual information, then the first term in the brackets of equation 3.31 works to increase the squared distance between sample pairs that are nearby in image intensity, while the second term decrease the squared distance between sample pairs that are nearby in both image intensity and the properties of the fixed image. The term $\frac{d}{dT}(m_i - m_j)$ generally involves gradients of the intensities of the image and the derivative of transformed coordinates with respect to the transformation.

When T is a linear operator, the following expression holds:

$$\frac{d}{dT} m(T(x_i)) = \nabla m(T(x_i)) x_i^T. \tag{3.35}$$

In the end, a different structure of gray levels in the images is not a problem when using mutual information, as it does not directly measure the similarity in the graylevels of the pixels, but rather how often the intensities appear together. Thus, the algorithm can match structures of low intensity in one image to structures of high intensity in another. By using mutual information, having structures that are bright in one modality and darker, or marked by edges, in the other is generally not a problem. A potential risks however is if each modality has two different anatomical structures that look similar and could lead the registration algorithm to try to match them.

4 Experiments and results

In this chapter we first have a look at the available image test data. We briefly discuss the use of watershed and fuzzy connectedness for segmentation, before the final implementation using level sets is presented in more detail. We end the chapter with a discussion of the use of mutual information for registration.

4.1 Image data

To test the schemes to be implemented, three postoperative CT images along with three corresponding US images were used.

4.1.1 CT image data

The CT image data has been acquired from patients with abdominal aortic aneurysm, injected with contrast to increase the visibility of the aortic structure. The original CT data to be analyzed were volumetric $512 \times 512 \times n$ images in 16 bit grayscale, with n, the number of slices or cross-sections, in the range between 100 and 200. These images are stored in DICOM format (Digital Imaging and COmmunications in Medicine), which is a standard image format for the exchange of medical images [NEMAX-93]. As ITK at the time of this work didn't have any kind of functionality to handle the DICOM format, the images had to be passed through an elaborate conversion procedure and stored in a much simpler raw data format more easily handled with ITK. The conversion software available to Sintef Unimed to do this conversions had the following unfortunate limitations:

- Each image slice was downsampled from 512×512 to 256×256 pixels.
- The grayscale depth was reduced from 16 bit to 8 bit.

- Each image slice was converted separately, its dynamic range being adjusted automatically, based on the intensity features of the slice.

Downsampling the images naturally means that a considerable amount of detail was lost. It is obviously more difficult to process images that contain only 25 % of the original spatial information.

Reducing the grayscale depth also reduces the amount of information available in the image, especially when combined with the spatial downsampling explained above.

As for the grayscale adjustment of each slice, the composition of intensity values in each slice is usually different from the next with regards to maximum and minimum intensity values and the number of distinct graylevels used. When the images were converted, this has the unfortunate consequence of introducing intensity gradients throughout the images in the vertical direction, through the cross-sectional plane, parallel to the general direction of the aorta. The result is that regions in different slices, which were originally of the same intensity, would sometimes end up with considerably different intensities. An example of this can be seen in figure 4.1.

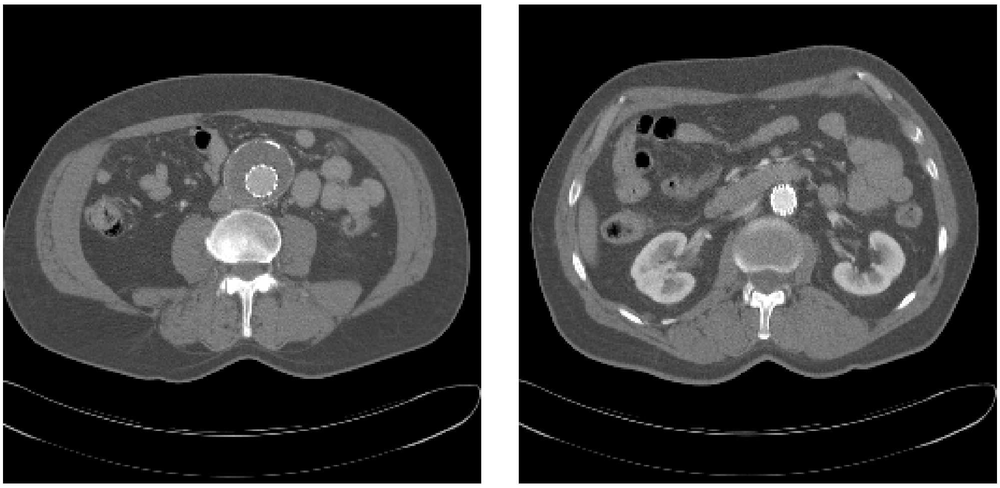


Figure 4.1: Two CT slices from the same volume showing the variation in intensities after conversion. Note especially the difference in graylevel value of the lumen. In the left picture, the lumen is the relatively dark region with a brighter circle around it, while in the right picture, the same region is relatively bright overall, almost to the point of being solid.

The problem with all of these issues, as they are a result of the image conversion only, is that they introduce artifacts that do not represent anything real. Since the resulting graylevels may vary considerably in a nonobvious way, it becomes more difficult to process the images. It should be stressed that these problems were beyond the author's control.

4.1.2 US image data

The ultrasound data were also volumetric images in 8 bit grayscale. The resolution of these images are variable, but smaller than for the CT images. The spatial resolution of the ultrasound images is frequently less than half of the corresponding values of the CT.

The issues with image conversion were not a problem with the US images. However, the available US data has been resampled to the equivalent of the CT image resolution. This means that the smaller ultrasound images have been padded with regions of black voxels (voxels with intensity equal to zero) to make them the same resolution as the CT images.

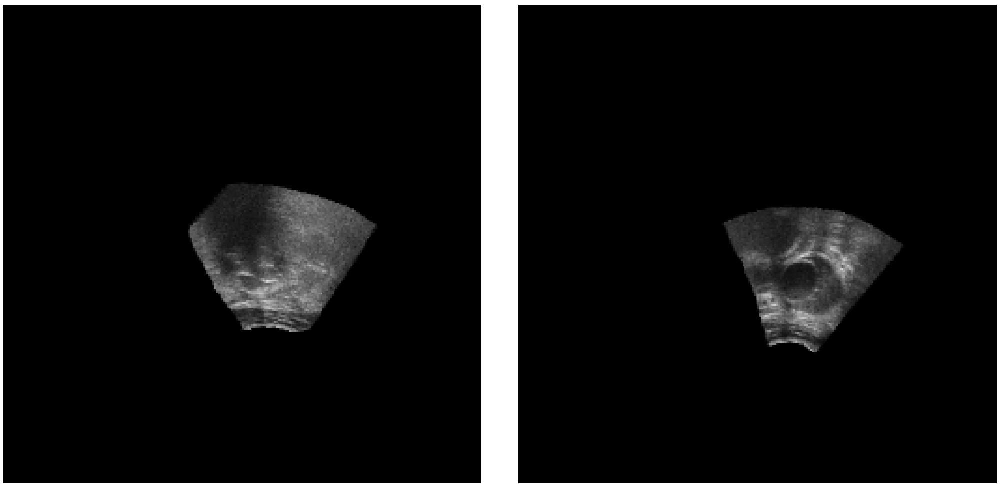


Figure 4.2: Two US slices from the same volume. In the left image, the bifurcated stent graft can be vaguely seen slightly to the left in the image. The dark region above it is caused by most of the soundwaves being reflected by the graft. In the right image, the graft can can be seen as a small dark circular area within the aneurysm, which is the bigger and slightly brighter region around it.

4.1.3 Initial registration

When the CT and US images were originally acquired, they were registered using a positioning system. Before the postoperative CT scan is acquired, physical markers called fiducials are attached in predetermined spots on the skin of the patient's body. The positioning system is then used to keep track of the fiducials in relation to a reference coordinate system. When the CT image is acquired, the markers will show up in the image, making it possible to obtain the position of the image relative to the reference coordinate system. Using this information, a transformation matrix $T_{CT \to ref}$ is calculated, which transforms a point from CT image space into the space of the reference coordinate system.

In a similar manner, when the US images are to be acquired, a tracking device is attached to the ultrasound probe, making it possible to track the probe's position with regard to the same reference coordinate system as was used for the CT. The ultrasound probe is moved by hand across the patient's body, acquiring a series of 2D image slices. After acquisition, the 2D US data is resampled to create a 3D ultrasound image. As the position relative to the reference coordinate system is known, a transformation matrix $T_{US \to ref}$ can be obtained, which transforms a point from US image space into the reference space.

Using these two matrices, it is now possible to map a point from US image space into CT image space, or vice versa, by means of the following transformations:

$$\begin{aligned} T_{CT \to US} &= T_{CT \to ref} T_{US \to ref}^{-1} \\ T_{US \to CT} &= T_{US \to ref} T_{CT \to ref}^{-1} \end{aligned} \tag{4.1}$$

If these operations were without fault, no further registration would be needed, as the transformed images would line up perfectly. However, the procedure is inexact for several reasons:

- The biggest source of error lies in tracking the position of the ultrasound probe.
- When acquiring the CT image, the fiducials are manually placed on the patient's body. This may be inaccurate because of patient respiration.
- The positioning devices used when acquiring both the CT and US images may not be perfectly calibrated, meaning that there are errors in the recorded positions.
- The patient may be physically positioned in a slightly different way during the two different examinations.

- Several other factors may variably affect the volume and position of the internal organs in the abdomen between the two examinations.

4.2 The watershed approach

4.2.1 The problems

The watershed algorithm was the first technique tested for segmenting the lumen in 3D. Watershed works best when the region to segment is composed of voxels which are very close to each other in intensity, so that they form a single region when the edge image is "flooded", as explained in section 3.5.2. The complications encountered with this method stems from its sensitivity to noise, both on the voxel level and the regional level. The persisting problem is that the resulting segmented image is divided into too many regions. Neighboring voxels (or small clusters of) within the region of interest, will frequently differ too much in intensity for smoothing and denoising to blend them together into a region sufficiently uniform for the watershed method to label it as just one region.

On the pixel scale, the result is that a number of small regions or single pixels are separated from the bigger aortic structure they really are a part of. This naturally tend to happen around the edges, where the variation in graylevels are highest.

On the regional scale, the slight gradients throughout the image, especially the intensity variations caused by the described conversion process, tend to cause the lumen region to be "fractured" into several smaller chunks, corresponding to regions of slightly different intensity in the original image.

The watershed algorithm was also found to be very sensitive to the choice of parameters, being heavily dependent on the threshold and level parameters to be within a small range in order to segment the desirable regions. This is illustrated in figure 4.3.

Consequently, intensity variations within the region of interest is something the watershed algorithm doesn't handle very well, and this is a phenomenon that occurs frequently in our problem. Thus the result is often an image which contains both oversegmentation and undersegmentation, in the sense that the image has lots of small regions around the edges and the whole aortic structure being "fractured" into several smaller regions. This in turn makes it extremely difficult to automatically determine which regions are actually part of the structure and which aren't. Thus, the segmented region will too easily contain undesirable portions of the image in some places and lack portions in other places.

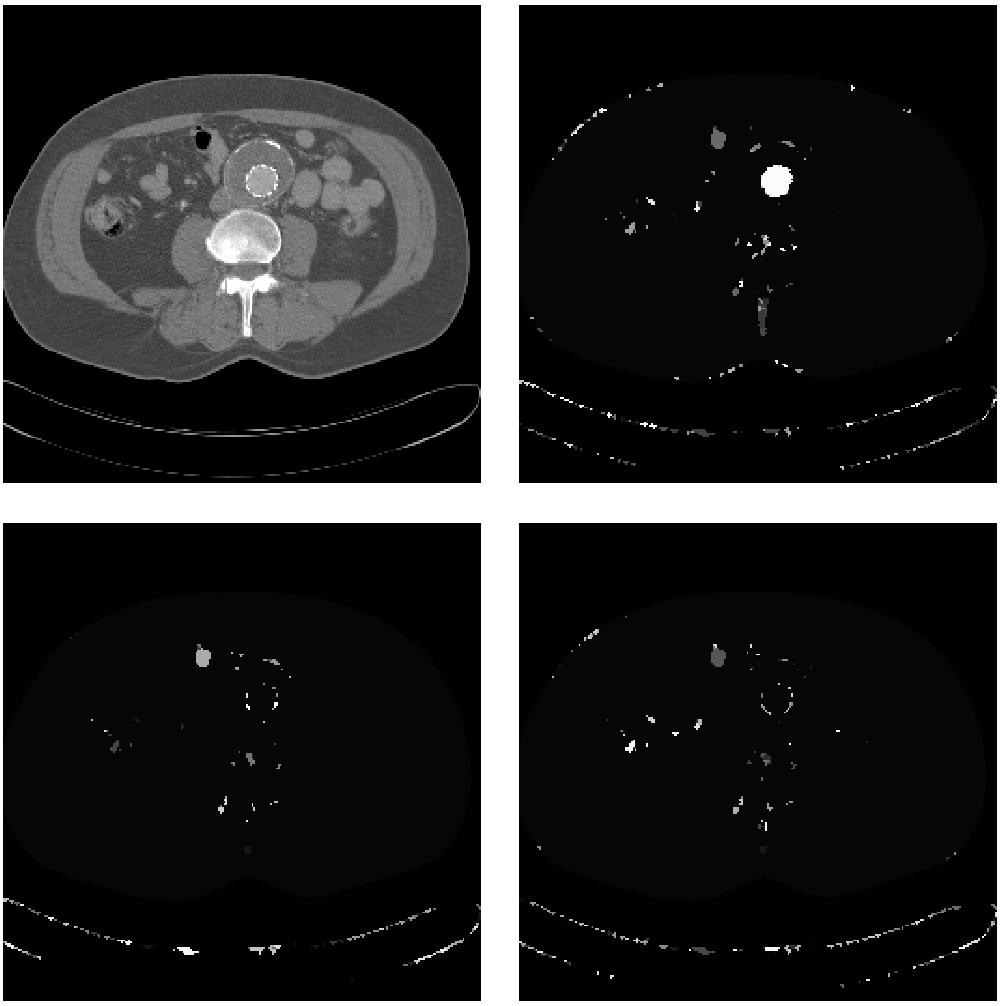


Figure 4.3: Examples of 3D watershed segmentation. The top left picture is the initial slice while the other three are the same slice segmented with different sets of parameters as follows: Top right: threshold = 0.08, level = 0.07. Bottom left: threshold = 0.10, level = 0.07. Bottom right: threshold = 0.08, level = 0.03. By varying the parameters only very slightly, significantly different segmentations are obtained

4.2.2 Attempted corrections

Gaussian filtering, median filtering and anisotropic diffusion were used in attempts to reduce noise and smooth the initial image to improve the performance of the scheme. However, these types of noise reduction and smoothing also result in degrading and obscuring the level of detail in the image. Relatively little noise reduction would tend to leave the big local gradients relatively untouched, resulting in severe oversegmentation (too many regions), while too much smoothing would lead to instances of edges blurring together, resulting in undersegmentation (too few regions).

4.2.3 Conclusion

The biggest problem with the watershed algorithm is that it is too sensitive to noise and too conspicuous with regard to the quality of the input image. It is also too dependent on images to have sharp edges delimiting monotonous regions to produce good results. The input image would have to undergo considerably more advanced forms of noise removal for the watershed algorithm to segment adequately. Thus, it made more sense to abandon the algorithm in favor of an approach less sensitive to noise in the CT images.

It becomes clear that the replacement algorithm must be very capable of classifying voxels as belonging to the same region despite some level of intensity variations and noise in the image.

4.3 The fuzzy connectedness approach

4.3.1 The problems

The fuzzy connectedness algorithm was the second technique tested for 3D lumen segmentation. Although more robust than watershed, the problem with this technique is mainly that it is based only on statistical measures of gray level similarity with no regards to the shape or boundary characteristics of the segmented region.

Generally, an image region which is close in proximity and intensity to the region of actual interest is too easily included in the final segmentation.

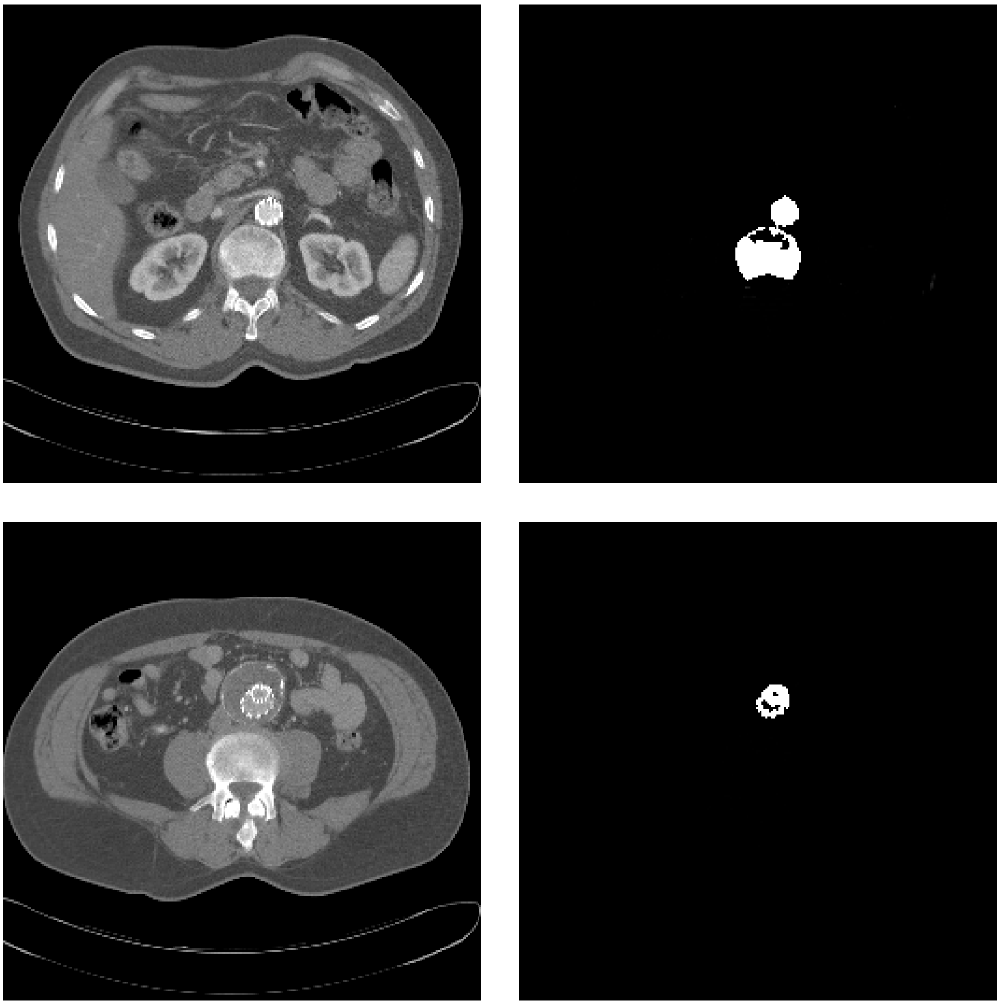


Figure 4.4: Example of 3D fuzzy connectedness segmentation. The right upper and lower images are initial slices from the same volume. The left upper and lower images are the same slices, segmented using the same set of parameters. This is a typical example of the algorithm segmenting both too little and too much.

As the fuzzy connectedness region expands, it will too easily tend to “leak” into parts of other structures with pixel regions sharing the same characteristics as the growing region. This is a frequent problem in the segmentation of the lumen, as the aorta runs parallel to the spinal column and through the hip. Parts of these regions consist of dense tissues and are often very similar in graylevel to the aortic lumen with injected contrast.

Although the end result of the segmentation is a single region, this region will frequently contain parts of vertebrae or parts of the hip or both. An example of this is shown in figure 4.4. A scheme would have to be implemented to automatically analyze the segmented image and remove such unwanted artifacts.

4.3.2 Attempted corrections

Techniques for smoothing, contrast adjustment and thresholding were explored in order to correct the problems with this scheme, with little success in removing the core problem. While regions with a degree of similarity are indeed detected and segmented despite variable intensity levels, the region grows too arbitrarily. As with the watershed approach, this results in both oversegmentation and undersegmentation of the input. As these problems are characteristics inherent to the segmentation algorithm and image data in question, it is something further preprocessing doesn’t easily solve. Also, the threshold parameters, used by the algorithm, has to be adjusted for every individual data set, and automatically estimating the thresholds that gives an appropriate segmentation for each individual data set is not a trivial task.

4.3.3 Conclusion

The biggest problem with fuzzy connectedness is that there are too few restrictions imposed on the development of the front of the expanding region. Although the algorithm shows promising signs of being able to cope with the type of images in question, it ultimately falls short of the goal of segmenting the lumen. Again, the input would have to undergo much more advanced preprocessing for this algorithm to produce good results.

Although the application of fuzzy connectedness overcome the fundamental problems of the watershed algorithm, there is no way to impose rules on the shape of the expanding front by using this scheme. This gives us few means to avoid the inclusion of irrelevant image regions with characteristics similar to what we’re interested in. Thus, we see that the ability to model and place restrictions on the evolving front is very desirable, and such a scheme will likely be more capable of producing a more satisfactory segmentation of the structure we are looking for.

4.4 Implementing level sets

4.4.1 Background

As the results of watershed and fuzzy connectedness methods were deemed unsatisfactory, and as their characteristics and capabilities were ultimately found to be insufficient for the purpose of solving the problem, they will not be discussed in further detail in this text. Level sets was the third technique tested for segmentation. Using the current framwork for level set segmentation in ITK, 3D segmentation of the lumen was done, and both 3D and 2D segmentation of the lumen was also performed. The results were generally very pleasing.

4.4.2 Manual initialization

The automatic segmentation procedure is initialized by manually selecting four initial values in the CT image.

Prior to segmentation, the CT image is examined and a subvolume containing the stent graft is delimited by manually selecting two slices, z_{upper} and z_{lower}. This subvolume consists of all voxels between, and including, these two delimiting slices, and contains the entire anatomical region we are interested in. The locations of the two slices are illustrated in figure 4.5. When the segmentation process is initiated, only this subvolume is actually processed.

Additionally, two seed points are selected manually to aid the process. The first seed point is a voxel located somewhere within the lumen region. This point is used to initiate the lumen segmentation procedure after preprocessing. The second seed point is a voxel located somewhere within the thrombus, between the lumen and the outer aortic wall. This point will be used to sample an appropriate grayvalue, which is used to mask the segmented lumen region, before segmenting the thrombus and outer aortic wall. The location of the two seed points are also illustrated in figure 4.5.

When the segmentation commences, the subvolume delimited by the slices z_{upper} and z_{lower} is extracted. The lumen is segmented first, and then the obtained structure is used to aid the segmentation of the thrombus.

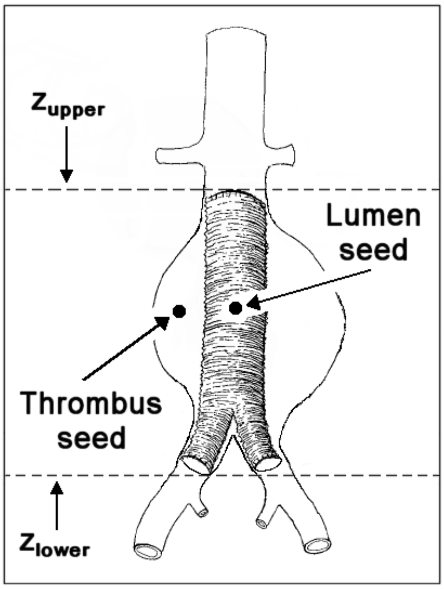


Figure 4.5: The four initial values selected through the manual initialization of the CT segmentation.

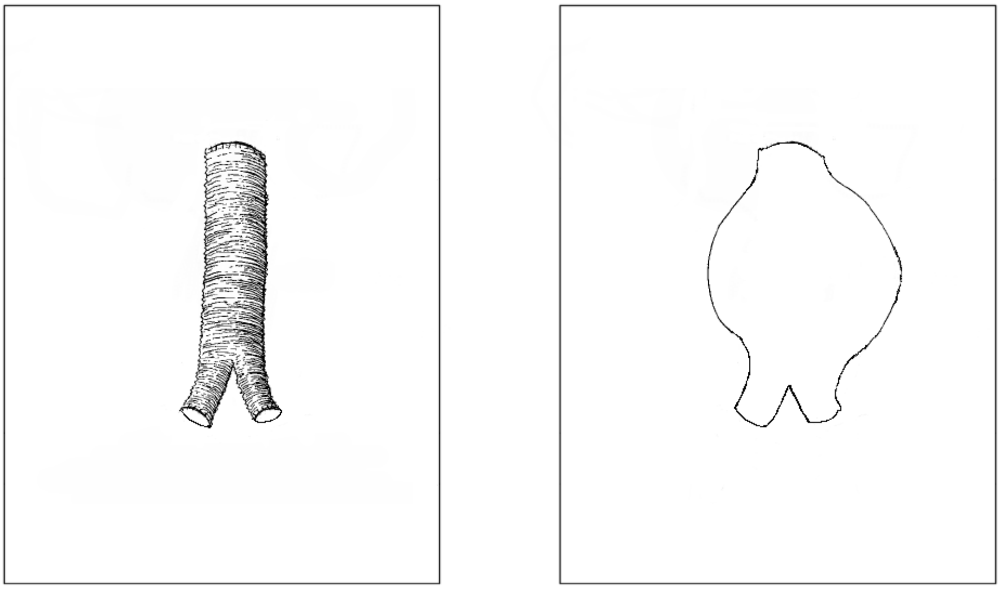


Figure 4.6: Illustration of the desired results of the segmentation process. To the left, the region acquired by lumen segmentation. To the right, the region acquired by thrombus segmentation.

The desired outcomes of these two steps are illustrated in figure 4.6. When the segmentation of a structure is completed, the processed subvolume is re-inserted in the correct position into a volume of the same size as the original input.

4.4.3 Automatic lumen segmentation

The lumen is segmented in 3D using the level set method.

The aortic vessel structure is generally not visible in all slices of the CT volumes. Especially, it tends to be obscure in the lower slices of the image, below the bifurcation point. If the image volume is not delimited by the two slices, as shown in figure 4.5, the evolving level set region may eventually tend to grow into irrelevant parts of the image or the segmentation may become inaccurate in image regions where traces of the vessel structure gradually disappear. This will be of no concern if a subvolume is properly delimited prior to segmentation.

When the lumen segmentation commences, a tiny initial volume is initialized around the seed point. The initial volume may be of any shape, as long as it is completely confined within the edges of the lumen. The level set algorithm will then grow a region from this initial volume until it occupies the aortic lumen in its entirety. At first, the small volume will expand evenly outwards. As the expanding region is constrained by the edges delimiting the lumen in the potential image, it will progress to expand mostly upwards and downwards. Eventually, it is contained by the two slices delimiting the subvolume, and the volume will stop growing. As the measured growth of the region drops below a specified limit, the lumen segmentation is ended.

4.4.4 Automatic thrombus segmentation

The thrombus is segmented in both 3D and 2D slice-by-slice using the level set method.

The thrombus is a much harder region to segment than the lumen. The outer aortic wall and the aneurysm may at times be very hard to distinguish from the surrounding soft tissues, and edge information may be scarce or even missing from the potential image. In such cases we hope that the possibilities of curvature restriction in the level set algorithm will prove to be useful.

To segment the thrombus, the intensity value sampled from the thrombus seed point is initially used to mask the previously segmented lumen structure, so that the traces of the lumen edges will be eliminated when a new potential image is produced.

When we segment the thrombus, the obtained segmentation of the lumen is used as the initial level set. This is practical as it is located completely within the borders of the outer aortic wall. Since the masking procedure has now removed the edges which constrained the growing region when we segmented the lumen, the initial level set will now expand freely until it is constrained by the next boundary, which is that of the outer aortic wall. The two selected slices delimiting the subvolume constrains the volume expansion, as it did before. As when segmenting the lumen, when the measured growth of the region falls below a specified limit, the segmentation stops and the thrombus segmentation is complete.

4.4.5 3D segmentation of the lumen

4.4.5.1 Overview

The goal of the 3D lumen segmentation is to segment the inner aortic wall. The segmented region will include the aorta through the lower abdomen, from below the renal arteries, to slightly past the bifurcation point in the lower abdomen, including the top of the iliac arteries.

4.4.5.2 Preprocessing

Figure 4.7 shows two examples of slices from the initial volume. As can be seen in this figure, the graylevel can vary quite substantially from voxel to voxel. This means that even in regions where the graylevel intensity is relatively monotonous on a big scale, the intensity values can be quite variable between single adjacent voxels. This is an unfortunate characteristic when we want to detect edges in the image later on, since edges are marked by intensity changes in the image.

To remedy this problem, we first employ a smoothing filter. The gaussian filter blurs an image by reducing detail and noise at the voxel level, while leaving structures and regions on a bigger scale relatively intact. The degree of smoothing is determined by the variance of the gaussian; a bigger variance means a greater degree of smoothing. Thus, a gaussian outputs a weighted average of each voxel's neighborhood, with the average weighted more towards the value of the central voxels. The result of gaussian filtering can be seen in figure 4.8.

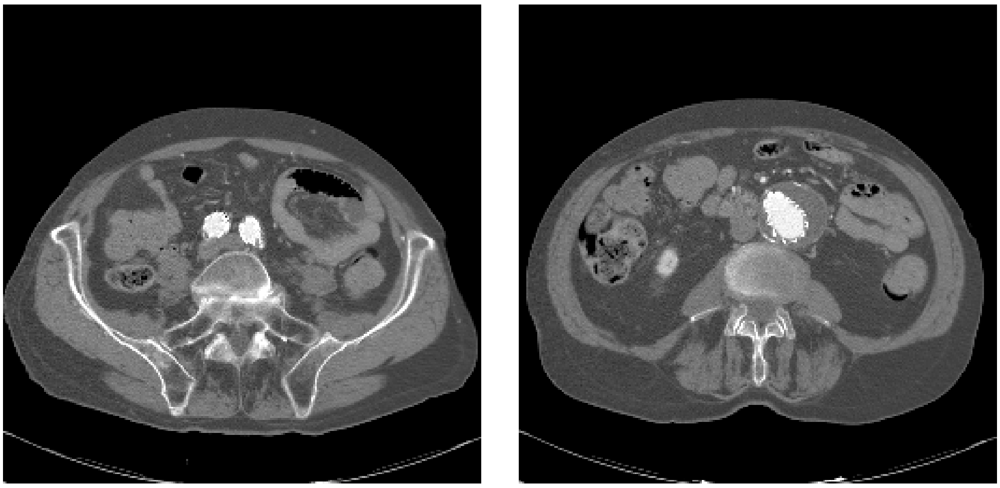


Figure 4.7: Two initial slices from the same unfiltered volume. In the left slice, taken from below the bifurcation point, the lumen can be seen as two bright round regions next to each other in the middle of the picture. In the right slice, taken from above the bifurcation point, the lumen is seen as a single bright region.

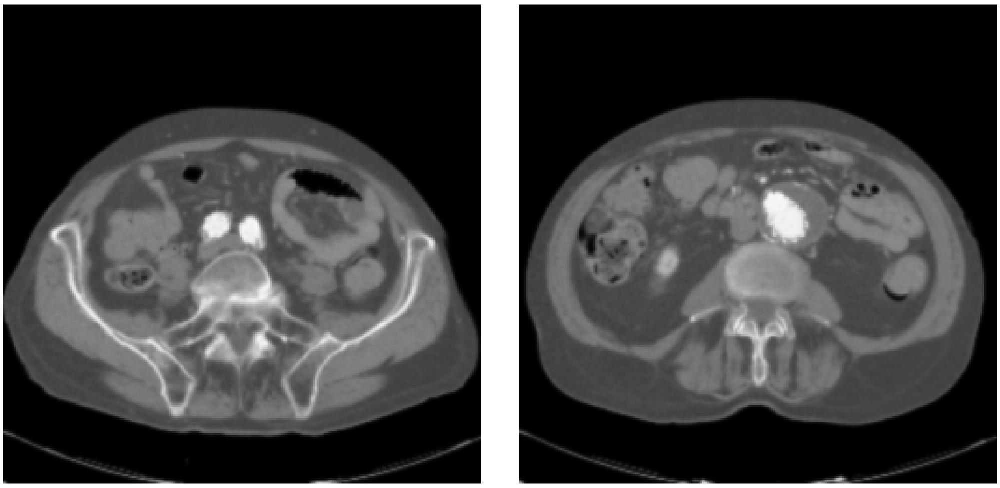


Figure 4.8: Gaussian filtered image, created with DiscreteGaussianImageFilter (variance = 0.9).

Through experiments, the range of values for the variance found to work best is in the range 0.75 to 1.0. By selecting values below this range, the potential image will tend to contain too much noise, reducing the quality of the segmentation. Selecting values above this range, leaves us with an image with washed-out edges, again reducing the quality of the segmentation, as the segmentation will tend to "leak" into surrounding areas more easily.

Experiments indicated that although an even greater degree of smoothing would be necessary to achieve a better segmentation, further use of the gaussian filter would reduce the quality of the edges too much. On the other hand, the median filter was found to accomplish the task of more smoothing quite well, and is therefore applied to the gaussian filtered image for further effect.

The median filter iterates over each voxel in the image. For each voxel a median value is calculated from a cubical region with radius r around the voxel, where r is defined by the user. For a three-dimensional image, this region is typically a cube containing $(2r + 1)^3$ voxels. The median filter is a robust average, since a single unrepresentative voxel in the neighborhood does not significantly affect the median value. Also, since the median value is actually one of the voxels in the neighborhood, the median filter doesn't create any new unrealistic voxel values when the filter is crossing an edge with large differences in graylevel values, for example. This makes the median filter much better at preserving edges than the very closely related mean filter and the gaussian filter.

Experiments showed that if the gaussian filter is applied after the median filter the result is an image with weaker edge characteristics, and, later on, the final image is much more likely to be oversegmented. The result of applying the median filter after gaussian filtering can be seen in figure 4.9.

After these initial steps to smooth the image, it's worth observing that, at this point, the areas in the image, which are similar in intensity on a regional level, contain voxels that are much closer in intensity values than they were to begin with. This is of course what we set out to achieve.

Thus, it makes sense to shift our attention from considering the image at the voxel-level to looking at things at the region-level. That is, in order to create the best possible potential image, we want the region of interest (the lumen in this case) to be as distinguishable as possible in intensity from the surrounding regions, thereby increasing the edge gradients. One way to accomplish this is to increase the image contrast.

Contrast enhancement of the image was implemented follows: First, two threshold intensities are selected. All voxels with intensities below the lower threshold t_{lower} is set to the minimum intensity, i_{min} (0 in our case), and all voxels above the upper threshold t_{upper} is set to the maximum intensity i_{max} (255 in our case). Voxel

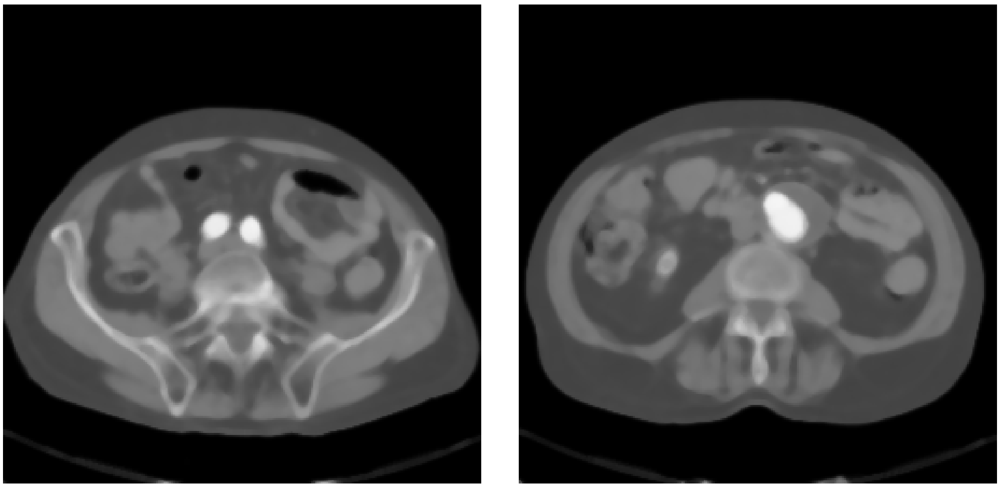


Figure 4.9: Median filtered image, created with MedianImageFilter (radius = 2).

intensities in the range between t_{lower} and t_{upper} are redistributed according to the following equation:

$$i_{voxel} = \frac{i_{voxel} - i_{lower}}{i_{upper} - i_{lower}} i_{max}.$$

As already noted, i_{max} is the maximum possible intensity value, and i_{voxel} is the intensity of the voxel to be recalculated. The upper and lower threshold intensities, i_{upper}, and i_{lower}, are given relative to i_{max} by

$$i_{lower} = t_{lower} \cdot i_{max}$$
$$i_{upper} = t_{upper} \cdot i_{max},$$

where t_{lower} and t_{upper} are values between 0 and 1.

Adjusting contrast generally helps to reduce the impact of the varying intensity levels introduced by the data conversion process described earlier. As a lot of voxels in the lumen region will be thresholded by the upper intensity limit in the process, the region becomes more uniform. Edges will also show up more clearly, and it is easier to distinguish between regions of significantly different intensities. Without adjusting the contrast at all, the segmentation will generally be poor or sometimes unacceptable with frequent undersegmentation. Adjusting it too much will usually lead to oversegmentation and generally a structure containing a significantly less detail than can be seen in the original image.

Thus, it is important to take great care when we adjust the intensity level using thresholds in this way. Generally, we have attempted to adjust the contrast as much as possible, but without losing important details in the image. Contrast adjustment is of great help in segmenting the image, and especially for images with weak intensity characteristics.

Through experiments, the values found to produce the best results are given by:

$$0.20 < t_{lower} < 0.35 \quad \text{and} \quad 0.65 < t_{upper} < 0.80.$$

Selecting values from these intervals will generally produce good segmentations. For the segmentations performed, the values are set to $t_{lower} = 0.275$ and $t_{upper} = 0.667$, giving $i_{lower} = 70$ and $i_{upper} = 170$. Figure 4.10 shows the output after increasing the contrast in the image.

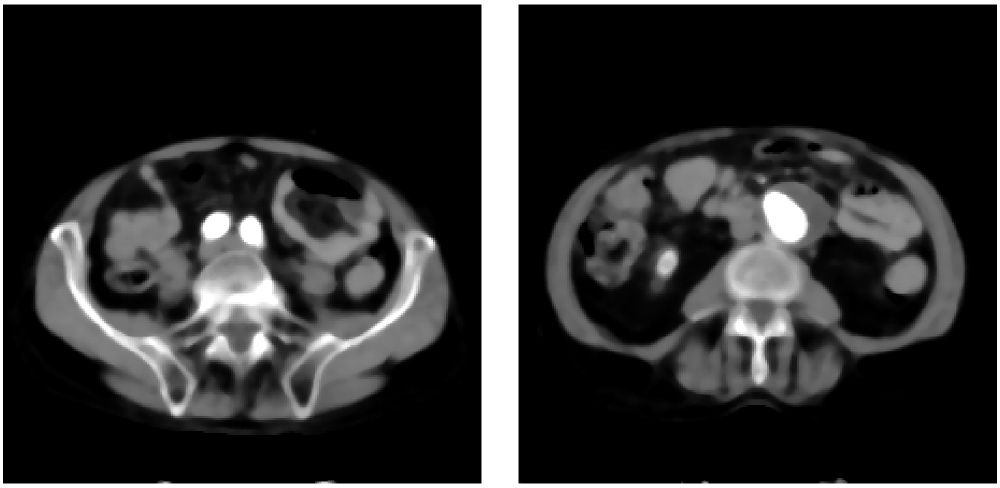


Figure 4.10: Contrast adjusted image ($i_{lower} = 70$ and $i_{upper} = 170$).

The gradient image is then calculated as seen in figure 4.11. As desired, the image features regions with low intensity in regions where the intensity in the original image is homogeneous and higher intensities in regions where the graylevel is changing rapidly.

The dynamic range of the image is then optimized, which means that we increase the span of the graylevels so that it covers its maximum possible range. The voxel intensity values are redistributed linearly so that the voxel with the lowest value gets intensity i_{min}, and the voxel with the highest value gets intensity i_{max}.

The image in figure 4.12 is what we convert to a potential image for the level set filter. Creating a potential image involves scaling all intensity values in the image according to the formula

$$P(I) = e^{-|I|},$$

where I is the intensity of the image used as input.

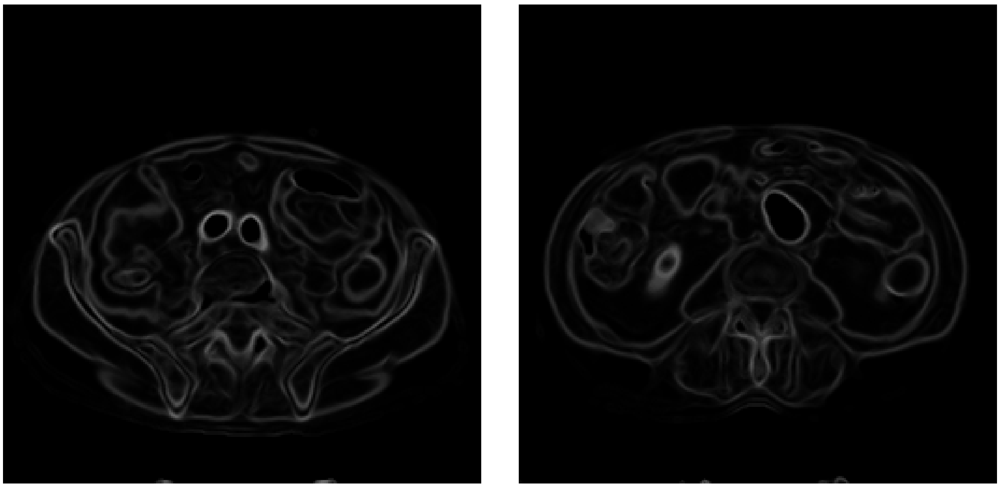


Figure 4.11: Gradient magnitude image, created with GradientMagnitudeImageFilter.

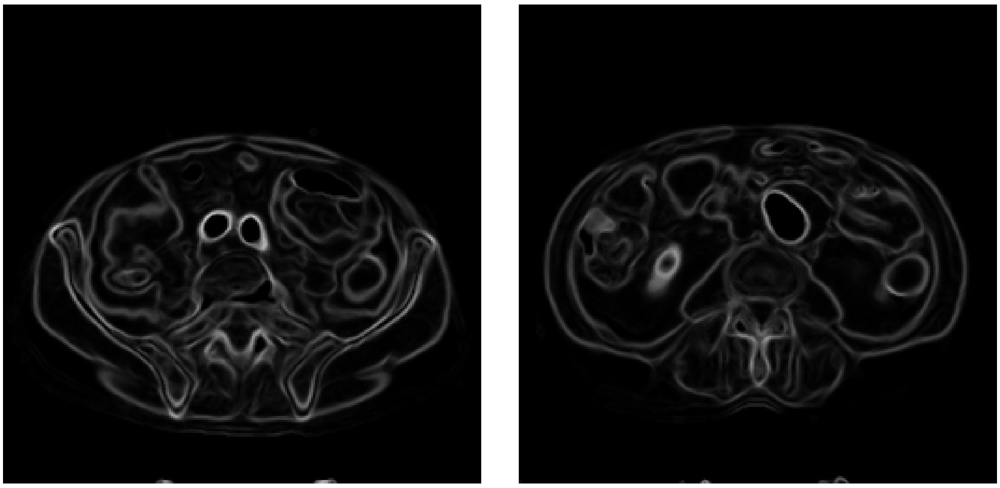


Figure 4.12: Gradient image with optimized dynamic range, created with RescaleIntensityImageFilter (OutputMinimum = 0, OutputMaximum = 255).

Ideally, the resulting potential image has values equal to zero in regions with constant intensity and intensities close to one at the edges. In practice, this is hard to achieve perfectly, but as long as the edges delimiting the region of interest are much stronger than edge traces found in the interior of the region, this will generally not be a problem.

4.4.5.3 Segmentation

After these preprocessing steps, we can finally initiate the segmentation. As a first attempt at applying the level set approach, the fast marching method, described in section 3.5.4.8 was tested (FastMarchingImageFilter in ITK). This special case applies to our problem because the speed of the evolving front we observe never changes sign, as it always moves outward from the initial level set. The advantage of this method is it's speed, as it can do a segmentation much faster than the general level set method. Its most serious drawback however, is that it does not encompass the same possibilities for restraining the curvature as does the general method. Thus, it turns out to have a weakness very similar to that of the fuzzy connectedness algorithm tested earlier, as the solution will tend to evolve into surrounding regions even if only very small sections of edges are missing in the potential image. This was quickly discovered upon testing the algorithm, as over-segmented images were frequently the result, and for this reason fast marching was abandoned in favor of the general level set approach.

The level set filter in ITK takes as input the potential image and the image of the initial level set surface. A tiny volume in the shape of a 3×3 cube is initialized around the seed point. This initial volume of 27 voxels serves as the initial level set. (A volume is initialized because the ShapeDetectionLevelSetFilter in ITK is unable to grow regions from a single voxel. This has no negative consequences for us, however.) A set of numerical parameters is also passed to the filter. These numerical parameters will be further explained in the following.

The level set algorithm itself doesn't specify any stopping criteria for the segmentation, as it is only a set of rules for how to evolve the relevant interface. Rather than just have the algorithm execute a large number of iterations, and then stop, stopping criteria are applied to halt the evolution of the solution to achieve the best possible segmentation.

The implemented method works by having the level set filter iterate on the image for a specified number of iterations, after which stopping criteria are checked. If the stopping criteria are met, no further segmentation is performed, and the segmented image is postprocessed. If not, the level set is reinitialized, using the currently segmented region as the new initial level set, and the iteration continues again for the specified number of iterations. This procedure goes on until the stopping criterion is met.

The implemented stopping criterion is based on a measure of the growth of the segmented region, which is checked after the specified number of iterations. The growth of the segmented region is measured as the ratio of the voxel count in the most recently segmented region relative to that of the previous one. This ratio is a number r, where $0 < r \leq 1$. The iteration stops when r reaches a certain specified limit, which depends on the application and is found through experiments. Naturally, the typical ratio is a number close to 1, which indicates that the process will stop iterating when the solution evolves only very slowly or has stopped evolving completely.

The exact parameters used for the 3D lumen segmentation are listed in table 4.1. These values were all determined by experimenting with the implemented setup until the values that produced the best results were found.

In the table, the parameters “iterations” and “ratio” are the same as those just described. The parameters “epsilon” and “narrowband” were described in chapter 3. The “timestep” parameter is related to the numerical implementation of the level set filter in ITK. Although details of the numerical implementation of the ITK filters are not the focus of this text, a brief description of this parameter is provided, since it is used explicitly: The timestep controls the speed at which the solution is allowed to converge from one iteration to the next. The parameter is used to maintain stability in the solution by limiting the maximum change that a voxel can make in any one particular iteration. It must be low enough to keep the solution stable, but should be high enough to keep the solution progressing at a reasonable rate. (At some point in the future, functionality will be implemented in ITK to determine this parameter automatically.)

parameter	iterations	epsilon	narrowband	timestep	ratio
value	10	0.4	4	3	0.998

Table 4.1: The parameters used for the 3D lumen segmentation.

Selecting good values for the timestep, the width of the narrow band and the number of iterations was done as follows: First, the value used for the timestep was found through experiments. For values higher than the one listed in table 4.1, the solution would run the risk of becoming unstable, and for lower values, the solution would progress at a very slow rate. Generally, the progress of the solution is slow, even for the highest timestep value that produces stable solutions. In this context, “slow” means that the region will grow with too few pixels over one iteration so that the ratio measure will easily fluctuate too much to be useful as a convergence criterion if it were to be checked after every single iteration. For this reason, the highest possible timestep value was used.

Having found a good value for the timestep, the next issue was to determine an appropriate number of iterations to be executed between each time the stopping criterion is checked. The difference between the solutions of two consecutive iterations, using the selected timestep, is still very small. To make it big enough for our stopping criterion to be viable, we increase the number of iteration between each check. Starting with one iteration, the number of iterations was increased until the solution evolved sufficiently between each condition check to make our ratio test for growth a reliable stopping criterion.

Having accomplished this, the last thing to do was to find the smallest possible narrow band that would contain the entire progress of the solution for the chosen timestep and iteration values. Starting with a narrow band of 10, this number was decreased as much as possible, while still containing the region evolved between the chosen number of iterations.

Through experiments, it was determined that, for the same preprocessing steps, the final solution would be very similar by selecting different (reasonable and working) sets of these three parameters. The primary gain lies in a much faster convergence to a final solution if a good set of parameters is selected. For an unfortunate selection of parameters, finding a solution may take hours, while for a good selection, a solution can be found in a matter of minutes (on a fast PC). Generally, for these three parameters, the same values were found to work equally well when segmenting the thrombus in 3D and 2D later on.

Regarding the epsilon value, the edges of the lumen structure are generally very clear and discernable, and the places where edge information is lacking are usually few and far between. As explained in chapter 3, the epsilon value regulates the curvature of the expanding front. A greater value indicates that the front will tend to evolve more smoothly and uniformly. This has the advantage of keeping the region from leaking through gaps in the potential image in places where edge features are weak. Choosing a too big epsilon, close to 1, can make the segmentation stop before the narrowest parts of the aortic structure are fully segmented. This can typically happen in the iliac arteries in the bottom of the image. Selecting very small values, closer to 0, might cause the solution to evolve into regions on the outside of the lumen, but because of the generally good quality of the potential images, this was found to occur rarely, even for small values of epsilon. In this case, finding a value for the parameter that produced a good solution was relatively easy.

As for the value of the ratio parameter, experiments showed that it was unproblematic to select a value very close to 1, as the progress of the solution (and hence the growth of the segmented region) would drop off rapidly as the confining edges of the potential image were reached. Several values were tested for segmenting the test data, and the value producing the most accurate segmentations was chosen.

The output of the level set filter is a binary image as seen in figure 4.14. A display of the evolution of front of the region can be seen in figures 4.15 and 4.16.

```
initialize 3D level set
while (stopping criterion not met)
{
  reinitialize 3D level set
  for (number of iterations)
  {
    do one level set iteration
  }
}
```

Figure 4.13: Pseudocode for the 3D level set segmentation of the lumen.

Figure 4.14: Slices from the 3D level set filtered image, created with ShapeDetectionLevelSetFilter. (The numerical parameters used are listed in table 4.1.)

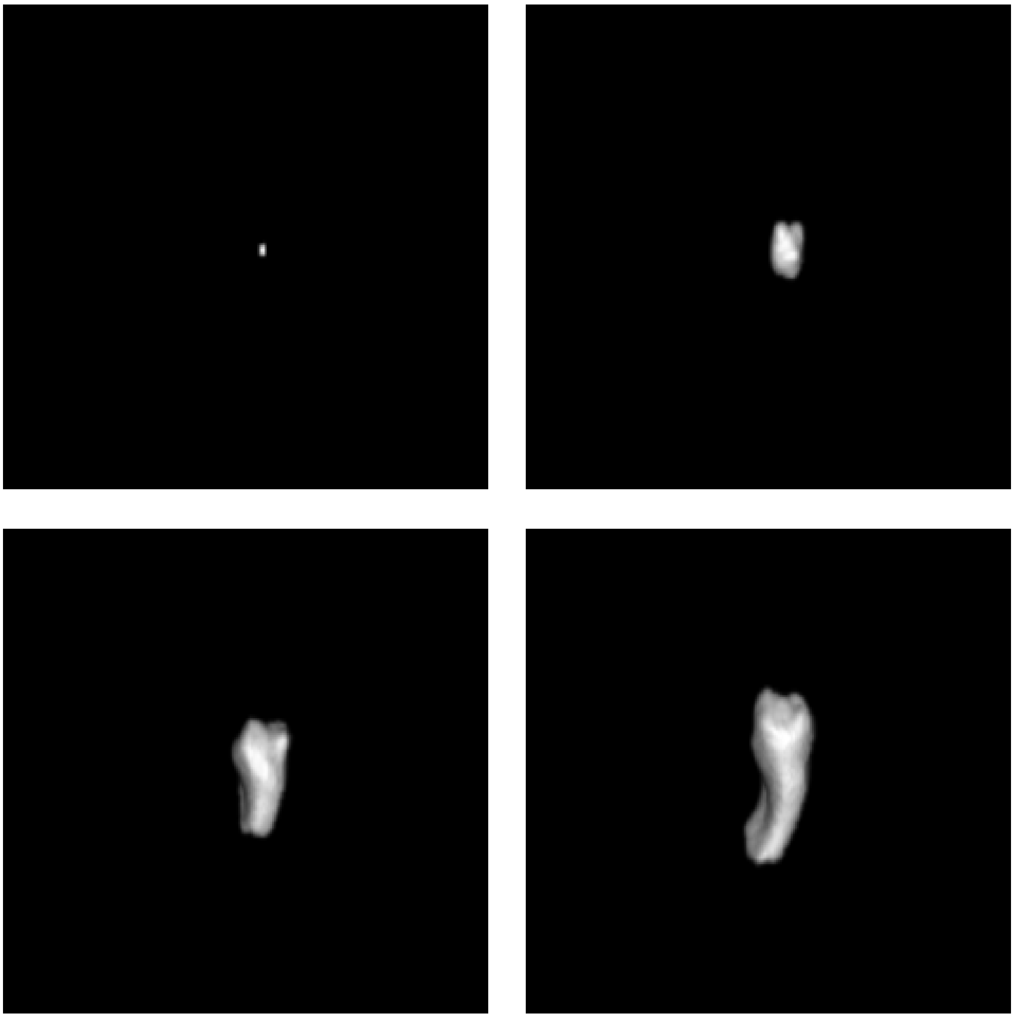


Figure 4.15: 3D level set filtering of the lumen, showing the evolution of the solution. From top left to bottom right, the images show the initial cubical level set and the segmented lumen region after 10, 20 and 30 iterations.

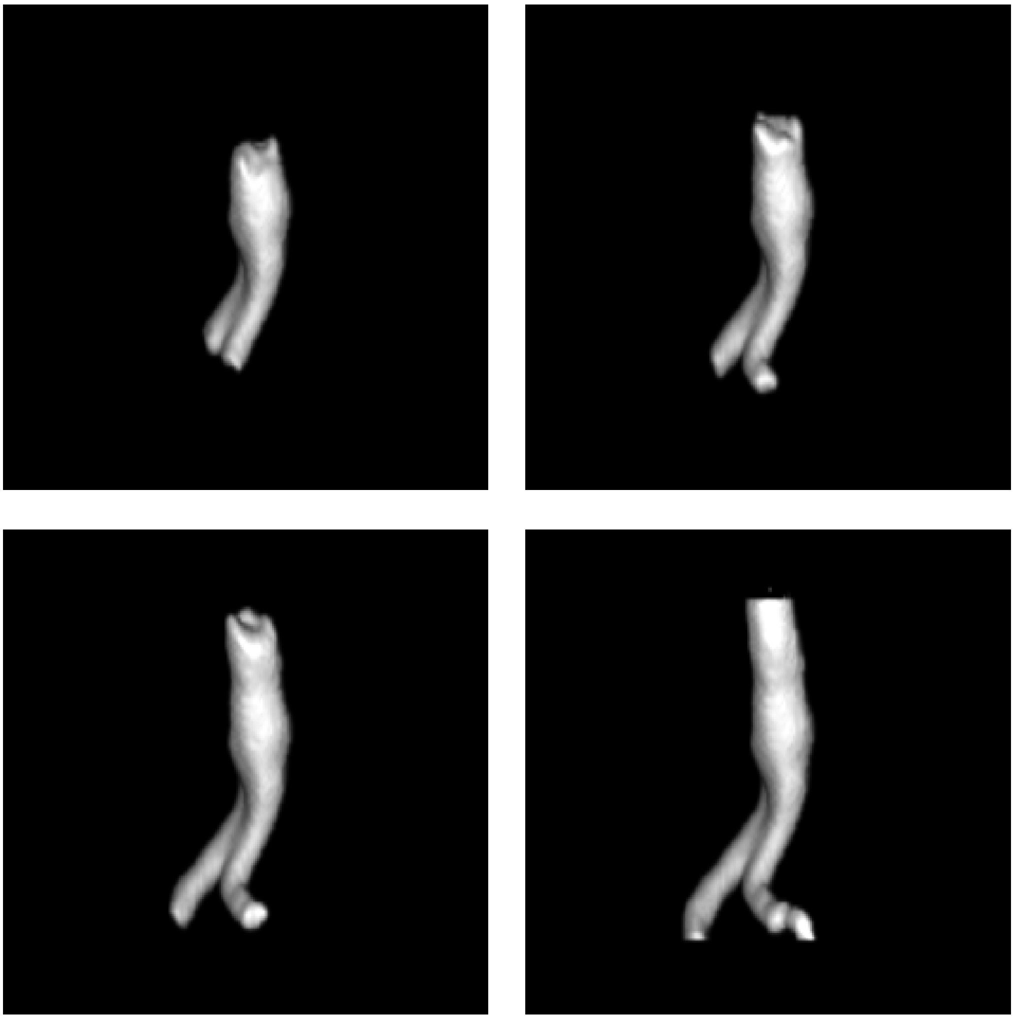


Figure 4.16: 3D level set filtering of the lumen, showing the evolution of the solution. From top left to bottom right, the images show the segmented lumen region after 40, 50, 60 and 100 iterations. The lower right image is also the final solution.

4.4.5.4 Postprocessing

In the postprocessing step, we wish to improve the shape of the segmented region, remove sharp corners and generally smooth the edges of the segmented region so that it conforms better to the actual anatomy of the lumen. A variety of techniques for smoothing the edges of the segmented region were tested, among them gaussian filtering with thresholding, dilation and erosion filtering and median filtering. (Using Fourier filters was also thought of, but ITK doesn't yet contain any functionality for Fourier filtering.) Of these methods, the median filter was the filter producing the best results. The problem with the median filter however, is that it is very slow, especially when the filter radius is big and the image to be filtered is in 3D. Specifically, even using a relatively small radius of 2 (or 3 later on) means that the filter, for each voxel in the image, has to sort a total of 125 and 343 voxels respectively. The typical CT test image contains about 10^8 voxels. Using the MedianImageFilter in ITK, with a radius of 3, this task took approximately 15 minutes (on a fast PC).

Figure 4.17: Binary median filtered image, created with BinaryMedianImageFilter (radius = 2).

Rather than dismiss the filter completely (which would also have implied less satisfactory results) because of its time penalty, it was decided to try to find a way of increasing performance specifically for filtering of binary images. The optimized filter developed uses the following two facts:

- Only the subportion of the image containing the segmented object needs to be filtered.
- Since the image is binary, the median can be obtained by simply counting

the voxels rather than sorting them. When more than half the voxels in a neighborhood has been found to be of either one of the two intensity values represented in the image, the median has been obtained.

Using these optimizations, the new BinaryMedianImageFilter processed the aforementioned image in about 30 seconds, producing the exact same results. The filter was subsequently submitted to the ITK developers and included in ITK [BINAR-02]. The binary median filtered image is shown in figure 4.17.

The end result of the lumen segmentation, a rendered 3D model of the obtained region, can be seen in figure 4.18.

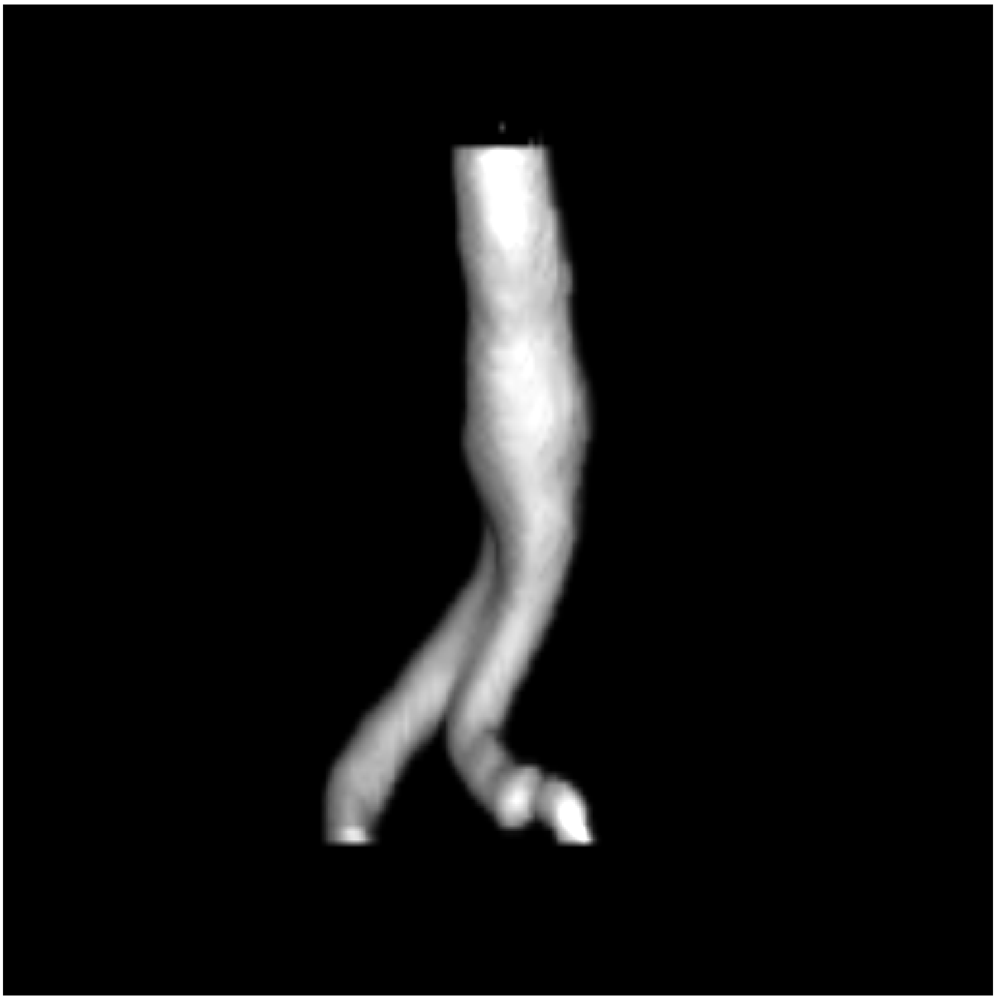


Figure 4.18: Rendered model of the lumen region segmented using level sets in 3D.

4.4.6 3D segmentation of the thrombus

4.4.6.1 Overview

The 3D thrombus segmentation is performed by using the segmentation of the lumen as the initial level set, advancing the position of the front outwards in a similar manner as before until it is constrained by the edges of the outer aortic wall and the stopping criterion is met.

4.4.6.2 Preprocessing

To begin with, the same preprocessing steps as were used for segmenting the lumen were tested. However, the resulting segmentation turned out to be unsatisfactory. This was mostly due to the contrast enhancement, which would tend to enhance differences between graylevels in the thrombus, thus amplifying intensity variations that lead to unfortunate characteristics in the potential image. A different preprocessing pipeline had to be devised for the thrombus segmentation.

Figure 4.19 shows two slices from the initial image.

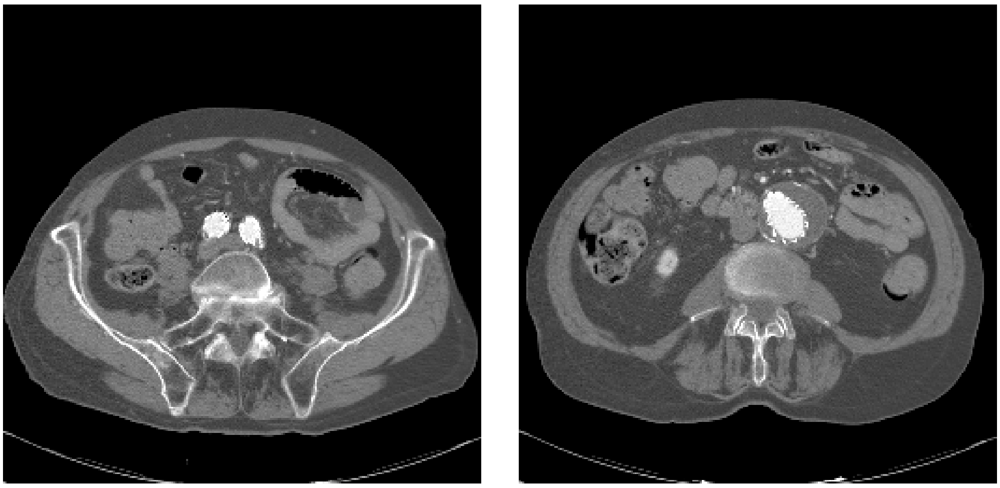


Figure 4.19: Slices from the initial unfiltered image.

When the thrombus is segmented, we want to make sure that there are no interfering edges between the initial level set and the thrombus border. For this reason, we need to mask the brighter voxels of the lumen region to the darker intensities characteristic of the thrombus region surrounding it. Thus, the edges of the lumen won't interfere when we segment the outer aortic wall, as they will be eliminated from the potential image.

The masking is done as follows: First, the segmented lumen structure is very slightly dilated by using a gaussian, so that it covers slightly more voxels than were actually segmented. This is done to ensure that all the brighter pixels that produce edge traces around the lumen region will be masked. The voxels of this dilated structure are then copied into the CT image subvolume using the intensity value of the thrombus seed. The actual intensity value of the thrombus seed point is determined by selecting the median intensity from a $3 \times 3 \times 3$ neighborhood in the initial image 4.19, with this neighborhood centered on the seed voxel. The result after pasting the slightly expanded lumen region into the initial image can be seen in figure 4.20.

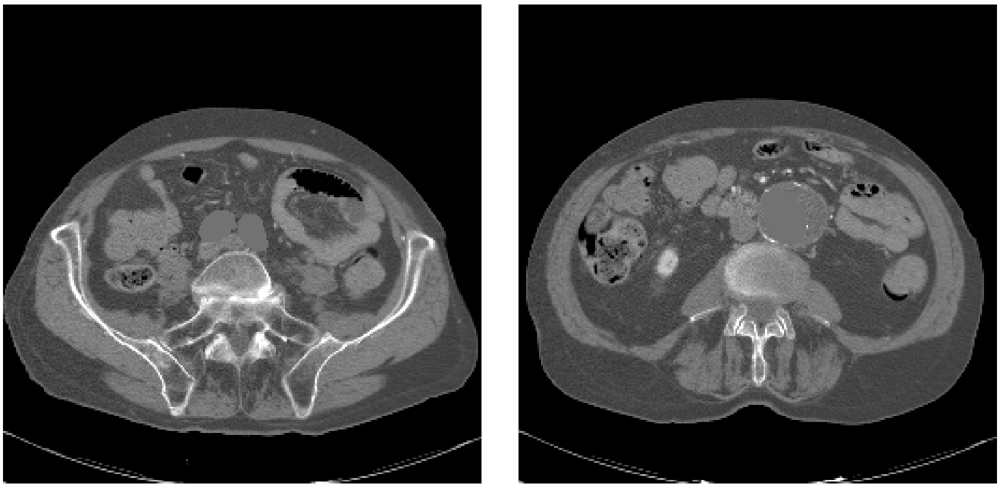


Figure 4.20: Slices from the initial image after the segmented region has been masked.

Care must be taken to not dilate the region too much, as we will run the risk of masking pixels outside the thrombus region. On the other hand it's important not to mask too small an area, as segmentation of the thrombus will be much more difficult if traces of the lumen edges remain inside the region of interest. From experiments, the value found to dilate the area by an appropriate amount is a variance of 0.1. (An obvious alternative to this procedure would be to mask the lumen area in the lumen gradient image directly, thereby eliminating the need for the intensity sample of the thrombus as well. In practice, this approach turned out to result in a very poor segmentation, as the edges of the thrombus are generally not very clear in the edge map of the lumen.)

After masking, all voxels with intensities above i_{upper} are thresholded by setting their intensity value to i_{upper}, as seen in figure 4.21. Note that this is the same value as used for the upper threshold for contrast enhancement in the lumen segmentation. As this is an automatically computed threshold, it is not accurate enough to be effective for removing the lumen edges without the prior masking. However, used in combination, these two steps were found to remove enough of the traces of the lumen region for the thrombus segmentation to proceed without significant problems.

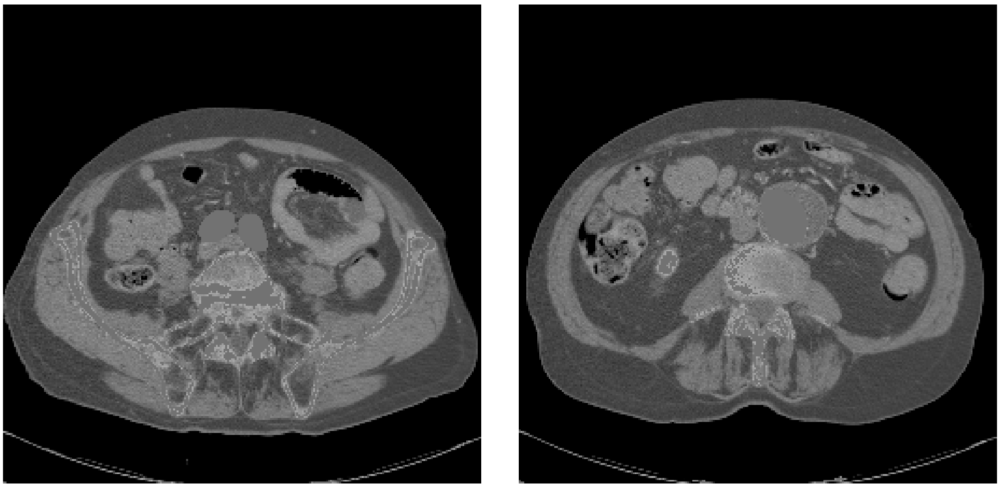


Figure 4.21: Intensity values above $i_{upper} = 170$ have been thresholded off. Voxels with intensities above this limit have all been set to i_{upper}.

We apply the gaussian operator on this image to reduce intensity variations on a voxel-to-voxel scale to obtain more uniform regions. The result of this step can be seen in figure 4.22.

In figure 4.23 we have used the median filter for further smoothing. Again, applying a median filter for smoothing proved to make regions with similar intensities more uniform and edges are better preserved compared to simply increasing the variance of the gaussian operator.

The gradient magnitude image calculated is shown in figure 4.24.

We then optimize the dynamic range of the edge image in exactly the same way as we did when we segmented the lumen. The result can be seen in figure 4.25.

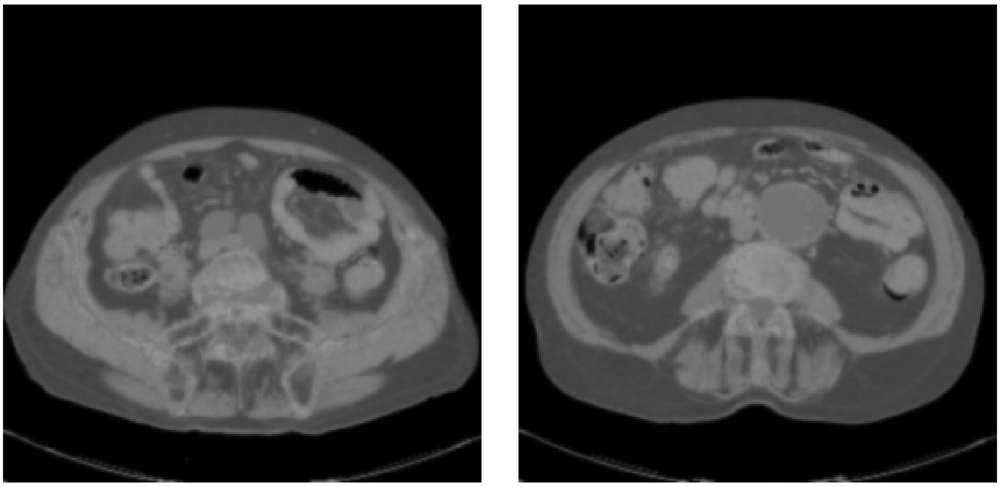


Figure 4.22: Gaussian filtered image, created with DiscreteGaussianImageFilter (variance = 1.0).

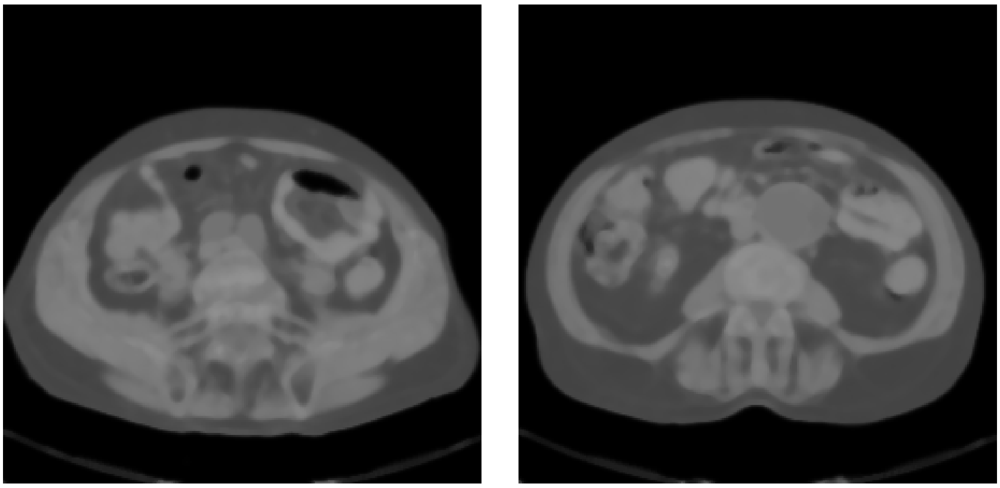


Figure 4.23: Median filtered image, created with MedianImageFilter (radius = 2).

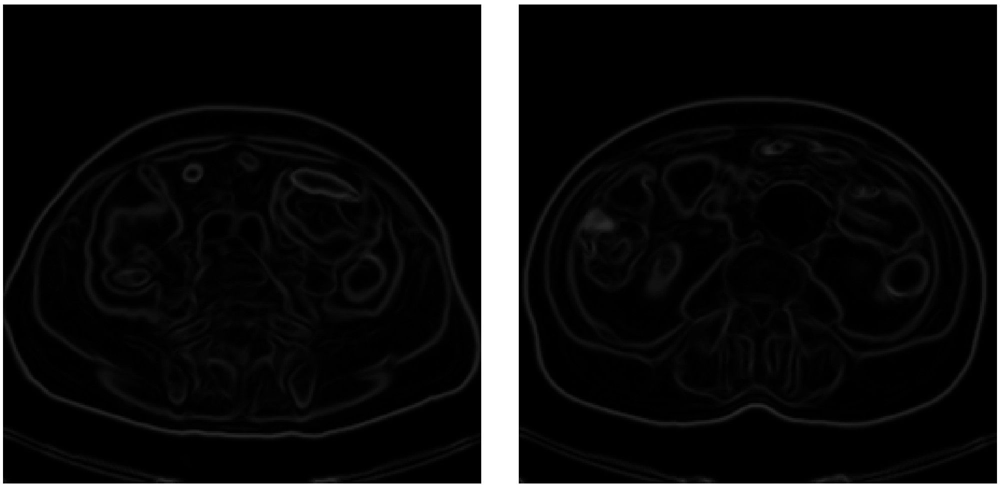


Figure 4.24: Gradient magnitude image, created with GradientMagnitudeImageFilter.

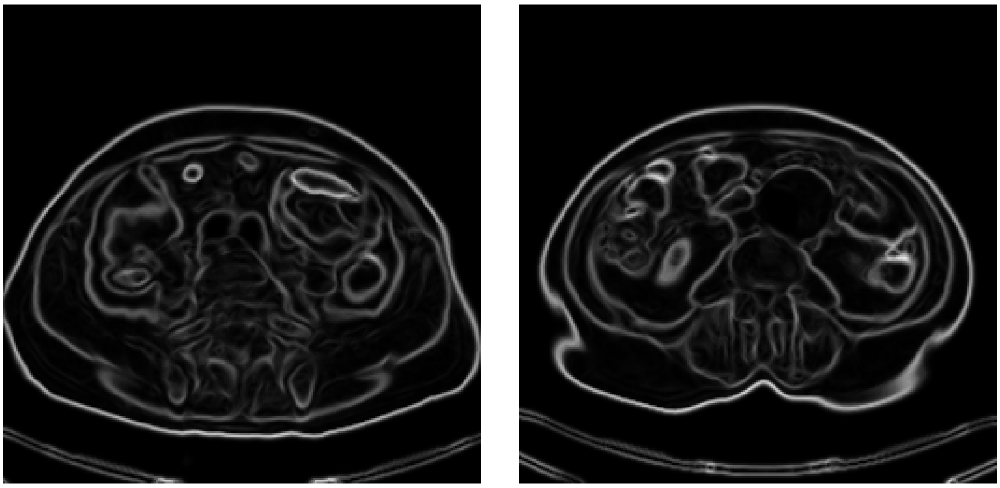


Figure 4.25: Gradient image with optimized dynamic range, created with RescaleIntensityImageFilter (OutputMinimum = 0, OutputMaximum = 255). Note how different the edge features in these slices are from those shown in figure 4.12, especially how the traces of the lumen have been removed and those of the thrombus are more prominent.

4.4.6.3 Segmentation

The preprocessed image is then passed on to the level set filter to be used as the potential image. The model of the lumen segmented in section 4.4.5 is also input to the filter, as the initial level set. The parameters used for this segmentation are listed in table 4.2.

parameter	iterations	epsilon	narrowband	timestep	ratio
value	10	0.6	4	3	0.994

Table 4.2: The parameters used for the 3D thrombus segmentation.

Note that a slightly bigger epsilon-value is used this time, compared to the 3D lumen segmentation. The reason for this is that the edges of the thrombus are much less prominent in the potential image than was the case with the lumen. Much bigger portions of the edges will be missing, and there is a greater need to restrain the expansion of the solution. Decreasing the parameter below the value in the table easily causes the segmented region to expand beyond the borders of the thrombus. Increasing it, however, sometimes lead to undersegmentation as the region stops expanding before the edges of the thrombus are reached.

The ratio parameter used for the stopping criterion is slightly smaller than the one used for the lumen segmentation. The reason for this is that the edges of the thrombus are of lower quality than those for the lumen, and it was observed that the solution can grow very slowly for a very long time before it stops and even expand beyond the edges if the ratio is set too high. The listed value was obtained through experiments and is set so that we get the most accurate segmentation. If set lower, the segmentation may terminate prematurely with undersegmentation as the result, and if set higher, oversegmentation is more likely to happen.

As mentioned previously, the other parameters values were found to work well, so no additional changes were needed. The resulting output of this step is shown in figure 4.27.

```
initialize 3D level set
while (stopping criterion not met)
{
  reinitialize 3D level set
  for (number of iterations)
  {
    do one level set iteration
  }
}
```

Figure 4.26: Pseudocode for the 3D level set segmentation of the thrombus.

Figure 4.27: Slices from the 3D level set filtered image, created with ShapeDetectionImageFilter. (The numerical parameters used are listed in table 4.2.)

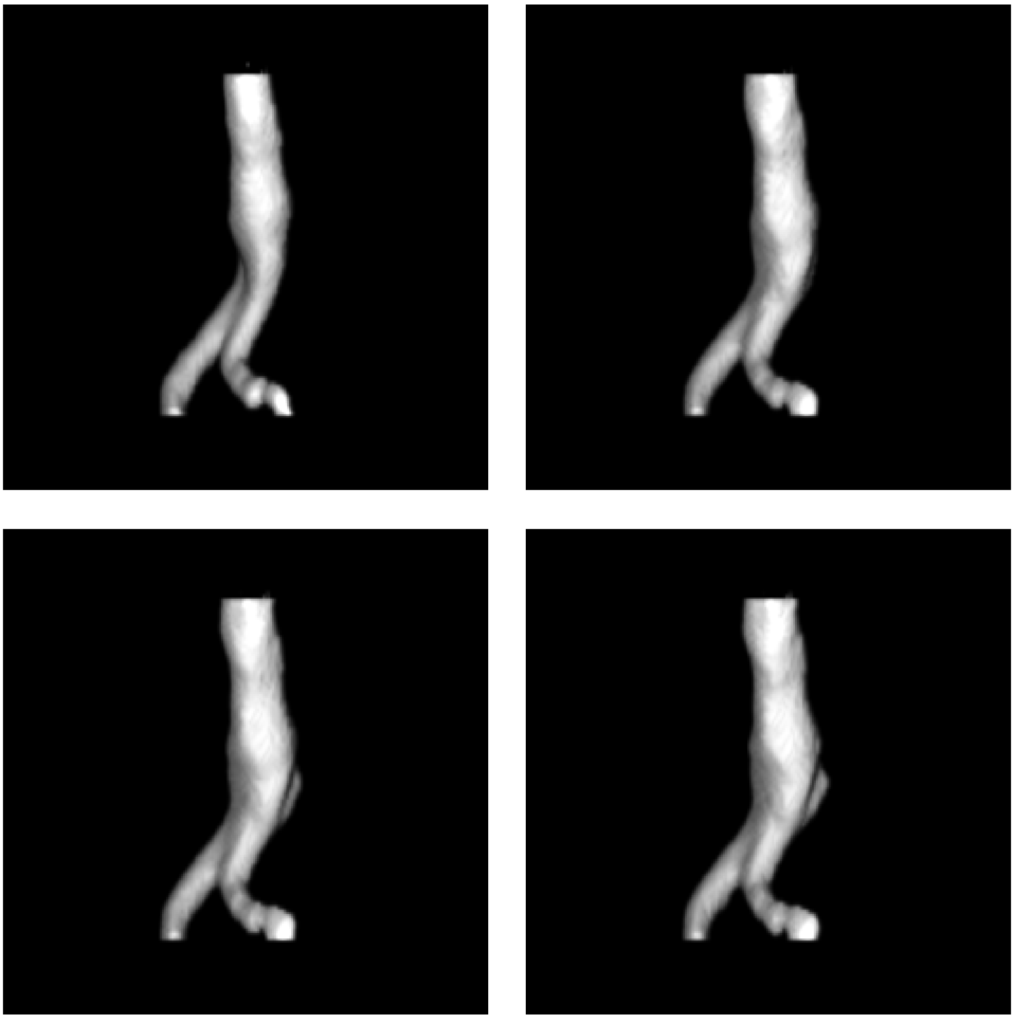


Figure 4.28: 3D level set filtering of the thrombus, showing the evolution of the solution. From top left to bottom right, the images show the initial level set and the segmented region after 10, 20 and 30 iterations. Notice that the initial level set is the same as the segmentation shown in figure 4.18

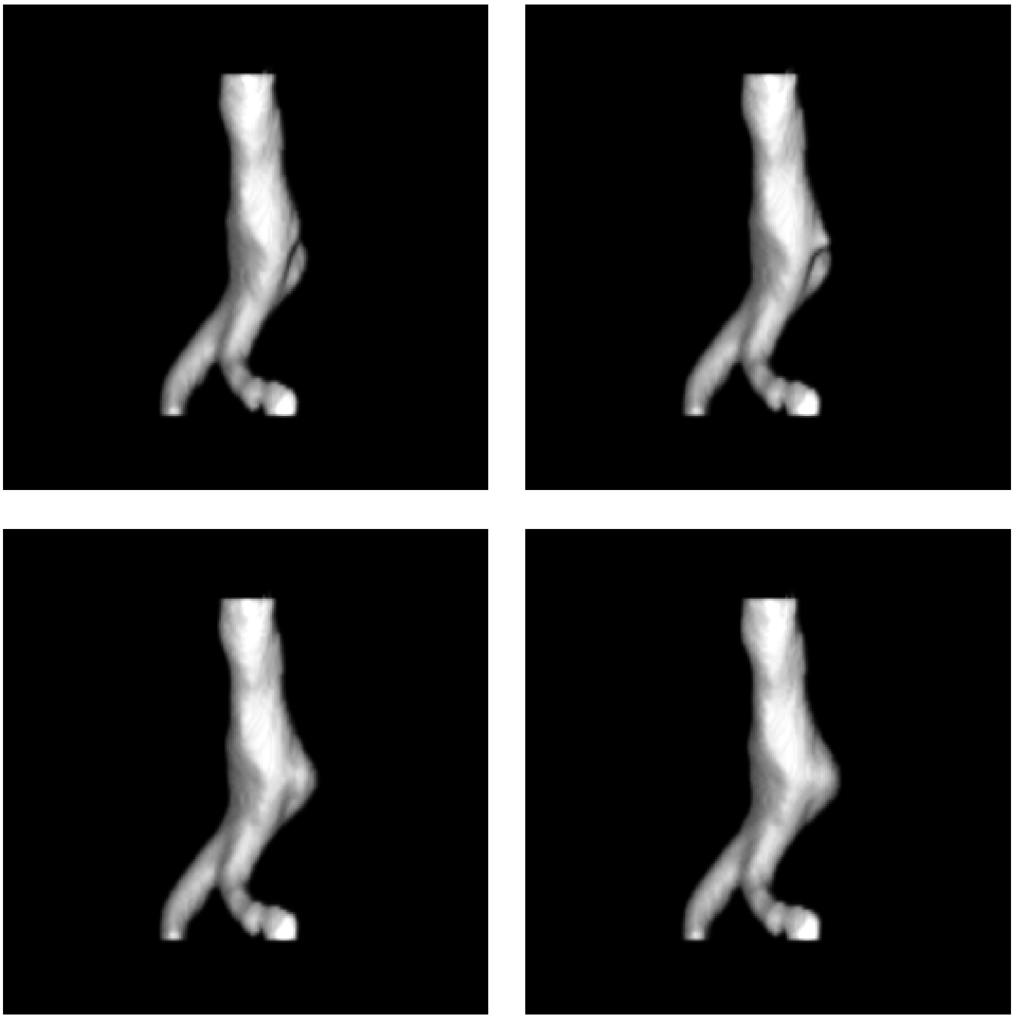


Figure 4.29: 3D level set filtering of the thrombus, showing the evolution of the solution. From top left to bottom right, the images show the segmented region after 40, 50, 70 and 90 iterations. The lower right image shows the final segmentation.

4.4.6.4 Postprocessing

Again, we smooth the edges using a binary median filter. The obtained region is more rugged this time, caused by the fact that the small variations in the potential image, inside the region of interest, are relatively big compared to the potential of the edges. (Remember that these variations was not a problem when segmenting the lumen because the characteristics of the lumen edges were so strong.) Thus, the need for postprocessing the result is more urgent, so a bigger radius is used in the binary median filter. The processed output can be seen in figure 4.30. The rendered end result of the thrombus segmentation is shown in figure 4.31.

Figure 4.30: Binary median filtered image, created using the BinaryMedianImageFilter (radius = 3).

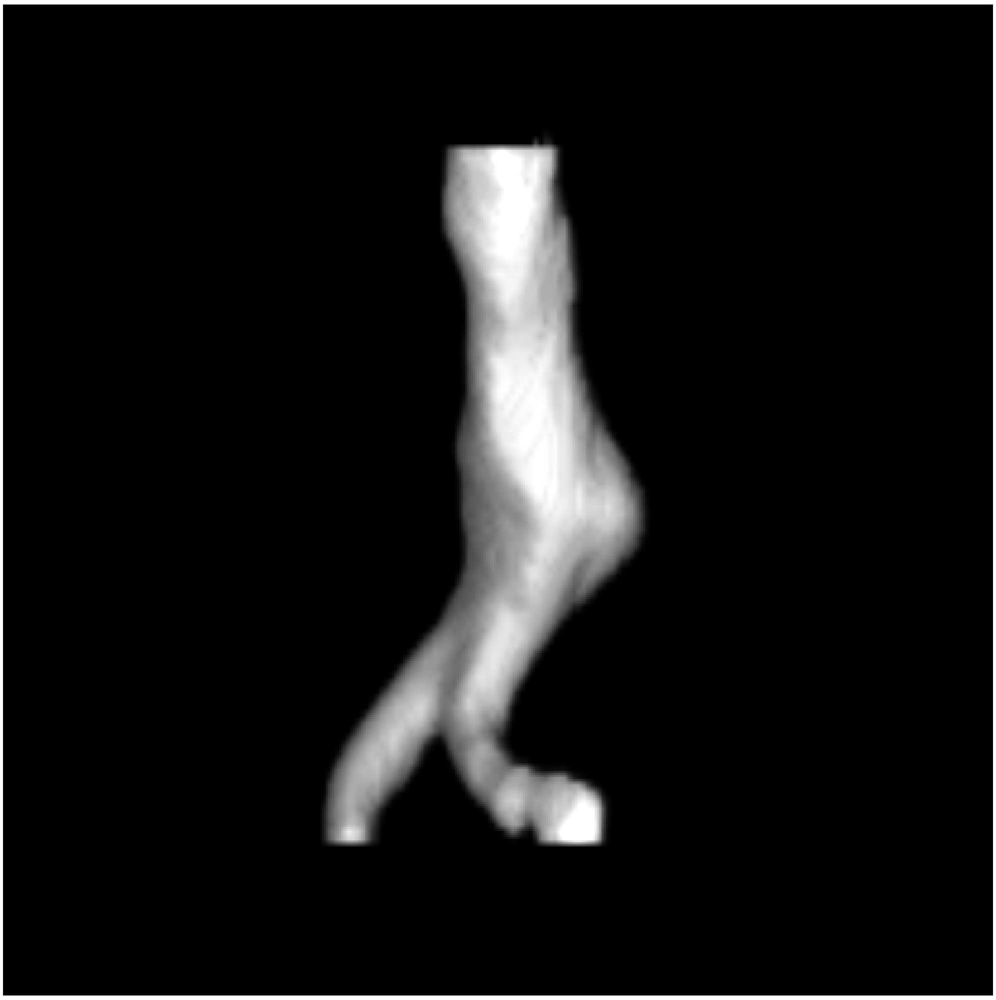


Figure 4.31: Rendered model of the thrombus region segmented using level sets in 3D.

4.4.7 2D segmentation of the thrombus

4.4.7.1 Overview

The 2D thrombus segmentation is also performed by using the segmentation of the lumen as the initial level set. This time, however, we segment the thrombus slice-by-slice, using the slices of the segmented lumen as the initial level sets. After all the slices are segmented, they are inserted into a volume of the same size as the original input to produce a 3D model.

4.4.7.2 Preprocessing

The preprocessing stage for this segmentation is exactly the same as for the 3D segmentation of the thrombus described in section 4.4.6.3.

4.4.7.3 Segmentation

The parameters used for this segmentation are listed in table 4.3.

parameter	iterations	epsilon	narrowband	timestep	ratio
value	10	0.8	4	3	0.995

Table 4.3: The parameters used for the 2D thrombus segmentation.

Compared to the 3D segmentation of the thrombus, the epsilon value has been increased. When segmenting in 2D, a bigger value for epsilon is needed to achieve similar results compared to when 3D segmentation is performed. In 3D, when the algorithm is confronted with a section of the thrombus where edge characteristics are weak, the algorithm will apply the curvature restraint globally. Frequently, edge traces are weak only in a few consecutive slices at a time, which often makes for areas with low potential that are small in height, but bigger in width. In these cases, it is clearly easier to contain the evolving front if the curvature restraint is applied to the whole region rather than independently to successive slices. Thus, a larger epsilon turned out to be required when segmenting in 2D.

Only a minor adjustment is performed on the ratio parameter, compared to the 3D thrombus segmentation. This is done mostly to compensate for the fact that the use of a bigger epsilon generally requires more iterations to evolve into a similar solution, mostly because the evolving front doesn't pass obstacles in the potential image as easily, and therefore expands at a slower rate.

Again, the other values were found to work well, so no changes were needed.

```
for (all slices)
{
  initialize 2D level set
  while (stopping criterion not met)
  {
    reinitialize 2D level set
    for (number of iterations)
    {
      do one level set iteration
    }
  }
}
```

Figure 4.32: Pseudocode for the 2D level set segmentation of the thrombus.

Figure 4.33: Slices from the 2D level set filtered image, created using the ShapeDetectionImageFilter. (The numerical parameters used are listed in table 4.3.)

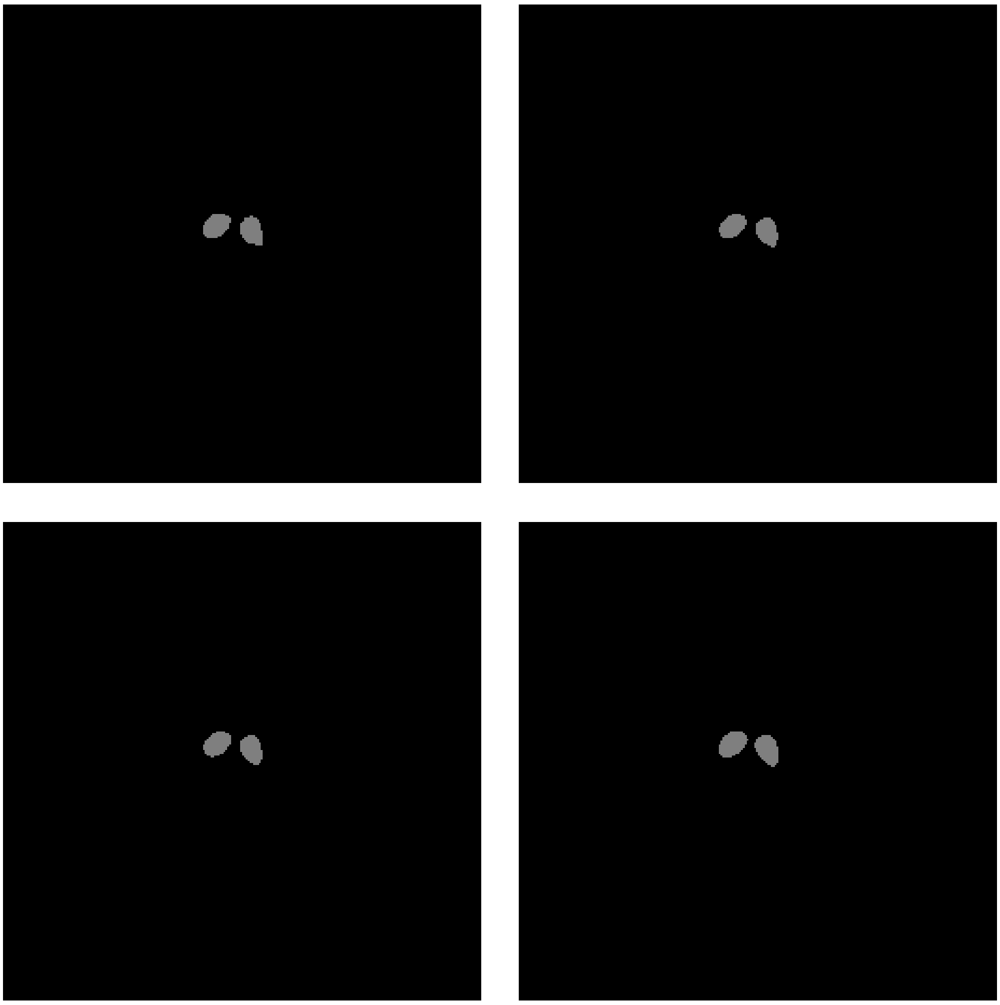


Figure 4.34: 2D level set filtering, showing the evolution of the solution in one of the slices. From top left to bottom right, the images show the initial level set and the segmented region after 10, 30 and 60 iterations. The lower right image shows the final segmentation.

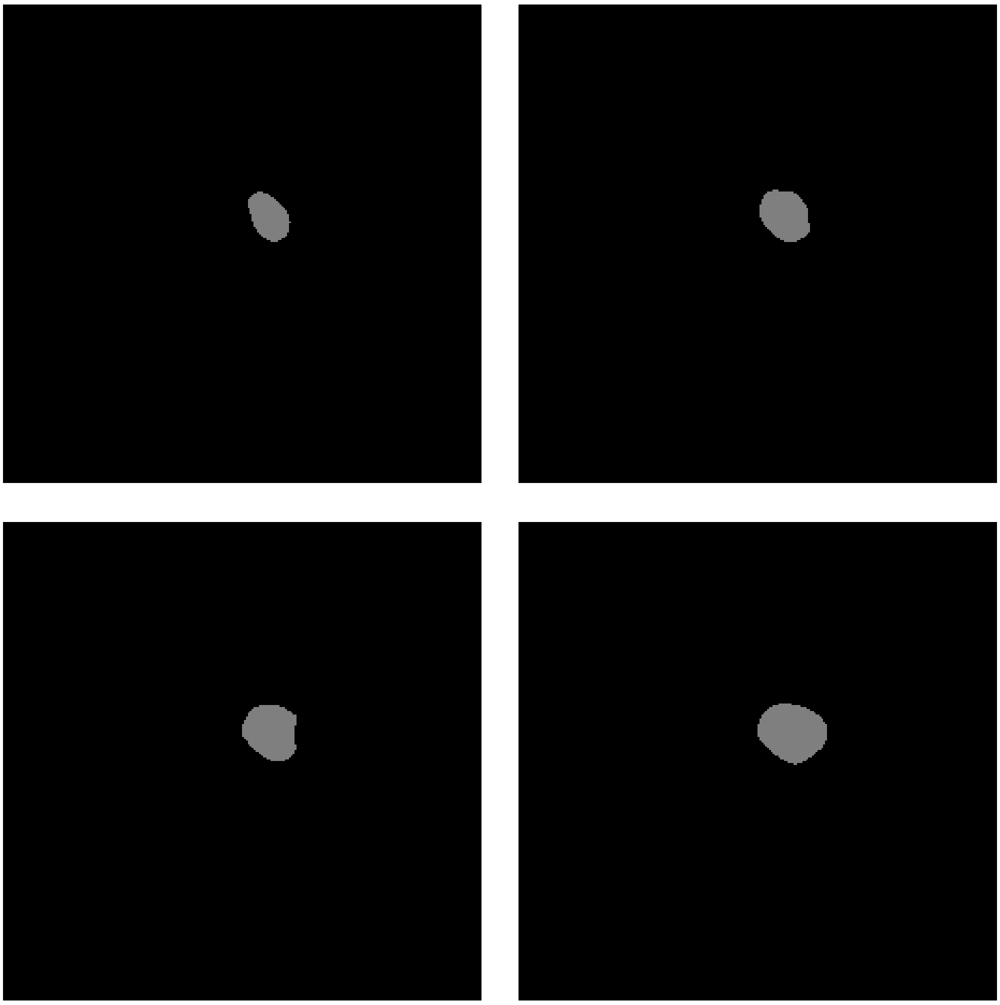


Figure 4.35: 2D level set filtering, showing the evolution of the solution in another of the slices. From top left to bottom right, the images show the segmented region after 10, 30 and 130 iterations. The lower right image shows the final segmentation.

4.4.7.4 Postprocessing

In this case, also, the segmented image is postprocessed in the same way as described in section 4.4.6.4. The binary median filter was found to be very good for processing the 2D segmented thrombus image, since using the median voxel of a neighborhood easily eliminates extreme protrusions or dents in the segmented region caused by single unrepresentative slices. The result of the median filtering is shown in figure 4.36. A rendered model of the outer aortic wall is displayed in figure 4.37.

Figure 4.36: Binary median filtered image, created with BinaryMedianImageFilter (radius = 3).

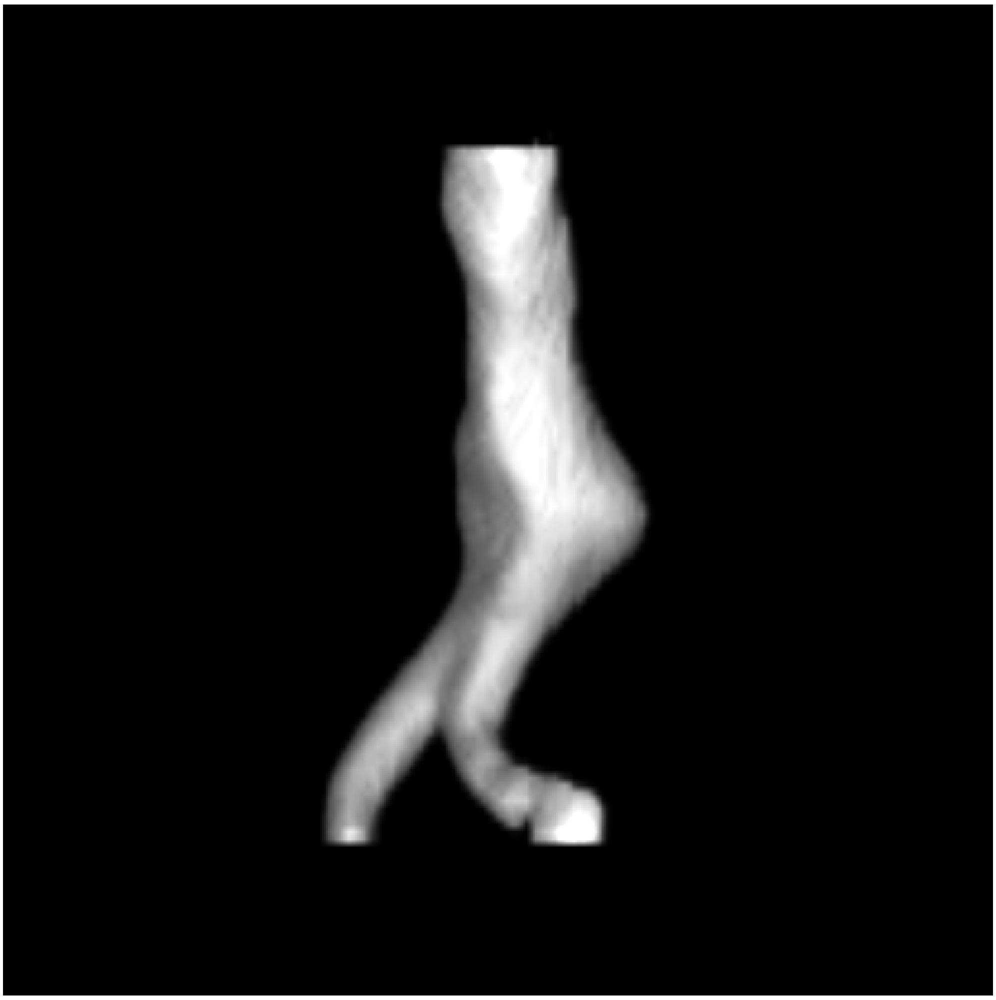


Figure 4.37: Rendered model of the thrombus region segmented using level sets in 2D.

4.4.7.5 The segmentation error

To evaluate the results of the volumetric segmentations performed, it is desirable to measure the difference of the segmented regions to the actual anatomical regions. The segmented region is usually compared slice by slice to the actual region segmented manually. As no manual segmentation of the test images existed, they would first have to be segmented manually if an error estimate was to be produced. However, since manually segmenting both the inner and outer aortic walls from all three test images (all of which contain well over 100 slices) would have been an excruciating and extremely time-consuming task, a compromise was made to select five slices from each test image. In these five slices, both the inner and outer aortic walls were extracted manually, and these manual segmentations were then compared with the equivalent slices from the automatically segmented data.

In all three cases the five slices were selected as follows: Slice 1 is the fifth from the top of the volume, while slice 5 is the fifth from the bottom of the volume. Slice 3 is located at the point where the diameter of the aneurysm is at a maximum, and slices 2 and 4 are located 15 slices above and below slice 3, respectively. Figure 4.38 shows an illustration of the location of the five slices.

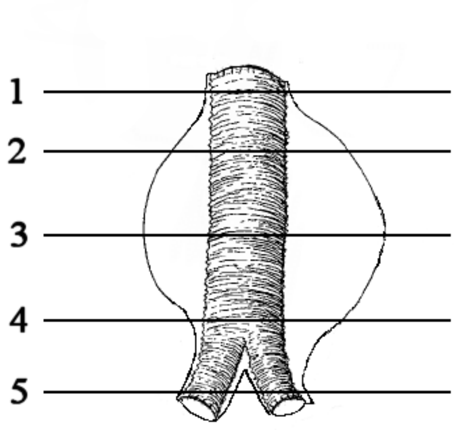


Figure 4.38: Illustration of the positions of the five slices used to calculate the segmentation error.

To measure the error in a slice, a center pixel is selected at the same index in both the manually and automatically segmented slices. Horizontal and vertical lines are then traced in both slices until they reach the edges of the manually and automatically segmented regions. The distance from the center, measured in pixels, is then recorded. The error, denoted by d, is the positive difference between the distances in the automatically segmented slice and the equivalent distances in the manually segmented slice. In the slices where we have a bifurcated structure, this process is done twice, once for each of the aortic branches. An illustration can bee seen in figures 4.39 and 4.40.

When the errors d_1 to d_4 (and d_5 to d_8 for slice 5) have been found, the mean error for each slice is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n} d_n,$$

where n is the number of distances measured in the slice (4 or 8).

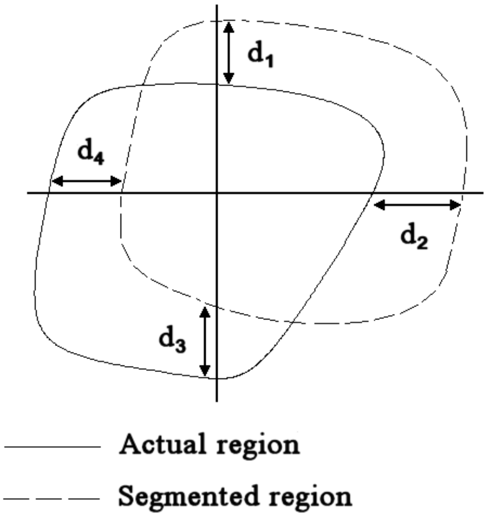


Figure 4.39: Illustration of the error measures in slices 1-4, above the bifurcation point.

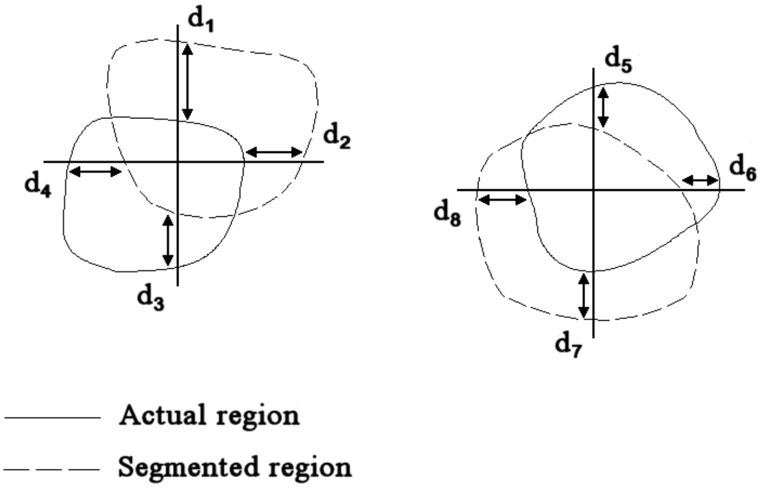


Figure 4.40: Illustration of the error measures in slice 5, below the bifurcation point. In this case, the segmentation error is measured separately for both of the iliac arteries.

The error in millimeters is found by multiplying each result by the pixel spacing. The pixel spacing is the spatial extent of one pixel, in our case measured in millimeters, given as one number for each spatial dimension.

The measured segmentation errors in millimeters are listed in table 4.4. More detailed data for the slices of each image can be found in appendix A.

segmentation	1	2	3	4	5	all
lumen 3D	0.47	0.22	0.71	0.70	1.05	0.63
thrombus 3D	1.95	1.05	1.41	1.61	2.02	1.61
thrombus 2D	2.17	1.17	1.41	1.36	1.82	1.59

Table 4.4: Error measures in millimeters of the different segmentations. The values in each row are averaged over the three segmented test images.

4.5 Implementing mutual information optimization

4.5.1 Background

We want to perform 3D rigid registration using the mutual information metric. Rigid registration means that only translation and rotation of the two images are performed to align them. No scaling is performed, as the scale of the CT and US images are already equivalent. This implies that if the distance between the same two anatomical points in the two images are measured in voxels and multiplied by their respective voxel spacings, the distance measured in millimeters will be the same. Unfortunately, no satisfactory results were obtained using the current framework for mutual information registration in ITK.

4.5.2 The CT and US image modalities

As noted earlier, the CT data is acquired from patients with injected contrast enhancement. In these images, the graft and the lumen appears as a solid region of bright voxels. The aneurysm around it appears as a region of darker voxels. The contrast between the lumen and the thrombus is generally good, while the contrast between the thrombus and surrounding tissues can be much more diffuse.

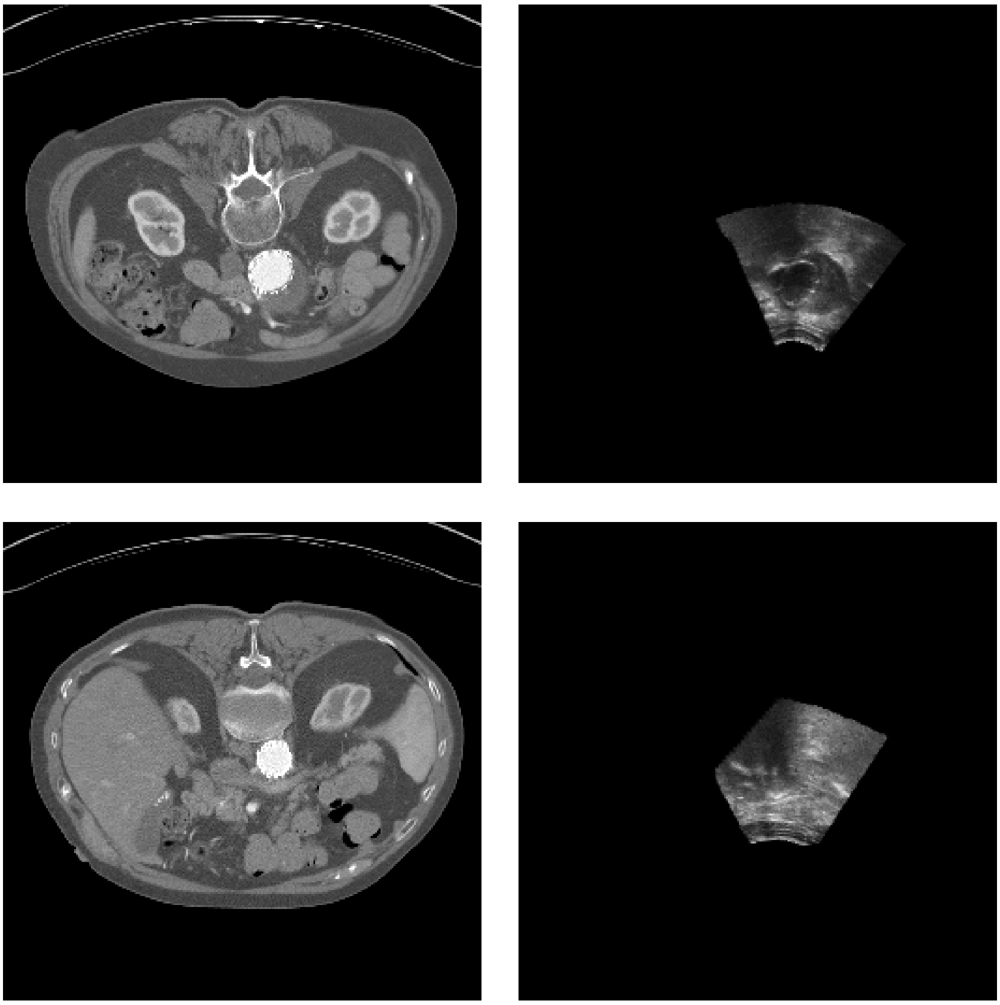


Figure 4.41: CT and US images of similar features in the abdomen prior to manual extraction of the subregions to be registered. The CT and US images are from corresponding data sets. While the lumen and the thrombus show up as solid regions in the CT images, it is the edges of these structures that are the most predominant features in the US images. The lower right US image also illustrates how US data is often very degraded by noise.

In the US images, the AAA and the stent graft are the most predominantly visible structures, and there is hope that these structures can be used as the main feature to register the images. The metal structure in the graft gives a high amplitude echo in the US images, and so does the outer aortic wall. These regions show up as bright voxels, while the space between the graft and the outer aortic wall has darker intensity.

These different characteristics are illustrated in the images in figure 4.41. In general, the intensity maps for CT and US images are very different in that structures that are bright in the CT volume may be less bright or even close to black in the US volume. The application of many other common registration techniques would easily lead to confusion as structure intensities and other image characteristics don't match directly. Thus they would have great difficulties matching the right structures to each other. The appearance of bright edges in the US image could easily be confused with structures that are bright in the CT volume, such as bone tissue, for example.

4.5.3 Registering CT and US images

4.5.3.1 Manual preparations

The available ultrasound test data were already resampled to the equivalent of the CT image resolution. Since the CT image is so big compared to the original US image (see section 4.1), the resampled ultrasound image will contain a majority of black voxels, with a relatively small region of actual ultrasound data inside it.

CT images generally also contain a lot of black voxels, as demonstrated in pictures throughout this text.

If these images are registered the way they are, the large regions of black voxels will influence the mutual information metric in an undesirable way, as they generally do not contain information about the anatomical region of interest. When voxels are sampled, as explained in section 3.6.2.3, there will be a significant number of (black, black) and (black, non-black) pairs which do not contain patient information. (Of course, the effect is negative not because the voxels are black, but because black voxels will mostly be sampled from regions in the images that doesn't contain information relevant for registering the images. Black voxels located within the image regions that actually represent anatomical data are not a problem.)

To exclude the problematic regions of black voxels from the computation, a small subvolume is extracted from both the CT and US images. These two subvolumes are of the same size and contain the CT and US voxels from corresponding voxel indices respectively. The volumes are delimited so that they contain only voxels that represent actual anatomical data obtained from the patient.

4.5.3.2 Full automatization

Since the issue of automatization is important in this report, an explanation is provided on how this procedure is intended to be fully automated, without the need for manual extraction of subvolumes from the CT and US images.

Although the US images available for use in this project were resampled to the same resolution as the CT images, the US images will generally not be resampled this way, and they will contain only voxels from within the patient's abdomen. When the initial transform is executed to initially align the two images before the registration procedure starts, the ultrasound image, which in the un-resampled form is much smaller than the CT image, will be transformed into a region of the CT that contains a majority of relevant voxels. Thus, the irrelevant black pixels initially causing problems here will generally not be an issue. (Naturally, the sample pairs used by the mutual information metric will always have to reside within the region of the smallest of the two images to be registered. The initial registration performed by the positioning system is accurate enough so that, after the initial transformation, the volume alignment will generally be good enough to make the risk of sampling too many irrelevant voxels a non-issue.)

When registering images in ITK, it is usually favorable to take advantage of the fact that the role of the moving image and the fixed image is not symmetric. This means that when the CT is the fixed image and the US is the moving image, the registration process will behave differently compared to when the roles of the two images are interchanged.

As described in section 1.7.1, the registration process basically takes voxels from the fixed image and map their spatial location through the registration transform into the geometric space of the moving image. It is best to use the US as the fixed image since such a setup is more favorable. The reasons for this has to do with the resolution of the image data. The US images typically contain around 10^7 voxels, while the converted CT data contain about 10^8 voxels. As mentioned earlier, both are 8 bit grayscale.

If the CT is used as the fixed image, each evaluation of the metric will need to visit 10^8 voxels. (When using the full-size DICOM CT data, this number will easily grow by an order of magnitude.) Most of the voxels will end up being mapped to the outside of the ultrasound image, thus not at all contributing to the metric, while still using a lot of processing time.

If, on the other hand, the US is used as the fixed image, the computation of the metric will be visiting about 10^7 voxels. Most of them will be mapped to regions of the CT image that contain relevant information, thus they are very likely to contribute to the metric.

(Specifically, when using the mutual information metric this not exactly true, since a precise number of voxels are selected and the metric is computed only for those pixels. For the use of most other metrics however, this is a fact important to be aware of.)

Another advantage of exchanging the role of the images is that the derivative of the metric uses the gradient of the moving image, and it is generally better to use the gradient of the CT image than the gradient of the US image. This is also another reason for using the CT image as the moving image.

Other than this, interchanging the role of the fixed and moving images will simply result either in a final transform T or its inverse T^{-1}.

4.5.4 The registration procedure

The registration framework in ITK is modular, as explained in section 3.4.3. Thus, when composing a registration method, each component is relatively independent of the others. The most critical components though, are the metric and the optimizer.

Before the registration process is started, the images are normalized to have a mean value of 0 and a standard deviation of 1. This is done to get the intensity values in the image on the standard form the mutual information metric works with, and is remniscent of the way we converted gradient images into the potential images required by the level set filter.

The registration process is started by first applying the initial offset transformation, which has been produced by the marker and positioning system and is described in section 4.1.3. This aligns the images as accurately as the system was able to record when the images were acquired. Of course, this initial transform is only applied once, after which the registration method takes over to further refine the image alignment.

All the voxels in the fixed image are then mapped to the moving image using the selected registration transform.

As the voxel values are mapped into the moving image, their spatial position will generally be mapped to non-grid positions. This means that a mapped voxel does not overlap exactly with one of the voxels in the other image. The interpolator then helps determine what the voxel intensity should be.

After this, the metric, that is the mutual information in our case, then evaluates how well features in the two images match each other. This is done by comparing the moving image with the transformed fixed image.

Lastly, the role of the optimizer is to keep changing the parameters of the registration transform, searching for a combination that gives the best value of the metric. Thus, the execution of the registration method is ultimately driven by the optimizer.

```
perform initial transform
for (number of iterations)
{
  transform fixed image to moving image
  interpolate voxels
  evaluate mutual information metric
  evaluate optimizer
}
```

Figure 4.42: Pseudocode for the registration procedure.

A technique known as multiresolution registration was also employed. To save processing time, both images to be registered may be downsampled to a lower resolution, before the registration starts. Registration is performed on the downsampled images for a number of iterations before it continues on the full-resolution images. The technique is generalized so that the original images may be subsampled any number of times, to different resolutions. When the registration is completed on one level, it steps up to the next level and the images are registered again, and so on, until the full-resolution images are registered. This increases the performance considerably compared to working with the full-resolution images all the time.

When multiresolution registration is used, the for-loop in figure 4.42 is executed once for each level, as every single level in the multiresolution framework is a complete registration in its own right.

The ultimate goal of the registration process is to find the set of values for the transformation that optimize the metric. The output of the registration is a transformation matrix which can be used to resample the moving image to align it with the fixed image.

Following the advice of the ITK developers, two different optimizers were tested: the GradientDescent optimizer and the RegularStepGradientDescent optimizer.

4.5.5 Using the GradientDescent optimizer

The GradientDescent optimizer implements one of the many variants of gradient descent. The central parameter in this optimizer is the learning rate. The learning rate is one of the hardest parameters to adapt in the entire process. It is used by the Gradient descent in order to compute how long each step should be in the parameter space of the optimizer.

When we perform rigid registration in 3D, the parameter space is six-dimensional, consisting of a translation (x, y, z) and a rotation (γ, θ, ϕ).

The optimizer starts with a particular combination of $(x, y, z, \gamma, \theta, \phi)$. It evaluates the metric for these values and the derivative of the metric with respect to x, y, z and γ, θ, ϕ. Using this derivative, the optimizer obtains a gradient direction in this 6D parameter space $(x, y, z, \gamma, \theta, \phi)$. The metric may then be improved by moving in the direction of this gradient.

To find out how far to move, the GradientDescent optimizer computes the length of this movement by multiplying the metric-derivative vector by the learning rate. Naturally, the larger the learning rate, the longer the step will be.

Initially, the easiest thing to do is to use relatively small values for the learning rate. The size will depend on the values of the metric derivative because what matters in the end is the product of the metric derivative and the learning rate. To find good values for the learning rate, the following procedure can be used:

- First, an arbitrary value is selected, for example in the order of 10^{-3}.
- A registration is performed and the values of the transform from one iteration to the next is tracked.
- If the distance moved from one iteration to the next is too big, the learning rate is reduced.
- If, on the other hand, the distance moved is too low, the learning rate is increased.

A movement in the range of 0.1 to 10 voxels per iteration is usually preferred, but this is dependent on the application and on the current degree of image alignment.

The value of the learning rate depends on the value of the metric derivative between the two images. The metric derivative can only be found through testing and depends on the nature and content of the images, as well as the type of transform used. Thus, the only way to find a good value is through experiments.

The learning rate of the GradientDescent optimizer has the serious drawback that the length of the step is proportional to the derivative of the metric, and it is pretty hard to get an idea of how this derivative will change as the registration evolves. For this reason, even if a good learning rate for the initial steps of the registration is selected, the value may become inappropriate at any moment. In the same way, low gradient regions in the metric can easily make the step size too big. The advantage of this optimizer is that it is fast, and it is useful to experiment to find good initial values for some of the registration parameters before switching to a different and more robust optimizer.

4.5.6 Using the RegularStepGradientDescent optimizer

The RegularStepGradientDescent optimizer also uses the metric derivative to determine in which direction to move. However, this optimizer applies a different strategy for computing the step size. The regularStepGradientDescent uses an independent, user provided, initial value for the step.

This optimizer computes the derivative of the metric with respect to $(x, y, z, \gamma, \theta, \phi)$ and normalizes it, so that only the direction of the derivative is used, and not its magnitude. This normal vector is then multiplied by the initial step value provided by the user. This way the first step will always be what the user specified amount of voxels, completely independent of the value of the metric derivative.

The optimizer will continue using this step size until it advances one step without any improvement of the metric value. At this point, the optimizer steps back, and the current step size is multiplied by -0.5, so that its direction is reversed and the step size is halved. It then proceeds as before, again advancing until the metric stops improving.

Every time that the direction changes more than 90 degrees, the step length is divided by two. A good initial value for rigid transforms is 0.1, which is equivalent to a moderate rotation of about 5 degrees.

The algorithm stops when the step size reaches a minimum value, also provided by the user.

This is generally a more robust method than the GradientDescent optimizer, with the drawback that it is much slower.

4.5.7 Parameter selection

After selecting a metric and an optimizer, the parameters for the chosen methods that produce the best results must be determined. In general, the most difficult task in registration is to find a good combination of parameters for the problem. The challenge lies in finding the parameters that fine-tune a particular approach for a particular application.

Through correspondence with the ITK developers, it was learned that the selection of these parameters is a huge and very lengthy process of trial and error. As of today, there are no documented rules or directions for tuning the parameters for the use of mutual information. According to ITK developers, it appears that in the past, the importance of documenting the process of parameter tuning for this technique has passed relatively unnoticed. As a consequence, the default parameters for mutual information provided in ITK are mostly those found through initial testing and experiments performed during development.

Mutual information is estimated using two sample sets: one to calculate the singular and joint probability density functions and one to calculate the entropy integral. By default 50 samples points are used in each set in ITK. In our tests, values between 50 and 80 were used, according to the recommendations of the developers. Using more sample points would make the process extremely slow, running for hours.

A gaussian kernel was used in the density estimation in the mutual information. The quality of the density estimate depends on the choice of the kernel's variance. The optimal choice will depend on the images in question. It can be shown that around the optimal variance, the mutual information estimate is relatively insensitive to small changes of the variance. Experiments performed by the ITK developers have shown that a variance of around 0.4 works well for images normalized to have a mean of zero and a standard deviation of 1.0. Variances in the range 0.3 to 0.5 were tested.

Selecting the right parameters for the optimizer is critical to reach a good result. According to the ITK developers, the only way to find what values give the best results is to experiment, using different values for the learning rate in the case of the GradientDescent, and the initial step value in the case of RegularStepGradientDescent. The step size and learning rate values used in our tests were both in the order of 10^{-6} to 10^{1}.

The number of iterations in the registration can be set to any amount. A number of iterations in the orders of 10^{3} to 10^{4} were tested, and using a number smaller than in the order of 10^{3} was not recommended, as the images would not have time to align.

The number of levels for multiresolution registration can also be varied. The number of levels used in our tests were between 1 and 4.

A parameter value is also used to scale between the translation and rotation parameters. This parameter is dependent on the images, and has to do with how much the edges of an image will move as a result of the image being rotated about its center. It was recommended that this value be set to approximately the voxel spacing in millimeters, and values of this order of the voxel spacings were tested.

4.5.8 The problems

Despite collaboration with the supervisors at Sintef Unimed and extensive correspondence with the developers of ITK to find a good set of parameters, it has so far been found extremely difficult to register US to CT using the registration methods currently available in ITK.

Using the GradientDescent optimizer with small learning rates of 10^{-6} to 10^{-5}, the registration will move arbitrarily. It never reaches the correct position, and the steps are very small, in the range 10^{-2} to 10^{-1}. The result is fluctuations around the starting point in a random way. Using multiresolution registration, or changing the other parameters did not produce any observable improvements in the outcome. When the learning rate is increased up to a certain threshold (about $5 \cdot 10^{-4}$ to $1 \cdot 10^{-3}$), the step size becomes too big, and the algorithm will consistently move the two images apart until their respective spatial regions are completely disjoint. The values of the transform were checked after each iteration, but there was generally a consistent lack of any pattern to the movement and direction of the image.

Tests performed with the RegularStepGradientDescent optimizer showed that, regardless of the initial learning rate used, the algorithm would consistently move the two images apart until their respective spatial regions were completely disjoint. Again, applying the multiresolution technique and considerably changing the set of parameters did not produce better results or observable improvements.

4.5.9 Attempted corrections

Initially, the CT and US images to be registered were not preprocessed in any way before the registration process started. Using a gaussian to blur the CT and the US images was tested, using variances between 0.6 and 1.0.

Smoothing either or both images before registering was also tried, but did not lead to observable improvements in the results. Attempts at registering the gradient map of gaussian smoothed CT rather than the original CT were also made. Unfortunately, none of these changes led to any observable improvement in results.

5 Discussion and conclusions

In this chapter, we discuss our findings and results on the use of ITK for medical segmentation and registration and sum up our work. A conclusive evaluation of ITK is also provided.

5.1 Segmentation

The segmentation scheme presented in this report performs automatic segmentation of both the inner and outer aortic wall in postoperative CT images of patients with AAA, using 2D and 3D level sets, requiring the manual initialization of only four values. The spatial extent of the segmentation includes a region from below the renal arteries to the top of the iliac arteries, including the aortic bifurcation point in the lower abdomen.

The inner aortic wall has been segmented using the level set method in 3D. The average segmentation error in this case is 0.63mm, which implies that sub-voxel accuracy has been achieved. The outer aortic wall has been segmented using the level set method in both 3D and 2D. In this case, the average errors measured were 1.61mm and 1.59mm respectively. This is an error margin of slightly more than one voxel. The segmentation error for the inner aortic wall is generally smaller than that of the outer aortic wall, and this can mainly be attributed to the much lower visibility of the outer aortic wall in the CT images.

Although schemes to segment AAA using level sets are also presented in [LONCA-01] and [MAGEE-00], these do not focus on, or detail, how to automate the presented methods. The deformable model scheme for semi-automatic segmentation presented in [BRUIN-02], produces a segmentation with a smaller error, compared to the one presented in this text, but also requires extensive manual initialization and user intervention. It also avoids the challenges associated with the aortic bifurcation by segmenting the dilated aortic tissue only. Further searches indicate that an automatic scheme for segmentation AAA such as the one documented in this report, has not previously been presented.

More testing needs to be performed to determine whether it is best to use level sets in 2D or 3D for segmentation of the outer aortic wall. The current results suggest that the error is very similar for both. However, it is easier to impose varying structure constraints when segmenting slice-by-slice, as conditions may be imposed on each individual slice when segmenting in 2D. This is in contrast to the 3D segmentation where the whole volume is handled as one single structure.

Using the current framework for level set segmentation in ITK and the available test data, it is likely that the achieved results are close to as good as one can realistically expect. As ITK is further developed, there are good prospects for improvement of the scheme in the future.

5.2 Registration

A scheme to perform automatic registration of CT and US images of AAA using mutual information has been discussed. The tests performed indicate that an approach using the framework currently available in ITK, may not by be sufficient. Despite intense collaboration at Sintef Unimed and extensive correspondence with the developers of ITK, no set of parameters providing satisfactory registration of the CT and US data was achieved.

While the process of determining the right parameters for the registration is indeed a huge process of trial and error, the fact that no improvement in results was observed, despite using parameter values as recommended by the ITK developers, is alarming. Between the extreme cases of the registration fluctuating randomly about its initial state and the two volumes moving completely apart, it would be reasonable to expect to see the registration moving in a way more or less consistent with our intentions. This was not found to be the case and was never observed. These results may be due to several factors.

The quality of the ultrasound data may be too poor, with too much noise, so that there is simply not enough mutual information in the sample pairs from the CT and US images for the mutual information algorithm to actually register them. Similarly, the converted CT data has been downsampled and contain only 25 % of the original information. Performing registration using the original CT data rather than the converted data would also be interesting. Using the original image data of higher resolution and without intensity artifacts may possibly produce better results. Although mutual information is basically immune to the differences in graylevels due to the different image modalities involved, it cannot do much if the images in question don't actually have enough potential matching regions.

Exploring other algorithms, such as the one described in [ROCHE-01], may be a worthwhile option over the risk of spending months attempting to find working parameters for a registration algorithm that may or may not be appropriate for the problem. At this time though, this method is not available in ITK.

The mutual information method for registration is the most promising method the ITK toolkit has for registration of CT and US images at the time of this work. Still, it may be that a different algorithm is needed to achieve better results. Although it may be desirable to perform further testing to confirm or deny the results of the work presented on registration of CT and US images in this text, it is likely that the methods of the existing registration framework available in ITK is insufficient for registering the image data used in this project.

5.3 Conclusive evaluation of ITK

As mentioned in section 3.4.6, the development of this project is for the most part based on intermittent versions of ITK released after the first beta. The first beta became available in February 2002, shortly after this project was begun, and the second beta was ready in June. Although it was initially seen as desirable to stay with the beta version of ITK to ensure full compatibility between all the pieces of software written for this project, the increasing requirements for more complex functionality as development went on, intermediate development versions of the toolkit had to be used instead. In practice, frequent upgrades were necessary to get access to new and essential functionality.

Although the first beta versions of ITK have been released, the software remains in very active development, and the first fully developed version is still months away. The sheer increase in the amount of source code well illustrates the pace of development of the toolkit. Over the span of four months, from March to June, the code base of ITK grew from 21MB to 32MB, an increase of more than 50 %. It is not always easy to keep up with the latest development of such a large software package, and the constant process of development, redevelopment and modification of existing functionality may sometimes lead to undesirable situations for the end user.

Since of the code in ITK is still relatively new, the coding style isn't always consistent, which means that it may be necessary to perform the same type of function calls in slightly different ways, depending on what filter or functionality is used.

This can make the API confusing, especially when the API is the only documentation available much of the time. Also, new functionality is added and changed on a regular basis, which can make software development difficult. The typical scenario encountered is the one where new features are added to the software, which the at some point is found to be necessary to solve the problem the user is working on. When an upgrade to the more recent version of the software is made, the new version turns out to contain other changes as well, requiring further modification of the user's own code.

The whole level set code framework in ITK has also been under redevelopment for the full length of this work, and the current level set implementation, used in this project, will eventually be removed from the codebase at some point in the future.

The lack of tutorials, books and documentation in general makes ITK difficult to learn and use. Thus, there was quite some time of learning before useful code was produced for this project. Also, the lack of a DICOM reader and a general lack of robust IO-modules at the time the development of this project started, meant that essential functionality to get data in and out of ITK had to be implemented before the relevant tasks could be considered. Time-consuming reading of source code to understand how to implement basic and absolutely necessary IO-functionality, with the API as the only documentation available, is not the kind of situation a user wants to be in when everything is new and time is the most scarce and valuable resource of all.

Despite these problems and drawbacks, it should be kept in mind that most of them are caused by the fact that ITK is still in a stage of relatively early development, and thus cannot by any means be expected to be fully functional for all intended purposes yet. Thus, for most medical imaging communities, it is a little too early to make full use of the toolkit today, but in the longer run, ITK has great potential. In time, it is quite possible that ITK will achieve a similar status in medical imaging communities as VTK (the visualization toolkit) enjoys in visualization communities today.

5.4 Conclusion

The starting point of this project was to perform segmentation and registration of abdominal aortic aneurysm as seen in CT and US images, using the existing framework provided by ITK. The achieved results of the segmentation indicate that ITK is a medical imaging tool with great potential. The achieved results of the registration indicate that it is a bit too early to make full use of the software in clinical applications. The current limitations of the ITK framework are thought to have been met for both of our specific problems, and thus the goals of this project have been achieved.

6 Future work

In this chapter a number of suggestions are made for possible improvements of the segmentation and registration. Some of the future improvements planned for ITK by the ITK development team are also mentioned.

6.1 Improving the segmentation scheme

Several improvements can be made to the segmentation scheme, some of which require only more time to implement, while others depend on the further development of ITK.

The robustness of the method may be improved by further testing it on more data sets and adjusting the various parameters to better fit a wider variety of input.

One of the most interesting improvements would be further improvement of the automatization to require fewer initialization values. One possibility would be to use some means of pattern recognition analysis (such as a neural network) to find a suitable seed point for the lumen segmentation. An initial analysis with the purpose of detecting characteristic regions could possibly eliminate the need to manually initialize the seed point to initiate the segmentation. The second seed point may be eliminated if some way of creating a gradient image for the thrombus can be found without relying on grayvalue masking. With such improvements, the new scheme would come very close to complete automatization.

It is also desirable to improve the accuracy of the method when segmenting data sets with severe lack of boundary information. The promising active shape model-based method presented for thrombus segmentation in [BRUIN-02], turns out to be robust in terms of improving accuracy with regard to this issue, but also has the formerly mentioned problems of the deformable model-based methods. Adding a deformable model-based method as a final refinement step to improve on the level set segmentation might still be an interesting possibility. This would also be a way of implementing a more prominent use of a priori knowledge of the shape of the (diseased) aortic structure. Implementing a method for inter-slice correalation when segmenting in 2D would also be very useful to improve the accuracy.

6.2 Improving the registration

According to the ITK developers, recent papers have proposed variations of mutual information in which gradient information is added. Such methods are not yet available in ITK, although they might well be included in the future, as published results using these methods seem to be positive so far. Especially, a much more efficient procedure for finding the optimal parameters for mutual information registration is desirable.

6.3 Improving ITK

Although improving ITK is not a part of this project, the following overview of things to come is provided for completeness. The ITK development team has reported to be aware of the following issues to be addressed in the near or foreseeable future:

- Better documentation and more examples will be added.
- A tutorial on ITK will be presented at the IEEE Visualization 2002 conference.
- The need for a book to provide organized and complete information on the toolkit has been acknowledged.
- The need for more robust and uniform code has been identified and is being worked on.
- The IO modules will be improved and DICOM support will be added.
- Wrapping for more scripting languages will be added.

Invitations are generally open for anyone to contribute new algorithms and functionality to the toolkit.

6.4 Further development

In the bigger picture, the work presented in this report is a step on the way to implement the following scheme:

1. Segment the CT image.
2. Register the CT and US images.
3. Segment the US image using deformable registration.

Exploiting multimodal image analysis and using both ultrasound and CT diagnostics rather than CT alone can pave the way for reduced time and costs in the treatment of diseases in general, not just AAA. The hope is that, by combining post-surgical CT image data with more recently gathered ultrasound images, condition assessment can be done by representing what is already known along with more recently acquired information in a single model, thus providing better information and a better overall picture. If this can be done, costs can be greatly reduced by relying more heavily on ultrasound equipment which is considerably cheaper, and both easier and safer to use than CT.

A Tables and charts

A.1 Error measurements of the 3D lumen segmentation

The following three tables give more detailed information about the measurements of the segmentation error. The d-values have been obtained as described in section 4.4.7.5.

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	0	0	1	0					1	0.2500	0.3467
2	1	0	0	0					1	0.2500	0.3467
3	0	0	1	1					2	0.5000	0.6934
4	0	0	1	1					2	0.5000	0.6934
5	1	1	0	1	1	2	0	1	7	0.8750	1.2134
All										0.4750	0.6587

Table A.1: Dataset 1 (AAA43).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	1	0	0	0					1	0.2500	0.3213
2	0	0	1	0					1	0.2500	0.3213
3	0	0	1	0					1	0.2500	0.3213
4	0	0	1	1					2	0.5000	0.6426
5	0	0	0	2	0	2	1	0	5	0.6250	0.8033
All										0.3750	0.4820

Table A.2: Dataset 2 (AAA52).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	0	1	0	1					2	0.5000	0.7520
2	0	0	0	0					0	0.0000	0.0000
3	0	1	1	1					3	0.7500	1.1279
4	0	1	1	0					2	0.5000	0.7520
5	1	1	0	1	1	1	0	1	6	0.7500	1.1279
All										0.5000	0.7520

Table A.3: Dataset 3 (AAA59).

A.2 Evolution of the lumen 3D segmentation

The three following tables include more detailed output of the 3D level set segmentation of the lumen of the three test datasets. The first column in the tables specifies the number of iterations performed. The second specifies the total number of voxels in the segmented region at the time of the previous check of the stopping criterion. The third column specifies the number of voxels in the segmented region at the following check of the stopping criterion (after 10 iterations). Finally, the fourth column specifies the ratio of the pixel count in the current region to the previous one. In the test setup, the initial level set was a $3 \times 3 \times 3$ cube, and thus the size of the first initial volume is always 27 (the first value in the second column). The iteration was halted when the ratio was equal to, or greater than, 0.998, as can be seen from the final ratio value in each table.

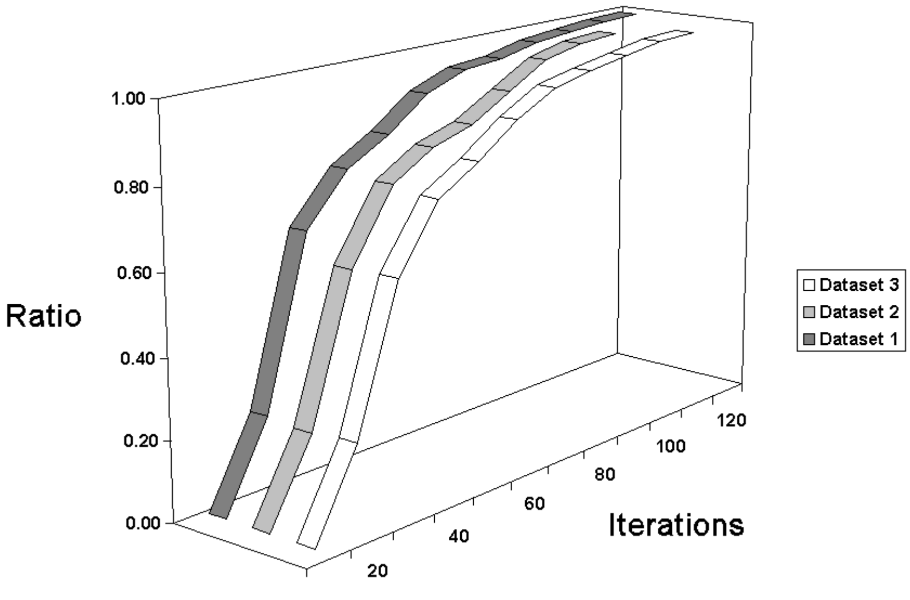


Figure A.1: The chart shows the growth measure ratio for each check of the stopping criterion, performed every 10 iterations. Notice how it converges almost asymptotically to 1.

iteration	initial	current	ratio
10	27	2023	0.013347
20	2023	8934	0.226438
30	8934	13855	0.644821
40	13855	18041	0.767973
50	18041	21733	0.830120
60	21733	23953	0.907318
70	23953	25296	0.946909
80	25296	26507	0.954314
90	26507	27128	0.977109
100	27128	27465	0.987730
110	27465	27566	0.996336
120	27566	27619	0.998081

Table A.4: Dataset 1 (AAA43).

iteration	previous	current	ratio
10	27	1980	0.013636
20	1980	9171	0.215898
30	9171	15926	0.575851
40	15926	21140	0.753359
50	21140	25877	0.816942
60	25877	30478	0.849039
70	30478	33512	0.909465
80	33512	34607	0.968359
90	34607	34920	0.991037
100	34920	34988	0.998056

Table A.5: Dataset 2 (AAA52).

iteration	previous	current	ratio
10	27	1984	0.013609
20	1984	8836	0.224536
30	8836	15190	0.581698
40	15190	20528	0.739965
50	20528	25584	0.802376
60	25584	29055	0.880537
70	29055	31024	0.936533
80	31024	32392	0.957767
90	32392	33204	0.975545
100	33204	33405	0.993983
110	33405	33466	0.998177

Table A.6: Dataset 3 (AAA59).

A.3 Error measurements of the 3D thrombus segmentation

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	4	1	1	3					9	2.2500	3.1201
2	0	1	1	0					2	0.5000	0.6934
3	1	0	1	0					2	0.5000	0.6934
4	1	2	1	1					5	1.2500	1.7334
5	1	2	1	2	2	2	0	1	11	1.3750	1.9067
All										1.1750	1.6294

Table A.7: Dataset 1 (AAA43).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	1	1	1	2					5	1.2500	1.6065
2	1	0	1	1					3	0.7500	0.9639
3	4	0	0	0					4	1.0000	1.2852
4	1	1	2	1					5	1.2500	1.6065
5	2	2	0	3	3	2	0	1	13	1.6250	2.0885
All										1.1750	1.5101

Table A.8: Dataset 2 (AAA52).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	0	1	2	0					3	0.7500	1.1279
2	2	1	0	1					4	1.0000	1.5039
3	1	0	3	2					6	1.5000	2.2559
4	1	1	1	1					4	1.0000	1.5039
5	2	1	1	1	2	1	1	2	11	1.3750	2.0679
All										1.1250	1.6919

Table A.9: Dataset 3 (AAA59).

A.4 Evolution of the thrombus 3D segmentation

The 3D segmentation of the thrombus proceeds in the same manner as for the lumen. The only differences are the initial level set, which in this case is the final preprocessed result of the lumen segmentation, and the segmentation parameters.

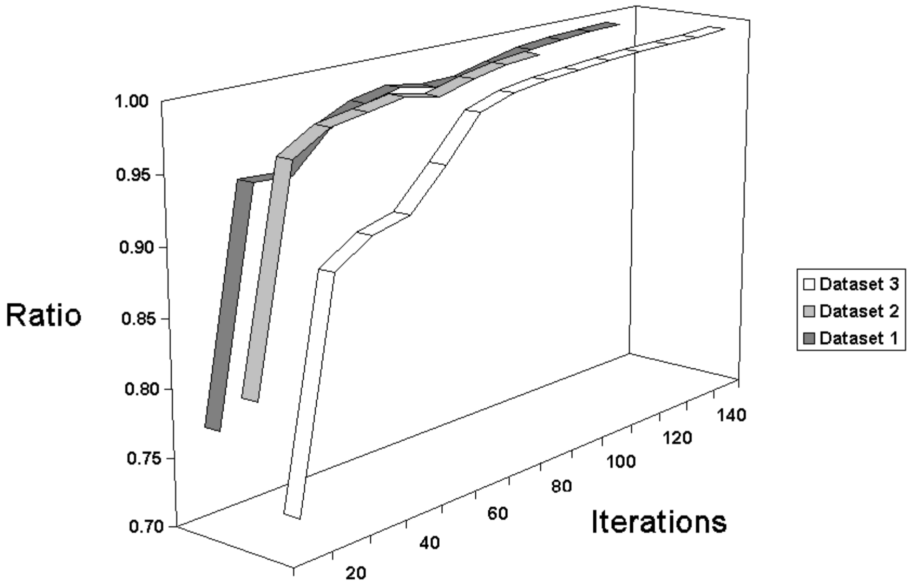


Figure A.2: The chart shows the growth measure ratio for each check of the stopping criterion, performed every 10 iterations.

iteration	previous	current	ratio
10	26466	34342	0.770660
20	34342	36583	0.938742
30	36583	39053	0.936753
40	39053	40546	0.963178
50	40546	41535	0.976189
60	41535	42297	0.981985
70	42297	43304	0.976746
80	43304	44301	0.977495
90	44301	45030	0.983811
100	45030	45516	0.989322
110	45516	45868	0.992326
120	45868	46160	0.993674
130	46160	46420	0.994399

Table A.10: Dataset 1 (AAA43).

iteration	previous	current	ratio
10	33632	42036	0.800076
20	42036	43798	0.959770
30	43798	44880	0.975891
40	44880	45765	0.980662
50	45765	46435	0.985571
60	46435	47380	0.980055
70	47380	47997	0.987145
80	47997	48363	0.992432
90	48363	48606	0.995001

Table A.11: Dataset 2 (AAA52).

iteration	previous	current	ratio
10	32185	44353	0.725656
20	44353	49901	0.888820
30	49901	55050	0.906467
40	55050	60251	0.913678
50	60251	63967	0.941908
60	63967	65708	0.973504
70	65708	66930	0.981742
80	66930	67881	0.985990
90	67881	68729	0.987662
100	68729	69426	0.989961
110	69426	70051	0.991078
120	70051	70608	0.992111
130	70608	71160	0.992243
140	71160	71562	0.994382

Table A.12: Dataset 3 (AAA59).

A.5 Error measurements of the 2D thrombus segmentation

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	3	2	1	4					10	2.5000	3.4668
2	0	1	1	1					3	0.7500	1.0400
3	0	1	0	1					2	0.5000	0.6934
4	1	3	1	0					5	1.2500	1.7334
5	1	2	0	2	2	2	1	1	11	1.3750	1.9067
All										1.2750	1.7680

Table A.13: Dataset 1 (AAA43).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	1	1	1	3					6	1.5000	1.9278
2	2	0	0	1					3	1.7500	0.9639
3	2	0	0	2					4	0.0000	1.2852
4	1	1	1	2					5	1.2500	1.6065
5	3	2	0	2	3	2	0	2	14	1.7500	2.2491
All										1.2500	1.6065

Table A.14: Dataset 2 (AAA52).

slice	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	sum	avg(vox)	avg(mm)
1	0	1	2	0					3	0.7500	1.1279
2	2	0	0	2					4	1.0000	1.5039
3	1	0	2	3					6	1.5000	2.2559
4	1	0	0	1					2	0.5000	0.7520
5	2	1	0	1	2	0	0	1	7	0.8750	1.3159
All										0.9250	1.3911

Table A.15: Dataset 3 (AAA59).

A.6 Evolution of the thrombus 2D segmentation

The three following tables include more detailed output of the 2D level set segmentation of the thrombus of the three test datasets. The first column specifies the slice number of the delimited subvolume. The second specifies the total number of voxels in the segmented region at the time of the second-to-last check of the stopping criterion. The third column specifies the number of voxels in the segmented region when the stopping criterion was met. The fourth column specifies the ratio of the pixel count in the current region to the previous one, and the fifth specifies the total number of iterations necessary to reach convergence. When segmenting the lumen in 2D, the iteration was halted when the ratio was equal to, or greater than, 0.995. At the end of the table is specified the average values for each of the four data columns.

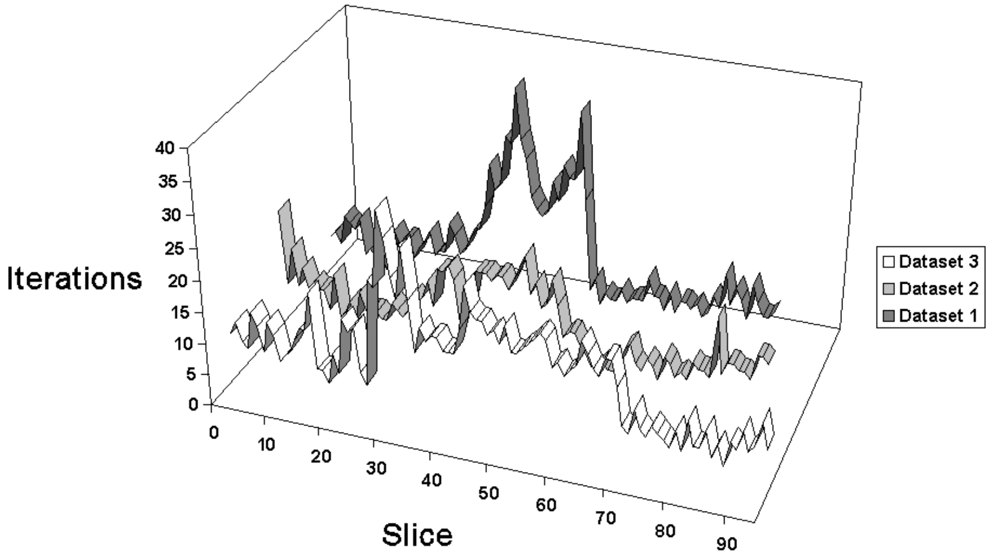


Figure A.3: The chart shows the total number of iterations per slice in each of the three data sets for the 2D thrombus segmentation.

Table A.16: Dataset 1 (AAA43).

slice	previous	current	ratio	iterations
1	431	432	0.997685	60
2	410	412	0.995146	70
3	415	417	0.995204	50
4	427	429	0.995338	100
5	411	413	0.995157	90
6	369	369	1.000000	80
7	348	348	1.000000	100
8	292	293	0.996587	40
9	286	287	0.996516	50
10	279	280	0.996429	40
11	270	271	0.996310	50
12	247	248	0.995968	20
13	273	274	0.996350	50
14	295	295	1.000000	90
15	298	298	1.000000	70
16	297	297	1.000000	80
17	308	309	0.996764	80
18	298	299	0.996656	50
19	309	310	0.996774	70
20	317	317	1.000000	80
21	334	335	0.997015	100
22	336	337	0.997033	60
23	349	350	0.997143	70
24	364	364	1.000000	70
25	411	413	0.995157	120
26	428	429	0.997669	100
27	450	451	0.997783	70
28	475	477	0.995807	80
29	508	510	0.996078	100
30	573	575	0.996522	120
31	619	622	0.995177	120
32	667	670	0.995522	160
33	760	762	0.997375	230
34	810	814	0.995086	190
35	850	854	0.995316	200
36	1042	1046	0.996176	280
37	1122	1127	0.995563	270
38	1243	1249	0.995196	360

slice	previous	current	ratio	iterations
39	1162	1165	0.997425	280
40	1180	1184	0.996622	250
41	1188	1191	0.997481	190
42	1191	1195	0.996653	170
43	1186	1189	0.997477	160
44	1194	1199	0.995830	170
45	1259	1264	0.996044	220
46	1205	1211	0.995045	190
47	1179	1181	0.998307	250
48	1023	1028	0.995136	230
49	891	895	0.995531	230
50	868	871	0.996556	340
51	655	655	1.000000	200
52	540	542	0.996310	50
53	550	552	0.996377	70
54	516	518	0.996139	30
55	507	508	0.998031	50
56	483	484	0.997934	40
57	471	471	1.000000	40
58	456	458	0.995633	60
59	433	435	0.995402	40
60	420	421	0.997625	50
61	413	415	0.995181	50
62	402	404	0.995050	50
63	411	412	0.997573	90
64	393	394	0.997462	60
65	373	374	0.997326	60
66	360	361	0.997230	40
67	379	379	1.000000	70
68	379	379	1.000000	40
69	401	403	0.995037	60
70	425	425	1.000000	60
71	426	427	0.997658	50
72	410	412	0.995146	60
73	401	403	0.995037	40
74	411	412	0.997573	50
75	403	404	0.997525	80
76	402	403	0.997519	60
77	433	434	0.997696	120
78	427	429	0.995338	70

slice	previous	current	ratio	iterations
79	444	445	0.997753	100
80	428	430	0.995349	80
81	421	423	0.995272	50
82	433	434	0.997696	110
83	409	411	0.995134	80
84	405	407	0.995086	60
85	400	402	0.995025	60
86	408	408	1.000000	70
Avg	547.38	549.19	0.996904	105.23

Table A.17: Dataset 2 (AAA52).

slice	previous	current	ratio	iterations
1	306	307	0.996743	190
2	258	259	0.996139	70
3	344	344	1.000000	130
4	408	409	0.997555	70
5	491	492	0.997967	100
6	517	519	0.996146	70
7	528	529	0.998110	80
8	515	517	0.996132	70
9	506	508	0.996063	70
10	442	444	0.995495	40
11	403	405	0.995062	100
12	357	358	0.997207	30
13	339	340	0.997059	40
14	331	332	0.996988	60
15	315	316	0.996835	30
16	334	335	0.997015	80
17	345	346	0.997110	50
18	343	343	1.000000	50
19	341	341	1.000000	50
20	336	337	0.997033	40
21	342	343	0.997085	40
22	350	351	0.997151	70
23	347	348	0.997126	60
24	346	347	0.997118	50
25	354	354	1.000000	60
26	376	376	1.000000	90
27	362	363	0.997245	80
28	385	385	1.000000	90
29	414	416	0.995192	120
30	417	419	0.995227	80
31	481	481	1.000000	140
32	525	527	0.996205	140
33	586	588	0.996599	150
34	610	612	0.996732	130
35	579	581	0.996558	60
36	643	646	0.995356	80
37	703	706	0.995751	100
38	788	789	0.998733	150

slice	previous	current	ratio	iterations
39	825	827	0.997582	150
40	875	878	0.996583	150
41	913	917	0.995638	140
42	939	942	0.996815	150
43	969	973	0.995889	150
44	965	969	0.995872	130
45	980	984	0.995935	150
46	965	968	0.996901	170
47	938	942	0.995754	190
48	842	846	0.995272	130
49	849	850	0.998824	150
50	826	830	0.995181	110
51	806	810	0.995062	120
52	778	781	0.996159	160
53	749	750	0.998667	120
54	728	729	0.998628	70
55	717	719	0.997218	80
56	697	698	0.998567	80
57	661	664	0.995482	60
58	605	607	0.996705	50
59	569	571	0.996497	30
60	548	550	0.996364	30
61	536	537	0.998138	40
62	510	511	0.998043	30
63	502	503	0.998012	30
64	494	496	0.995968	40
65	495	497	0.995976	70
66	485	487	0.995893	80
67	455	457	0.995624	50
68	447	447	1.000000	40
69	443	444	0.997748	60
70	449	450	0.997778	50
71	451	453	0.995585	30
72	487	488	0.997951	70
73	498	500	0.996000	60
74	478	480	0.995833	40
75	466	468	0.995726	50
76	464	466	0.995708	40
77	478	479	0.997912	70
78	481	481	1.000000	60

slice	previous	current	ratio	iterations
79	484	486	0.995885	60
80	486	487	0.997947	50
81	511	512	0.998047	150
82	476	478	0.995816	60
83	469	470	0.997872	80
84	471	472	0.997881	80
85	462	464	0.995690	70
86	459	461	0.995662	70
87	461	463	0.995680	60
88	483	485	0.995876	100
89	481	482	0.997925	100
90	475	477	0.995807	90
Avg	538.31	539.99	0.997004	84.00

Table A.18: Dataset 3 (AAA59).

slice	previous	current	ratio	iterations
1	234	235	0.995745	80
2	242	243	0.995885	100
3	248	248	1.000000	70
4	265	266	0.996241	60
5	344	344	1.000000	120
6	438	439	0.997722	100
7	473	475	0.995789	60
8	527	528	0.998106	110
9	503	504	0.998016	90
10	475	476	0.997899	50
11	454	456	0.995614	60
12	426	427	0.997658	70
13	380	381	0.997375	90
14	349	350	0.997143	110
15	321	322	0.996894	110
16	346	346	1.000000	170
17	285	286	0.996503	50
18	293	294	0.996599	50
19	282	283	0.996466	30
20	306	307	0.996743	80
21	287	288	0.996528	50
22	331	332	0.996988	140
23	308	309	0.996764	120
24	298	299	0.996656	130
25	261	262	0.996183	60
26	260	261	0.996169	40
27	367	368	0.997283	210
28	421	423	0.995272	200
29	617	620	0.995161	320
30	606	607	0.998353	280
31	619	622	0.995177	220
32	737	739	0.997294	190
33	890	892	0.997758	270
34	883	887	0.995490	190
35	863	866	0.996536	110
36	964	966	0.997930	150
37	1021	1025	0.996098	130
38	1084	1089	0.995409	140

slice	previous	current	ratio	iterations
39	1125	1128	0.997340	140
40	1178	1183	0.995773	120
41	1234	1239	0.995964	120
42	1295	1300	0.996154	120
43	1363	1368	0.996345	160
44	1535	1541	0.996106	230
45	1567	1573	0.996186	190
46	1525	1529	0.997384	180
47	1572	1578	0.996198	180
48	1550	1557	0.995504	160
49	1565	1572	0.995547	170
50	1544	1551	0.995487	150
51	1584	1588	0.997481	180
52	1539	1545	0.996117	140
53	1528	1534	0.996089	140
54	1547	1552	0.996778	150
55	1546	1552	0.996134	160
56	1530	1535	0.996743	170
57	1500	1505	0.996678	180
58	1349	1352	0.997781	170
59	1280	1283	0.997662	130
60	1259	1260	0.999206	140
61	1178	1182	0.996616	120
62	1127	1131	0.996463	130
63	1038	1041	0.997118	140
64	948	949	0.998946	180
65	857	861	0.995354	160
66	824	827	0.996372	140
67	781	784	0.996173	130
68	730	733	0.995907	160
69	669	672	0.995536	160
70	595	596	0.998322	120
71	561	562	0.998221	60
72	561	561	1.000000	50
73	579	579	1.000000	90
74	574	575	0.998261	60
75	572	572	1.000000	50
76	577	577	1.000000	70
77	574	576	0.996528	60
78	559	561	0.996435	50

slice	previous	current	ratio	iterations
79	526	528	0.996212	40
80	515	516	0.998062	70
81	496	497	0.997988	40
82	492	494	0.995951	90
83	471	473	0.995772	50
84	452	454	0.995595	50
85	458	459	0.997821	80
86	429	431	0.995360	40
87	439	440	0.997727	60
88	426	427	0.997658	30
89	431	432	0.997685	70
90	415	417	0.995204	60
91	429	431	0.995360	90
92	401	403	0.995037	60
93	402	404	0.995050	70
94	403	404	0.997525	110
95	387	388	0.997423	70
Avg	743.15	745.55	0.996903	116.84

B Glossary

AAA Abdominal Aortic Aneurysm.

aneurysm An abnormal blood-filled dilatation of a blood vessel, and especially an artery, resulting from disease of the vessel wall.

bifurcation point In this text, this term generally denotes the point in the lower abdomen where the aorta splits into the iliac arteries leading blood to the legs.

CAT Computer Assisted Tomography

CT Computer Tomography

DICOM Digital Imaging and COmmunications in Medicine

ITK The Insight segmentation and registration ToolKit

image registration The process of subdividing an image into its constituent parts or objects.

image segmentation The process of bringing the involved pictures into spatial alignment.

lumen In this text, the inner aortic wall is often referred to as the lumen for the sake of simplicity.

MR Magnetic Resonance

multimodal imaging Combining, in some way, two or more techniques for image acquisition

PET Photon Emission Tomography

thrombus In this text, the outer aortic wall is often referred to as the thrombus for the sake of simplicity.

US UltraSound

VTK The Visualization ToolKit

voxel The three-dimensional equivalent of a pixel

Bibliography

[BELKI-94] M. Belkin, M. C. Donaldson and A. D. Whittermore, ***Abdominal aortic aneurysms***, Current Opt. Capitol., (1994), pp. 581–590.

[BINAR-02] ***BinaryMedianImageFilter***, 2002. <URL:http://www.itk.org/Doxygen/html/ classitk_1_1BinaryMedianImageFilter.html>.

[BLANK-00] J. D. Blankensteijn, ***Imaging techniques for endovascular repair of abdominal aortic aneurysms***, 2000.

[BROWN-92] P. M. Brown, R. Pattenden and J. R. Gutelius, ***The selective management of small abdominal aortic aneurysms: the Kingston study***, Journal of Vascular Surgery, (1992), pp. 21–27.

[BRUIN-01] M. de Bruijne, Bram van Ginneken, Wiro J. Niessen, J. B. Antoine Maintz and Max A. Viergever, ***Active shape models exploiting slice-to-slice correlation in segmentation of 3D CTA AAA images***, 2001.

[BRUIN-02] Marleen de Bruijne, Bram van Ginneken, Wiro J. Niessen, J. B. Antoine Maintz and Max A. Viergever, ***Active shape model based segmentation of abdominal aortic aneurysms in CTA images***, 2002.

[COVER-91] Thomas M. Cover and Joy A. Thomas, ***Elements of Information Theory***, Wiley, 1991.

[ECALL-97] H. Schumacher, H. H. Eckstein, F. Kallinowski and J. R. Allenberg, ***Morphometry and classification in abdominal aortic aneurysms: Patient selection for endovascular and open surgery***, Journal of Endovascular Surgery, 4 (1997), pp. 39–44.

[FMAES-97] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal and Paul Suetens, ***Multimodality Image Registration by***

Maximization of Mutual Information, IEEE Transactions on Medical Imaging, 16 (1997), pp. 187–198.

[GERHA-92] Philip M. Gerhart, Richard J. Gross and John I. Hochstein, ***Fundamentals of Fluid Mechanics***, Addison Wesley, second ed., 1992.

[GRIMS-00] L. Zollei, E. Grimson, A. Norbash and W. Wells, ***2D-3D Rigid Registration of X-Ray Fluoroscopy and CT Images Using Mutual Information and Sparsely Sampled Histogram Estimators***, 2000.

[ITKSR-02] ***NLM Insight Segmentation and Registration Toolkit (ITK)***, 2002. <URL:http://www.itk.org/>.

[JASET-02] ***J. A. Sethian's pages on level set and fast marching methods***, 2002. <URL:http://math.berkeley.edu/~sethian/>.

[JASET-96] J. A. Sethian, ***Level Set Methods: An Act of Violence***, 1996.

[JASET-99] J. Sethian, ***Level Set Methods and Fast Marching Methods***, Cambridge University Press, 1992.

[JENSE-01] Tormod Omholt-Jensen, ***Segmentation of the hepatic vessels as seen in MR or CT images***, 2001.

[LONCA-01] Sven loncaric, Marko Subasic and Erich Sontarin, ***3-D deformable model for abdominal aortic aneurysm segmentation from CT images***, 2000.

[MAGEE-00] Derek Magee and Andrew Buliptt and Elizabeth Berry, ***Level Set Methods for the 3D Segmentation of CT Images of Abdominal Aortic Aneurysms***, 2000.

[MAINT-98] J. B. Antoine Maintz and M. A. Viergever, ***A survey of medical image registration***, 1998.

[MALLA-95] R. Malladi, J. A. Sethian and B. Vemuri, ***Shape modeling with front propagation: A level set approach.***, IEEE Trans. on Pattern analysis and Machine Intelligence, 17 (1995), pp. 158–175.

[NEMAX-93] National Electrical Manufacturers Association (NEMA), ***Digital imaging and communications in medicine***, 1993.

[OWINK-00] O. Wink, W. J. Niessen and M. A. Viergever, ***Fast delineation and visualization of vessels in 3-D angiographic images***, IEEE Transactions on Medical Imaging, 19 (2000), pp. 337–346.

[POHLE-00] Regina Pohle and Klaus D. Toennies, ***Segmentation of medical images using adaptive region growing***, 2000.

[RAVHO-98] R. Ravhon and D. Adam and L. Zelmanovitch, ***Validation of Ultrasonic Image Boundary Recognition in Abdominal Aortic Aneurysm***, IEEE Transactions on Medical Imaging, 20 (2001), pp. 751–763.

[RCEGW-93] Rafael E. Gonzalez and Richard E. Woods, ***Digital Image Processing***, Addison Wesley, 1993.

[ROCHE-01] Alexis Roche, Xavier Pennec, Gregoire Malandain and Nicholas Ayache, ***Rigid Registration of 3-D Ultrasound With MR Images: A New Approach Combining Intensity and Gradient Information***, IEEE Transactions on Medical Imaging, 20 (2001), pp. 1038–1049.

[RSNAW-02] ***RadiologyInfo***, 2002. The Public Information web site developed and funded by the American College of Radiology and the Radiological Society of North America. <URL:http://www.radiologyresource.com/>.

[SAPIR-01] Guillermo Sapiro, ***Geometric Partial Differential Equations and Image Analysis***, Cambridge University Press, 2001.

[SHOLM-98] Sverre Holm, ***Medisinsk ultralydavbildning (Medical ultrasound imaging)***, 1998.

[SMADA-95] S. Shiffman, G. D. Rubin, S. Napel, ***Semiautomated editing of computed tomography sections for visualization of vasculature***, 1995.

[TAYLO-00] T. F. Cootes and C. J. Taylor, ***Statistical models of appearance for computer vision***, 2000.

[TAYLO-01] T. Cootes and C. J. Taylor, ***Statistical models of appearance for medical image analysis and computer vision***, Medical Imaging: Image Processing, Proceedings SPIE, (2001), pp. 236–248.

[TAYLO-95] T. F. Cootes, C. J. Taylor and D. H. Cooper and J. Graham, ***Active shape models: their training and application.***, Computer Vision and Image Understanding, 61 (1995), pp. 38–59.

[TILSO-02] ***Answers to FAQs about abdominal aortic aneurysms***, 2002. <URL:http://www.columbia.edu/~mdt1/faqs.html>.

[TNRPA-02] ***The Norwegian Radiation Protection Authority (Statens strålevern)***, 2002. <URL:http://www.nrpa.no/>.

[UDUPA-00] Jayaram K. Udupa and Gabor T. Herman, ***3D Imaging in Medicine***, CRC Press, second ed., 2000.

[UNSER-00] Philippe Thevenaz and Michael Unser, ***Optimization of Mutual Information for Multiresolution Image Registration***, IEEE Transactions on Image Processing, 3 (2000), pp. 2083–2099.

[VERDO-96] B. Verdonck, I. Bloch, H. Maitre, D. vandermeulen, P. Suetens and G. Marchal, ***Accurate segmentation of blood vessels from 3D medical images***, IEEE International Conference on Image Processing, 3 (1996), pp. 311–314.

[VHPRO-02] ***The National Library of Medicine's Visible Human Project***, 2002. <URL:http://www.nlm.nih.gov/research/visible/visible_human.html>.

[VIERG-97] J. B. Antoine Maintz and Max A. Viergever, ***A survey of Medical Image Registration***, 1997.