

Abstract

In this master thesis a novel image segmentation method is presented. Segmentation is the process of dividing images into interesting regions, or objects. The prototype based image segmentation method proposed in this thesis introduces *a priori* information about the object to be segmented, from a set of few manually segmented image examples to a general segmentation method, the level set method. The *a priori* information is extracted as an intensity function, an initial segmentation, a spatial map, and finally some optimised parameters for the level set method. The intensity function is derived from signal intensity statistics of the image examples. The initial segmentation is an approximation of the shape of the object to be segmented. The spatial map constrains the segmentation based upon a correction map and a distance map derived from the image examples. The prototype is used for image segmentation with a fast level set method without having to adjust any parameters. The only manual interaction needed is to set anatomical landmarks, typically 3 or 4, in the images. The method is validated on different image segmentation problems in MR images, the aorta, left ventricle wall volume, left ventricle blood volume, right ventricle blood volume and pig lungs. The validation shows promising results and that a major advantage of the method is the introduction of the spatial map. The strength of the segmentation method is that only a few image examples are required to introduce *a priori* information that allows to tailor a general purpose segmentation algorithm for specific applications.

Acknowledgements

Firstly I would like to thank my supervisor Einar Heiberg for fruitful discussions and extraordinary support, my supervisor Kalle Åström and Erik Bergvall for mathematical guidance. Secondly I thank Johan Renner for the images and segmentation of the aorta, Joey Ubachs for manual segmentation of the left ventricle, Sverrir Stephensen for manual segmentation of the right ventricle blood volume and Christian Torbrand and Olof Gålne for manual segmentation of pig lungs. Finally I would like to give a big thanks to the whole Cardiac MR Group at Lund University for an inspiring and encouraging environment.

Contents

1	Introduction	1
2	Problem formulation	3
3	Background	5
3.1	Statistical methods	5
3.2	Deformable model methods	6
3.3	Active Models	6
3.4	Level set method	6
3.4.1	Fast marching formulation	8
3.4.2	Level set formulation	9
3.4.3	Advantages of the level set and fast marching methods	10
3.4.4	Implementation of the level set method	11
3.4.5	Implementation of the fast marching method	11
3.4.6	Fast approximation of level set computations	12
3.5	The software Segment	15
4	Methods	17
4.1	Speed law	20
4.2	The prototype	22
4.2.1	Intensity function	22
4.2.2	Alignment	26
4.2.3	Initialisation	27
4.2.4	Spatial map	28
4.2.5	Parameter optimisation	30
4.2.6	Generation of the prototype	31
4.3	Segmentation	35
4.4	Validation	37
4.4.1	Segmentation results	37
4.4.2	Method sensitivity	38

5	Results	41
5.1	Quantitative results	46
6	Discussion & Conclusions	55
6.1	Segmentation of aorta	56
6.2	Segmentation of left ventricle wall volume	57
6.3	Segmentation of left ventricle blood volume	58
6.4	Segmentation of right ventricle blood volume	59
6.5	Conclusions	60
7	Future work	61

Chapter 1

Introduction

Medical images, such as magnetic resonance imaging (MRI) and computed tomography (CT), contain wealth of information, both anatomical and functional, which can be used for a diagnosis, surgery planning and research. These images can be two-dimensional time resolved images, or three-dimensional images both time resolved and non-time resolved. A lot of the information in these images can be assessed by simple visual inspection, but to make quantitative measurements objects in the images have to be segmented. Segmentation is the process of separating objects in an image by adding a contour, or a surface in 3D images, that separates objects as good as possible in some sense. Quantitative measurements that can be derived are for instance size of organs, size of a heart infarct, contraction of heart muscle, size of tumours and also the progression or regression of a tumour or other disease progress. From the segmentations even the blood flow in for instance the aorta can be simulated using Computational Fluid Dynamics (CFD) [1].

Image segmentation is a central challenge in medical image analysis, since there are so many different image types and the solution to the problem differs with the application. Segmentation can be done manually which is time consuming, for instance it takes 20 minutes at least to segment the complete 3D left ventricle manually. Manual segmentation also suffers from being subjective with both inter and intra subject variability. Therefore it is advantageous to be able to do automated or at least semi-automated segmentation. Automatic methods exist only for a few medical image segmentation problems. One reason for this is that the development of new methods is time consuming and often challenging.

This master thesis has been written within the Cardiac MR Group at Lund University. In this group the research is focused on cardiac MR images and quantitative measurement that can be derived from these images. With constantly new research projects and a large amount of three-dimensional

images to be segmented an automatic image segmentation method would be of great advantage.

General segmentation methods often give poor results since they do not utilise any *a priori* information about the object to be segmented, nor the imaging modality. Methods which do incorporate *a priori* information often require a relatively large set of manually segmented image examples, usually over 50 images [2], [3].

It would be desirable to have an accurate automatic segmentation which can easily handle new applications that is to say, can be built fast and does not need a large set of manually segmented images to incorporate *a priori* information, and this is the rationale for this thesis.

The method developed during this master thesis has also resulted in a conference paper presented at the Swedish Society for Automated Image Analysis conference in Lund 2008.

Chapter 2

Problem formulation

The aim of this master thesis is to construct a method to extract *a priori* information for image segmentation from a few example images. The method shall be general in the sense of being able to extract and use information from different kind of images. The *a priori* information should then be used to semi-automatically perform application tailored image segmentation.

Chapter 3

Background

In this chapter a few common segmentation methods will be presented briefly and the level set method used in this master thesis will be more thoroughly reviewed.

Image segmentation is the process of dividing an image into different regions, or objects, based upon some features. These features can for example, be intensity, colour or location of the pixels.

One of the simplest segmentation methods is to threshold the image at a certain intensity level. This method can only be used if the objects in the image has well defined intensities, that is to say one object has intensities within a defined interval and this interval does not overlap with another object.

More sophisticated segmentation methods are level set methods, deformable model methods, statistical methods, like the EM algorithm, and Active Model methods, like the Active Shape Model and Active Appearance Model.

3.1 Statistical methods

The EM-algorithm is a statistical method [4], where solely the number of different objects is given beforehand, called clusters. For example two clusters are used if one kind of object should be segmented from the background. The segmentation starts by picking two pixels to initialise the clusters, the rest of the pixels are then divided into the clusters. The dividing into clusters is done so that the variance of the clusters is minimised. The variance can be measured for example of the pixel intensity or location.

3.2 Deformable model methods

Deformable model methods [5], are based on the idea of an initial segmentation, a curve or a surface, and propagating it in a certain way specific for the method and the application.

Often this is applied by discretising the curve, which divides the image into regions, or objects. After the discretisation the curve is parameterised by points which can be moved. The curve segments between the points are calculated as linear segments or as splines. The way of moving the curve can be done in several ways. One such idea is to define an energy which can depend on the position and shape of the curve and the propagation of the curve is done in a way that minimises the energy of the curve. This method is a marker based deformable model often referred to as a snake when applied in 2D [6]. The method can be extracted to 3D and 3D+T, three-dimensional time resolved images, but the parameterisation becomes much more difficult [7]. A disadvantage with the marker based methods is that it can not handle change of topology in its general form [8].

3.3 Active Models

Active Models refers to the two methods Active Shape Model [9] and Active Appearance Model [2]. Active Shape Models are a deformable model containing *a priori* information about the shape of the object and the Active Appearance Model is a further development which contains information about both shape and intensity of the object. The model has been built on a set of segmented images. To segment a new object the model is deformed by adjusting parameters in the model to minimise the difference between the image and the model. The model and parameters are based on a principal component analysis of the image examples. Both the active shape model and active appearance model is well functioning but the disadvantage is that a relatively large set of images, usually over 50 images [2], [3], is needed to build an accurate model.

3.4 Level set method

The level set method, [10], is another way of propagating an initial curve or surface. Since the level set method will be used in this thesis it will be described in a more extensive way than the other methods. The level set method is a general method which will be made application specific by the use of *a priori* information. The fast marching method, which is a special case of

the level set method, will be described since the two methods are similar and the fast marching method will be used when extracting *a priori* information.

Level set methods and fast marching methods are numerical methods for tracking moving boundaries. The method can be used for image segmentation and the boundaries are the curve which separates the regions, or objects. The curve moves under the influence of a speed law and the speed law depends on different factors, which in image segmentation often are features in the image. Figure 3.1 shows an example of a propagating curve segmenting an image into different regions, one inside the curve and one outside the curve. In the level set method and fast marching method the moving boundary is the

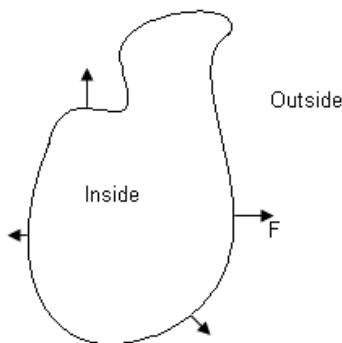


Figure 3.1: A curve moving under influence of a speed law and segmenting an image into the two regions, inside and outside.

implicit level set surface acquired from the solution to a partial differential equation. This means that the curve is not discretised in the level set method, nor in the fast marching method. The partial differential equation to solve differs between the level set method and the fast marching method and thus some properties of the methods differ. Common for the two methods is that the curve, which in image segmentation separates the image into regions, is allowed to change topology. The curve can for example split or two curves can merge into one. The fast marching method is a very fast method but it can only solve problems with a speed law which is positive, that is to say the curve can only propagate outwards. The level set method can handle a more complicated speed law, without any restrictions to the sign of the speed law, but on the other hand this method is not that fast.

3.4.1 Fast marching formulation

The problem of tracking moving boundaries, under the influence of a speed law, F , is easier to solve if the propagation of the curve always is in the same direction. If the speed is strictly positive the problem can be solved with the fast marching method. Since the speed law, F , is strictly positive the curve will only propagate outwards and therefore pass through each position, \mathbf{x} , only once. Instead of computing the position of the curve Γ in every time step the arrival time T in every position \mathbf{x} , in a grid, can be calculated. The arrival time function $T(\mathbf{x})$ creates a surface which has the property

$$\Gamma(t) = \{\mathbf{x} \mid T(\mathbf{x}) = t\}. \quad (3.1)$$

This means that the initial curve $\Gamma(0)$ corresponds to the zero level set of $T(\mathbf{x})$ and for every time t the curve $\Gamma(t)$ can be extracted from $T(\mathbf{x})$ as the t level set. An example on how the function T and the level set at different times can look like is given in Figure 3.2.

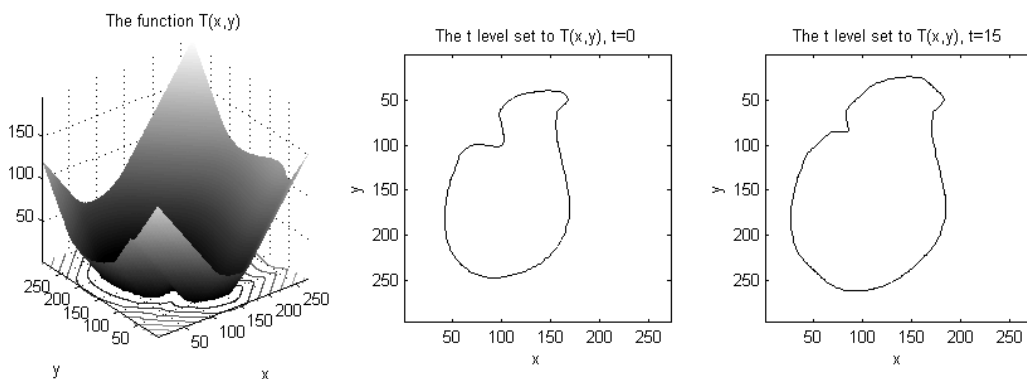


Figure 3.2: The function $T(\mathbf{x})$ (left), the level set at time $t = 0$ (middle) and the level set at time $t = 15$ (right).

The arrival time function $T(\mathbf{x})$ depends on the speed law F . F is the speed of the curve propagation in the direction normal to the surface. In the one-dimensional problem the equation

$$\frac{dT}{dx} = \frac{1}{F}, \quad T(0) = \Gamma(0) \quad (3.2)$$

is the equation to solve to calculate T . In two, or higher, dimensions the equation can be written as

$$|\nabla T|F = 1, \quad \Gamma(t) = \{\mathbf{x} \mid T(\mathbf{x}) = t\}. \quad (3.3)$$

∇T is orthogonal to the level sets of the equation and the magnitude is inversely proportional to the speed function F . In the fast marching method the speed law, F is required to be strictly positive. If the problem is reversed, that is to say the definition of outside and inside is reversed, the method can also be applied to curves which are shrinking.

3.4.2 Level set formulation

Since it is desirable to be able to solve problems with a speed law which is not strictly positive nor strictly negative another method is needed. A method appropriate for solving this problem is the level set method. When the speed law is able to change sign the arrival time function will not be a single valued function and the tracking of the moving boundary can not be done by calculating the arrival time function. Instead the initial curve $\Gamma(0)$ is built into a surface ϕ similar to the arrival time function but this surface must account for the fact that the speed law is neither strictly positive nor negative. If the surface is linked to an initial value problem the zero level set of $\phi(\mathbf{x}, t)$ will give the curve Γ as

$$\Gamma(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\} . \quad (3.4)$$

If a point $\mathbf{x}(t)$ belongs to the curve $\Gamma(t)$ then the point will also belong to the zero level set of ϕ at time t

$$\phi(\mathbf{x}(t), t) = 0 . \quad (3.5)$$

Differentiation of this equation gives

$$\phi_t + \nabla\phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0 . \quad (3.6)$$

To link this equation to the speed law F we observe that the speed law F is the speed orthogonal to the curve, $F = \mathbf{n} \cdot \mathbf{x}'(t)$. Also observe that the normal vector is calculated from the function ϕ , $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$. This gives

$$F = \mathbf{n} \cdot \mathbf{x}'(t) = \frac{\nabla\phi}{|\nabla\phi|} \cdot \mathbf{x}'(t) . \quad (3.7)$$

This observation gives that the equation to be solved for the level set method is

$$\phi_t + F|\nabla\phi| = 0 . \quad (3.8)$$

The propagating curve Γ is the implicit solution to the zero level set of the function ϕ

$$\Gamma(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\} . \quad (3.9)$$

The differences between the level set method and the fast marching method are that the fast marching method can only handle a speed law, F , which is strictly positive and the fast level set method can handle more complex speed laws, which for example can include the local curvature. The fast marching method is however a much faster method since it only passes through each grid point once.

3.4.3 Advantages of the level set and fast marching methods

The level set method and the fast marching method have several common advantages compared to other segmentation methods. Both can be extended to three or higher dimensions. This property is very important since most medical images are either two-dimensional and time resolved, that is to say three-dimensional in some sense, or three-dimensional images as used in this master thesis, or even three-dimensional and time resolved. Marker based methods like snake are parameterised and the parameterisation becomes much more difficult in three or higher dimensions. Another advantage with the level set method and fast marching method is that the curve, or surface, Γ can easily change topology, that is to say the curve is allowed to split and merge. Figure 3.3 shows an example of two curves merging into one. For marker based curves it is hard or not even possible to change topology since it is difficult to remove markers.

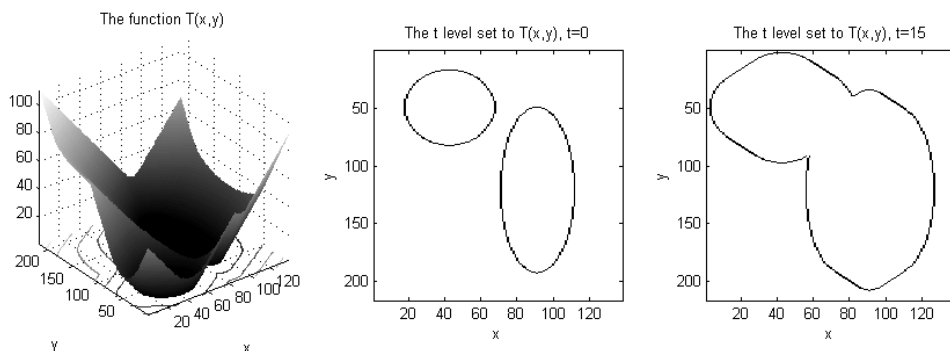


Figure 3.3: The function $T(\mathbf{x})$ (left), the level set at time $t = 0$ (middle) and the level set at time $t = 15$ (right) showing how two curves can merge into one.

Other advantages are that the curve is implicit, that is to say the curve is not discretised, and the methods are accurately approximated with schemes on a discretised grid. Also the geometric properties normal vector and curvature can be calculated. The normal vector is calculated as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \text{ or, } \mathbf{n} = \frac{\nabla T}{|\nabla T|} \quad (3.10)$$

and the curvature is calculated as the divergence of the normal vector

$$\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}, \text{ or, } \kappa = \nabla \cdot \frac{\nabla T}{|\nabla T|}. \quad (3.11)$$

3.4.4 Implementation of the level set method

The level set method is a numerical method built upon a partial differential equation. In numerical methods the derivatives are estimated on a discretised grid and this can be done in several different ways. The derivatives must be accurately approximated to be able to maintain smoothness. The spatial derivatives can be estimated with forward, backward and central differences. A scheme is used to decide which derivative approximation to use depending on the sign of the speed law F in the level set method. For example if the curve propagates towards a point from the left the backward difference should be used since information travels that way. For the time derivative it is natural to use the forward difference since the next time step is calculated from the current step. In the level set method it is always the zero level set of ϕ that specifies the curve, Γ , and because of that it is possible to limit the calculations of ϕ to a region near the zero level set. This operation specifies a faster implementation of the level set method, called the narrow band level set method [10].

3.4.5 Implementation of the fast marching method

The fast marching method can be seen as a stationary problem and from the initial curve $\Gamma(0)$ the surface $T(\mathbf{x})$ can be iteratively calculated. Since the curve Γ only passes through each grid point \mathbf{x} once it is important to calculate the first arrival time to that point, that is to say the lowest value of $T(\mathbf{x})$. To iteratively calculate T , propagate the curve according to the speed law F and find out which grid point holds the lowest arrival time. From this grid point propagate further and compare these new arrival times to the arrival times of the grid points which did not have the lowest arrival time in the previous step. In this iterative scheme it is important to keep track

of the grid points and arrival times for the grid points from which the curve has not propagated further.

The grid points are divided into three different groups. The first group consist of grid points for which the arrival time has been calculated and the curve has propagated further away from, these are not necessary to keep track of apart from the calculated arrival time which is stored in $T(\mathbf{x})$. The second group is the one containing the grid points for which the arrival time T has not been calculated. The third group consists of the grid points for which the arrival time T has been calculated but the curve has not propagated further since the arrival time T was lower in another grid point. It is important to keep track of both the arrival time and the grid point in the third group and it can be done with a min-heap.

A min-heap is a data structure in which it is very fast and easy to find the minimum value, [11]. The min-heap is reminding of a tree turned upside down. At the top of the min-heap the smallest value is stored and one level beneath higher values is stored. The value at one level is always smaller than the value on the level beneath. It is fairly easy to store new values into the heap and finding the minimum value is a very fast operation. Since both finding the minimum value and storing new values is fast in the min-heap the min-heap is well suited for the fast marching algorithm.

The grid points in group three are sorted into a min-heap and the grid point to propagate further will always be at the top of the heap. After propagation from a grid point the grid point and its arrival time value is removed from the heap, and added to group one, and the grid points to which the curve propagated are sorted into the heap. The calculation of the arrival time is done by iterating the process of picking a grid point from the heap, propagating the curve according to the speed law to new grid points, and sorting these new grid points into the heap. Eventually the arrival time function will be calculated for all grid points or the calculation stops when the arrival time T reaches a specified value.

3.4.6 Fast approximation of level set computations

The level set method is more general than the fast marching method since the speed law can be both positive and negative but on the other hand the fast marching method is much faster than the level set method. Preferably the method for the image segmentation should be both general and fast. Nilsson and Heyden have combined the two methods in [12]. An implementation of their ideas is used in this master thesis to do the segmentation. Their approach is to use the idea of an iterative scheme from the fast marching method and to approximate the level set calculations.

The grid points are divided into three different groups, points inside, point outside and points on the curve. The points on the curve have an arrival time T_a and from this arrival time and the speed law F a departure time T_d is calculated as

$$T_d = T_a + \frac{1}{|F|} . \quad (3.12)$$

The departure time T_d is then stored in a min-heap and from the grid point with the lowest departure time the curve is propagated. If the speed law is negative the curve will propagate inwards. To propagate the curve inwards the grid point with the lowest departure time will be added to the group of points outside and the neighbouring grid points which belonged to the inside will be added to the group of points on the curve. The grid points which are added to the curve get an arrival time T_a equal to the departure time of the grid point from which the curve propagated. The process of propagating the curve by picking the grid point with the lowest departure time continues until a halting criterion is reached. In Figure 3.4 the fast level set method, as proposed by Nilsson *et al.* is represented as a flow chart.

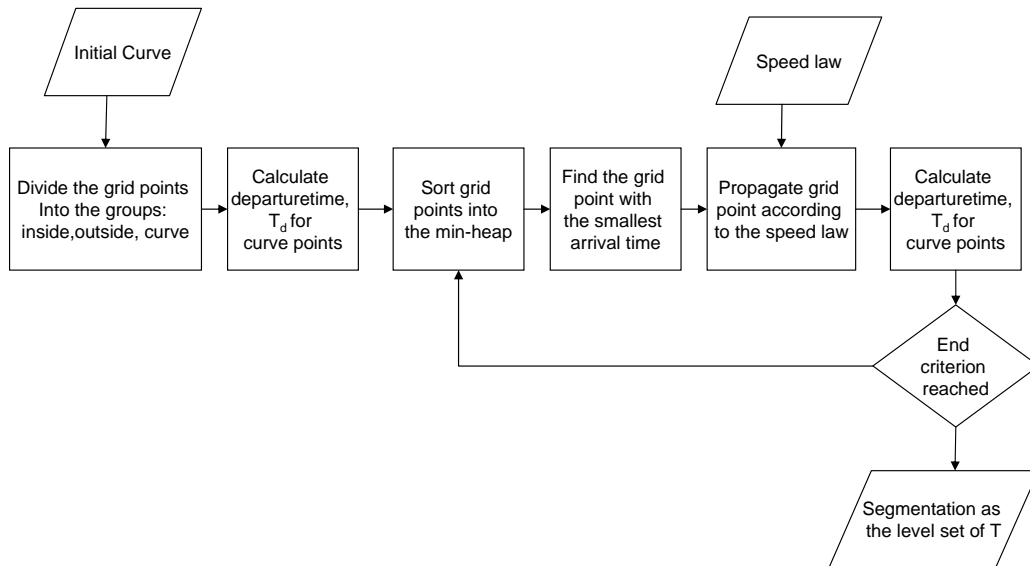


Figure 3.4: Flow chart of the fast level set method.

The speed law F can contain the curvature and the normal direction of the curve and these are in a level set method estimated from the neighbourhood of the level set. In this algorithm the value of ϕ is not calculated and the curvature and normal direction must be approximated in another way. To do this a function m is introduced,

$$m(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \text{inside} , \\ 0, & \mathbf{x} \in \text{curve} , \\ -1, & \mathbf{x} \in \text{outside} . \end{cases}$$

The curvature is a measurement in the level set method used to smooth the curve and therefore it might not be necessary to have an accurate approximation. The approximation is done by counting the number of grid points belonging to the group inside points within a circular neighbourhood of the grid point belonging to the curve. The number of such grid points is referred to as n_{inside} and from this quantity the curvature is estimated as

$$\tilde{\kappa} = \frac{n_{inside}}{n} - \frac{1}{2}, \quad (3.13)$$

where n is the number of grid points in the circular neighbourhood.

The speed law can also contain the normal vector and this is in a level set method calculated from the level set function ϕ . Since ϕ is not calculated in this algorithm the normal vector is instead approximated from the function m . The algorithm suggested is a fast approximation of the level set method. It is fast since it only calculates the values on the zero level set, the approximations of curvature and normal vector are fast and a heap is used to keep track of how to propagate the curve.

3.5 The software Segment

Segment is a freely available software for cardiovascular image analysis (<http://segment.heiberg.se>). This software has been developed within the Cardiac MR Group at Lund University. It is an extensive software, and only two of the tools have been used within this master thesis, the manual segmentation tool and the general segmentation tool which has been further developed and improved as a consequence of this master thesis. The general segmentation tool had an implementation of the fast level set for general segmentation with an interface for visualisation of result, adjustment of the speed law, and tools for manual interactions. All the methods described in this thesis are implemented in Segment. Segment has thus been extended to handle the novel prototype based segmentation scheme developed in this thesis. In Figure 4.10 an example of the general segmentation tool can be seen. A tool for generating a prototype is also implemented and in Figure 4.9 the user interface is shown. Either the tool for manual segmentation or the general segmentation tool in Segment has been used to segment the set of image examples from which the *a priori* information is extracted.

Chapter 4

Methods

In the method developed a general segmentation method, the level set method is used and made application specific by the use of *a priori* information. The *a priori* information is extracted from a few manually segmented example images and stored in a 'prototype'. The prototype is then used with a fast level set method [12] for segmentation of new images.

To use the level set method a speed law and an initial segmentation has to be defined. It is crucial to have a speed law which is well suited for the image to be segmented. However, it is desirable to have a general speed law so that a new speed law does not have to be manually adjusted for each image segmentation problem. In order to accomplish this, the speed law can have a general form where the parameters are adjusted to fit to a specific image segmentation problem. To adjust the speed law to a specific segmentation problem *a priori* information about the object can be used. The general appearance and how to use *a priori* information in the speed law is described in Section 4.4.

The information is stored in the prototype both as intensity and spatial information about the object. The prototype defines both the speed law and initialisation of the level set method. In Section 4.2 the generation of a prototype is described, and in Figure 4.1 an overview is given as a flow chart.

The prototype is generated for a specific image segmentation problem and used with the fast level set method for segmentation of new images. How the segmentation is done is described in Section 4.3, and in Figure 4.2 an overview is given as a flow chart.

The method is also validated by generating prototypes for different applications and segmenting new images from these applications. In Section 4.4 the validation of the method is described.

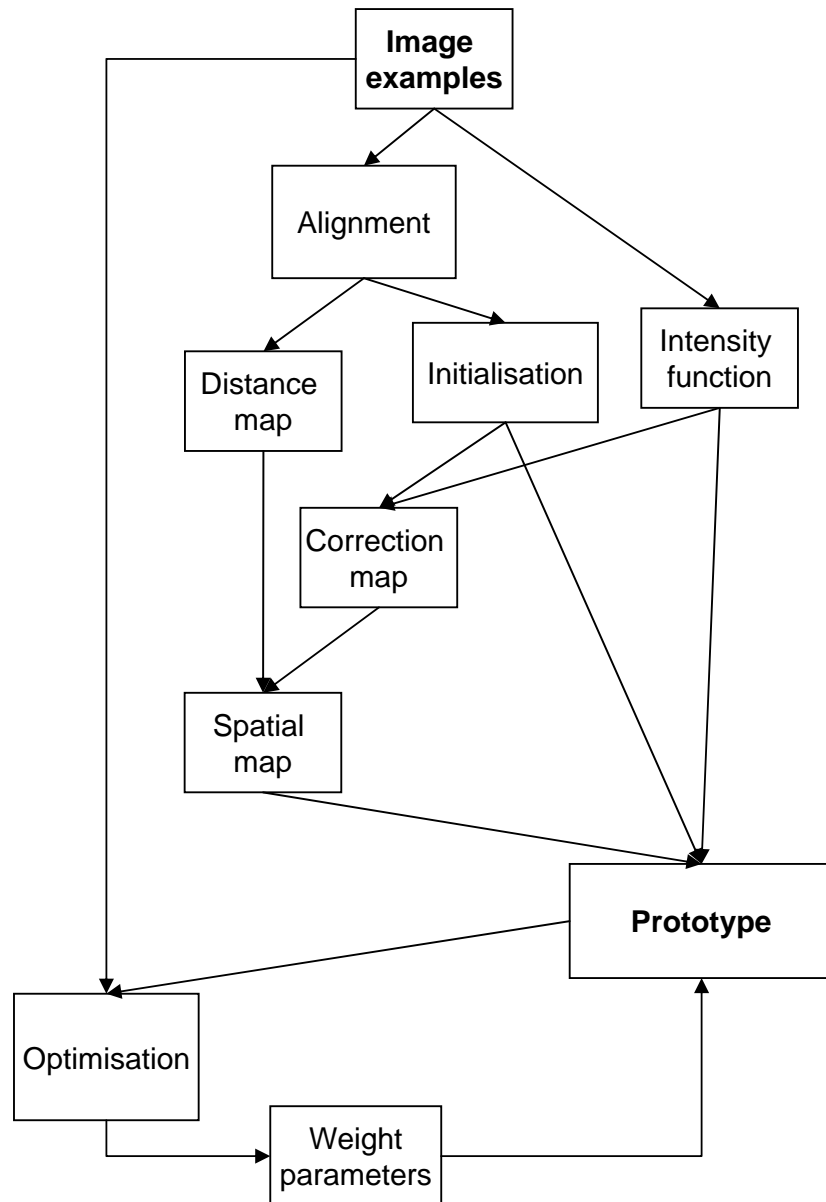


Figure 4.1: Flow chart of how the prototype is generated. The process starts with image examples (top) and ends with a prototype (bottom right).

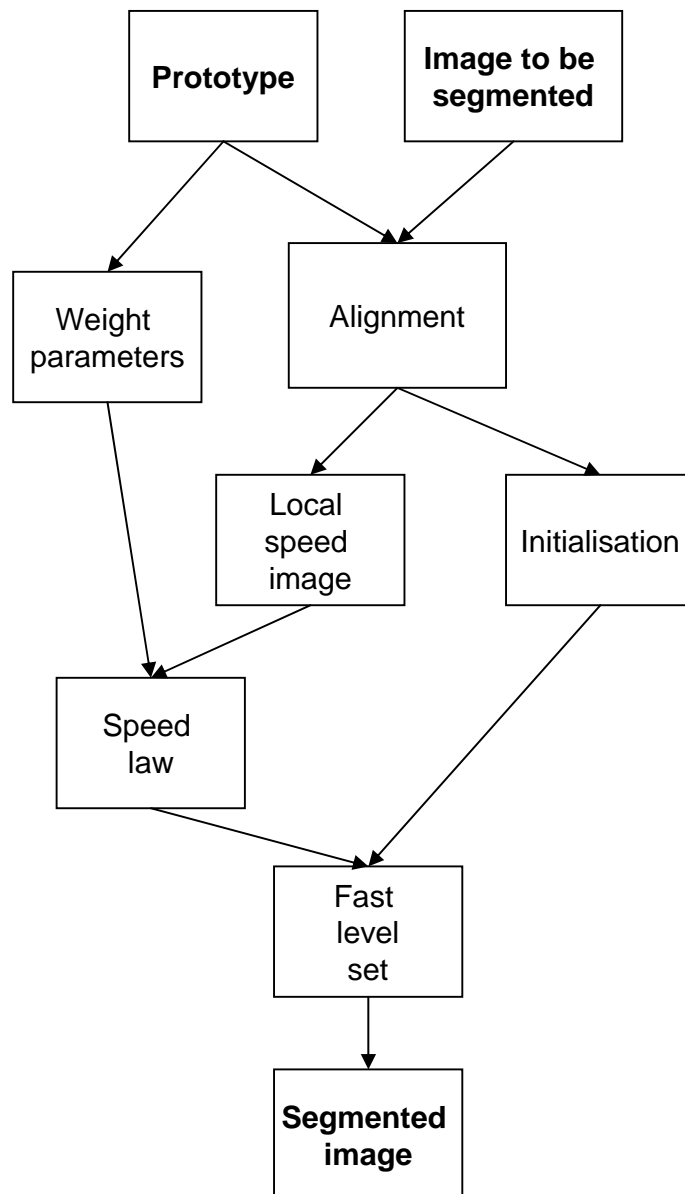


Figure 4.2: Flow chart of how the segmentation is done using the prototype.

4.1 Speed law

In level set methods the speed law can have many different appearances since it can solve many different kinds of problems [10]. In image processing the speed law often has a local propagation part depending on the image to be segmented, a smoothing part which depends on the curvature and a stabilising part which depends on where the edges in the image are located. The weights of these parameters can be adjusted to suit the specific image segmentation problem, and even more important is that the local propagation part can be adjusted to the problem.

The speed law derived in this section will be general in its appearance to be able to handle different applications and be adjusted to a specific application by the use of *a priori* information.

The *a priori* information is introduced to the local propagation part, which will further be referred to as the local speed image I_{speed} . How to incorporate the *a priori* information from the prototype will be further discussed but first the general appearance of the speed law will be derived.

The speed law will have both a smoothing and stabilising part. The curvature κ is used as the smoothing part together with a weight parameter β . The curvature is calculated by the level set method and the weight β specifies how much the curvature will affect the propagation speed.

The stabilising part is used to increase the likelihood of the segmentation converging to the edges. This can be done by introducing a gradient part which decreases the propagation speed near the edges. The factor $\frac{1}{1+\alpha|\nabla I|}$, where $|\nabla I|$ is the magnitude of the gradient of the image I to be segmented and α is a weight parameter, is multiplied with the local speed image to decrease the speed near the edges in the image. The propagation will be decreased near the edges since the gradient is high near edges and the propagation speed will be nearly unaffected otherwise since the gradient is close to zero otherwise. The parameter α will decide how much the speed is lowered near the edges. With the use of the gradient factor and the curvature the speed law is given as

$$F = I_{speed} \frac{1}{1 + \alpha |\nabla I|} + \beta \kappa , \quad (4.1)$$

where the local speed image I_{speed} is based upon the *a priori* information from the set of example images and the image I to be segmented.

When the speed law is positive the curve will expand and the curve will strive to include all grid points, pixels, with a positive value of the speed law. The local speed image, I_{speed} , is the main propagating part of the speed law

and has positive values for pixels that should be included in the region and negative values for the pixels that should not be included.

The local speed image can be based upon pixel intensity, colour, texture and location. In most applications pixel intensity is the most obvious choice, and therefore also used in this implementation. The intensities in the image to be segmented will be mapped to a speed with a lookup-table. This lookup-table is a one-dimensional function, the intensity function $h_{intensity}$, which is based upon the intensities in the example images and stored in the prototype. The segmentation of the object is thus based upon the intensities in the image to be segmented and in the example images.

The location of a pixel can also be used to calculate the speed in the local speed image. This is particularly useful where pixels that should not belong to the segmentation have the same intensity as pixels that should belong to the segmentation. The location of a pixel can be used to lower the speed in the local speed image I_{speed} if the pixel should not belong to the object. This is done by subtracting a spatial map, $M_{spatial}$, from the image calculated with the intensity function. The spatial map is based upon the set of example images and stored in the prototype.

The local speed image I_{speed} is thus calculated as

$$I_{speed} = G(I, h_{intensity}) - \lambda M_{spatial} , \quad (4.2)$$

where I is the intensity image, G is a conversion function of the intensities in the image to the values of the intensity function $h_{intensity}$, and λ is a weight parameter deciding how much the spatial map $M_{spatial}$, will influence the local speed image, I_{speed} .

The segmentation is based upon both intensity and location of a pixel. The object must have a clear mapping to intensities used in the intensity function or a clear spatial mapping. Applications where there is no clear mapping of the intensities in some parts of the object and that the position of that part varies will not be feasible to segment with the method proposed in this thesis. Example of such an application is cardiac SPECT images where infarcts, or other perfusion defects, make some parts of the cardiac nearly invisible in the image and the location of the perfusion loss differs. To handle these kinds of images application specific methods will have to be developed. These kind of applications are, however, luckily rare since most image types has a clear intensity and spatial mapping.

The prototype used for segmentation with the level set method contains *a priori* information in the intensity function $h_{intensity}$, and the spatial map $M_{spatial}$, as well as the parameter weights α , β and λ so that the speed law is completely adjusted by the prototype.

4.2 The prototype

To be able to extract *a priori* information a set of segmented image examples is needed. The segmentation can be done either manually or with help from the general segmentation tool in Segment. The example images should also contain a set of landmarks used for alignment.

The prototype contains information needed in the segmentation process, such as, the intensity function, spatial map and parameter weights which define the speed law, and also an initialisation of the level set method.

4.2.1 Intensity function

The intensity function is a one-dimensional function which maps pixel intensities to a speed used in the level set method. The function is based upon statistics from the set of example images.

The probability distribution of interest is the probability of a pixel belonging to the object given the pixels intensity value. The probability distribution for the pixel intensities can easily be approximated from the histogram by normalising the histogram by the sum of the histogram. The probability of a pixel belonging to the object given its intensity value is calculated with the usage of Bayes law as

$$p(s|i) = \frac{p(i|s)p(s)}{p(i)}, \quad (4.3)$$

where s is the event of the pixel belonging to the object and i is the intensity value. The probability distributions $p(i|s)$ and $p(i)$ is calculated from the intensity histograms with the small difference being from which region the histograms are calculated.

For the distribution $p(i|s)$ the histogram is calculated from the object and for the distribution $p(i)$ the histogram is calculated from a larger region. This larger region contains the object and a region outside the object. The probability $p(s)$ is the probability of a pixel belonging to the object, without any information of the intensity value. This probability is approximated from the quotient of number of pixels belonging to the object and the number of pixels belonging to the larger region.

The probability of a pixel belonging to the object given the pixels intensity value $p(s|i)$ is compared to the probability of the pixel not belonging to the object given the pixels intensity value, $p(s^c|i)$. This probability distribution can be expressed as

$$p(s^c|i) = \frac{p(i|s^c)p(s^c)}{p(i)}, \quad (4.4)$$

where s^c is the event of the pixel not belonging to the object and i is the intensity value. The probability distribution $p(s^c|i)$ is calculated in the same way as the probability distribution $p(s|i)$ with the small difference being from which region the histograms are calculated. For the probability $p(s^c|i)$ the region outside the object is used instead of the object. The larger area, used to calculate $p(i)$, is the same for both distributions and this area is the sum of the object and the region outside the object.

The region outside the object can be all pixels which do not belong to the segmentation or it can be a smaller region containing the pixels closest to the object. For this implementation a smaller region has been used. The reason for this is that the level set method will not propagate through the whole image, but rather in the neighbourhood of the object. Also the probability of belonging to the object given the pixel intensity, $p(s|i)$, is dependent of the probability of belonging to the object $p(s)$. This probability is low if the region outside the object is much larger than the object. If the probability $p(s)$ is low, the probability $p(s|i)$ also is low and it will generate a lower speed in the local speed image.

To create the region outside the object, the binary mask of the object is being dilated with a ball with a specified radius. Dilation is a morphological operation which makes white regions in a black and white image larger. Pixels within a distance of the specified radius from the binary mask become white just like the mask. The newly created image is a mask defining the sum of the outside region and the object, that is to say the larger region from which $p(i)$ is calculated. The outside region is simply the difference between the new mask and the object. The radius of the ball, which the object mask is dilated with, is decided by the user when the prototype is being created. In Figure 4.3 an example of the regions can be seen calculated for a segmentation problem of the aorta in 3-D. The images show a slice of the original object mask, the dilated mask and the difference between the masks, that is to say the region from which $p(s|i)$, $p(i)$ and $p(s^c|i)$ are calculated.

The idea of using an intensity function approximated from a set of example images is based on the assumption that all images have approximately the same distribution of intensities. Unfortunately this is not the case for all images, especially not for medical images such as MR images, since the images could have been differently normalised. On the other hand, if the images are supposed to represent the same type of object the true intensities is approximately the same in the images. If this assumption can be made, it is possible to make a new normalisation of the image intensities. After this normalisation, the distributions of pixel intensity should be approximately the same.

The normalisation of the histogram can be done in several ways. The

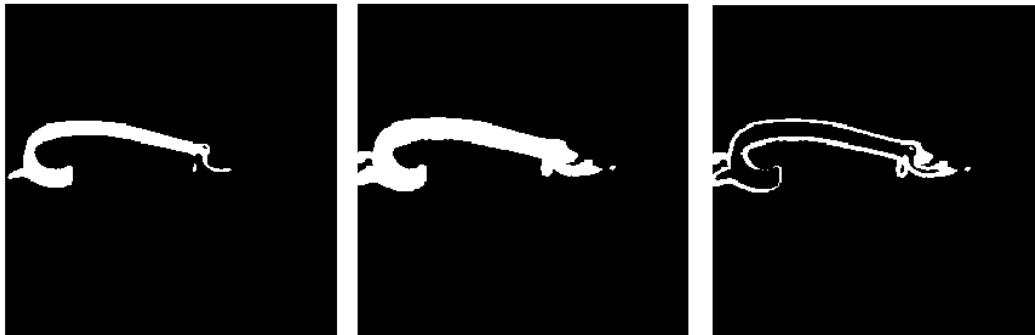


Figure 4.3: Example of what the regions, from which the histograms are calculated, can look like. The region for $p(s|i)$ (left), the region for $p(i)$ (middle) and the region for $p(s^c|i)$ (right).

easiest approach is to normalise the histogram to fill the entire interval $[0, 1]$ of intensities by subtracting the minimum intensity value and normalising with the maximum intensity value. The drawbacks of this approach are that it is possible that the intensities is saturated in one of the images but not in another and therefore the maximum or minimum intensities might not be corresponding true intensities. Another approach is to make histogram equalisation. After such an equalisation the histograms for all the example images will be very similar, unfortunately this equalisation can distort the true image intensities just to make the histograms resemble each other.

The adjustment which is implemented is to saturate the histograms highest and lowest intensities to, respectively, intensity 1 and 0. Intensities to be saturated are set as a percentage of the intensity range. For example the highest and lowest 2% of the pixel intensities are saturated. This percentage can be chosen when the prototype is created. In Figure 4.4 the histogram of four images is first shown without adjustment and then with a saturation of the 2% highest and lowest intensities. The example is from a segmentation of pig lungs. From the adjusted histograms of the example images, the probability distributions are calculated. The difference between the probability distribution for a pixel belonging to the object given the intensity value $p(s|i)$ and the probability distribution of not belonging to the object given the intensity, $p(s^c|i)$, is calculated for each example image. The intensity function is the normalised mean of the difference between $p(s|i)$ and $p(s^c|i)$ taken over the example images. The normalisation is done by dividing by the maximum value of the absolute value of the calculated mean. The intensity function is

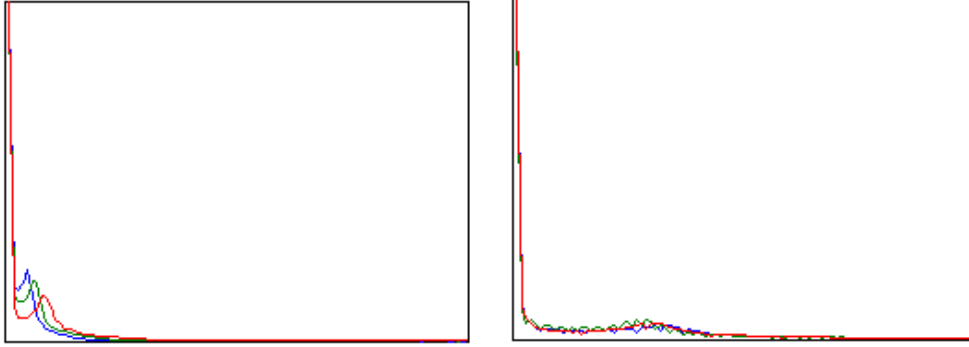


Figure 4.4: The histograms of four images. In the left image the histograms are unadjusted and the left image the histograms are shown with a saturation of the 2% highest and lowest intensities.

thus calculated as

$$h_{intensity} = \frac{\overline{p(s|i) - p(s^c|i)}}{\max(|\overline{p(s|i) - p(s^c|i)}|)} \quad (4.5)$$

The normalisation cause the intensity function to be in the interval $[-1, 1]$. The normalisation is done to make sure that the local speed image, which is generated from the intensity function, always has values within a given interval. In the previous implementation of the level set method in Segment this interval was $[-2, 2]$. Since it is desirable to a certain extent to retain similarities, between the implementation with prototype and the one without, the intensity function is also being multiplied with the factor 2. In Figure 4.5 the differences $p(s|i) - p(s^c|i)$ is shown for the five image examples of the aorta in the left image and in the right image the resulting intensity function $h_{intensity}$ is shown.

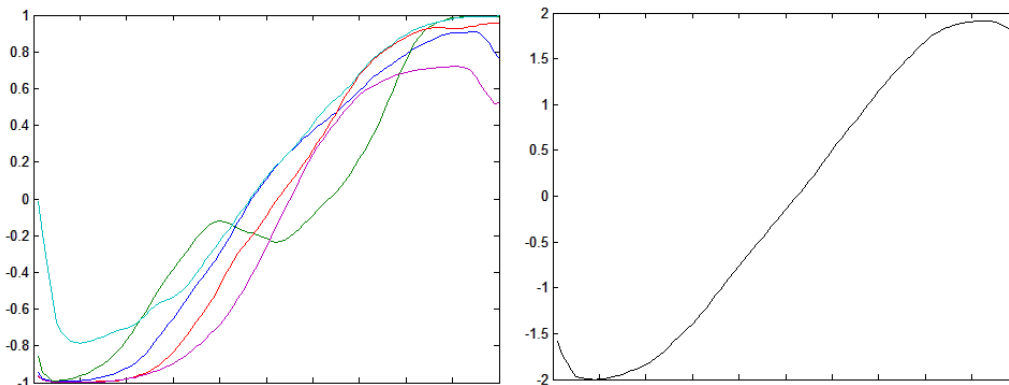


Figure 4.5: The differences $p(s|i) - p(s^c|i)$ is shown for five image examples of the aorta in the left image and in the right image the resulting intensity function is shown.

4.2.2 Alignment

To be able to compare shape information between the example images, and the images to be segmented, an alignment must be done. An alignment of the images is done to minimise the difference between the object shapes in rotation, scaling and translation. The alignment is done by an affine transformation.

Affine transformations consist of rotation, scaling, shearing and translation. The parameters in the transformation can be calculated or approximated from three or more landmarks. Three landmarks is the minimum number and the affine transformation is calculated as the exact solution for the three pairs of landmarks. If there are more than three landmarks in the images a least squares solution is calculated. It is desirable to use few landmarks since these landmarks will have to be set, not only in the example images but in each image that uses the prototype for segmentation as well. At the same time the number of landmarks must be enough to make the transformation unambiguous. For a 3D image it is necessary to use more landmarks than for a 2D image. If the image is time-resolved the same affine transformation will be applied to all timeframes.

The shape to align to is chosen as one of the objects, that is to say the landmarks in one of the example images. It is also possible to put some constraints on the affine transformation. The constraints implemented are to use an isotropic scaling or to put constraints on the degrees of rotation. The constraints and landmarks are saved in the prototype and used when

aligning the prototype to the landmarks in the image to be segmented.

4.2.3 Initialisation

To start the segmentation of an image with the level set method, there must be an initial curve, or surface. This initialisation is calculated from the aligned image examples.

The idea behind this implementation is that the initialisation should approximate a mean shape. After alignment the shape of the objects can be compared. The objects are represented as binary masks where white indicate that the pixel is inside the object and black indicate pixels outside the object.

Since the correspondences between pixels in the example images are not known it is not possible to calculate the mean position of each pixel or even the border pixels. Instead the initialisation is calculated from the binary masks smoothed with a Gaussian kernel. The mean of the smoothed masks, gives a probability map, which indicates the probability of a pixel belonging to the object given its position. The standard deviation of the Gaussian kernel is estimated by the user when the initialisation is calculated, and the variance is given in millimetres.

The probability map is an image that takes values in the interval $[0, 1]$ and from this image an image consisting of solely the values 0 and 1 should be created. To create this map the probability map must be thresholded. The threshold is calculated so that the number of pixels in the initialisation is approximately as many as the mean number of pixels in the objects. The number of pixels in the initialisation can also be chosen as a percentage of the mean number of pixels in the objects. This percentage is decided when the prototype is created. A map of initialisation pixels is created by thresholding the probability map. In Figure 4.6 a slice of a probability map for aorta and the same slice of the map of initialisation pixels can be seen. The initialisation is used to create the initial segmentation by aligning the initialisation stored in the prototype to the landmarks in the image to be segmented.

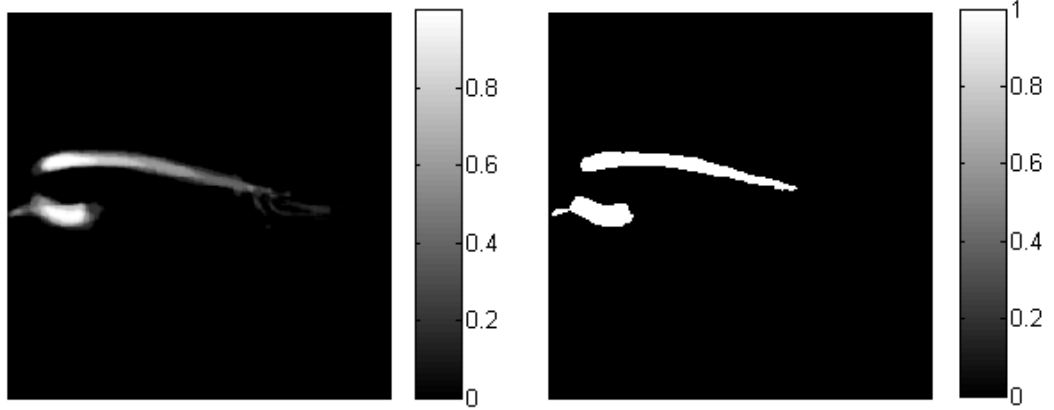


Figure 4.6: A slice of the probability map for an aorta (left) and the same slice of the initialisation (right).

4.2.4 Spatial map

A spatial map is used to introduce spatial *a priori* information into the speed law F used by the level set method. The spatial map is the sum of a distance map and a correction map. The distance map is a normalised map of the distance to the object. The correction map introduces information about where the segmentation had to be further spatially corrected if the local propagation speed I_{speed} was solely based upon intensity information.

The distances in from the objects in the example images are calculated by a fast marching algorithm. The speed law of the fast marching method is constantly one. The result from the fast marching method, M_{fm} , has values from zero inside the object, that is to say the initialisation for the fast marching method, to the maximum distance between a pixel and the object. The fast marching method is used to calculate the map, M_{fm} , for all the aligned example images. The distance map is the normalised mean of the fast marching maps taken over the example images. The normalisation is done by dividing by the maximum of the calculated mean map. The distance map thus has values in the interval $[0, 1]$ and is calculated as

$$M_{distance} = \frac{\overline{M_{fm}}}{\max(\overline{M_{fm}})}, \quad (4.6)$$

where $\overline{M_{fm}}$ is the mean over the fast marching maps taken over the example images. The correction map introduces information about where the segmentation has to be further spatially corrected. A fast level set segmentation is

done for each example image using a speed law based upon solely the intensity information. The resulting segmentations are compared to the manual segmentation of the object. The spatial map is based upon the difference between the segmentations represented as binary masks. The difference is modified so that no corrections are done in the pixels closest to the object border. These modified differences between the masks are smoothed with a Gaussian kernel. The correction map is the smoothed mean of the differences. The correction map is in the interval $[-1, 1]$. Figure 4.7 shows an example of the segmentations of the aorta based on solely the intensity function in the top left, which is compared to the manual segmentations, shown in the top right, and finally the same slice of the resulting correction map is shown in the bottom. The correction map has high values where it is common that the segmentation fails if solely intensity information is used, such as in this example where the heart is falsely included in the segmentation.

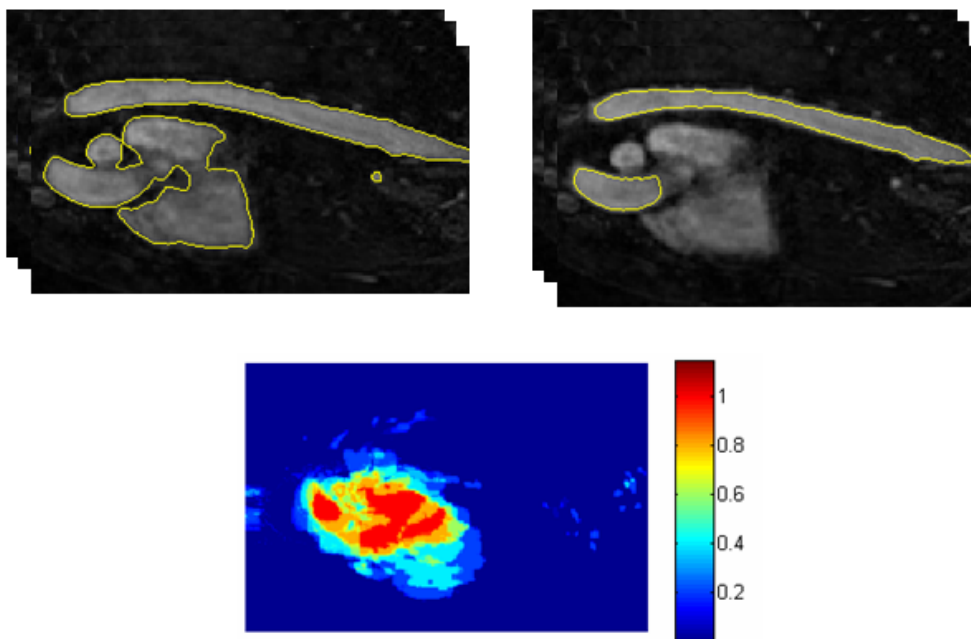


Figure 4.7: The image in the top left shows a slice of the segmentations based on solely intensity function, in the top right the manual segmentations are shown and in the bottom an image of the same slice of the correction map is shown.

The spatial map is the sum of the distance map and the correction map and can hold values in the interval $[-1, 2]$. To be able to adjust the influence of the distance map, compared to the correction map in the spatial map, the

distance map can be transformed to be another monotonic function, than a linear function of the distances. Thus the spatial map is calculated as

$$M_{spatial} = M_{correction} + f(M_{distance}) , \quad (4.7)$$

where f is a monotonic function. In this implementation f has been tested to be the function $f(M) = M$, $f(M) = \sqrt{M}$ and $f(M) = M^2$. In Figure 4.8 an example of the different functions f can be seen on a slice of a distance map for an aorta. The advantage of having the function $f(M) = \sqrt{M}$ is that a lot of the surrounding regions will have an immediately lowered speed but the drawback is that it also lowers the speed very close to the initial curve, that is to say this function is good to use when the objects are very similar.

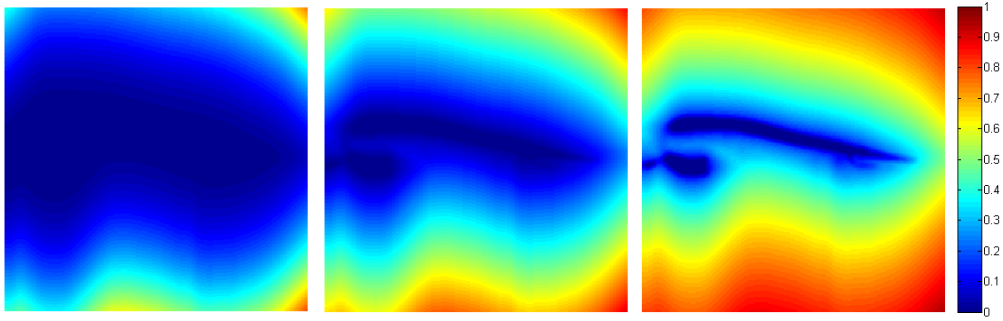


Figure 4.8: An example of the distance map with different functions from left to right $f(M) = M^2$, $f(M) = M$ and $f(M) = \sqrt{M}$. The images show a slice of the distance map for an aorta.

In the speed image the spatial map is multiplied with a parameter λ . This parameter λ decides how much the spatial map will influence the local speed image I_{speed} . If the parameter λ and the function f is chosen manually it should be chosen so that as few pixels as possible outside the wanted object has a positive speed value without assigning pixels inside the wanted object a negative speed value. The parameter λ and the function f can also be chosen by optimisation described in the next section.

4.2.5 Parameter optimisation

The speed law F in the level set method consists of the local speed image I_{speed} , the curvature and the gradient. The influence of the curvature and gradient is decided by the parameters α and β . These parameters should be set so that the propagation of the level set method stops at the edges in the

image and the segmentation becomes smooth. The local speed image I_{speed} can also be adjusted with the parameter λ , which decides the influence of the spatial map and the function f , which decides the influence of the distances in the spatial map.

The parameters α , β and λ can be adjusted in the tool for general segmentation in Segment and the function f has to be set when creating the spatial map and is as default chosen to be the linear function. The values of α , β and λ and the function f can be set by optimisation.

The optimisation is done by segmenting the example images and comparing these segmented objects to the true object contours given beforehand. In the optimisation the following error measurement is used

$$\epsilon_{V,abs} = \frac{V_{|true-segmented|}}{V_{true}}, \quad (4.8)$$

where V_{true} is the volume of the true object and $V_{|true-segmented|}$ is the volume of the absolute difference between the two objects. This difference is calculated as the absolute value of the difference between the binary masks of the object. The error measurement $\epsilon_{V,abs}$ takes into account both the error of the segmentation where it is too big and where it is too small. The error measurement is also normalised so that it is expressed as a percentage of the true object volume. The parameters are chosen to be the parameters which minimises the mean of the error measurement $\epsilon_{V,abs}$ for the images in the set.

The method used for optimising the parameters α , β and λ is the Nelder-Mead simplex method [13]. The function f is chosen by exhaustive search. The function is set before optimisation of the other parameters and function f , and the other parameters, are chosen to be the one which generated the smallest segmentation error. In this implementation the function f tested is $f(M) = \{\sqrt{M}, M, M^2\}$. The optimisations are done on the set of images from which the intensity function and spatial map was built, to be able to build the prototype on as few images as possible.

The optimisation typically takes between 4 and 10 hours on an ordinary PC depending on the application.

4.2.6 Generation of the prototype

Not only the general segmentation tool in Segment has been adjusted to handle prototype based image segmentation but a user friendly tool for generation of prototypes has also been developed, see Figure 4.9 for an example of the graphical user interface. In this section all the parameters used to generate the prototype is described. Besides the intensity function, spatial map and initialisation the prototype contains other parameters used for both

generation of the prototype and the segmentation, and some parameters are used solely for the generation of the prototype.

First a brief summary of the parameters which are used to do segmentation with the prototype.

- *Saturation of intensities*

Here the values of the lowest and highest percentage to saturate are set manually and stored in the prototype. The values are set so that the intensity histograms of the example images resemble each other and the saturation is as small as possible. A default of 2 % is suggested.

- *Landmarks*

The landmarks should be at least three and they should be set so that they easily can be reproduced in the images to segment with the prototype. To generate unambiguous transformations the landmarks shall be set far away from each other and to have a shape defining transformation the landmarks shall be set in places which are relatively stable.

- *Alignment constraints*

As a constraint it is almost always good to have uniform scaling but the rotation constraints shall only be used if it is clear that the transformation is unambiguous in the set of example images and the images to segment will be rotated in a similar way as the example images.

- α, β, λ

The parameters in the speed law which are set by optimisation.

- *Radius in the fast level set method*

This radius is used to define the neighbourhood in which the curvature is estimated. The radius shall be set so that it is as large as possible to be able to do more accurate approximations and at the same time be less than the smallest structure in the object to segment.

- *Smoothing radius*

After segmentation the object is smoothed. This radius can be set to be the same as the radius in the fast level set method since this radius also shall be less than the smallest structure in the object.

Secondly there are parameters which are only used for the generation of the prototype.

- *Dilation radius*

The parameter defining the regions from which the histograms are calculated and the intensity function is based upon. After setting this radius the intensity function is calculated and the mapping to intensities in the example images can be seen. The radius shall be adjusted so that the intensity mapping is positive inside the objects and negative outside. If the intensity function is low the dilation radius should be decreased and on the contrary the dilation radius should be increased if the intensity function is high where it ought not to be.

- *Standard deviation of object shape*

This parameter is used to smooth the probability map from which the initialisation is defined. The parameter is application dependent on how much the delineation differs. The standard deviation is given in millimetres. When setting the standard deviation the initialisation is calculated and shown. The standard deviation can be set by trial and error so that the initialisation shape is similar to the object shape. If the delineation differs substantially between the objects the standard deviation should be increased.

- *Percentage of mean volume*

The default of the initialisation is that its volume is as close as possible to the mean of the object volumes in the set of example images. In some application it can, however, be preferable to have an initialisation which is smaller and the percentage can then be lowered. The lowered percentage might be better if the object to segment is very close both spatially and in intensity to another object.

- *Function f*

The distance map is by default set to be a linear function but it can also be optimised as described in the previous section.

- *Standard deviation of correction map smoothing*

This parameter can by default be set to the same value as the smoothing radius used after segmentation.

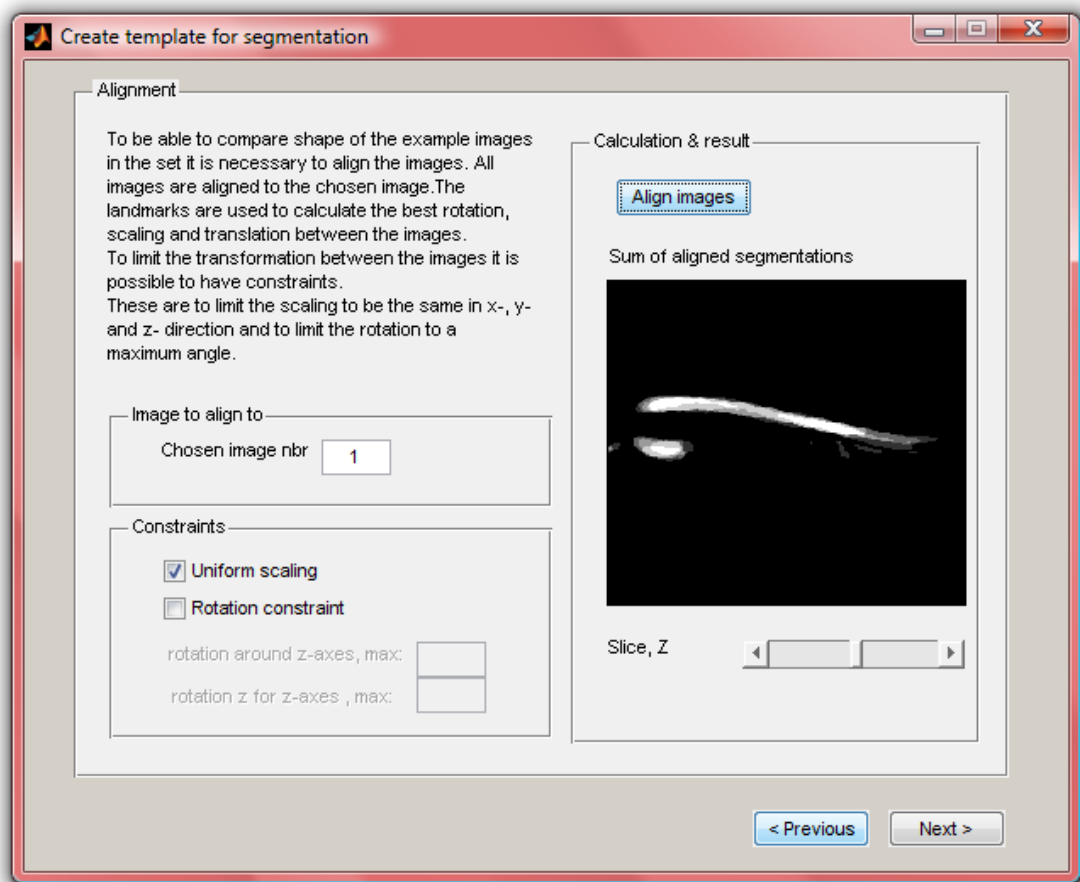


Figure 4.9: Example of the graphical user interface of the tool for creating a prototype (alignment step).

4.3 Segmentation

To segment an image the landmarks must be set in the image, this is, however, the only manual interaction needed. The landmarks are used to align the spatial map and initialisation in the prototype to the image to be segmented. The image is also intensity adjusted by saturation of the intensities defined in the prototype. After alignment and intensity adjustment the local propagation speed is calculated as

$$I_{speed} = G(I, h_{intensity}) - \lambda M_{spatial} , \quad (4.9)$$

where I is the intensity adjusted image, G is a conversion function of the intensities in the image to the values in the intensity function $h_{intensity}$, $M_{spatial}$ is the aligned spatial map and λ is the weight parameter stored in the prototype. The gradient of the image I is also calculated before segmentation with the level set method. The gradient is calculated by central differences and the magnitude of the gradient is also smoothed so that the influence of noise in the image is lowered. After calculation of the speed image and gradient the segmentation can start from the aligned initialisation, with the usage of the speed law

$$F = I_{speed} \frac{1}{1 + \alpha |\nabla I|} + \beta \kappa , \quad (4.10)$$

where κ is the curvature estimated in the level set method and α and β are the weight parameters stored in the prototype. The segmentation is performed with an implementation of the fast level set method proposed by Nilsson and Heyden [12].

The prototype based image segmentation is implemented in Segment and an example of the graphical user interface is shown in Figure 4.10.

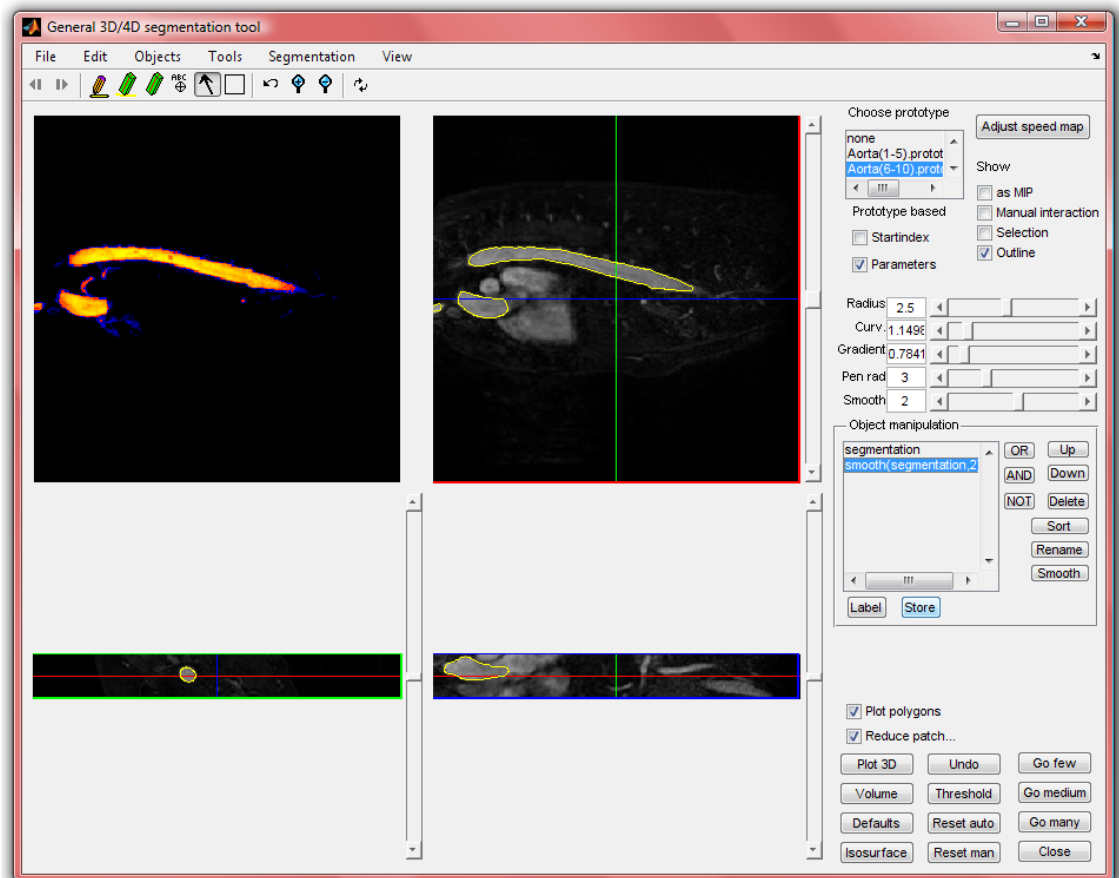


Figure 4.10: Image showing the graphical user interface of the general segmentation tool in which segmentation can be done with the use of a prototype.

4.4 Validation

The prototype based image segmentation proposed in this thesis should be able to be built upon few images and be general. This section will describe how to validate the method. The method is tested on the following five applications, segmentation of the aorta, left ventricle blood volume, left ventricle wall volume, right ventricle blood volume and pig lungs in MR images. These applications have been chosen since they differ in shape, intensity and complexity. Furthermore, manual segmentations for these images were available.

The validation will not only be done by visual inspection of the segmentations but quantitative measurements will be performed. Both the accuracy of the method will be measured and segmentations will also be compared to investigate the sensitivity of the method. How to evaluate the accuracy of the segmentation will be described in Section 4.4.1 and how to evaluate the sensitivity of the method will be described in Section 4.4.2.

4.4.1 Segmentation results

The accuracy of the method will be evaluated both as a percentage volumetric error, ϵ_V and as an absolute distance error, $\epsilon_{mm,abs}$. The measurement ϵ_V is acquired by calculating the volume of the prototype based segmentation and the manual segmentation, here referred to as the true segmentation. The difference between these volumes are then normalised with the volume of the true segmentation to be able to compare the error between different images. The error measurement is calculated as

$$\epsilon_V = \frac{V_{segmented} - V_{true}}{V_{true}}. \quad (4.11)$$

This error measurement is chosen since it is the most physiologically interesting measurement in most applications. In the optimisation another error measurement

$$\epsilon_{V,abs} = \frac{V_{|segmented-true|}}{V_{true}} \quad (4.12)$$

was used. These two error measures differ. The measure $\epsilon_{V,abs}$ takes into account every pixel which is segmented in the wrong way. This measurement is better when optimising to make sure that the segmentation is situated in the right place and not only has the right volume. Since the segmentation is done with those optimised parameters the segmentation will not be in a location very different from the true location and it is feasible to use the physiological interesting error measurement ϵ_V .

The measurement $\epsilon_{mm,abs}$ is interesting since it shows how much the border of the segmentation differs between the generated segmentation and the true segmentation measured in millimetres. This measurement is closer related to the measurement $\epsilon_{V,abs}$. To calculate this measurement the difference volume, $V_{|segmented-true|}$, and the surface area of the true segmentation, A_{true} is used. The measurement is calculated as

$$\epsilon_{mm,abs} = \frac{V_{|segmented-true|}}{A_{true}}. \quad (4.13)$$

To calculate the surface area of the true object, the binary mask of the true object is dilated and eroded with a ball of radius 1 pixel. This is done since the edge of an object can be calculated as either the dilation of the object minus the object to get an outer edge or by subtracting the eroded object from the object to get an inner edge. The surface area is then calculated as half the difference of the eroded object mask and the dilated object mask to get an average of the inner and outer edge.

4.4.2 Method sensitivity

In order to investigate the sensitivity of the method three tests will be performed. To investigate if the method can be built on only a few segmented images it is important to ensure that the segmentation result does not depend on which images the prototype was built upon. This is tested by building two prototypes on different image sets. The two prototypes will then be used on yet another set of images to compare the segmentation errors.

The second test is done to investigate if it makes any difference if the parameters in the prototype are optimised on the same set as the intensity function, spatial map and initialisation are calculated from. To investigate this the optimisation of the parameters is done both with the set used to generate the rest of the prototype and on another set of images. The two prototypes are then used for segmentation of another set of images and the segmentation errors are compared.

Finally it is also desirable to compare how interaction sensitive the method is. This is done by letting one person set the landmarks in a set of images twice and the segmentation errors are then measured.

The error measurement shall compare the two segmentations and the measurement shall preferably be given so that it can be related to the error measurement used to get quantitative results for the segmentations, ϵ_V . This is done by measuring the volume of the two segmentations, and normalising the difference with the volume of the true segmentation. The new error

measurement, $\epsilon_{V,diff}$ is calculated as

$$\epsilon_{V,diff} = \frac{V_{segmented,1} - V_{segmented,2}}{V_{true}} , \quad (4.14)$$

where $V_{segmented,1}$ and $V_{segmented,2}$ is the volume of segmentation number one respectively number two and V_{true} is the volume of the true segmentation. This new error measurement is actually equal to the difference of the error measurement ϵ_V for the segmentations

$$\epsilon_{V,diff} = \epsilon_{V,1} - \epsilon_{V,2} . \quad (4.15)$$

Chapter 5

Results

The segmentation results of the prototype based image segmentation method will be shown for aorta, left ventricle wall volume, left ventricle blood volume, right ventricle blood volume and segmentation of pig lungs in Magnetic Resonance Imaging, (MRI). First some visual examples are given, and in Section 5.1 quantitative results are given.

Visual examples of the segmentations, manual compared to prototype based segmentations, are given in Figure 5.1- 5.5.

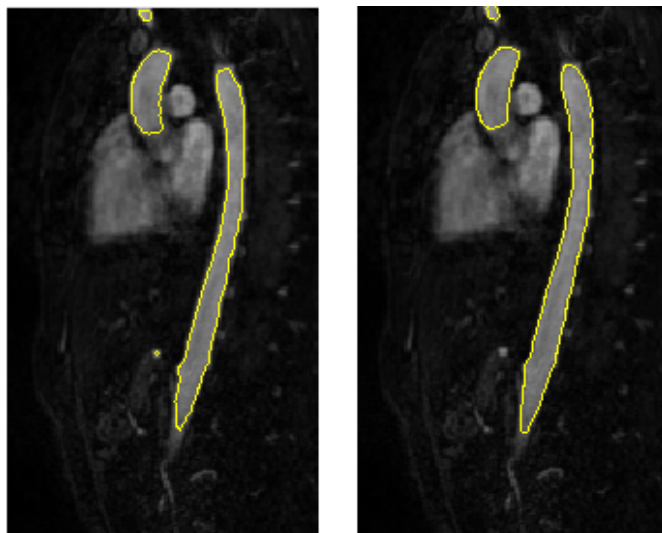


Figure 5.1: A slice of the segmentation of the aorta in an MR image. The left image is the manual segmentation and the right image is the prototype based image segmentation.

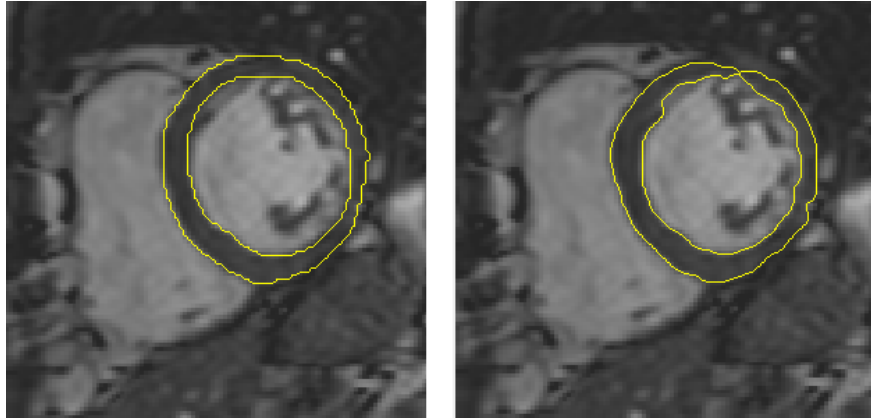


Figure 5.2: A slice of the segmentation of the left ventricle wall volume, in a short axis view in an MR image. The left image is the manual segmentation and the right image is the prototype based image segmentation.

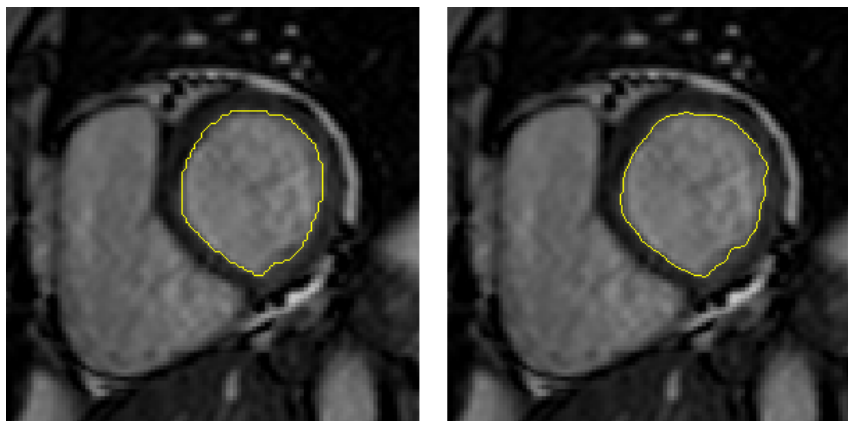


Figure 5.3: A slice of the segmentation of the left ventricle blood volume, in a short axis view in an MR image. The left image is the manual segmentation and the right image is the prototype based image segmentation.

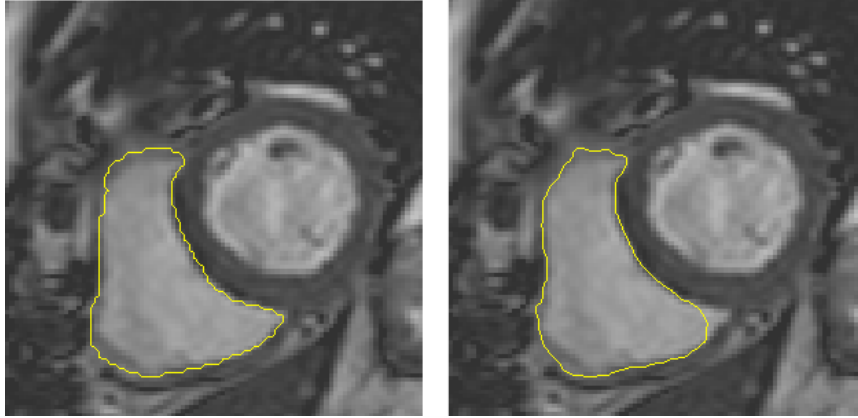


Figure 5.4: A slice of the segmentation of the right ventricle blood volume, in a short axis view in an MR image. The left image is the manual segmentation and the right image is the prototype based image segmentation.

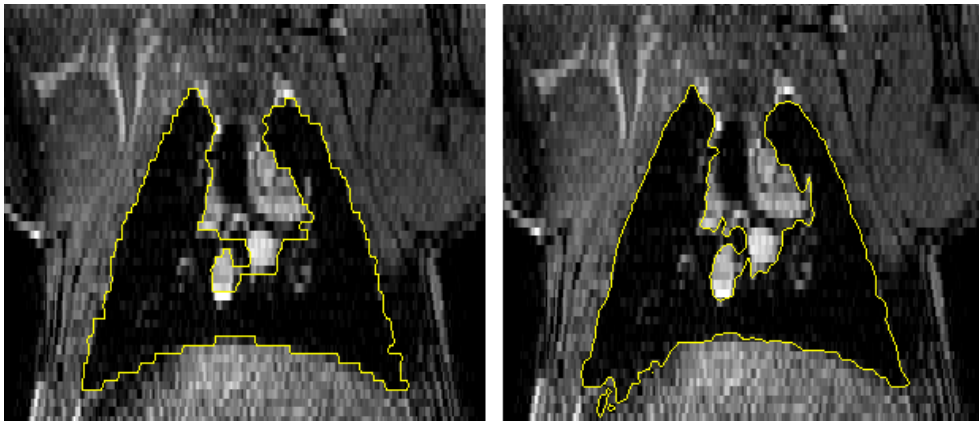


Figure 5.5: A slice of the segmentation of pig lungs in an MR image. The left image is the manual segmentation and the right image is the prototype based image segmentation.

An example of segmentation initialisation and the manual segmentation can be seen for the aorta in Figure 5.6. It can clearly be seen that the initialisation is quite close to the manual segmentation. A main advantage

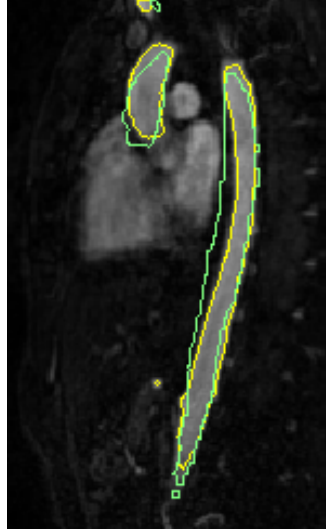


Figure 5.6: A slice of the initialisation of the aorta segmentation in a MR image. The initialisation, (green line) is close to the manual segmentation, (yellow line).

with the method has shown to be the spatial map. In Figure 5.7 a local speed image solely based upon the intensity can be seen, that is to say the speed image as

$$I_{speed} = G(I, h_{intensity}) , \quad (5.1)$$

and the speed image with the optimised influence of spatial map, that is to say with the speed image

$$I_{speed} = G(I, h_{intensity}) - \lambda M_{spatial} . \quad (5.2)$$

Figure 5.8 shows a slice of the resulting segmentations and Figure 5.9 shows a 3D visualisation of the resulting segmentations using the speed images in Equation 5.1 and 5.2. When the spatial map is not used the segmentation falsely includes the heart and when the segmentation is done with the usage of the optimised spatial map the segmentation only includes the aorta as it should.

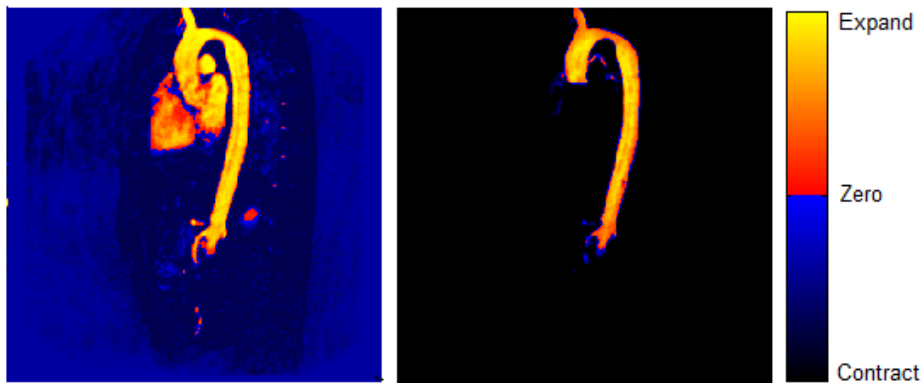


Figure 5.7: A slice of the speed image I_{speed} without a spatial map (left) and with the optimised spatial map (right) for the segmentation of the aorta in a MR image.

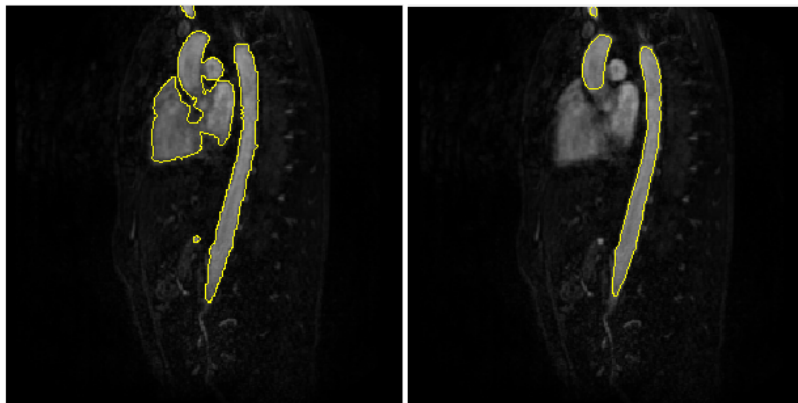


Figure 5.8: A slice of the segmentation of an aorta without the usage of a spatial map, (left) and with the usage of the optimised spatial map (right). It can be seen how the heart is falsely included in the segmentation when the spatial map is not used.

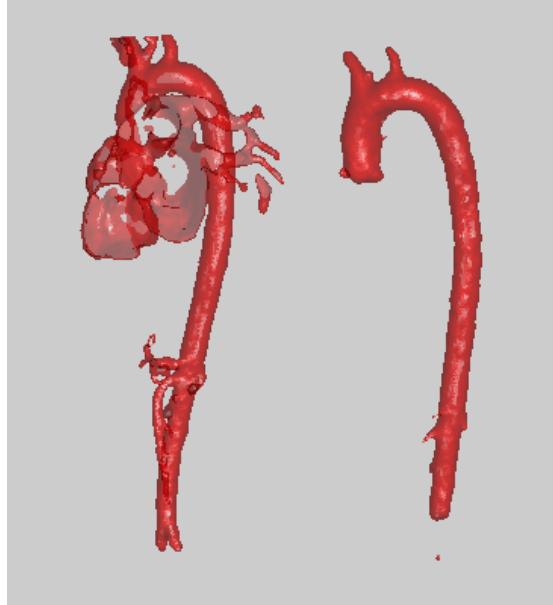


Figure 5.9: A 3D visualisation of the resulting segmentation without the usage of a spatial map (left) and with the usage of a spatial map (right). It can be seen how the heart is falsely included in the segmentation when the spatial map is not used.

5.1 Quantitative results

The quantitative results are shown in tables sorted by application. First three tables showing results for the aorta followed by two tables with result for the segmentation of Left Ventricle Wall Volume, (LVWV), four tables with results for the Left Ventricle Blood Volume, (LVBV), and finally three tables showing result for segmentation of the Right Ventricle Blood Volume, (RVBV). For the segmentation of pig lungs quantitative test has not been performed since the number of images was too few.

Result for the first test of method sensitivity, comparing segmentation built upon one set of images compared to segmentation with a prototype built on another set of images, can be seen in Table 5.4 for the left ventricle wall volume, in Table 5.6 for the left ventricle blood volume and in Table 5.10 for the right ventricle blood volume.

Results of the second test of method sensitivity, segmentation with a prototype built on image 1 to 5 and optimised on image 6 to 10 compared to segmentation with a prototype built and optimised on images 1 to 10, is

shown in Table 5.6 for the left ventricle blood volume and in Table 5.10 for the right ventricle blood volume.

Results of the third test of method sensitivity, segmentation with landmarks set twice in the images to be segmented, is shown in Table 5.3 for the aorta and in Table 5.9 for the right ventricle blood volume.

In Tables 5.2, 5.5, 5.8 and 5.12 the segmentation results for the applications aorta, left ventricle wall volume, left ventricle blood volume and right ventricle blood volume is shown measured in millimetres for all the different prototypes tested.

Aorta, error ϵ_V [%]

image #	Prototype	
	1-5	6-10
1	-	5,6
2	-	5,7
3	-	11,1
4	-	-1,0
5	-	-0,4
6	-4,6	-
7	5,6	-
8	-5,6	-
9	-4,1	-
10	-22,7	-
mean	-6,3	4,2
std	10,2	5,0

Table 5.1: Quantitative results, ϵ_V , for segmentation of the aorta. Percentage volumetric segmentation error of images 1 to 5 and 6 to 10 segmented with the use of prototypes generated from images 6 to 10 and 1 to 5.

Aorta, error $\epsilon_{mm,abs}[mm]$

image #	Prototype	
	1-5	6-10
1	-	0,58
2	-	0,65
3	-	0,63
4	-	0,53
5	-	0,50
6	0,52	-
7	0,44	-
8	0,65	-
9	0,48	-
10	0,79	-
mean	0,58	0,58
std	0,14	0,06

Table 5.2: Quantitative results for segmentation of the aorta. Segmentation error measured in millimetres for images 1 to 5 and 6 to 10 segmented with the use of prototypes generated from images 6 to 10, and 1 to 5.

Aorta, error $\epsilon_V[\%]$

image #	Landmarks		$\epsilon_{V,diff}$
	first time	second time	
1	5,6	-5,6	11,2
2	5,7	3,1	2,6
3	11,1	4,7	6,4
4	-1,0	-8,5	7,5
5	-0,4	-1,3	0,9
mean	4,2	-1,5	5,7
std	5,0	5,6	4,1

Table 5.3: Comparison of segmentation results, ϵ_V , for the aorta using a prototype built upon images 6 to 10 and landmarks set twice in images 1 to 5.

LVWV, error ϵ_V [%]

image #	Prototype		$\epsilon_{V,diff}$
	1-5	6-10	
11	-8,6	-21,3	12,7
12	-10,4	-14,4	4,0
13	-24,8	-38,0	13,2
14	-22,1	-30,6	8,5
15	-36,7	-33,4	-3,3
16	-18,8	-36,1	17,4
17	-19,9	-37,8	17,9
18	-64,4	-65,3	0,9
19	-42,0	-45,3	3,3
20	-4,9	-22,4	17,5
mean	-25,3	-34,5	9,2
std	18,1	14,3	7,7

Table 5.4: Comparison of segmentation results, ϵ_V , for the left ventricle wall volume with prototype built upon images 1 to 5 and prototype built upon images 6 to 10.

LVWV, error $\epsilon_{mm,abs}$ [mm]

image #	Prototype	
	1-5	6-10
11	0,45	0,53
12	0,48	0,45
13	0,43	0,48
14	0,47	0,45
15	0,47	0,45
16	0,81	0,80
17	0,56	0,52
18	0,84	0,80
19	0,64	0,63
20	0,48	0,55
mean	0,56	0,57
std	0,15	0,14

Table 5.5: Quantitative results, $\epsilon_{mm,abs}$, for segmentation of the left ventricle wall volume. Segmentation error measured in millimetres for images 11-20 segmented with the use of prototypes generated from images 1 to 5 and 6 to 10.

LVBV, error ϵ_V [%]

image #	Prototype		$\epsilon_{V,diff}$
	1-5	6-10	
11	-10,5	-20,1	9,6
12	-16,3	-23,8	7,5
13	5,2	-3,4	8,7
14	15,9	1,1	14,8
15	-11,8	-19,7	8,0
16	-10,2	-13,5	3,2
17	-14,2	-24,9	10,7
18	5,1	-7,0	12,1
19	-5,9	-11,6	5,8
20	-19,7	-28,8	9,1
mean	-6,2	-15,2	8,9
std	11,4	9,9	3,2

Table 5.6: Comparison of segmentation results, ϵ_V , for the left ventricle blood volume with prototype built upon images 1 to 5 and with prototype built upon images 6 to 10.

LVBV, error ϵ_V [%]

image #	Prototype		$\epsilon_{V,diff}$
	1-5,6-10	1-10	
11	-13,4	-17,6	4,2
12	-19,6	-23,1	3,6
13	-8,5	-16,6	8,1
14	6,7	1,9	4,8
15	-15,0	-18,3	3,2
16	-15,5	-19,0	3,5
17	-20,9	-26,2	5,4
18	-13,5	-14,4	0,9
19	-10,3	-13,0	2,7
20	-23,0	-25,8	2,8
mean	-13,3	-17,2	3,9
std	8,4	8,1	1,9

Table 5.7: Comparison of segmentation results, ϵ_V , for the left ventricle blood volume when using a prototype built upon images 1 to 5 and optimised on images 6 to 10 compared to using a prototype built and optimised on images 1 to 10.

image #	Prototype			
	1-5	6-10	1-5,6-10	1-10
11	0,63	0,64	0,65	0,50
12	0,47	0,65	0,48	0,51
13	0,58	0,58	0,35	0,43
14	0,73	0,56	0,61	0,43
15	0,31	0,50	0,36	0,45
16	0,82	0,87	0,83	0,69
17	0,45	0,57	0,45	0,57
18	0,74	0,56	0,53	0,78
19	0,52	0,49	0,52	0,38
20	0,46	0,69	0,53	0,61
mean	0,57	0,61	0,53	0,53
std	0,16	0,11	0,14	0,13

Table 5.8: Quantitative results, $\epsilon_{mm,abs}$, for segmentation of left ventricle blood volume. Segmentation error measured in millimetres for images 11 to 20 segmented with the use of prototypes generated from images 1 to 5, 6 to 10, generated from images 1 to 5 and optimised on images 6 to 10 respectively generated from images 1 to 10.

image #	Landmarks		$\epsilon_{V,diff}$
	first time	second time	
11	-10,54	-10,59	0,06
12	-16,28	-15,53	-0,74
13	5,21	4,95	0,26
14	15,88	11,57	4,31
15	-11,75	-10,96	-0,79
mean	-3,50	-4,11	0,62
std	13,53	11,70	2,12

Table 5.9: Comparison of segmentation results, ϵ_V , for the left ventricle blood volume using a prototype built upon images 1 to 5 and landmarks set twice in images 11 to 15.

RVBV, error ϵ_V [%]

image #	Prototype		$\epsilon_{V,diff}$
	1-5	6-10	
11	-43,8	-33,0	-10,8
12	-27,0	-21,1	-5,9
13	-19,5	-12,0	-7,6
14	-11,1	-11,5	0,4
15	-18,4	-14,5	-3,9
16	-14,8	-12,2	-2,5
17	-32,3	-23,4	-9,0
18	-39,3	-44,7	5,4
19	-17,5	-25,2	7,7
20	-15,2	-17,0	1,7
mean	-23,9	-21,5	-2,4
std	11,2	10,7	6,2

Table 5.10: Comparison of segmentation results, ϵ_V , for the right ventricle blood volume with prototype built upon images 1 to 5 and with prototype built upon images 6 to 10.

RVBV, error ϵ_V [%]

image #	Prototype		$\epsilon_{V,diff}$
	1-5,6-10	1-10	
11	-28,2	-42,5	14,3
12	-18,4	-29,6	11,1
13	-7,6	-21,7	14,2
14	-4,7	-11,9	7,3
15	-7,7	-22,8	15,0
16	1,0	-25,4	26,4
17	-10,3	-37,1	26,8
18	-29,2	-49,7	20,5
19	-7,9	-27,4	19,5
20	-5,3	-0,3	21,9
mean	-11,8	-26,8	17,7
std	10,1	14,4	6,4

Table 5.11: Comparison of segmentation results, ϵ_V , for the right ventricle blood volume when using a prototype built upon images 1 to 5 and optimised on image 6 to 10 compared to a prototype built and optimised upon on images 1 to 10.

RVBV, error $\epsilon_{mm,abs}$ [mm]

image #	Prototype			
	1-5	6-10	1-5,6-10	1-10
11	0,89	0,51	0,85	0,69
12	0,57	0,51	0,47	0,56
13	0,46	0,38	0,50	0,52
14	0,62	0,63	0,62	0,62
15	0,77	0,71	0,72	0,73
16	0,70	0,65	0,80	0,63
17	0,76	0,69	0,93	0,85
18	1,36	1,02	1,34	1,29
19	1,01	0,63	1,04	1,03
20	0,57	0,50	0,62	0,56
mean	0,77	0,62	0,79	0,75
std	0,26	0,17	0,27	0,25

Table 5.12: Quantitative results, $\epsilon_{mm,abs}$, for segmentation of right ventricle blood volume. Segmentation error measured in millimetres for images 11-20 segmented with the use of prototypes generated from images 1-5, 6-10, generated from images 1-5 and optimised on images 6-10 respectively generated from images 1-10.

Chapter 6

Discussion & Conclusions

In this chapter the results will be discussed, and in the next chapter future work will be suggested based upon discussion and conclusions in this chapter. The discussion will go through the quantitative and visual results for the segmentation of the aorta, left ventricle wall volume, left ventricle blood volume, and the right ventricle blood volume. Finally, conclusions will be drawn from the discussion of these four applications.

The proposed prototype based image segmentation incorporates *a priori* information to a level set method from a set of few example images. The *a priori* information is extracted as an intensity function, initialisation, a spatial map and optimised parameters, all stored into a prototype. The prototype is generated once for each desired application and the information in the prototype is then used in a general segmentation method, the level set method. The segmentation of an image can then be done with the only manual interaction being to set a number of landmarks in the image, typically 3 or 4. The setting of landmarks is a very small manual interaction compared to doing manual segmentation of the image which may take 20 minutes or more.

A big advantage of the method is the use of a spatial map which constrains the segmentation. The intensity function defines how to map intensities in the image to speed values. This intensity function is based upon an intensity adjustment and statistics about the pixel intensities on the inside and outside of the object to be segmented. The spatial map contains both a distance map and a correction map to constrain the segmentation both far away and in locations where the intensities is mapped to speed values which would cause an unconstrained level set algorithm to produce result that would substantially differ from a manual segmentation. The initialisation and the optimised parameters also constrain the segmentation. With

an initialisation approximating the object shape the gradient and curvature influence, decided by the optimised parameters, can be used in a sensible way. The gradient is used to lower the speed near the edges and since the nearest edge from the initialisation most probably is the edge of the object the use of the gradient will most likely constrain the segmentation to be the object searched for. The use of the curvature is also more sensible to use when the initialisation is close to the object searched for since the use of curvature makes sure that the segmentation becomes smooth but it can also prevent leakage of the segmentation into other structures in the image.

The method is general in its appearance to be able to use it on different image applications.

6.1 Segmentation of aorta

The segmentation of the aorta is both visually and quantitatively very satisfying. The quantitative error can unfortunately not be compared to manual segmentation error since no accurate enough ground truth exists and comparison of manual segmentations of the aorta has not been done during this thesis, nor has a study been found on the subject. For the aorta also the volume is hard to define since the volume depends on how much of the aorta is included. The aorta is divided into smaller arteries and finally capillaries which supply the whole body with blood. For the aorta it is more important to capture the shape of the object, since simulation of the blood flow will be done using this segmentation approach. Probably the mean error measured in millimetres is the most important parameter. The mean error measured in millimetres is as small as 0.58 millimetres which should be compared to a pixel resolution of 1.25 millimetres. The segmentation is visually very satisfying and it is hard to say which one of the segmentations is the most accurate, the manual or the prototype based.

The volumetric error measurement shows slightly more variation and in Table 5.1 it can be seen that images number 3 and 10 have a slightly larger respectively a significantly larger segmentation error than the other images. In image 3 the segmentation error mainly reflects how far down the aorta has been segmented, and this seems to have been arbitrarily chosen in the manual segmentations which the prototype and especially the spatial map has been built upon. In image 3 the aorta is visible further down than in the other images, however, in this application this is not of a mayor concern, since the interesting part of the aorta is the part closest to the heart and the aortic arch. This error might not arise when the manual segmentations include the whole object.

The segmentation error in image 10 can not be explained in the same way as the segmentation error of image 3, instead this error seems to arise from a poor intensity adjustment or intensity mapping, since some bright pixels inside the aorta are not bright enough to be mapped to positive speed values. This poor mapping might arise from the fact that there is no smoothing of the intensity function so that a deviation from the intensities in the example images is not included in the intensity function or from a poor adjustment of the intensities. Adjustment of intensities has to be done for MR images since they are normalised in different ways. How to improve the adjustment is further discussed in the next chapter, future work.

The mean volumetric segmentation error of -6% and 4% is, however, satisfying and the segmentation error is even more satisfactory when noticing that the absolute segmentation error measured in millimetres has a mean of 0.58 millimetres for both prototypes and set of images. Even the sensitivity to landmark setting for the aortic application seems to be low since the error of the two different landmark settings is in the same order of magnitude.

6.2 Segmentation of left ventricle wall volume

The segmentation of the left ventricle wall volume has few error measurement within the range of $\pm 10\%$ and it can also be seen that all prototype based segmentations is smaller than the manual segmentations. By visual inspection of the segmentations it is clear that the medical expertise doing manual segmentations use anatomical *a priori* information on that the left ventricle is a nearly circular structure in almost all slices and coherent. The prototype based segmentation is not coherent either as a consequence of poor intensity mapping or as a consequence of smoothing. The smoothing unfortunately makes thin structures disappear and where the wall is very thin the wall breaks into two pieces.

The underestimation of the wall volume might depend on the fact that the optimisation of parameters are done with regard to the error measurement $\epsilon_{V,abs}$. This error measurement does take into account all falsely segmented and falsely excluded pixels but the total error of falsely segmenting pixels within a distance of one pixel outside the object is greater than the error of falsely including pixels within a distance of one pixel inside the object. This fact might encounter for the underestimation of the volume. It is also clear when looking at the segmentation error $\epsilon_{mm,abs}$ that a small mean error of 0.57 and 0.58 millimetres in the border of the object is of great importance. A fairly large volumetric error measurement is given even though the absolute mean error measured in millimetres is on sub pixel level. This application is

probably the most challenging segmentation application tested in this thesis.

6.3 Segmentation of left ventricle blood volume

Segmentation of the left ventricle blood volume have been done for 10 images with four different prototypes and for five of these images landmarks have been set twice and the segmentation done with one of the prototypes. The segmentation error depends on difficulties to handle the papillaries, which is a muscle structure inside the left ventricle blood volume with the same intensity as the left ventricle wall and the abrupt transition of the left ventricle to the left atrium with the same blood pool signal intensities. Also the segmentation sometimes falsely includes bright pixels on the outside of the heart. The three different error sources can be handled in different ways.

The opportunity of choosing a smaller initialisation has not been used when creating these prototypes but when looking at the results it may eliminate or at least decrease the problem of the segmentation leaking out through the left ventricle wall. A smaller initialisation would be completely inside the left ventricle and probably the optimised value of the alpha parameter controlling the influence of the edges in the image would be larger.

The intensity function can not in its current appearance handle the inclusion of the papillaries in the left ventricle blood volume and the papillaries is actually not always encountered as a part of the blood volume but rather as a part of the wall volume and therefore it might be better to compare the prototype based segmentation to a manual segmentation of the blood volume without the papillaries [14]. A better adjustment and mapping of the intensities, as suggested in the chapter Future work, might also lead to a better optimisation of the parameters to control both the leakage into the left atrium and the inclusion or exclusion of the papillaries.

Comparison of the different prototypes might not be feasible to do since the segmentation results is poor and the difference between the segmentation errors depends on how much of the papillaries that are included, leakage into left atrium or leakage through the left ventricle wall. The results may though indicate that it does not make a big difference that the parameters in the prototype is optimised on the same set as the intensity function, initialisation and spatial map was generated from. Also the difference in segmentation error when setting landmarks twice in the same image is below 1% which might indicate that the segmentation of the left ventricle blood volume might not be sensitive to the landmark setting.

6.4 Segmentation of right ventricle blood volume

The fourth application tested was segmentation of the right ventricle blood volume. For this application four prototypes have been built and tested on 10 images. The right ventricle has a thinner wall structure than the left ventricle and thus it is more likely that the segmentation leaks through the wall. The optimisation is though done to minimise the segmentation error. With a large value of λ more spatial constraints will be used, which might prevent the leakage unfortunately it is hard to optimise a good value of λ since the variation in shape of the right ventricle is fairly large. The curvature weight β can also prevent the leakage but the curvature should also be allowed to vary where it is needed otherwise the parts of the object with larger curvature will not be included. The parameter α can also prevent the leakage since it lowers the speed near the edges.

When optimising the parameters the absolute volumetric segmentation error $\epsilon_{V,abs}$ is used which includes all falsely included and excluded pixels, but the influence of leakage is larger than an underestimation of the right ventricle. This leads to a compromise which gives the smallest absolute error but unfortunately underestimates the left ventricle blood volume. It can though be seen that the optimisation has been done so that a small absolute error is acquired since the mean absolute error measurement is between 0.62 and 0.79 millimetres for the four prototypes.

For the segmentation of the right ventricle four prototypes has been built to be able to draw conclusions on the sensitivity of the method, however, the relatively poor result might not make this feasible. The results also point in another direction than the results for the segmentation of the left ventricle blood volume.

Comparing the prototypes built on different images, that is to say image 1 to 5 respectively image 6 to 10, gives a small difference indicating that it does not matter which set of images the prototype is built upon. Comparison on the optimisation on different sets of images also shows a result contradicting the result for the left ventricle blood volume. The segmentation error for the prototype built upon image 1 to 5 and optimised on image 6 to 10 compared to the prototype built and optimised on image 1 to 10 indicates that it does matter that the optimisation is done on the same set as it was built upon. This further reinforces the argument that it is not feasible to draw any conclusions on the method sensitivity on these fairly poor results. To be able to draw certain conclusions one need to improve the intensity mapping function and run the parameter optimisation once more.

6.5 Conclusions

The segmentation result of the aorta is very satisfying and the results indicate that the segmentation can be improved for the other applications. To acquire better results for the other applications the usage of the *a priori* information about intensity needs to be improved. First, after such improvement and re-optimisation, the suggested way of validating the method sensitivity can be applied and conclusions about the method sensitivity can be drawn.

In conclusion, the prototype based segmentation is a method to incorporate *a priori* information to a level set method. The segmentation result is promising since the segmentation of the aorta has an error measurement comparable to the one an experienced observer might do.

Chapter 7

Future work

The prototype based method shows promising results, and further development of the method will be done. Commercial interest exist to use the method for segmentation of the whole heart in both MR and computed tomography, (CT), segmentation of the aorta for run off studies in CT and segmentation of renal artery and other peripheral vessels. The development of the method will therefore be continued. Primarily the intensity adjustment and intensity function will be further developed. The intensity adjustment will be further developed to be able to handle the large variations in intensity normalisation, in especially MR images. The intensity function will be further developed to handle deviations from the intensities most present in the example images. Ideas on these improvements and other improvements suggested in the discussion chapter will be discussed below.

The adjustment of intensities can be done in several ways, some previously discussed in the method chapter. All three methods discussed were methods of having a predefined way of adjusting the intensities. The intensity can also be adjusted depending on the intensities in the image to be segmented.

The adjustment which will be tested, to improve the intensity mapping, is to make a histogram of the intensities inside the initialisation and compare the peak of this new histogram to the peak of the intensity function. The intensity function is then calculated by adjusting the peak of the histograms of the example images. It might also be feasible to develop the method so that it is possible to use two different histogram peaks for the segmentation. Of course these histograms should be compared to the histograms of the intensities outside the object to segment. With this calculation of the intensity function it, might also be feasible to encounter for some uncertainty or deviation of the intensities. To encounter for this the width of the peaks for the histograms can be compared.

Another adjustment of the method will be to investigate the importance of the size of the initialisation. A smaller initialisation will most probably be contained in the object to segment and the risk of segmentation leakage through the object edges will be reduced. After optimisation with a smaller initialisation the spatial map might be even more helpful in for example the segmentation of the left and right ventricle blood volume so that even the leakage into the atriums is reduced. The leakage of the segmentations into other structures could also be completely eliminated by using landmarks which defines a slice or position through no propagation of the segmentation is allowed. In the case of the ventricle blood volume a slice could be an excellent choice of such constraint. In the aorta though such a constraint could not have been used instead of the spatial map since the heart is imbedded in the same slices as the aorta. After adjustment of the intensity function and usage of smaller initialisation such a constraint most probably will not be needed though.

One explanation of the underestimation of the volumes has been that the error becomes larger if the segmentation falsely includes pixels at a distance of one pixel outside the object than falsely excluding pixels at the same distance inside the segmentation since the volume outside is larger. This might be corrected by weighting the errors differently. The optimisation of the parameters for the aorta, however, has resulted in segmentation errors both positive and negative. Since the intensity function works better for the segmentation of the aorta the problem of underestimation might also be reduced when the intensity function is improved.

Yet another improvement of the method would be to use a smoothing which does not split structures in two when the structure is thin, like for example the left ventricle wall. This can be accomplished by using an anisotropic smoothing and smooth more along the segmentation and not across the border. Such a method can for example be an image enhancement method [15].

Further development of the method will focus on improving the intensity function and testing segmentation with a smaller initialisation. When these improvements have been done, re-optimisation of the parameters and further analysis of the method sensitivity will be done.

Bibliography

- [1] J. Renner, T. Ebbers, M. Karlsson, E. Heiberg, D. Loyd, T. Länne, and R. Gårdhagen, “Feasibility of patient specific aortic blood flow CFD simulation,” in *MICCAI 2006*, (Copenhagen, Denmark), pp. 77–86, 2006.
- [2] T. Cootes, C. Beeston, G. Edwards, and C. Taylor, “A unified framework for atlas matching using active appearance models,” *Information Processing in Medical Imaging*, pp. 322–333, 1999.
- [3] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, “3-D active appearance models: Segmentation of cardiac MR and ultrasound images,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, pp. 1167–1178, 2002.
- [4] M. Lorenzo-Valdés, G. Sanchez-Ortiz, A. Elkington, R. Mohiaddin, and D. Rueckert, “Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm,” *Medical Image Analysis*, vol. 8, no. 1-4, pp. 255–265, 2004.
- [5] T. McInerney and D. Terzopoulos, “Deformable models in medical image analysis: A survey,” *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.
- [6] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 312–331, 1988.
- [7] T. McInerney and D. Terzopoulos, “A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis,” *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, pp. 69–83, 1995.
- [8] T. McInerney and D. Terzopoulos, “T-snakes: Topology adaptive snakes,” *Medical Image Analysis*, vol. 4, pp. 73–91, 2000.

BIBLIOGRAPHY

- [9] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models - their training and applications,” *Computer Vision and Image Processing*, vol. 61, no. 1, pp. 38–95, 1995.
- [10] J. Sethian, *Level Set Methods and Fast Marching Methods*. Berkeley, California, USA: Cambridge University Press, 1999.
- [11] A. V. Aho, J. E. Hopcroft, and J. Ullman, *Data Structures and algorithms*. Addison-Wesley publishing company, 1987.
- [12] B. Nilsson and A. Heyden, “A fast algorithm for level set-like active contours,” *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1331–1337, 2003.
- [13] L.-C. Böiers, *Lectures on optimization*. Lund, Sweden: KFS AB, 2001.
- [14] B. Sievers, S. Kirchberg, A. Bakan, U. Franken, and H.-J. Trappe, “Impact of papillary muscles in ventricular volume and ejection fraction assessment by cardiovascular magnetic resonance,” *Journal of Cardiovascular Magnetic Resonance*, vol. 6, no. 1, pp. 9–16, 2004.
- [15] C. F. Westin, L. Wigstrom, T. Loock, L. Sjoqvist, R. Kikinis, and H. Knutsson, “Three-dimensional adaptive filtering in magnetic resonance angiography,” *Journal of Magnetic Resonance Imaging*, vol. 14, no. 1, pp. 63–71, 2001.