

Diploma Thesis in Computer Science

**Segmentation, 2D-3D Registration,
and Uncertainty Propagation
for Dynamic Roadmapping
in Angiographic Interventions**

Frederik T. Bender

Technische Universität München
Department of Computer Science
Chair for Computer Aided Medical Procedures and Augmented Reality
in collaboration with
Siemens Corporate Research, Princeton

Frederik T. Bender

Segmentation, 2D-3D Registration, and Uncertainty Propagation for Dynamic Roadmapping in Angiographic Interventions

Diploma Thesis in Computer Science



Technische Universität München

Fakultät für Informatik

Chair for Computer Aided Medical Procedures and Augmented Reality

This thesis was conducted in collaboration with
Siemens Corporate Research, Princeton

Diplomarbeit: Segmentation, 2D-3D Registration, and Uncertainty Propagation
for Dynamic Roadmapping in Angiographic Interventions
Autor: Frederik T. Bender(benderf@in.tum.de)
Aufgabensteller: Prof. Dr. Nassir Navab
Betreuer: Martin Groher, Ali Khamene
Abgabedatum: September 28, 2007

Ich versichere, dass ich diese Diplomarbeit selbständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Garching, den 28. September 2007

Frederik T. Bender

Abstract

Minimal invasive catheter-guided interventions play an important role in most hospitals all over the world. During the treatment physicians have to rely on 2D information that is acquired through a fluoroscopic imaging device. In those fluoroscopic images, catheter, bone structures, and (if contrast agent added) blood vessels are visualized in real-time. Those low resolution images can be enhanced by high resolution DSAs where only vessel structures are visible. In difficult interventions these 2D views might not be adequate and it is challenging to navigate the catheter to a region of interest. In this thesis we present a method that allows the physician to use information from a pre-operatively taken 3D CTA scan during the intervention. In the first part we introduce an innovative approach for 2D/3D registration which forms the basis for the second part of the thesis, the development of an estimation procedure to resolve the 3D location of the catheter tip in real-time. This transfers the navigation process to the three-dimensional space and leads to a faster and smoother workflow.

Zusammenfassung

Minimal invasive Katheter-geführte Chirurgie ist weltweit ein sehr verbreitetes Verfahren. Während des Eingriffs navigiert der Chirurg einen Katheter zu einer bestimmten Stelle im Körper mit Hilfe von sogenannten Fluoroskopien. Das sind schnell aufeinanderfolgende Röntgenaufnahmen die dem Arzt 2D Informationen des Körperinneren in Echtzeit liefern. Sichtbar sind vor allem knöcherne Strukturen, der Katheter und falls Kontrastmittel gespritzt wird, auch die Blutgefäße. Bei schwierigen Eingriffen sind diese 2D Informationen jedoch nicht ausreichend für eine einfache Navigation des Katheters. Das liegt hauptsächlich an der Komplexität der Blutgefäßstrukturen und eventuellen Überschneidungen einzelner Gefäße. In 2D ist dann schwer zu entscheiden, welcher Weg in welches Gefäß führt. Aus diesem Grund stellen wir einen Ansatz vor, der es erlaubt, vor dem Eingriff angefertigte 3-dimensionale Aufnahmen des Patienten auch während der Operation zu verwenden. Der erste Teil besteht dabei aus einem Registrierverfahren, das das 3D Volumen und die 2D Aufnahmen während des Eingriffs zueinander ausrichtet. Dies wird dann dazu verwendet, die 3D Position der Katheterspitze zu jedem Zeitpunkt in Echtzeit zu berechnen, so dass der Navigationsprozess in 3 Dimensionen durchgeführt werden kann. Dies führt zu einer erheblichen Erleichterung der Navigation und zu einem schnelleren und reibungloserem Ablauf der Intervention.

Acknowledgments

First of all I want to thank my supervisor Martin Groher who supported me all the way. He introduced me to the topic of angiographic image processing during my time as a student assistant, worked out a nice topic for my diploma thesis, and helped me to realize it. Lots of thanks also to my professor Nassir Navab and to Ali Khamene from Siemens Corporate Research in Princeton who gave me the opportunity to work in such an interesting field of research. Many thanks to Wolfgang Wein who was a helpful hand in Princeton whenever Ali was not available. Thanks to the people from the chair for computer aided medical procedures who were always very cooperative and perservering when I came to their office and asked them questions. Thanks to Joerg Traub who let me win in soccer so many times, and to my friends who helped me to remember and enjoy the social life after work. Last but not least, very special thanks to my parents and my sister who supported me my whole life, and were always there for me when I needed them the most. They always encouraged me to go the extra mile and take the challenges on the way. Therefore, I dedicate this thesis to Susanne, Manfred, and Martina Bender. Shall this be the proof that your effort was not in vain. Thank you for everything!

Contents

1. Introduction	1
2. Background	3
2.1. Angiography	3
2.1.1. Fluoroscopic Sequences	3
2.1.2. Digital Subtraction Angiography	3
2.1.3. Computed Tomography Angiography	4
2.2. Mathematical Morphology	4
2.2.1. Basic Set Theory	5
2.2.2. Primitive Operators - Dilation and Erosion	5
2.2.3. Advanced Operators - Opening and Closing	7
2.3. Segmentation	8
2.3.1. Digital Image	8
2.3.2. Neighborhood and Homogeneity	9
2.3.3. Segmentation	9
2.3.4. Region-Based Segmentation	10
2.3.5. Alternative Segmentation Approaches	10
2.4. Registration	11
2.4.1. Transformations	12
2.4.2. Cost Function	14
2.4.3. Optimizer	14
2.4.4. Intensity-Based Registration	15
2.4.5. Feature-Based Registration	16
2.5. 2D/3D Registration	17
2.5.1. Projection	17
2.5.2. Pinhole Camera	18
2.5.3. Intensity-Based Registration	19
2.5.4. Feature-Based Registration	20
2.6. Data Analysis	22
2.6.1. Random Variable	22
2.6.2. Mean and Covariance	22
2.6.3. Kernel Density Estimation	23
3. Related Work	25
3.1. Vessel Segmentation	25
3.2. 2D/3D Registration	26
4. Method for 2D/3D Registration	29

4.1. Preprocessing	29
4.1.1. Bothat Filtering	30
4.1.2. Histogram Equalization	31
4.1.3. Preprocessing of DSA Images	32
4.2. Vessel Extraction	32
4.2.1. Automatic Seed Point Detection	33
4.2.2. Region Growing	39
4.2.3. Model Extraction	41
4.3. Initial Registration	42
4.4. Iterative Approach	44
4.4.1. Probability Map	44
4.4.2. Iteration	46
5. Dynamic Roadmapping	51
5.1. Respiratory Motion Compensation	51
5.2. Evaluation of 3D Catheter Tip Position	52
5.2.1. Uncertainty Propagation	53
5.2.2. Non-Parametric Information Fusion	55
6. Evaluation and Results	61
6.1. 2D/3D Registration	61
6.1.1. Evaluation	61
6.2. Dynamic Roadmapping	63
6.2.1. Generating Data Sets	63
6.2.2. Evaluation	64
7. Conclusion	67
Appendix	71
A. List of Symbols	71
B. List of Abbreviations	73
Bibliography	75

1. Introduction

With recent developments in medical technology, interventions and surgeries tend to be more and more minimally invasive. In this procedure, optical instruments, catheters, or other interventional instruments are inserted into the body through very tiny cuts. This avoids the need for an open invasive surgery for many medical procedures. Main advantages of minimally invasive surgery are less operative trauma, faster recovery, and less post-surgical complications. The main disadvantage is that surgery becomes more complicated since the physician's view is very restricted. He has to rely on information obtained through the optical instruments or on image data acquired through external imaging devices like X-Ray or computed tomography.

We will focus on an interventional procedure called transarterial chemoembolization (TACE). It is applied to patients that suffer from hepatocellular carcinoma (HCC) which is a primary malignancy (cancer) of the liver. During this procedure, the physician inserts a catheter into the abdominal aorta and navigates it into the proper hepatic artery (which supplies the liver). By injecting contrast agent, the tumor supplying vessel is detected, and occluded by introducing emboli, such that the tumor's blood supply is interrupted. The goal of the procedure is to stop tumor growing and finally to kill the tumor.

The difficulty in this procedure is the navigation of the catheter. The physician has to navigate through a complicated vessel system with several branches and vessel overlaps. Thereby, he has to rely on information acquired through an X-Ray imaging device which visualizes the vessel structures and the catheter in 2D. Compared to 3D this is a loss of information since overlapping vessels are visualized in a single plane. Therefore, a 3D volume of the region of interest is obtained pre-operatively to plan the intervention and extract a road map in advance. Recent approaches yield to use this 3D information also intra-operatively by overlaying the road map onto the acquired 2D images. This is already an enhancement to the navigation process, but it is still not a perfect solution because the problem of overlaps is still present.

In this thesis we introduce an approach which transfers the navigation procedure to the three-dimensional space. In the first part, we develop a registration method to compute a correct spatial relationship between the three-dimensional volume and the intra-operatively acquired two-dimensional image data. We use the information obtained from the registration to compute the on-line 3D position of the catheter tip using an statistical approach which takes the uncertainties of our algorithms into account. This allows the physician to use the complete information of the 3D volume which yields a higher accuracy in navigation, and therefore speeds up the intervention significantly.

2. Background

In this chapter we give an introduction to the basic terms, concepts, and procedures used throughout the thesis. We start with a short overview of current image modalities regarding blood vessel visualization, and an introduction to mathematical morphology. Afterwards, we explain the concepts of segmentation, registration, and statistical analysis which are fundamental to understand the algorithms presented in chapters 4 and 5.

2.1. Angiography

Angiography is a technique to visualize blood vessels in medical image data like X-Ray or Computed Tomography images. Since blood has a similar radiodensity as the surrounding tissue, they can usually not be identified in a regular X-Ray image. The solution to this problem is to inject a contrast agent into the blood flow. This contrast agent absorbs X-Rays and therefore differs from the surrounding tissue in the final image (compare figure 2.1(a) and 2.1(b)). The contrast agent can be added either globally or locally. If added globally it is injected directly into a vessel with the help of a needle and from there distributed through the whole blood system. Often though specific regions of the body are subject to examination. Then a catheter (a thin flexible tube which can be inserted directly into and moved inside a vessel) is used to administer the contrast agent exactly where it is needed.

2.1.1. Fluoroscopic Sequences

A fluoroscopy sequence is a series of X-Ray images which are taken consecutively in real-time. During a fluoroscopic intervention the patient is placed under a fluoroscope. This consists of an X-Ray source and a computer screen where the images can be displayed. A continuous acquisition of X-Ray images allows the surgeon to actually see inside the body on line, this means he sees real-time images of the patient's inside. This might be helpful to navigate a catheter, to fix bone fractures, or to place a stent. Although the radiation dose used for fluoroscopy is usually less compared to a regular X-Ray image, the exposure for the patient and also for the surgeon is still not negligible. Therefore, lots of research is done to minimize the hazard of radiation exposure.

2.1.2. Digital Subtraction Angiography

Digital subtraction angiography (DSA) is a special form of angiography which improves the visibility of blood vessels to a greater extent (figure 2.1(c)). During this

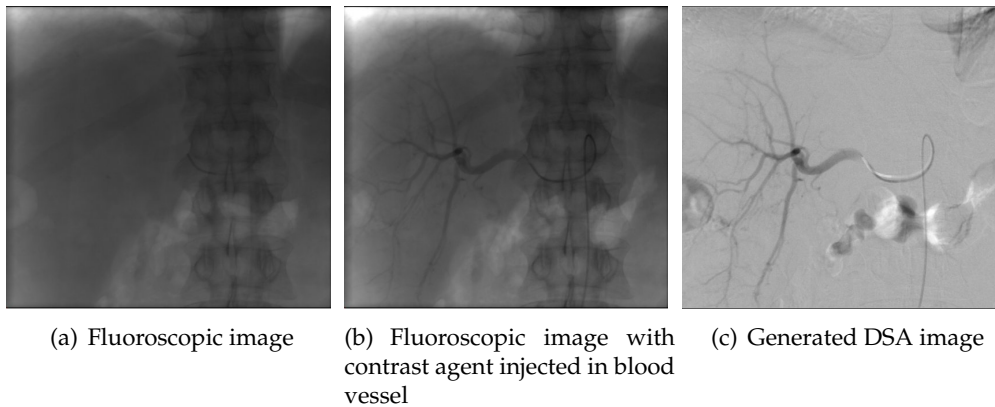


Figure 2.1.: Comparison of fluoroscopic and DSA images.

process two digital X-Ray images of the same part of the body are taken, one with and one without contrast agent injected. In its simplest form the two images are subtracted and the resulting image only shows vessels without any other background structures like bones, etc. Since there might be small patient movements in-between the acquisition of the images, techniques for automatic spatial alignment or other pre-processing steps might be applied before the subtraction to improve the quality of the output [26] [25].

2.1.3. Computed Tomography Angiography

Computed tomography is a method of medical imaging where a series of two-dimensional X-Ray images is used to generate a three-dimensional volume of the body or a body part. The X-Ray source (which is part of the CT-scanner) is rotated on an axis around the patient and emits X-Rays in a given interval. These rays are assembled by detectors and a computer is used to combine all those images to a three-dimensional volume. In computed tomography angiography (CTA) a contrast agent is added additionally. This makes blood vessels again visible, and if segmentation (see section 2.3) is applied, a three-dimensional model of the blood vessel structures can be extracted.

2.2. Mathematical Morphology

Mathematical morphology is a powerful tool for various image processing tasks, based on set theory. Boundary extraction, thinning, segmentation, or image enhancement are just a few examples to mention. Before we start with the definition of several morphological operators, we will give a short introduction to set theory following [15].

2.2.1. Basic Set Theory

Assume we have two sets A and B . The *union* C of two sets is then denoted as

$$C = A \cup B$$

and contains all elements that are either element of A or B .

The *intersection* D is denoted by

$$D = A \cap B$$

and contains all elements that are element of A and B .

We say two sets A and B are *disjoint* if their intersection is the *empty set* \emptyset , which means

$$A \cap B = \emptyset$$

To specify the elements of a set we use two braces: $\{\bullet\}$. As an example, a set E consisting of the elements 1, 2, and 3 can be written as $E = \{1, 2, 3\}$. If we want to define a set F with an infinite number of elements, for example a set containing all natural numbers x larger than 5, we can write $F = \{x \mid x \in \mathbb{N}, x > 5\}$.

2.2.2. Primitive Operators - Dilation and Erosion

After this short introduction to set theory, we will now describe some morphological operators. We will start with binary morphology and extend it to greyscale morphology later on. A morphological operator is based on a structuring element which itself is a set. Two very basic operators which form the basis to many advanced operators are dilation and erosion. Dilation between an object A and structuring element B is defined as

$$A \oplus B = \{z \mid (\widehat{B})_z \cap A \neq \emptyset\}$$

This equation indicates that dilation of A and the reflection \widehat{B} of B results in the set $A \oplus B$ of all points z such that A and \widehat{B} , translated by z , overlap by at least one element. The reflection \widehat{B} is obtained by flipping B around its origin (see figure 2.2).

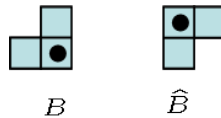


Figure 2.2.: Structural element B and its reflection \widehat{B} .

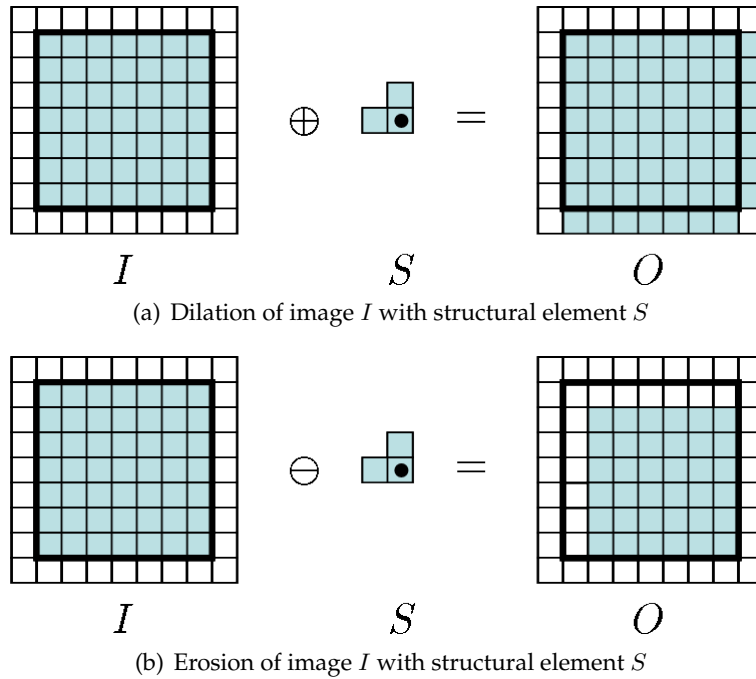


Figure 2.3.: Primitive morphological operators (a) dilation and (b) erosion.

Using the same denotation, erosion is defined as

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

which is the set of all points z such that B , translated by z , is contained in A .

Since this notation might be a little confusing, we will illustrate the application of dilation and erosion of a 2D image I with a structuring element S with an example (see figure 2.3). We start with an input image I , a structuring element S , and an output image O . Dark (object) pixels represent the value 1, while white (background) pixels represent the value 0. In the initial state, all pixels of output O are set to 0. We apply the structuring element to each pixel of input image I , this means we translate the origin of S (marked with a black dot) to each image pixel successively. If $T_{i,j}$ denotes the transformation of the origin of S to image pixel (i, j) , and $v_{i,j} = \{v_1, v_2, \dots\}$ is the set of image pixels which are covered by S at translation $T_{i,j}$, the intensity value $O(i, j)$ of pixel (i, j) of O is computed as $O(i, j) = \max(v_{i,j})$. The algorithm is summarized in table 2.1.

This notation of binary dilation has some big advantages. First of all, it is usable for greyscale images, which means it is also an implementation of greyscale dilation. Furthermore, if we replace the maximum notation in table 2.1 by a minimum notation, we obtain an algorithm for binary and greyscale erosion.

The effect of dilation and erosion is visualized in figure 2.3. While dilation can be used to fill gaps between two regions, erosion can be used to delete small unwanted

<ul style="list-style-type: none"> • Input input image I, structuring element S • Output output image O • Constants w, h width and height of input/output image • Algorithm <ol style="list-style-type: none"> 1. Initialize O with width and height of I 2. for $i = 1$ to w and $j = 1$ to h 3. do <ul style="list-style-type: none"> Translate origin of S to image pixel (i, j) Compute maximum value max of all pixels (a, b) where (a, b) is covered by S Set $O(i, j) = max$ 4. od

Table 2.1.: Implementation of the dilation operator.

details, e.g. background noise.

2.2.3. Advanced Operators - Opening and Closing

Based on the fundamental morphological operators dilation and erosion we introduce two more advanced operators, opening and closing. As you can see in figure 2.3, dilation and erosion expand and shrink objects in the image. To countervail this effect, opening and erosion include two consecutive executions of dilation and erosion. Opening, denoted by \circ , is defined as

$$A \circ B = (A \ominus B) \oplus B$$

which represents an erosion of A with structuring element B followed by a dilation with B .

Closing, on the other hand, is defined as

$$A \bullet B = (A \oplus B) \ominus B$$

and consists of a dilation of A with B followed by an erosion with B . The effect of opening and closing compared to dilation and erosion is illustrated in figure 2.4. Input image 1 can be seen as the result of a bad photocopy. If we want to detect and interpret the letters automatically, we need to preprocess the image such that all parts of a letter are connected, i.e. we need to fill the gaps between the letter parts. Both, dilation and closing, are able to fulfill this task, but as you can see, the letters in 2.4(b) are thicker than in the input image. This is not the case in figure 2.4(c) were we apply a closing operation. A similar observation can be obtained from the second row in the figure where we tried to reduce the noise of input image 2. Unlike erosion (figure 2.4(e)), the opening operation applied in figure 2.4(f) retains the letter size.

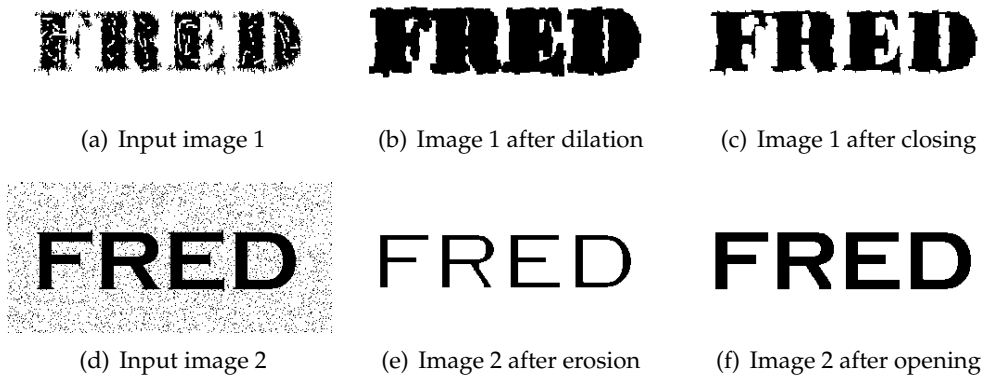


Figure 2.4.: Comparison of multiple morphological operators applied to binary images.

2.3. Segmentation

Before we describe the term segmentation in detail, we will explain some basic terms which are essential to understand the process of segmentation, and continue with some specific segmentation algorithms. We will start with a formal definition of a digital image and give some examples for image properties. Afterwards, we will finally explain what segmentation does and what we can use it for. We will then have a closer look at the segmentation approach relevant to our work and give a short overview of alternatives in section 2.3.5.

2.3.1. Digital Image

Pham et al. [34] define a digital image as a 'collection of measurements in two- or more dimensional space', also called pixels in 2D or voxels in 3D. In this thesis segmentation is mainly applied to two-dimensional images. Therefore, all the definitions and examples are given for the 2D case, even though most of it can be transferred to three-dimensional space as well. Also, we will solely talk about grey value images and neglect the existence of color images.

In a more formal definition an image in its discrete form is a function $I : D \mapsto \mathbb{R}$ with D being a subset of \mathbb{N}^2 . This function describes a mapping of data points $(x, y) \in D$ to intensity values. These are again discrete values and dependent on the accuracy of the data type used for the implementation. Very common are 8 bit character [0..255] and 32 bit float values. The latter ones are used in our work, and the intensities are additionally normalized to the interval [0..1].

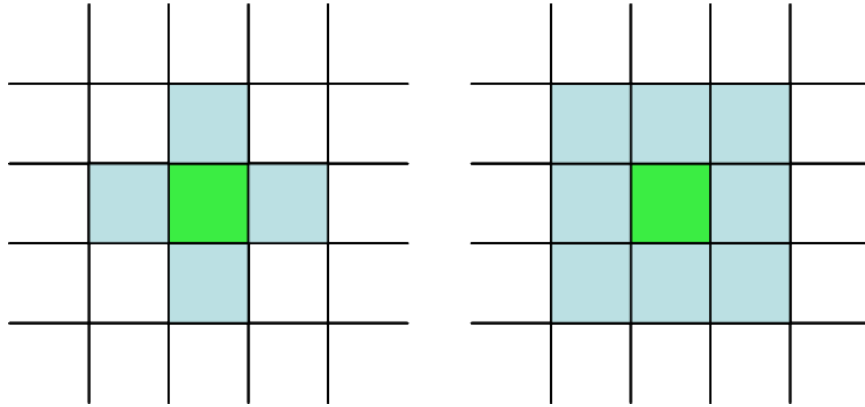


Figure 2.5.: 4-neighborhood (left) vs. 8-neighborhood (right).

2.3.2. Neighborhood and Homogeneity

As already defined above, an image is a set of pixels which form a grid. What we want to define now are two image properties which are relevant for the segmentation technique we describe in section 2.3.4. The first one is called neighborhood and determines what pixels are declared as directly connected to a center pixel. In 2D there are usually two cases, the 4-neighborhood and the 8-neighborhood. For an illustration see figure 2.5.

Homogeneity describes the similarity of pixel intensities. A set of pixels $x_i, i = 1..n$ is called homogeneous if the following formula is true for all i

$$\forall i : T_l \leq x_i \leq T_u$$

where T_l defines a lower and T_u an upper boundary for pixel intensities (threshold). The threshold selection is crucial since it directly affects the quality of the segmentation result. This is an issue of many segmentation algorithms, including the one we use. If we do not set correct values for T_l and T_u , we will end up in anything but a good segmentation. Therefore automatic threshold computation will be one of the important topics of this thesis.

2.3.3. Segmentation

Now it is time to give a definition of the term segmentation. As already mentioned, an image is a function $I : D \mapsto \mathbb{R}$. Segmentation is then defined as the process of partitioning the domain D into subsets $D_i, i = 1..k$ where $D = \bigcup_{i=1}^k D_i$ [34]. $\bigcap_{i=1}^k D_i = \emptyset$ might or might not be true, depending on the constraints of the problem. In some cases it makes sense to restrict the quantity of subsets D_i . A common case is to set k to 2. Then one of the sets is called background, and the other one is called object. This is done in this thesis where we divide the image into vessel structures and background objects.

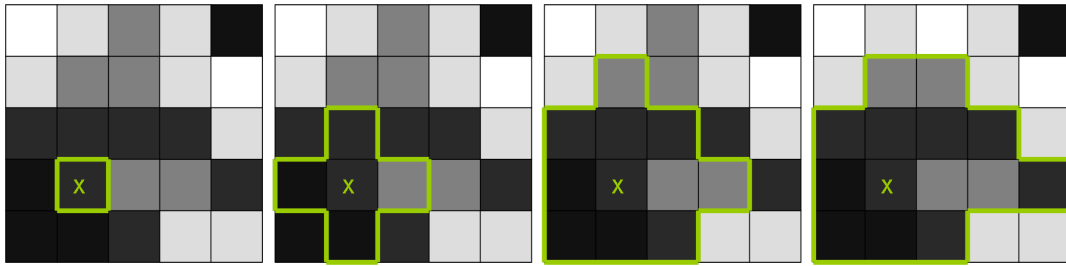


Figure 2.6.: Example for region growing. The green x marks the seed point.

2.3.4. Region-Based Segmentation

The segmentation concept used in our approach is region-based. It is based on the so-called region growing algorithm and enhances it to get over the issues coming with such a trivial method. Here we first describe the basic algorithm and specify the extended version features in more detail in a later section.

The meaning of a region can be easily understood if we use the expressions from section 2.3.2. Then a region R is a set of connected and homogeneous pixels and the pixels $x_i \in R$ are called region pixels. Region growing works as follows. Starting from a given seed point the segmentation (or region) is expanded iteratively. In each step the neighbors of all region pixels are examined for homogeneity and if the result is positive they are added to the region. The examination for homogeneity can be either done by comparing the pixel intensities to the intensity value of the seed point, or to a given threshold border. An example for region growing is given in figure 2.6.

As you may have noticed, there are two crucial things concerning the success of region growing. The correct position of an initial seed point and a good threshold determination. In the most trivial case this is both done by manual user input, for example by a mouse click in the image. Often though, manual user input is not an option and should be avoided. You can imagine, it might be distracting for a doctor if he has to click on a specific position in an image several times during an intervention and sometimes, if we think about fluoroscopic sequences, it is not even possible due to real-time constraints. Automatic seed point detection and threshold computation is then inevitable.

2.3.5. Alternative Segmentation Approaches

The segmentation approach from the previous section is solely based on intensity values. If we have a closer look at a picture itself, we may detect a couple more properties than just intensity. Edge-based algorithms, for instance, use strong intensity gaps in a local neighborhood to reach the final goal. Those jumps of intensities may refer to an edge which marks the border between two distinguished objects. Another group is formed by model-based approaches. A very common one is the

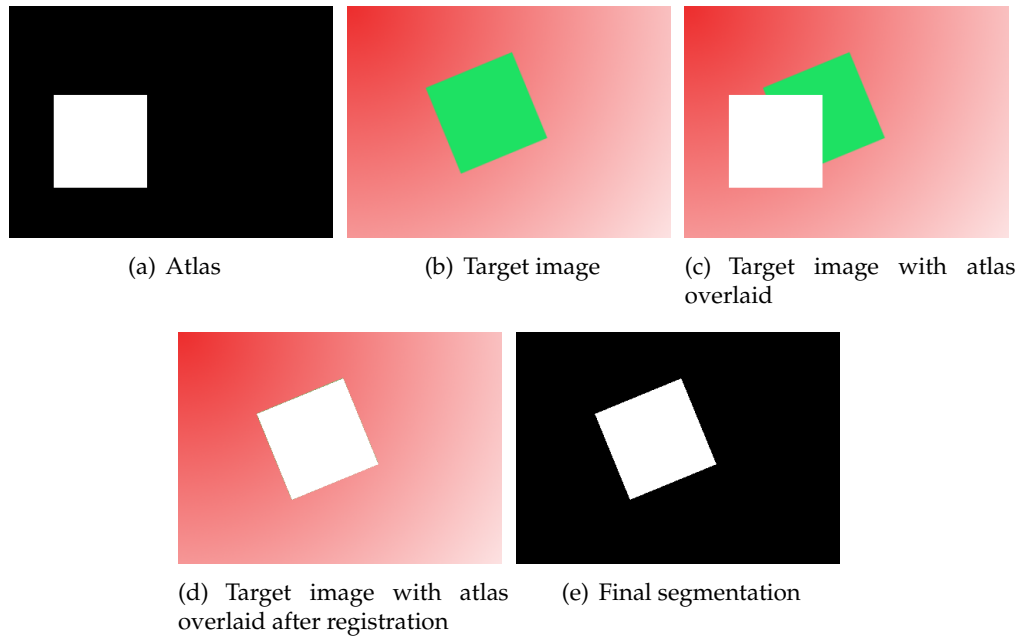


Figure 2.7.: Example for atlas-based segmentation.

Hough transform which can be used to detect simple shapes like circles or lines. Examples for more advanced approaches are Snakes and the Level Set method [30]. Both start with a simple model which is then deformed iteratively to fit the feature in demand. Main difference between those latter two is the representation of the model. They differ by an explicit (Snakes) vs. an implicit (Level Sets) representation. The deformation in both approaches is driven by internal and external energy functions which are dependent on the problem statement. Atlas-based approaches form the last category we want to mention here. Those are based on a reference image (called atlas) which can be acquired from some training data for example. This atlas is then matched to the current image (in meaning of spatial alignment, this is called registration) so information from the atlas can be transferred and used for segmentation. An example to illustrate the latter one is given in figure 2.7.

2.4. Registration

Image registration is the process of aligning images so that corresponding features can easily be related [18]. We will explain this in more detail with the help of some examples. Let us assume a CT image of a person is taken for some diagnostic reasons. Suspicious structure is detected, and another CT image is taken a couple of weeks or months later to check if the structure is growing or if its shape is changing. For a better comparison of the two data sets, and especially of the size and shape of the suspicious structure, it is necessary to bring them into spatial alignment. If we

take two different modalities into account, it might be even more obvious. Different modalities usually have different properties in highlighting tissues. CT, for example, is useful to display bony structures or the shape of a tumor, while PET (positron emission tomography) is well suited to display the activity of a tumor but does not give lots of information about its shape. Accordingly, it makes sense to join information of the CT and the PET data sets which requires registration.

2.4.1. Transformations

In general the process of registration involves the search for correct values for a set of parameters which describe a transformation between the images to be registered. The type of transformation, and hence the number of parameters, depends on the dimension of the input data as well as the type of transformation. The most trivial 3D transformation is the Euclidean or rigid transformation which includes three rotations around the three coordinate axis x , y , and z and three translations along the coordinate axis. This sums up to six parameters in total. If $X = (x, y, z)^T$ denotes the original 3D point we want to translate, and $X' = (x', y', z')^T$ the new, translated point, then a Euclidean transformation can be written as

$$X' = RX + t \quad (2.1)$$

where R is a rotation matrix and t a translation vector with $t = (t_x, t_y, t_z)^T$. R can be based on Euler angles and is the composition $R = R_z R_y R_x$ of

$$\text{a rotation by } \psi \text{ about the } z\text{-axis: } R_z = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{a rotation by } \theta \text{ about the } y\text{-axis: } R_y = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

$$\text{a rotation by } \phi \text{ about the } x\text{-axis: } R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$

This transformation equation is not always convenient since it is not linear and therefore not always easy to use in computations. Therefore we introduce the concept of homogeneous coordinates. The only thing we need to do is to add an additional column to all our vectors, coordinates, and matrices. Our points X and X' are then represented by:

$$X = (x, y, z, 1)^T$$

$$X' = (x', y', z', 1)^T$$

Rotation and translation are represented by:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$t = \begin{pmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We can now combine rotation and translation into one single translation matrix of the form

$$H = \begin{pmatrix} & R & t \\ & & \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

which leads to a new formulation of equation 2.1:

$$X' = HX \quad (2.3)$$

Besides rigid transformations there exist several non-rigid ones. A metric or similarity transformation adds a scaling factor to the Euclidean transformation and has therefore seven degrees of freedom (DOF) instead of six as the Euclidean. If we have different scaling factors for each of the three directions x , y , and z , and additionally a shearing, we are talking about an affine transformation with twelve DOF. This transformation still keeps parallel lines parallel but it does not conserve angles. Last but not least we introduce projective transformation which has fifteen DOF and does not conserve parallelism anymore. For a better understanding of those four types of transformations, have a look at figure 2.8 which shows a 2D example for each of them.

For the matter of completeness, we should also mention deformable transformations. Deformable transformations are special and do not necessarily map lines to lines or keep structure shapes. In medical image registration this might be important if we want to register structures that deform over time, for example a beating heart. Even though we also face the problem of deformation in this work, we do not use deformable registration and will therefore not explain it in more detail. If you are interested in the topic we refer to the work of Zikic [47] which gives a nice introduction to deformable registration at the beginning of the thesis.

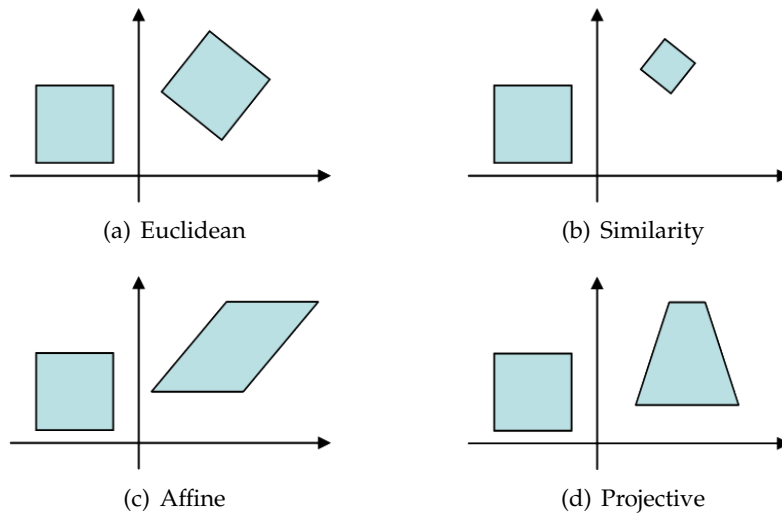


Figure 2.8.: Four different types of 2D transformations.

2.4.2. Cost Function

To decide if the parameters we determine yield a good registration we need some way to rate them. This is done by evaluating a cost function. A cost function is a formula that maps the registration parameters and the input images to a single value. Depending on the definition of the cost function, this value needs to be maximized or minimized to gain a good registration. Some concrete examples of cost functions will be given in [2.4.4](#) and [2.4.5](#).

2.4.3. Optimizer

By now, our registration process comprises input images, a set of parameters, and a cost function. It is quite evident that we still need some method to manipulate our parameter values and drive them towards a minimization/maximization of our cost function. This is done by an optimizer. An optimizer takes an initial guess of registration parameters, evaluates a cost function, and tries to change the parameters such that the registration gets better, i.e. the cost function value increases/decreases. The initial parameters must be close enough to the correct solution otherwise the optimizer might fail to return a correct registration. What 'close enough' actually means is dependent on the optimizer itself. Have a look at figure [2.9\(a\)](#). The blue curve represents a one dimensional parameterized cost function with its input value in x-direction and its output value in y-direction. The position of the grey ball represents our registration, and gravity our optimizer. If we throw the ball into the scene, gravity will force it to roll downwards along the curve until it reaches a minimum. The position where we drop the ball represents the initial guess of our parameter. In this example the ball will follow the red dotted line and end up in a local minimum

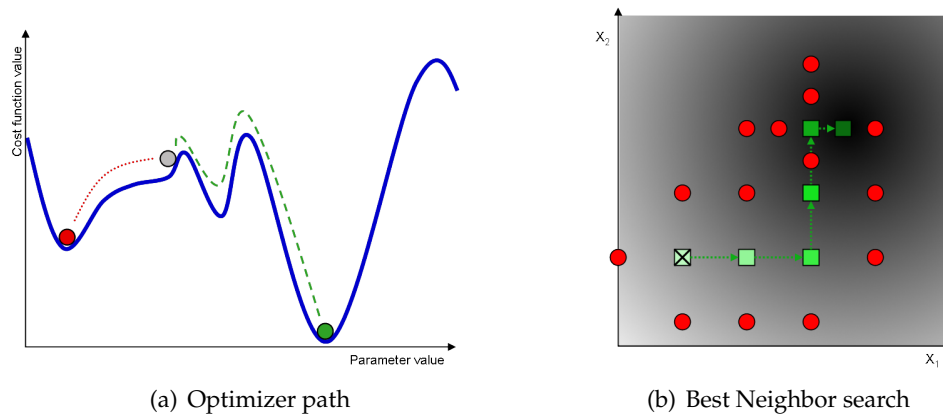


Figure 2.9.: Optimizer problems.

(red ball). This is far away from the global minimum (green ball) which is the correct solution to our registration problem. To find the global minimum, the grey ball has to follow the green dashed line. In our trivial case this will never happen unless we bring our initial guess closer to the correct solution, which means further to the right, or if we choose another optimizer, one that is not so much influenced by gravity but is also able to overcome obstacles like a small bump in the curve. Some optimizers are able to handle local minima better than others and an important step towards a good registration is therefore a good choice of optimizer.

To illustrate how an optimizer actually decides what parameters need to be changed, we will explain a very trivial optimizer in more detail, the Best Neighbor optimizer. Figure 2.9(b) shows the optimization problem of a registration with two parameters. The parameter values are given by the two coordinate axis, the cost function value is given by the brightness of the surfaces inside the coordinate system, whereas black must be interpreted as the optimal parametrization and white as the least optimal. The optimization starts at the position where the rectangle with the X inside is located. From there the Best Neighbor optimizer evaluates a cost function for all of its $2n$ neighbors. n specifies the number of parameters, which is two in our case. If any of the $2n$ positions yield a better cost function value than the value of the current one, the parameters are changed to match the position of the neighbor with the best cost function result. If no neighbor returns a better result the step size is halved and the optimization process is continued with the new step size. The algorithm stops when the step size or the cost function value reaches a given threshold.

Some other more advanced optimizers are the Downhill Simplex, the Powell-Brent, or the Gradient-Descent optimizer.

2.4.4. Intensity-Based Registration

Intensity-based registration (as the name already implies) works on the intensity values of images. We declare one of the images to be the reference image and the other

one to be the target image. The latter one stays fixed during the registration process while the former one is transformed to match the other image in terms of spatial alignment. The whole registration procedure includes three main steps which are applied iteratively until a decent result is obtained. The first one applies a transformation to the reference image. The second one evaluates a cost function, and the last step includes an optimization of the transformation parameters. This general description is actually true for most registration processes, also for a feature-based approach. The main difference is the cost function. In intensity-based registration a so-called (dis-)similarity measure is used as a cost function. A (dis-)similarity measure provides information about the (dis-)similarity of grey values of relating pixels in the two images. For instance, if pixel (1, 1) is black in the reference image and white in the target image, a similarity measure will return a small value for this pixel. A dissimilarity measure on the other hand will return a high value. As you can see, a good registration yields a maximization of the cost function if the cost function is a similarity measure, and it yields a minimization if it is a dissimilarity measure. For a better understanding we will give an example of a simple dissimilarity measure, the sum of squared differences (SSD), which describes the intensity dissimilarity between image X and Y :

$$SSD = \frac{1}{wh} \sum_{i=1}^w \sum_{j=1}^h (X(i, j) - Y(i, j))^2$$

Here w and h are the image width and height in pixels.

While some measures work relatively good on images of the same modality (e.g. CT-CT or MRI-MRI) they do not work on images of different modalities (e.g. CT-MRI). Measures of the former case are SSD, sum of absolute differences (SAD), or normalized cross-correlation (NCC). Examples for the latter one are joint entropy and mutual information (MI) [18].

2.4.5. Feature-Based Registration

More important for this work is the feature-based approach. Instead of intensities, it works on features. A feature can be everything ranging from an edge, over a bifurcation, to a centerline of a vessel. Which one fits the best is once again dependent on the image. Since we try to register vessel structures, we use vessel centerlines to accomplish the registration task as accurate as possible. The cycle of feature-based registration is very similar to the one mentioned in 2.4.4. However, the input of the registration is not necessarily two images. A model-to-image or a model-to-model approach is possible as well. Then the first step of the procedure includes a transformation of a reference model rather than a reference image. A good example, which in a modified version is also relevant for this thesis, is the Iterative Closest Point algorithm (ICP), originally proposed by Besl and McKay, 1992 [3]. We give now a short general overview of the algorithm and go into more detail when we explain our method in the next chapter. Let us assume we want to register two point sets X and Y . This is the input to our algorithm. The output will be a transformation that

registers X to Y . First of all we need a definition of closest point. Closest point $y \in Y$ to a point $x \in X$ is the solution to the function

$$CP(x, Y) = \operatorname{argmin}_{y \in Y} d(x, y) \quad (2.4)$$

where $d(x, y)$ is the Euclidean distance between the points x and y , i.e. $d(x, y) = \|x - y\|$. Now we can do a closest point search which results in a set of correspondences $\{x_i \leftrightarrow y_i\}$ of size N where $x_i \in X$ and $y_i \in Y$. If the size of point set X is m , and the size of Y is n , the worst cost for this search will be $O(mn)$. Fortunately, methods like k-D tress [27][35] for example can speed it up significantly. After we have a first set of correspondences it is time to compute the transformation between those sets. The best solution to this problem is the pair (R, t) that minimizes the mean-squares objective function [46]

$$F(R, t) = \frac{1}{N} \sum_{i=1}^N \|Rx_i + t - y_i\|^2 \quad (2.5)$$

The newly computed transformation parameters R and t are used to transform the point set X and the algorithm starts over again with the determination of closest points. An overview of the algorithm in pseudo code is given in figure 2.10.

- **input** point sets X and Y
- **output** transformation parameters R and t that register X to Y
- **algorithm** iterate until convergence:
 1. Compute closest point correspondences using equation 2.4
 2. Compute rotation R and translation t that minimize equation 2.5
 3. Apply transformation to all points $x \in X$

Figure 2.10.: Iterative Closest Point Registration.

2.5. 2D/3D Registration

2D/3D registration is a very important part of nowadays image-guided interventions. It involves the task of aligning 3D data sets like CT or MRI with 2D X-Ray projection images, such that corresponding structures in both data sets are properly aligned. Applications for this are, for instance, intra-operative navigation and patient positioning for radiation therapy. In addition to a rigid body transformation as described in chapter 2.4, 2D/3D registration involves a projection of the 3D data set onto a plane in order to be able to align it with the 2D data set.

2.5.1. Projection

When using a basic pinhole projection model, it is common to consider the origin of a Euclidean coordinate system as the center of projection, and define the so-called

image plane as a plane parallel to the (X,Y) -plane with distance f , i.e. the image plane can be written as $Z = f$. The coordinate axis Z is also called the principal axis and the point where the principal axis meets the image plane is called the principal point [19]. This scenario is illustrated in figure 2.11. A 3D point X in our Euclidean space is then mapped to a point x in the image plane such that x is the point where the image plane intersects a line joining X and the projection center C . This means a point $X = (X_x, X_y, X_z)^T$ is mapped to the point $x = (fX_x/X_z, fX_y/X_z)^T$ on the image plane which is a mapping

$$(X_x, X_y, X_z)^T \mapsto (fX_x/X_z, fX_y/X_z)^T$$

from Euclidean 3-space to Euclidean 2-space.

If we write this in matrix form with homogeneous coordinates we get

$$P = K[I|0] \tag{2.6}$$

where P is called the camera projection matrix which holds for

$$x = PX \tag{2.7}$$

K is called the camera calibration matrix and has the form

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

In equation 2.8 we make two assumptions. First, we assume the principal point to be the origin of the coordinate system of our image plane. The second assumption is, the number of image pixels per unit distance is the same in x - and y -direction. Both do not have to hold though. The former misconception leads to a vector $(p_x, p_y)^T$ which describes the principal point offset in the image plane coordinate frame. The latter one introduces two scaling factors which must be multiplied with our focal distance f which altogether leads to the final calibration matrix

$$K = \begin{pmatrix} a_x & 0 & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.9}$$

2.5.2. Pinhole Camera

In the previous section, we defined P to be a camera projection matrix. More precisely, it is the projection matrix of a pinhole camera. Since the pinhole camera model defines its own coordinate system, the camera coordinate frame, we need to relate the world coordinate frame, in which 3D points are expressed, to the camera position and orientation. This is done via a rotation R and a translation t , that lead to the final projection matrix

$$P = K[R|t] \tag{2.10}$$

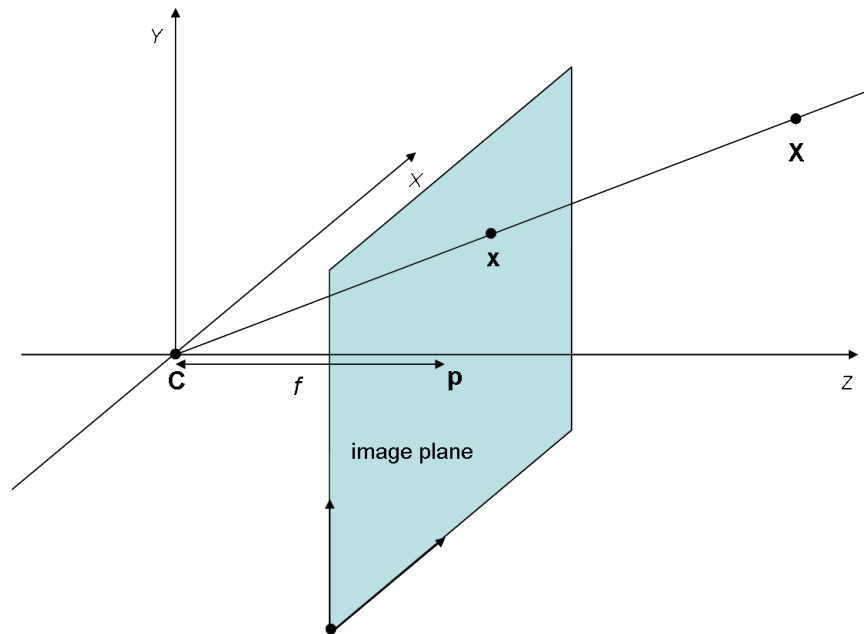


Figure 2.11.: Pinhole projection scenario.

R and t describe the position and orientation of the camera in terms of the Euclidean world coordinate frame, and K describes the internal camera settings. R and t are also called the extrinsic camera parameters, K is called the intrinsic camera parameters. For our 2D/3D registration problem this means we have a total of 10 DOF, i.e. our registration problem consists of solving for 10 parameters. Fortunately, in medical imaging, the intrinsic parameters are usually known since modern imaging devices can provide us this information. If the intrinsic parameters are known, we say the camera is calibrated. For calibrated cameras the problem of 2D/3D registration reduces to solve for 6 unknowns, the extrinsic parameters R and t (see section 2.4.1).

2.5.3. Intensity-Based Registration

As mentioned in chapter 2.4, there are two main classes of registration algorithms, intensity-based and feature-based ones. But compared to 2D/2D or 3D/3D registration, 2D/3D requires a little more effort due to the differing dimensions of the data sets. It does not make sense to compare a pixel intensity (x_{2D}, y_{2D}) to an intensity of pixel (X_{3D}, Y_{3D}, Z_{3D}) since there is no relationship between the two locations given. This is exactly why we project our 3D data into 2D. The way we do this differs a lot between intensity- and feature-based approaches. A common thing to do within the former one, is to produce a digitally reconstructed radiograph image (DRR). DRRs are computed by integrating along rays from a virtual point source through the 3D

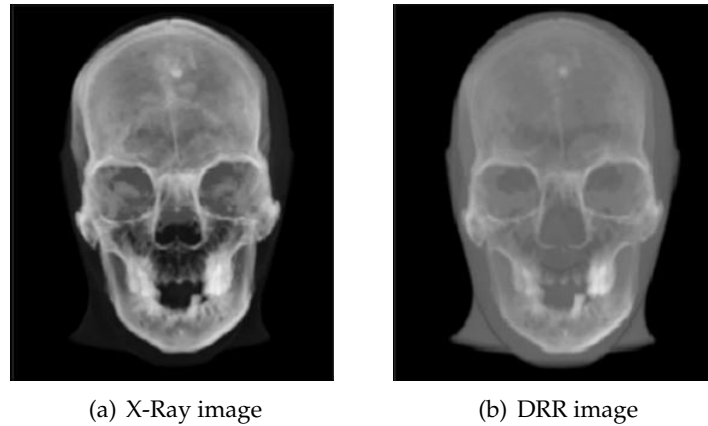


Figure 2.12.: These images (courtesy of Wolfgang Wein) demonstrate the similarity of X-Ray and digitally reconstructed radiographs (DRR).

volume. The resulting image is a simulation of an X-Ray image (see figure 2.12). These two data sets can then be registered with a usual 2D registration algorithm with a proper similarity measure [33]. The main issue here is the time consuming DRR generation, since a new DRR has to be generated every time the registration parameters change. Approaches to speed up this process can be found in Lacroute et al. [23] or La Rose[24]. Other approaches which directly work on the gradient vector volume of a CT data set are introduced by Tomazevic et al. [42] and Wein et al. [44].

2.5.4. Feature-Based Registration

Irrespective the time issues mentioned in 2.5.3, there are scenarios where intensity-based approaches might not suit well for 2D/3D registration. These are scenarios where the image content of the two data sets differs too much. We will illustrate this with an example relevant for this thesis. Imagine we want to register a pre-operatively taken 3D CTA volume of a patient’s liver to a 2D fluoroscopic image of the same body part which is acquired intra-operatively during a catheter-guided intervention. The first difference is certainly the catheter, visible in the 2D image only. Since contrast agent for the CTA acquisition is added globally (see section 2.1), there is no need for a catheter, and therefore, it is not existent in the 3D image. The global vs. local injection of contrast agent is also the reason for different vessel shapes. Some vessels which are visible in one image are not visible in the other, and vice versa. The third issue is due to breathing. The vessel structures of the liver are influenced by breathing motion and will therefore deform between two distinct image frames. All of these observations combined might cause an intensity-based approach to fail. A feature-based technique which knows how to deal with those problems is introduced in this thesis. But before we explain our algorithm in chapter

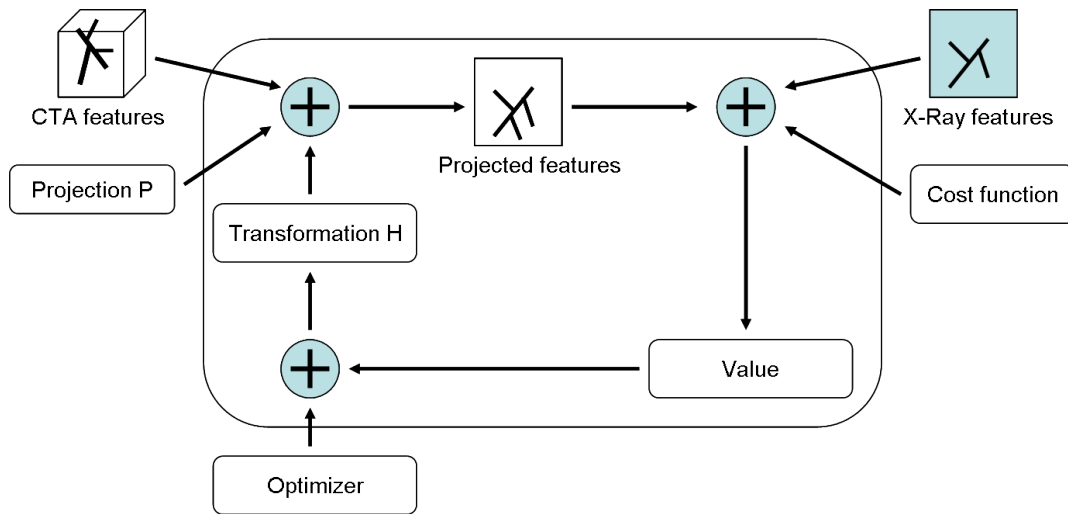


Figure 2.13.: Feature-based 2D/3D registration.

4, we will give a general overview of how 2D/3D feature-based registration works.

A crucial part of feature-based registration is feature extraction. In our example, where we use vessel centerlines for the registration task, feature extraction is the process of vessel segmentation and subsequent centerline detection. This step is the reason why intensity-based approaches might be preferable for many registration problems, since the accuracy of the final result is highly dependent on the accuracy of the extracted features. Once the features are extracted, we can start with an iterative search for the best parameters R and t which transform the 3D model such that it is properly aligned to the 2D X-Ray image. For this task we can use, for example, the iterative closest point algorithm which we have already described in section 2.4.5. The only difference is, before we can do the closest point search and evaluate the cost function, we need to project our 3D vessel model using projection matrix P to retrieve a corresponding vessel model in 2D (see figure 2.13). Here we optimize the 3D transformation based on cost function evaluations in 2D. This is not a perfect solution and might lead to inaccuracies, especially with regard to the translation parameter in z -direction. Though, it is the best thing we can do and the resulting accuracy is sufficient for many applications. Devices which are able to take two 2D images of the same anatomy simultaneously might help to further improve the result. When two images are available, a 3D model can be computed, and hence, we can neglect the projection step and do the registration on 3D features instead. An algorithm doing feature-based registration closely related to our example is introduced by Groher et al. [17]. In addition, it forms the basis for this diploma thesis. Another approach which focuses on solving the 2D/3D problem for neuro-interventional applications can be found in Sundar et al. [41].

2.6. Data Analysis

Data analysis is the act of extracting information from a data set. This can be information about quality, distribution, etc. Often, it is not enough to know the result of an algorithm. Especially in medical applications, where wrong or inaccurate results can cause serious harm, we want to know their quality or their uncertainty. Therefore, we will use these methods in chapter 5 to derive uncertainties of the algorithms developed in chapters 4 and 5. In this section, we will give a short introduction to the basic tools and concepts used in data analysis [36], and will describe them in more detail in the appropriate chapters.

For a better understanding, we will explain them based on an example scenario. Imagine we want to collect information about people visiting a new kids' movie in a movie theater which is rated FSK-6 (which means this movie can be seen by children older than 6 years).

2.6.1. Random Variable

A random variable is a function that maps the result of a random trial (which are elements of the so-called sample space) to values of the measurable space. Let us assume, we want to find out the number of adults visiting the movie. We define our sample space to include the ages $\{6..100\}$. A proper random variable A which is applicable to count the number of adults would be

$$A(\omega) = \begin{cases} 0, & \text{if } \omega < 18 \\ 1, & \text{otherwise} \end{cases} \quad (2.11)$$

where ω is the age of an observed person.

If we define a random variable A_1, A_2, \dots, A_n for each person entering the movie theater, we can count the number of adults entering the theater as the sum of A_i s using the equation

$$Adults = \sum_{i=1}^n A_i \quad (2.12)$$

2.6.2. Mean and Covariance

From the previous section, we know the number of adults visiting the movie. Now we want to know the average age of all people. In statistics, this value is called the statistical mean. To compute the average age, we introduce another random variable X which is the identity function, i.e. it maps the age of each person to its age:

$$X(\omega) = \omega \quad (2.13)$$

The statistical mean \bar{X} is then computed as

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad (2.14)$$

We are still not satisfied and want to know how the ages are spread around the average, their variability. The variance is a measure to compute such a distribution and is defined to be the average of the squared differences between data points and their mean. It can be computed as

$$Var(X) = \frac{1}{n} \sum_{i=1}^n (\bar{X} - X_i)^2 \quad (2.15)$$

A problem of the variance is that its values are specified in a different unit than its input data. This problem can be solved by taking the root of the variance. This is called the standard deviation σ and can be computed as

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (\bar{X} - X_i)^2} \quad (2.16)$$

Since statistics is so much fun, we explore the gender of the visitors using a third random variable Y . We want to know if there is any dependency between age and gender. That means we want to know, if when the number of females (Y) tend to be above its expected value, does the age (X) tend to be above its expected value, too? Or does it tend to be below? This dependency is called the covariance of X and Y and is computed as

$$Cov(X, Y) = \frac{1}{n} \sum_{i=1}^n (\bar{X} - X_i)(\bar{Y} - Y_i) \quad (2.17)$$

The higher the value of $Cov(X, Y)$, the higher is the dependency of the two variables. If $Cov(X, Y) = 0$ we say X and Y are totally uncorrelated.

2.6.3. Kernel Density Estimation

After examining the age of n persons, we want to know the probability of the next person to be at the age of 30. Therefore, we estimate a probability density function based on the previous observations. A probability density function describes the density according to which a large population is distributed. The easiest way to estimate such a density function is based on a histogram [38]. In this approach, we sum up the occurrences of a certain event and divide it by the total number of events. The probability of the next person x to be thirty can then be computed as

$$P(x = 30) = \frac{1}{n} \sum_{i \in \{j: X_j=30, j=1..n\}} X_i \quad (2.18)$$

This probability estimation might be sufficient for some applications, but it is not in most cases, since it is not smooth, and it is discontinuous.

To derive a smooth, continuous density function we use an approach called kernel density estimation [12]. A kernel is simply a function that is centered at location μ . It is usually symmetric and has finite variance. An example for a commonly used kernel is the gauss-kernel

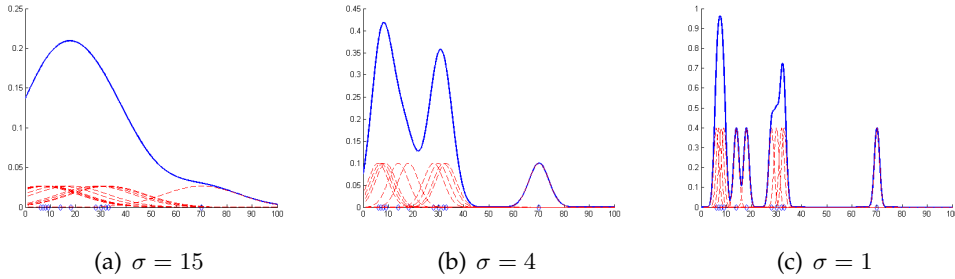


Figure 2.14.: Estimated density function using different bandwidths σ .

$$K_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.19)$$

where σ is called the bandwidth (or smoothing parameter) of the kernel. Other examples for popular kernel functions are the triangle, epanechnikov, and cosine function.

To estimate a density function, we center a kernel K at each measurement, and sum them together to get a composite curve:

$$f_K(x) = \sum_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-X_i}{\sigma}\right)^2} \quad (2.20)$$

The probability of the next person to be 30 can then be derived from function $f_K(x)$.

An important part of good density estimation is a the specification of the bandwidth parameter σ . If you take a look at the example in figure 2.14, you can see three density functions which were estimated based on the same input data, but differ in the choice of σ . The blue diamonds at the horizontal coordinate axis mark occurrences of an event. The red dashed curves are the kernel functions, and the blue solid curves are their sums. From our knowledge, we would expect a high density at the ages around 7 and 32, since most people who go to see a kids' movie are usually kids or young parents. Figure 2.14(a) shows a curve with only one extrema (also called mode) which is not a good representation for the density of the data set. We say the function is too smooth which is the result of bandwidth that is too large. Figure 2.14(c), on the other hand, shows a curve where we picked the bandwidth value too small. The best result is achieved in figure 2.14(b) which shows a good representation of the probability distribution with its two highest maxima at around 7 and 32. The task of choosing an appropriate bandwidth is not trivial. An approach to overcome this issue will be presented in chapter 5.2.

3. Related Work

The related work part gives an overview of existing techniques for image segmentation and registration that are related to this thesis. In terms of segmentation, these are techniques that deal with the extraction of blood vessels from angiographic images in 2D. For registration, these are approaches that align vessel structures of a 2D image with vessel structures of a 3D image.

3.1. Vessel Segmentation

To extract vessel centerlines from a 2D image, we can follow two schemes. We can either segment the complete vessel structure and extract the centerline from this segmentation, or we can directly explore the centerline using a vessel tracing algorithm.

Can et al. [5] propose a very fast tracing algorithm that directly processes on the image intensities of retinal fundus images. In a first step, seed points on multiple vessel branches are extracted, and used as starting points for a vessel centerline exploration. The exploration algorithm is lead by the response of a set of directional matched low-pass differentiator templates. It is also able to handle branching points and connect the extracted centerlines of different vessel branches to one connected vessel tree.

Steger [39] [40] detects curvilinear structures based on an explicit representation of lines. In a scale-space approach, using a Gaussian kernel mask and its first and second derivative, positions and widths of vessel centerline and edge points are extracted, and a bias, introduced by the Gaussian smoothing, is removed. In a second step, the extracted line points are linked to form the final centerline.

Quek and Kirbas [37] propose an algorithm based on wave-propagation. Using a dual-sigmoid function, they assign an index to every image pixel, expressing the reciprocal of its probability of being a vessel point. Points that are certain to be no vessel points are set to infinity. Afterwards, they apply a wave-propagation algorithm, starting from the base of the vessel tree. The wave travels fast through pixels with a high vessel probability (i.e. small index), and slow through pixels with a low vessel probability. The final vessel extraction is done by simply tracing the wave back along the local normals to the wavefront.

Condurache et al. [11] use an advanced form of thresholding to achieve a segmentation of vessel structures. In a first step, they enhance the vessel structures using a multiscale approach proposed by Frangi et al. [14]. This is accomplished by assigning a vesselness measure (i.e. the probability of being a vessel point) to every image pixel. This measure is derived from an analysis of the eigenvalues of the Hessian matrix. The vessel enhancement follow two consecutive threshold segmentations, called hysteresis thresholding. The first one is applied with a strong threshold

value, such that only pixels with a high vesselness factor are segmented. The second thresholding yields a segmentation of pixels with a lower vesselness factor. While the first segmentation only contains the strongest vessel points, the second segmentation contains all, or at least almost all, vessel points. Since the second segmentation also contains many background pixels, the final segmentation is achieved by selecting only those pixels from the second segmentation, that are somehow connected to pixels from the first one.

Vasilevskiy et al. [43] propose an active contours based approach called 'Flux Maximizing Geometric Approach'. A closed curve is placed in a dense vector field which is computed from the gradient of the intensity image. The inward flux through the curve, based on magnitude and direction of the gradient, defines a measurement which needs to be maximized. For vessel-like structures, where the gradient at vessel borders is perpendicular to the vessel border direction, the maximization process converges to a curve whose inward normals are everywhere aligned with the direction of the vector field. The implementation of the flow is done using the level-set method.

Another active contours approach is introduced by Nain et al. [28]. Unlike methods without shape priors, which are solely based on image intensities, and methods with high shape priors, where the segmentation is constraint to a pre-defined shape space, a soft shape prior is used. The segmentation is not constraint to a pre-defined shape space, but it is penalized if its deviance to tubular structures is too high. The advantage is, the algorithm does not lack accuracy at vessel branches or abnormalities where the tubular property is not fulfilled and is still robust to leaks where the intensity disparity of vessel and background is too small.

Even though those approaches give good results for specific vascular image types, they do not fulfill all properties for an intra-operative segmentation of blood vessels. Region growing and thresholding algorithms often require user input. Tracing algorithms can be fully automatic but they often lack in terms of accuracy and end up in non-connected centerline segments. Active contours yield very accurate results and are usually able to deal with low contrast images. However, they do not qualify for real-time applications due to speed deficits.

3.2. 2D/3D Registration

2D/3D registration plays a significant role in today's literature due to the importance of minimal invasive image-guided surgery. Recent approaches can be divided into three main topics: (a) image-to-image registration where a 3D image is projected (e.g. as DRRs) and the registration optimized based an intensity metrics, (b) in a model-to-image a model is generated from the 3D image in advance, (c) model-to-model registration requires features in 2D as well as in 3D. Those features are used for the optimization process.

Alperin et al. [1] use vessel centerlines of a 2D X-Ray image and a 3D MRI volume for 2D/3D registration. In an iterative approach, landmark vessel points in 2D and their corresponding 3D vessel points are detected, and the MRI volume transformed,

until its projected vessel points match the position of their 2D correspondences.

The approach of Feldmar et al. [13] is based on a similar idea. First, vessel centerlines of two 2D X-Ray images are extracted using the eigenvalues of the Hessian and an edge detector. An extension of the Iterative Closest Point algorithm is used to compute a rigid 3D transformation, such that the projection of a vessel tree extracted from a 3D MRI volume matches the centerlines of both X-Ray images. An outlier removal procedure yields a further improvement in terms of accuracy.

Kita et al.[22] assume position and orientation of a 3D vessel model to be approximately known. In an atlas-based approach, using the projection of the 3D model, segmentation of 2D vessel points of a DSA image is performed. To compute the registration parameters, 2D/3D correspondences are computed using Voronoi regions, and a minimization of their Euclidean distance is performed.

Chan et al. [6] focus on the real-time constraint of a registration algorithm which is similar to an intensity based approach. A pre-operatively extracted 3D vessel model is sampled as a set of spheres whose diameter represent the local vessel width. Those spheres are projected and all pixels that are covered by the projected spheres are assigned a maximum value, all other pixels are set to a minimum value. The output is a binary image which is a projection of the 3D vessel model which (in combination with an intra-operatively taken DSA) can be used to optimize the registration parameters by evaluating the sum of squared differences.

Hipwell et al. [20] generate digitally reconstructed radiographs (DRRs) from an MRA volume to register them with 2D DSAs, testing six different similarity measures.

The approaches described above solve the problem of 2D/3D registration for neurovascular images where we do not have to deal with deformations of the vascular tree. In abdominal image registration, where the transformations are not rigid anymore, they do not give satisfying results.

Groher et al. [17] proposes a method for abdominal registration based on a bifurcation in the vascular tree which is visible in 2D as well as in 3D. This bifurcation is set as an anchor point and the registration is performed around this anchor, solving for four registration parameters only, three rotational and one translational. This is achieved by a Iterative Closest Point registration, taking topological properties into account.

The method of Jomier et al. [21] requires a pre-operatively extracted 3D vessel model which is registered to two orthogonal fluoroscopic images. The 2D location of each projected vessel centerline point is computed and weighted with the local vessel diameter. The minimization of the sum of the weighted intensities yields the best registration.

Even though the two latter approaches achieve better results than the algorithms described above, they still have difficulties to deal with vessel branches that are visible in 3D but not in 2D.

4. Method for 2D/3D Registration

In chapter 2 we introduced some basic concepts of medical image processing, including segmentation and registration. In this chapter we will give a detailed description of our method of 2D/3D feature-based registration of a 2-dimensional DSA and a 3-dimensional CTA image, supported by a segmentation approach that yields a maximum correspondence of features of the resulting 2D and 3D vessel model. The first part describes the automatic vessel extraction technique, including a preprocessing of the input image, and the actual segmentation procedure. The second part presents our method of robust 2D/3D registration [16].

4.1. Preprocessing

In practice we cannot assume the images we work with to be perfect. This is due to the environmental conditions during the acquisition and also dependent on the imaging device's properties and quality. One reason for an imperfect image is noise. This is an irregular fluctuation of pixel values which is not connected to the actual image content. An example for a noisy image is given in figure 4.1(b). It has been generated from the image in figure 4.1(a) by adding noise. There are various approaches for noise removal. Linear filters use information of the neighboring pixels to derive a probable value for the center pixel. This is done by a convolution (see section 4.2.1) of a filter mask with the original image. The simplest version of such a filter is the mean filter which computes the mean intensity value of pixels in a given neighborhood (sum of intensities of n neighboring pixels divided by n). A better filter is the Gauss filter which computes a weighted mean, such that closer neighbors have more influence on the final result. The disadvantage of those filters is the blurring effect. While blurring might be desirable in homogeneous regions, it is not desirable on edges or similar structures. Non-linear filters do a better job in preserving edges. An example for a non-linear filter is the median filter. Instead of computing a (weighted) mean, it computes the median value of the neighbor's intensities. The median value of a set of n values is obtained by arranging them in ascending order and selecting the middle one, i.e. value $\frac{n}{2} + 1$. This leads to noise reduction while maintaining edges in the same time. Since noise is not a main issue in this thesis, we will not further outline the details.

More relevant is the problem of different intensity levels within a single object. This can be caused by changing light conditions (compare figure 4.1(c)) or, as in the case of angiographic images, by a non-uniform distribution of contrast agent. If you go back to figure 2.1(c) you can see that thicker vessels appear darker than thin vessels. This is because thicker blood vessels can hold more contrast agent than smaller ones. Additionally, the distribution alters during time since the contrast agent is

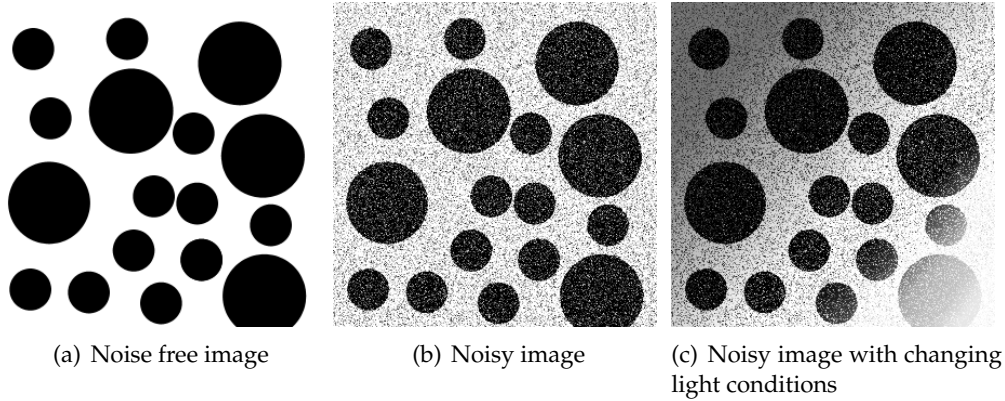


Figure 4.1.: Images of different qualities.

washed out by the regular blood flow introduced by heart beat. As a result, the intensity ratio of object and background is not constant and makes a segmentation more difficult. In the following sections we describe our method to enhance the input image before we apply any segmentation or registration algorithms.

4.1.1. Bothat Filtering

In a first step we try to remove the background from the input image as good as possible. Good results are obtained by using a bothat filter. A bothat filter (in literature sometimes also called tophat filter) is based on mathematical morphology as described in section 2.2, and is defined as the difference between the closing of an input image I and I itself:

$$I_B = (I \bullet S) - I$$

I_B denotes the bothat filtered image and S the structuring element used for the closing procedure.

To remove the background with a bothat filter, the filter takes advantage of some properties of angiographic images. Vessels usually appear darker in the image than the surrounding background and they have a certain size (i.e. diameter). To further isolate the dark vessels we select the size of the structuring element slightly larger than the maximum vessel diameter. This completely disposes the vessels from the closed image while it keeps the background information. The following subtraction therefore removes the background information and highlights vessel structures [11]. It should be noticed that vessels appear dark in the input image but light in the filtered image. If this is not desirable the output image can just be inverted. The advantage of this preprocessing step is visualized in figures 4.2(a), which displays the input image I , and 4.2(b), which displays the bothat filtered image. The window-level of the images in the figures has been adjusted such that the filtering effect is apparent. Additionally, the bothat image has been inverted for the same reason.

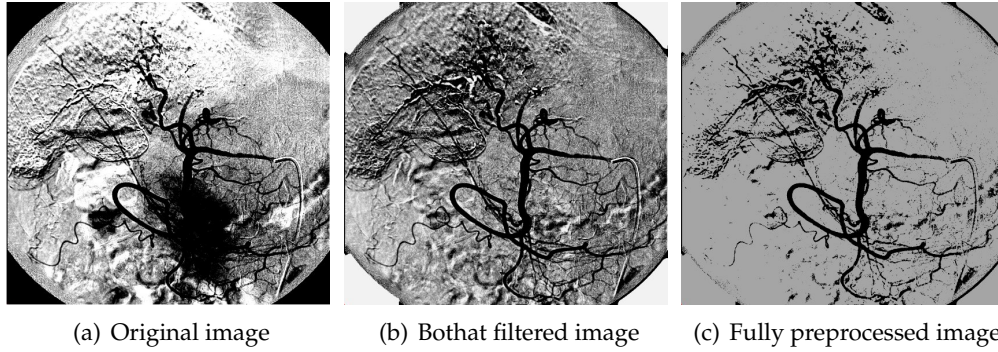


Figure 4.2.: Image during different preprocessing steps.

4.1.2. Histogram Equalization

Histogram equalization is the process of enhancing the local contrast of images where object and background have similar intensities. The process involves the uniform distribution of intensity values along the whole range of intensities. This is done using a histogram. A histogram of a gray scale image with intensity values in the interval $[0, L - 1]$ is a discrete function $h(r_k) = n_k$ with $k = 0, 1, 2, \dots, L - 1$ [15]. r_k denotes the k th gray value of a discrete image and n_k is the number of occurrences of r_k in the image. If we normalize the histogram by dividing the output of $h(r_k)$ by the total number of image pixels, n , we receive the probability of an occurrence of a pixel with intensity r_k . We therefore denote the normalized histogram as

$$p_r(r_k) = \frac{n_k}{n} \quad (4.1)$$

In images with bad contrast, the majority of pixel intensities is concentrated in a relatively small area of the histogram (compare figure 4.3(a) and 4.3(b)). For contrast enhancement we have to distribute the intensities over the whole range $[0, L - 1]$. This is done using the discrete transformation function T

$$T(r_k) = \sum_{i=0}^k p_r(r_i) = \sum_{i=0}^k \frac{n_i}{n} \quad (4.2)$$

which is also called the *cumulative distribution function*.

The equalized image can be obtained by mapping each intensity value r_k of the input image to the intensity value $T(r_k)$ of the output image. This can be summarized in the equation

$$I_{out}(x, y) = T(I_{in}(x, y)) \quad (4.3)$$

The result of a histogram equalization of the image in figure 4.3(a) is shown in figure 4.3(c) and 4.3(d). As you can see, its intensity values are well distributed in the histogram.

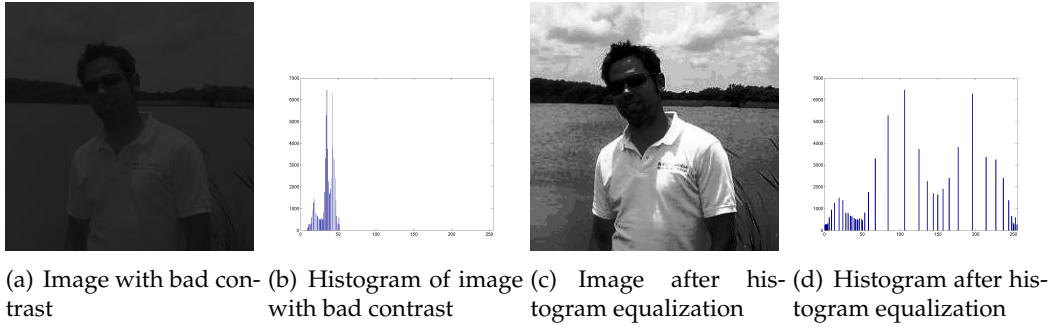


Figure 4.3.: Histogram equalization of an image with bad contrast.

4.1.3. Preprocessing of DSA Images

To further refine the enhancement of the both filtered DSA image I_B , we combine the methods from section 4.1.1 and 4.1.2 in a two step approach. The intention is to sharpen the edges of vessel structures and improve the contrast between vessel and the remaining background. The first step includes a histogram equalization of I_B resulting in the image I_{HB} . If we invert this image, we receive image $I_{HB}^{-1} = 1 - I_{HB}$ (recall we work with images whose intensities are normalized to $[0,1]$). While vessel intensities in I_{HB}^{-1} are close to zero, the background level is relatively high. We use this knowledge for a further background suppression by subtracting I_{HB}^{-1} from I_B . Background pixels, which have a high value in I_{HB}^{-1} and a low value in I_B , are thereby significantly decreased while vessel pixels, with an intensity in I_{BH}^{-1} close to zero, almost maintain their brightness level. This can be summarized as

$$I_P = I_B - I_{HB}^{-1} \quad (4.4)$$

where I_P is our final preprocessed image.

The final preprocessed image is visualized in figure 4.2(c). Please note the image in the figure has been inverted and its window-level adjusted for the reason of comparability to the other two images.

4.2. Vessel Extraction

In this section we describe the segmentation method we use to extract vessel structures from the 2D DSA image. Since the 2D segmentation is done intra-operatively, the method has to be fast and should need as little user input as possible. The approach we present in this section is based on the basic region growing algorithm described in section 2.3.4. The main enhancements to the basic algorithm are an automatic seed point detection and an automatic threshold computation. The only user input required is the specification of the maximum vessel diameter which is needed in the preprocessing step of the previous section. Since this parameter does

not change during the intervention, it can be set once at the beginning. It might even be possible to set a default value for a class of interventions. In the following subsections we will describe our method in more detail.

4.2.1. Automatic Seed Point Detection

Seed point detection is one of the key components of our segmentation algorithm. Seed points are used during the region growing procedure, both as starting points and for threshold computation. Therefore, it is necessary to obtain a high amount of accurate seed points while keeping the amount of outliers (artificial seed points) as low as possible. We will follow an approach for automatic seed point acquisition originally proposed by Collorec et al. [9] and further improved by Can et al. in [5]. It starts with an exploration of the image along a grid in horizontal and vertical direction. Potential seed points are extracted by dividing the grid lines into segments and searching for the local minimum inside each segment. To eliminate minima which do not represent a true seed point, all potential seed points are validated using several validation rules which are based on a set of derivative-mean filters. We will now describe this procedure in more detail and start with the definition of convolution.

Convolution

Linear filtering, as introduced in 4.1, is done by convolving an image with a filter mask (also called kernel or template). Convolution, denoted as $*$, is an integral which gives the amount of overlap between a function g while it is shifted over another function f . The convolution of functions f and g over the finite interval $[a, b]$ is expressed as

$$(f * g)(t) = \int_a^b f(\tau)g(t - \tau)d\tau \quad (4.5)$$

From 4.5 we can derive a discrete equation to convolve an image I of dimension $M \times N$ with a filter mask f of dimension $m \times n$

$$O(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b f(s, t)I(x + s, y + t) \quad (4.6)$$

where $a = \frac{m-1}{2}$ and $b = \frac{n-1}{2}$. The coordinates of f are relative to the filter's center point. To obtain the complete output image O , expression 4.6 has to be applied to all $x = 0..M$ and $y = 0..N$ [15]. Additional care must be taken along the image borders where the filter exceeds the image domain. An example for a convolution of an image I with a 1D Gauss kernel f is given in figure 4.4. Note the border pixels of output image O . They contain no value since we do not treat the edge pixels at all. This is usually not a good border treatment but it is sufficient for illustration. For a correct Gaussian filtering, we would also have to divide the values in O by the sum of filter elements, which is 4 in this example ($1 + 2 + 1 = 4$).

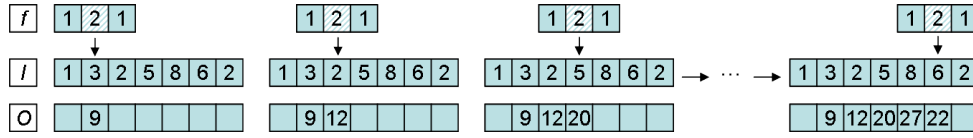


Figure 4.4.: Convolution of image I with 1D Gauss kernel f .

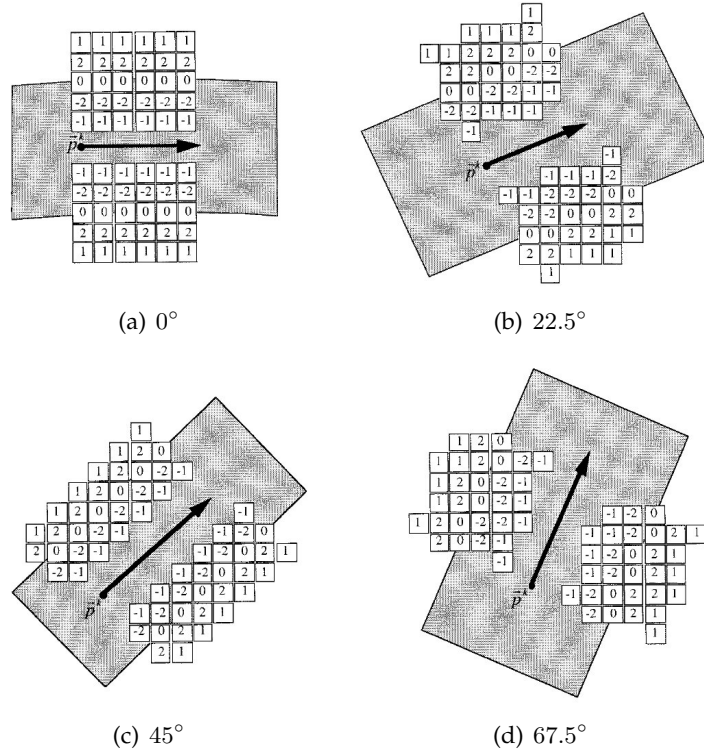
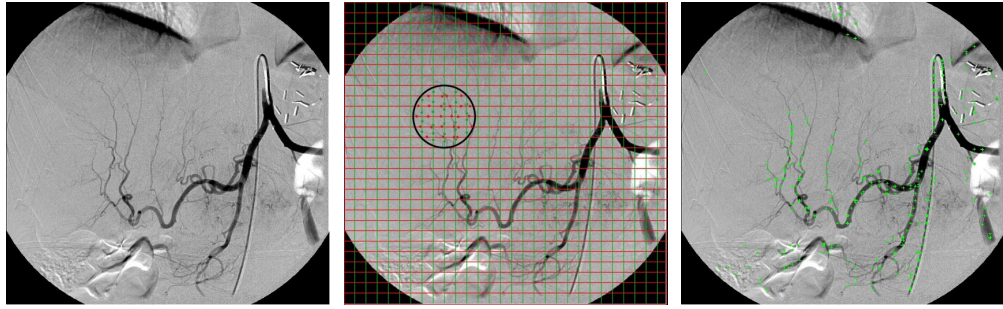


Figure 4.5.: Left and right template filters for the first 4 directions. The black arrow points along the filter direction. p^k indicates the vessel point at which the template is computed. Images are taken from [5].

Derivative-Mean Filters

The seed point validation rules we mentioned above are based on a set of linear filters. Those filters act as a derivative filter in one direction, and a mean filter in the orthogonal direction. Since derivative-mean filters yield a high response for pixels with a high gradient (change of intensities), if applied in the correct direction, we can use them to detect vessel borders. We cannot expect vessels to run in only one direction, and we therefore need a filter template for all possible directions, 0° to 360° . Because this results in an infinite number of templates, we discretize the angles with a step size of 22.5° which results in a total of 16 filter templates. As an example, the following equation specifies a kernel which is oriented along the x-direction. This



(a) Original image.

(b) Image with grid lines and potential seed points.

(c) Extracted seed points are marked as green crosses.

Figure 4.6.: This figure shows the different stages of seed point extraction. (a) Original image. (b) Extracted seed point candidates before validation. Candidates are visualized as small dots inside the black circle. (c) Extracted seed points after validation.

means it acts as a derivative filter in the y -direction and does a smoothing along the x -axis. It therefore returns a high filter response on vessel borders oriented the same direction as the filter itself.

$$h(x, y) = \frac{1}{6K} \{-\delta(y-2) - 2\delta(y-1) + 2\delta(y+1) + \delta(y+2)\} * \left\{ \sum_{i=0}^K \delta(x-i) \right\} \quad (4.7)$$

The filter template corresponding to equation 4.7 for $K = 6$ (filter length), and templates for three other directions, are visualized in figure 4.5. The figure contains a left and right filter template for each direction since we want to detect both, the left and right border of a vessel. Of course, this is just for convenience of exposition. In practice, we simply implement the right template filter and simulate the left filter response by negating the response of the right one. Moreover, these kernels are only approximations to the correct kernels resulting from a rotation of 4.7. The exact rotation would lead to non-integer weights which implies a computational slow-down. Still, the approximations yield some important properties, the sum of weights is the same for all directions, and the angular quantization error (i.e. angle between approximated and exact direction) is below 1° . The remaining eight kernels can be derived by flipping the kernels of figure 4.5 around the origin and around the direction-axis (black arrow) respectively.

Seed Point Detection

The seed point detection is carried out in a two step approach. In the first step we extract a set of potential seed points which are validated in the second step, and either discarded or accepted.

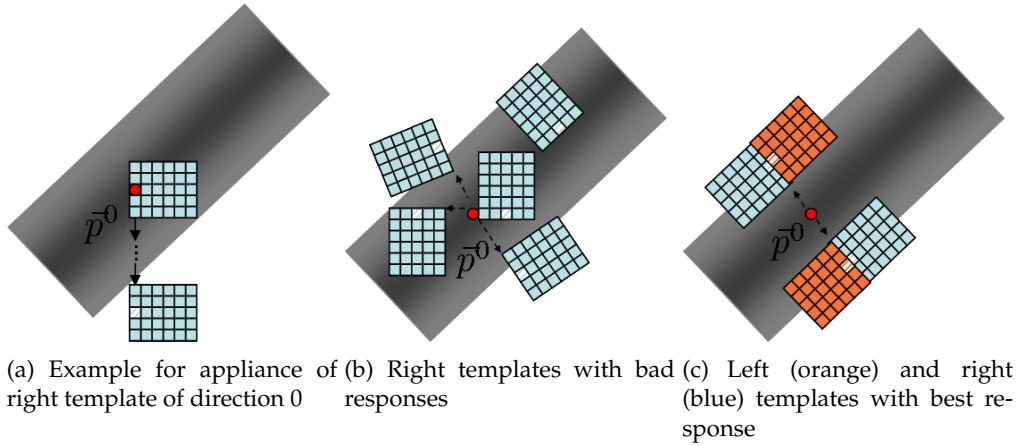


Figure 4.7.: Visualization of different templates (position and direction). The grey area represents the vessel, the red dot represents the vessel point at which the templates are computed.

Step 1 - Potential Seed Point Extraction: To extract seed point candidates, we perform a grid search on the input image. We place 1-pixel wide horizontal and vertical lines on the image and divide them into segments, such that each crossing of two grid lines marks a segment. This is illustrated in figure 4.6(b). The spacing between the grid lines should be chosen based on the maximum vessel width expected. We work with abdominal image data of size 1024×1024 whose maximum vessel width is between 25 and 45 pixels, for instance. Good results were obtained with a grid step size of 25 pixels. In a next step the intensity values of each segment are examined to determine a local minimum. To minimize the influence of noise during the examination step, we are using a 1D Gauss kernel of size three for the determination of the minimum (compare figure 4.4). It takes the intensities of its direct neighbors into account to perform a smoothing which eliminates a distortion of result. The output of the grid search is visualized as small dots in figure 4.6(b) (on the grid lines for the area surrounded by the black circle).

Step 2 - Seed Point Validation: As you can infer from figure 4.6(b), not every minimum detected in step 1 is a valid seed point. There are many segments which do not cross a vessel at all but still have a minimum, and other minima might be misled by noise which could not be eliminated by the Gauss kernel. Therefore, we use the filter templates, introduced above, to apply a set of validation rules. To understand the background of those rules, we will first explain how the filters are applied to the image. From now on, right templates of direction k will be denoted as T^k . Furthermore, if we mention templates in the following sections, we always talk about right templates if not otherwise stated. Now take a look at figure 4.7. The red dot, marked as \vec{p}^0 , represents a potential seed point that we want to validate. We start with the template T^0 and place it as shown in figure 4.7(a). Note that the origin of the templates is thereby not the center of the templates, but rather the bottom center

point pixel (visualized with a line pattern filling). Then, we evaluate the current position using equation 4.6. We keep this value and move the kernel one step outwards along the direction perpendicular to the kernel direction. For the right template this is the direction obtained by a clockwise rotation, and for the left template by a anti-clockwise rotation of 90° . To enhance the performance of the algorithm, the unit vectors representing the 16 template directions can be pre-computed as

$$\vec{u}^k = \begin{bmatrix} \vec{u}_x^k \\ \vec{u}_y^k \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{2\pi k}{16}\right) \\ \sin\left(\frac{2\pi k}{16}\right) \end{bmatrix} \quad (4.8)$$

where k represents one of the 16 directions 0..15. For template T^0 , this means, we move the kernel along the unit vector \vec{u}^3 . We iterate the moving and evaluation procedures until we reach a certain distance to \vec{p}^0 , which is half the maximum vessel width. The maximum value we obtained during this process is declared to be the filter response R^0 for template T^0 and repeated for all 16 directions, both for left and right templates. As you can see in figure 4.7(c), if the seed point candidate lies inside a vessel, this results in two maximum responses for the left (orange), and two maximum responses for the right templates (blue). These are denoted as L_1^{max}/L_2^{max} and R_1^{max}/R_2^{max} . Moreover, if \vec{p}^0 is a vessel point, the directions corresponding to the detected maxima R_1^{max} and R_2^{max} (and also L_1^{max} and L_2^{max}) should differ by approximately 180° and the directions corresponding to the left and right maxima should be pairwise the same. These observations already form 5 of the 6 final validation rules. The last rule introduces a sensitivity threshold T_{seed} which should be exceeded by the sums $R_1^{max} + R_2^{max}$ and $L_1^{max} + L_2^{max}$ respectively. Here is a summary of the six validation rules:

1. The output of the right templates must have 2 maxima R_1^{max} and R_2^{max} .
2. The output of the left templates must have 2 maxima L_1^{max} and L_2^{max} .
3. The directions corresponding to the maxima from 1. must differ by approximately 180° .
4. The directions corresponding to the maxima from 2. must differ by approximately 180° .
5. The maxima from 1. and 2. must be pairwise the same (approximately).
6. The sums of maxima from 1. and 2. must exceed a sensitivity threshold T_{seed} , i.e. $(R_1^{max} + R_2^{max}) > T_{seed}$ and $(L_1^{max} + L_2^{max}) > T_{seed}$.

If a seed point candidate fails to fulfill one of these rules, it is discarded and the validation continues for the next candidate. After the process is completed, the remaining seed points have a high probability of being true vessel points. The result of a seed point detection using this algorithm is shown in figure 4.6(c).

To compute threshold T_{seed} introduced by rule number 6, we use information obtained during the grid search. Instead of solely searching for local minima, we also collect the local maxima of each segment and combine them altogether in a set M .

This results in a set of pixels which are either probable to be foreground or background pixels. Based on this set, we compute a value that separates the two pixel classes as good as possible and define this value to be T_{seed} . This is done using an approach originally proposed by Otsu in [31], and therefore also called Otsu's method. Based on the assumption that a set of pixels contains only two classes, foreground and background, Otsu tries to find an optimal threshold which separates the two classes such that the variance of elements within each class is minimized, while the variance between the classes is maximized. As described in section 4.1.2, the probability of occurrence of a pixel intensity r_k can be computed as

$$p_r(r_k) = \frac{n_k}{n}$$

If we have one set of foreground pixels M_F and one set of background pixels M_B , we can compute the probability of occurrence for each class as

$$P_F(t) = \sum_{i=0}^t p_r(i) \quad (4.9)$$

and

$$P_B(t) = \sum_{i=t+1}^{L-1} p_r(i) = 1 - P_F(t) \quad (4.10)$$

where t is the threshold we are searching for, and L the number of grey levels of the image domain. At the beginning, t can be initialized with a random value, for example $t = \frac{L}{2}$.

If \bar{m} represents the mean intensity value over the whole image and \bar{m}_F and \bar{m}_B the mean value of foreground and background intensities, we can compute the weighted variance of each pixel class as

$$\sigma_F^2(t) = \sum_{i=0}^t p_r(i)(i - \bar{m}_F)^2 \quad (4.11)$$

and

$$\sigma_B^2(t) = \sum_{i=t+1}^{L-1} p_r(i)(i - \bar{m}_B)^2 \quad (4.12)$$

Using those equations we can describe the within-class variance σ_{intra}^2 as

$$\sigma_{intra}^2(t) = P_F(t)\sigma_F^2(t) + P_B(t)\sigma_B^2(t) \quad (4.13)$$

If we subtract the within-class variance from the total variance we get the between-class variance $\sigma_{inter}^2(t)$:

$$\sigma_{inter}^2(t) = \sigma^2 - \sigma_{intra}^2(t) = P_F(t)(\bar{m}_F - \bar{m})^2 + P_B(t)(\bar{m}_B - \bar{m})^2 \quad (4.14)$$

Since the total of $\sigma_{intra}^2(t) + \sigma_{inter}^2(t)$ is constant, a maximization of $\sigma_{inter}^2(t)$ yields a minimization of $\sigma_{intra}^2(t)$. Therefore, all we have to do is to step through all possible values for t and pick the one that maximizes $\sigma_{inter}^2(t)$. The final t is the threshold in demand, i.e. $T_{seed} = t$. This can be done efficiently using the following recursive scheme:

$$P_F(t+1) = P_F(t) + p_r(t+1)$$

$$P_B(t+1) = P_B(t) - p_r(t+1)$$

$$\bar{m}_F(t+1) = \frac{\bar{m}_F(t)P_F(t) + (t+1)p_r(t+1)}{P_F(t+1)}$$

$$\bar{m}_B(t+1) = \frac{\bar{m}_B(t)P_B(t) + (t+1)p_r(t+1)}{P_B(t+1)}$$

In the case where we deal with float values instead of integer values, we have to redefine the discretization of the image domain since the float data type can represent more than 400 million grey levels. Therefore, we define a value s which describes the range between intensity value t and value $t+1$. Then, value $t=0$ represents the intensity values in the range $[0, s]$, $t=1$ represents the values $[s, 2s]$ and so on. The smaller the value of s , the higher the accuracy. To obtain the accuracy of an 8-Bit Byte image, for example, s will be set to $s = \frac{1}{256}$.

4.2.2. Region Growing

An accurate extraction of seed points is the first of two essentials for a successful segmentation using region growing. The second important thing is a good threshold which is used to distinguish between vessel and background pixels. In the following part of this section, we will describe our method to automatically compute an adequate threshold, the actual region growing procedure, and finally, a connected component analysis which yields a segmentation containing only one single object.

Threshold Computation

The automatic threshold computation is done using tools from statistics. More precisely, it is based on the arithmetic mean and the standard deviation of the seed point intensities obtained during the procedure described in section 4.2.1. Those seed points have a high probability of being vessel points and their intensity values therefore provide valuable information about the distribution of vessel point intensities. To derive the borders of this distribution, we compute the arithmetic mean of the seed points and add/subtract their standard deviation. If $S = \{s_1, s_2, \dots, s_n\}$ is the set of extracted seed points of size n , the arithmetic mean \bar{s} of seed point intensities is computed (as described in [36]) as

$$\bar{s} = \frac{1}{n} \sum_{i=1}^n I_p(s_i) \quad (4.15)$$

The standard deviation σ of the same set is computed as

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (I_p(s_i) - \bar{s})^2} \quad (4.16)$$

The range of points which are considered vessel points during the segmentation process is then set to $[\bar{s} - \sigma, \bar{s} + \sigma]$. Since background pixels are usually brighter than vessel pixels, we can extend the interval to $[0, \bar{s} + \sigma]$ and define every pixel with an intensity value below $\bar{s} + \sigma$ to be a vessel point. This results in a single threshold

$$T_r = \bar{s} + \sigma \quad (4.17)$$

that separates object from background pixels during the region growing procedure.

Region Growing

After we have obtained all required initial parameters for a region growing algorithm, we can start the actual segmentation process. The segmentation is applied to the preprocessed DSA image I_P from section 4.1.3. In a first run, we start the exploration of image pixels from the seed point s_1 . We examine all pixels of its 8-neighborhood (compare figure 2.5) by comparing them to threshold T_r . If a pixel's intensity is below T_r , and if it has not been marked in a previous iteration, we mark it as object and put it into a queue Q . We repeat this procedure for each image point in Q until the queue is empty. When Q is empty we continue with seed point s_2 , then s_3 , and so on. When all seed points are processed and the queue is empty, the region growing algorithm stops. The image pixels that are marked as object form the segmentation of the vessel tree. A summary of this algorithm is given in figure 4.8.

Connected Component Analysis

Since we start the region growing from several seed points, it is not guaranteed that we end up with a single region. It is not even guaranteed that all segmented regions actually belong to a vessel tree. Even though we applied strict validation rules to the seed candidates obtained from the grid search, there might still be outliers in the validated set of seed points. These outliers lead to a segmentation of regions that are not part of a vessel. Fortunately, these false-detected regions are relatively small in most cases. Therefore, to get rid of them, we run a connected component analysis and only keep the biggest region, which means the region with most connected image points. In this approach we follow the assumption of the vessel tree in demand being the biggest region. Another reason for the connected component analysis will be explained in the next section where we extract a graph model from the segmented image.

```

1. extract seed points as described in 4.2.1  $\Rightarrow S = \{s_1, s_2, \dots, s_n\}$ 
2. preprocess DSA image as described in 4.1.3  $\Rightarrow I_p$ 
3. compute threshold  $T_r$ 
4. for each seed point  $s_i \in S$ 
5. do
5.1. if intensity  $I_p(s_i)$  is below  $T_r$  and  $s_i$  is not yet marked as
    object:
    a) add  $s_i$  to  $Q$ 
    b) mark as object
    c) assign region ID  $rid_i$ 
5.2. while  $Q$  is not empty
5.3. do
5.3.1. pop item  $q$  from  $Q$ 
5.3.2. examine each neighbor  $p$  of  $q$ . if intensity  $I_p(p)$  is below  $T_r$ :
    a) add  $p$  to  $Q$ 
    b) mark as object
    c) assign region ID  $rid_i$ 
5.4. od
6. od
7. determine biggest connected region:
    a) identify region ID with most frequent occurrence rate,  $rid_{max}$ 
    b) unmark all image pixels with region ID  $rid_i$  where  $rid_i \neq rid_{max}$ 
    c) remaining marked pixels form the final segmentation

```

Figure 4.8.: Region growing algorithm with connected component analysis.

The basis of the analysis is established within the segmentation algorithm. We assign a number, called region ID, to every seed point that we push in queue Q . Every pixel that we add to the queue during the neighborhood exploration will be assigned the same value. This results in all pixels that refer to the same region having the same region ID, since a region arises from a neighborhood exploration started from exactly one seed point. If, in a later iteration, we encounter a seed point that has already been marked as object, it will not be added to the queue and it will keep the region ID that has been assigned before. This avoids an occurrence of small holes in the segmented image where seed points are located after running the connected component analysis. At the end, every pixel marked as object belongs to a region with a unique region ID. The only thing left to do to extract the biggest region is counting the pixels of each region, and unmark all points which do not belong to the region with the highest amount of members. The pixels that remain marked after this procedure give us the final segmentation of the vessel tree in which all points are connected. This is also summarized in figure 4.8.

4.2.3. Model Extraction

For a feature-based registration method we need a model that represents the vessel structures in 2D and 3D. The model type we choose is a graph model. Bifurcations,

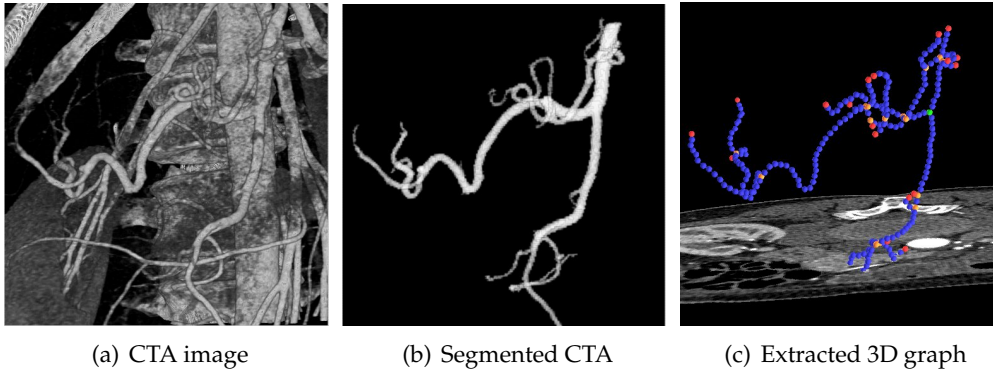


Figure 4.9.: Model extraction work flow. Images taken from [17].

where several vessels intersect, are represented as so-called A-Nodes, which is short for articulation nodes. The vessel points that connect two A-Nodes are called edges and are sampled by I-Nodes, which stands for interpolation nodes. The whole graph $G = (V, E)$ is then a set of (A-)nodes V and edges E . We extract the graph model from the segmented vessel images (both 2D and 3D) as follows. The first step involves a thinning of the segmentation to extract the vessel centerlines. This is done using the algorithm proposed in [32]. Based on the center line, we run a wave propagation algorithm [45] to create an initial graph. Due to image noise, wrong branches might be part of the vessel graph. Therefore, we use an Euclidean distance transform to remove branches which are smaller than the maximum vessel width. Bifurcations which are located very closely to each other are fused for the same reason. Last but not least, we compute the vessel diameter at each node and add this information to our final graph G . An example for the work flow and an extracted 3D graph is given in figure 4.9.

4.3. Initial Registration

As already mentioned in section 2.4.3, a registration algorithm needs to be initialized. This means, before we start the algorithm, we have to specify initial values for the registration parameters Θ , consisting of rotation R and translation t . These values have to be good enough such that the optimizer is able to direct the registration towards the correct solution. This can be done via manual user input. But since we want our algorithm to be fully automatic, we pick an approach proposed by Groher et al. in [17], and extend it to fit our initialization problem.

Recall the information from section 2.5.2. The 2D/3D feature-based registration problem involves the task of finding values for ten registration parameters, such that the projection of a 3D model, that is transformed and projected by matrix P , is spatially aligned to a 2D model as good as possible. $P = K[R|t]$ consists of the six extrinsic parameters of R and t (see section 2.4.1), and the four intrinsic parameters

of K (see equation 2.9). Just like Groher, we assume state-of-the-art imaging devices (C-arms) to be used for the acquisition of the DSA images. Those devices provide information about the intrinsic parameters, a source-to-object distance (SOD), and the rotation angles of the device. Using this information we can compute K , an initial guess of translation parameter t_z , and an initial guess of the rotation angles ψ , θ , and ϕ . This sums up to a calibrated camera (four parameters) and four initial estimates for extrinsic parameters. The two remaining parameters that we have to specify, are the translation parameters t_x and t_y . Those are estimated by an exhaustive approach, similar to the one used in [17]. Groher defines a so-called root node (3D bifurcation with largest vessel attached) in the 3D vessel model, and iteratively translates the 3D model parallel to the image plane, such that the root node is overlaid to 2D bifurcations. For each of those translations he evaluates a cost function and uses the transformation which yields the lowest cost function value as an initial guess of the translation parameters t_x and t_y . Unfortunately, we cannot use this root-based approach since we cannot assure that the 2D correspondence of the 3D root node is part of the extracted 2D vessel model. This is the drawback of a fully automatic model extraction technique as we use it in our approach. Anyway, there is a high probability of having at least one pair of matching bifurcations between the 2D and the 3D vessel model. Therefore, we do not translate a single 3D root node over the 2D bifurcations, we rather translate *every* 3D bifurcation over the 2D bifurcations. For each translation we project the 3D model into 2D and evaluate a cost function f_{ex} which is based on a ratio of inliers and outliers. For each node of the projected vessel graph (denoted x_i) we compute the distance to the 2D vessel model. This is the smallest Euclidean distance to one of the 2D sampling nodes of the model (denoted y_j). The nodes x_i which have a distance smaller than a given threshold T_d are declared inliers and their distances are summed up to the error value $E = \sum_{(x_i, y_j)} d(x_i, y_j)$, where $d(x_i, y_j)$ is the Euclidean distance between a projected point x_i and its closest 2D vessel point y_j . All other nodes are declared outliers. We set the threshold T_d to the value of the maximum expected vessel width. The cost function f_{ex} is defined as

$$f_{ex} = \frac{1}{2} \left(\frac{E}{N_{in} * T_d} + \frac{N_{out}}{N_{in} + N_{out}} \right) \quad (4.18)$$

where N_{in} specifies the number of inliers and N_{out} the number of outliers. The first part of the equation is the mean error value (mean distance) of an inlier, while the second part is the percentage of outliers in regards to the total number of nodes. For a good estimate of t_x and t_y , we need to minimize the average of those values. The need for the first part of the equation should be pretty obvious. The smaller the sum of distances, the better the registration. In addition, we need the second part of the equation since we only take the distances of inliers into account. If the maximization of inliers was not part of the equation, the final estimate might be a translation which yields only one inlier (the bifurcation itself) and therefore has the distance 0. This is not desirable and is avoided by the definition of equation 4.18.

4.4. Iterative Approach

As mentioned in chapter 1, the motivation of this thesis is to present an algorithm for 2D/3D registration that requires as little user input as possible, and, in the same time, is able to deal with the difficulties introduced by patient movement and the different ways of injecting contrast agent. Those issues mainly result in vessel structures that are only visible in one of the two images, and deformations of common vessel features. A solution to the former problem, fully automatic registration, is given in sections 4.2 and 4.3. Our approach of dealing with the latter one is part of this section.

4.4.1. Probability Map

Variation of features between the two vessel models will cause naive registration approaches to fail. Outliers will drive the registration towards a totally wrong direction, or it will, at least, lead to drastic inaccuracies in the final result. We solve this problem using a combination of segmentation and registration. In an iterative approach, we use the segmented vessel images to register the two data sets. The information obtained from the registration is used to enhance the segmentation which then leads to a further enhancement of the registration. To accomplish this task, we need to bring segmentation and registration into a common feature space. In our work, this is done with the help of a probability map. During the iteration process, our object is to improve the quality of this probability map, such that vessel features that are visible in both data sets, are assigned high probability values, while background and outliers (vessels only visible in one data set) are assigned low probability values. This is done by a fusion of information gained from image preprocessing, and a so-called distance map (or distance transform), which is computed from the result of the registration. The distance map provides information about the closest distance of an image pixel to an image feature (i.e. pixels that are marked as object). In our case, we define the features to be the projected centerlines of the 3D vessel graph. An example of a distance transform is given in figure 4.10.

Since the computational effort of the exact Euclidean distance transform is relatively high, we use an algorithm proposed in [4] which is based on a Chamfer metric. The Chamfer distance reduces the global problem of distance computation to a local problem by repeated propagation of local distances. Euclidean distances are approximated, such that the maximum difference between Euclidean and Chamfer distance is minimized. Borgefors suggests to do a forward- and a consecutive backward scan with the two non-linear filter masks shown in figure 4.11(a). The zero pixel of the mask is placed on every pixel of the image successively, and the intensities of the image pixels that are covered by the mask are added to the corresponding mask value. This results in five sums of which we pick the minimum and set it as the new value for the image pixel covered by the mask's zero pixel. In a first run, we start from the upper left corner, using mask 1, and move pixelwise to the lower right corner in a row major order. This is called the forward scan. The next run is the backward scan, which is the same thing vice versa, i.e. moving mask 2 from the lower right corner

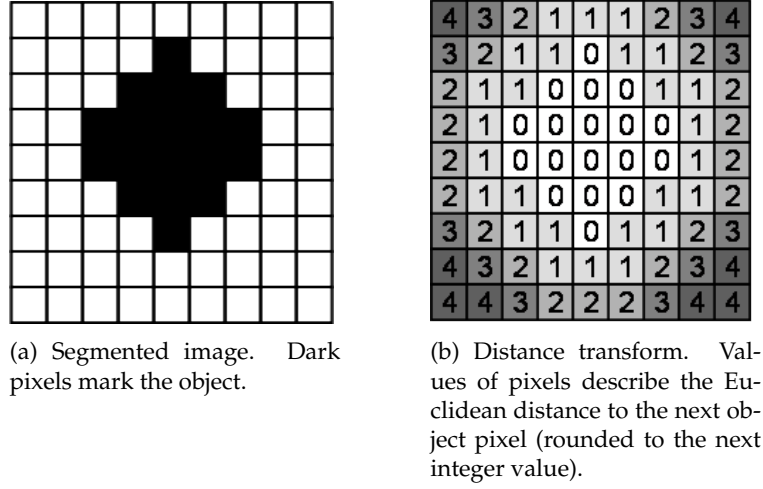


Figure 4.10.: Example for Euclidean distance transform.

pixelwise to the upper left one. This procedure is visualized in figure 4.11. If we divide the resulting integer valued distance transform by 3, we get distances which do not differ by more than $\sqrt{2} - \frac{4}{3}$ from the Euclidean distances.

If we want to use the distance map to compute a probability map, we need to convert the distances to probability values. If we suppose the probabilities of pixels on the projected centerline to be 1, and a gaussian like decreasing of probabilities, we can compute a probability map P_d from the distance map as

$$P_d(x, y) = e^{-d((x,y), M_\Theta)/\sigma^2} \quad (4.19)$$

$$\propto \frac{1}{\sigma\sqrt{2\pi}} e^{-d((x,y), M_\Theta)/\sigma^2} \quad (4.20)$$

where $d((x, y), M_\Theta)$ is the distance of a distance map pixel (x, y) to the projected 3D centerline M_Θ (approximated by values of Chamfer distance transform) and σ is the deviation proportional to the maximum expected vessel width and dependent on the quality of the current registration, i.e. cost function evaluation.

The second input for the computation of the final probability map is the preprocessed image from section 4.1.3, image I_P . Since the values of I_P are normalized to $[0, 1]$, its intensities can be interpreted as probabilities. If we assume the probabilities of P_d and I_P to be independent, we can combine them to form the equation of the final probability map P_v which is

$$P_v(x, y) = P_d(x, y) \cdot I_P(x, y) \quad (4.21)$$

$$= e^{-d((x,y), M_\Theta)/\sigma^2} \cdot I_P(x, y) \quad (4.22)$$

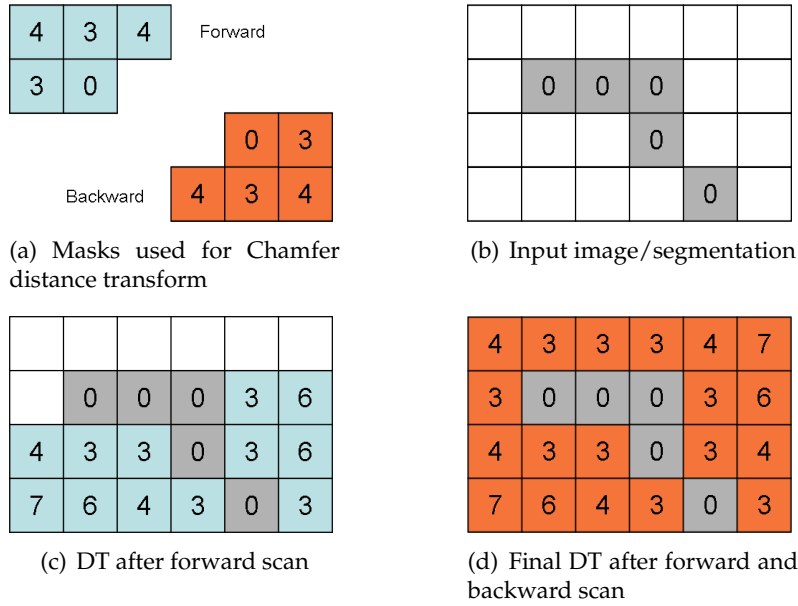


Figure 4.11.: Chamfer distance transform computation.

To obtain a 2D vessel model which is highly related to the 3D vessel model, we apply the segmentation algorithm from 4.2 to the probability map P_v . This means, the threshold computation and the region growing algorithm are executed on the intensity values of P_v . The seed point detection from 4.2.1 is not repeated since we can still use the previously extracted seed points. However, we will exclude seed points from the threshold computation which yield an intensity value of zero in the probability map. They cannot be vessel points and would only lead to a deterioration of threshold T_r . This whole procedure is illustrated in figure 4.12. If you compare the initial segmentation of figure 4.12(a), which has been created without a probability map, with the probability map based segmentation of figure 4.12(h), you can observe that the features of 4.12(h) are much better related to the features of the 3D model of figure 4.12(b).

4.4.2. Iteration

The iterative part of the algorithm comprises an alternation of vessel model extraction and registration refinement. We use the estimated registration parameters of section 4.3 to compute a probability map P_v as described in the previous section 4.4.1. The 2D vessel model extraction of 4.2 is then directly applied to P_v . The resulting vessel graph is highly related to the 3D model, and the initial parameter estimates are already close to the global optimum. We use it for an optimization of the registration parameters which is an iterative process that alternates between the determination of correspondences and the evaluation of the non-linear cost function

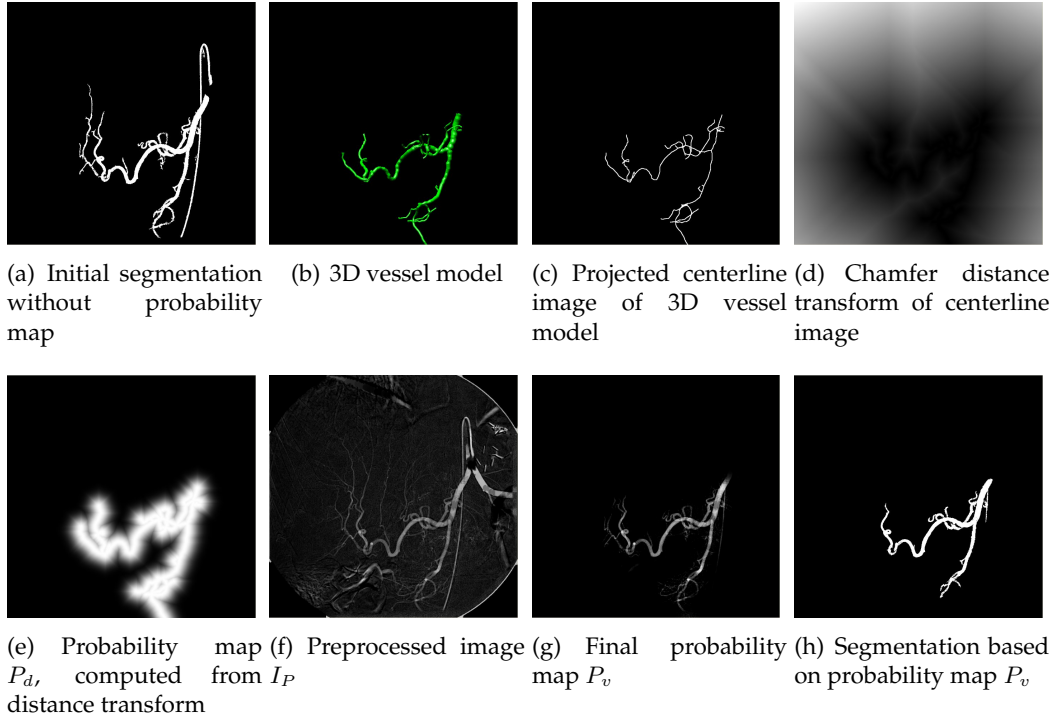


Figure 4.12.: Probability map based segmentation process.

$$f(\Theta) = \sum_{(x,Y)} \epsilon_{\Theta}(x, Y) \quad (4.23)$$

$$= \sum_{(x,Y)} d(x, P_{\Theta}Y) \quad (4.24)$$

which is the sum of Euclidean distances of points x and the projections of their correspondences Y using projection matrix P_{Θ} . P_{Θ} consists of the intrinsic parameters K , derived from the imaging device, and the extrinsic parameters R and t , summarized in the pose Θ .

To determine the correspondences, we search for closest point matches between the 2D vessel model M_{2D} and the projection of the 3D vessel model with P_{Θ} . The projected model is denoted as M_{Θ} . The closest point matching is done by iterating through the sampling nodes of M_{2D} , and searching for the sampling node of M_{Θ} that solves equation 2.4. The currently processed node of M_{2D} corresponds to x , and the set of sampling nodes of M_{Θ} corresponds to Y . In a first implementation, we wanted to assure one-to-one correspondences. Therefore, if the search returned a node y that was already matched to another node, the matches were updated such that y was assigned to the node which yielded the smaller point distance. This leads to a loss of robustness compared to n-to-one correspondences, and we thus use an

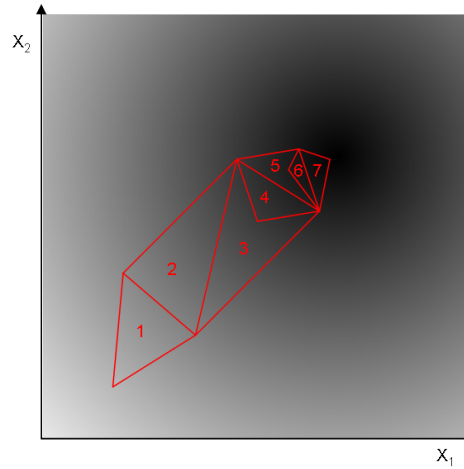


Figure 4.13.: Downhill Simplex optimizer.

implementation of the latter one.

The optimization itself is driven by a Downhill Simplex optimizer introduced by Nelder and Mead [29]. The optimizer is based on a geometric structure, the Simplex, which is an n -dimensional polytope with $n + 1$ vertices. Each vertex represents a configuration for a function's n parameters. The Downhill Simplex optimizer starts with $n + 1$ initial vertices for which the cost function is evaluated. The worst vertex, the one that yields the worst cost function value, is replaced by a new vertex which is obtained by reflecting the bad point through the centroid of the remaining vertices. If the new vertex is worse than the old one, we shrink the step size, which means we pick a vertex between the new and the old one. If the new vertex yields a better value than all remaining vertices, we do not keep it either, but rather replace it by a vertex that is even further away from the old one, i.e. we increase the step size. Those steps are repeated until we reach a given convergence criteria. A visualization of the Downhill Simplex approach for two parameters is given in figure 4.13. The initial Simplex (a triangle in 2D) is marked with 1, all following with an increasing number. The intensity level of the image pixels represents the quality of the parameter estimates. The darker the pixels are, the better is the result of the evaluated cost function

The result the registration is an optimal configuration of pose Θ , such that the current model M_{2D} and model M_{3D} , projected with P_{Θ} , are spatially aligned. This configuration of Θ is also supposed to be a better estimate than the initial configuration which we used to compute the probability map P_v . Thus, we use the new configuration to compute an improved probability map and start the vessel extraction algorithm again, using the new P_v . This should yield a further improvement of M_{2D} in terms of corresponding features compared to model M_{3D} . A better vessel model implies a better 2D/3D registration result which, once more, can be used to compute

a better probability map. This iteration continues until we reach a convergence criteria. The convergence criteria we use, is based on the segmentation of P_v . We say two segmentations are very similar, i.e. the difference is visually insignificant, when the absolute pixel difference is smaller or equal than 0.05% of the total pixel amount. For an image of dimension 1024×1024 , this results in a threshold of 524 image pixels. This value might be adjusted depending on the image size, and the requirements for speed and accuracy. The full algorithm structure, including initialization phase and main loop, is summarized in figure [4.14](#).

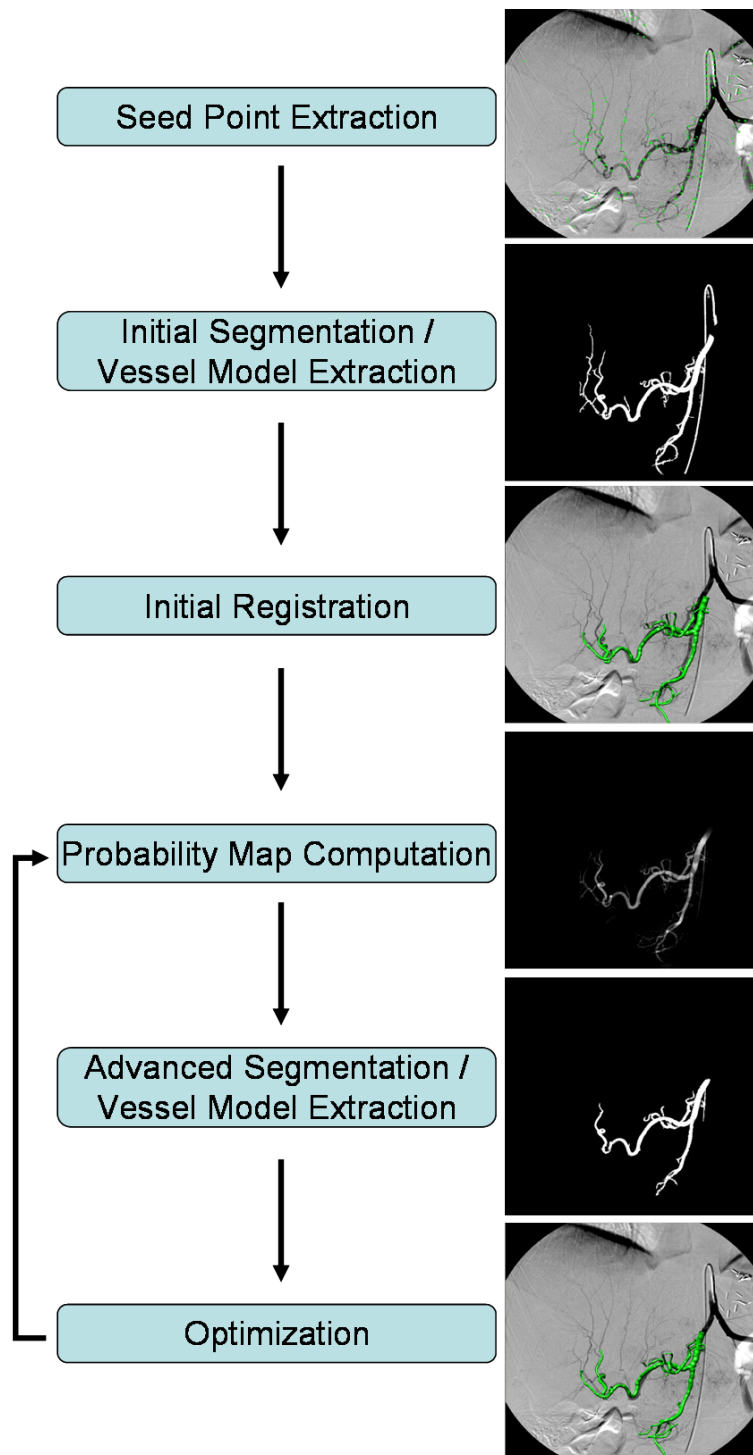


Figure 4.14.: Algorithm overview.

5. Dynamic Roadmapping

The 2D/3D registration of chapter 4 can be seen as an initial step for a procedure called dynamic roadmapping. During minimal invasive catheter-guided interventions physicians have to rely on 2D information to navigate the catheter to a region of interest. Multiple branches and vessel crossings make this a difficult task. Current approaches to solve this problem are based on pre-operatively generated road maps that are overlaid onto the fluoroscopic images. The challenge here is to keep the spatial alignment between the road map and the moving 2D image while the patient is breathing. Even if this issue can be solved, it still does not solve the problem of vessel crossings, and the respiratory motion is still visible which might be distracting for the physician during the navigation process. Dynamic roadmapping is an alternative with many advantages. The idea is to compensate the respiratory motion in 2D, detect the catheter tip with respect to a static 2D reference frame, and compute its correct 3D position within a pre-operatively extracted 3D vessel model. If we can display the catheter tip in 3D in real-time, we can transfer the whole navigation process into 3D-space which solves all the problems mentioned above. The following sections will give a detailed description of our method for dynamic roadmapping.

5.1. Respiratory Motion Compensation

As already mentioned, we want to find the catheter tip in the current fluoroscopic image frame, and compute its 2D position with respect to a static 2D reference frame. We define this reference frame as the coordinate system of the first frame of the fluoroscopic sequence. This determination proceeds in two steps. The first step involves the detection of the catheter tip in the current fluoro image. This can be accomplished in many ways, for example via magnetic tracking where a magnet is attached to the catheter, or via optical tracking where image processing methods are applied to detect the tip. Those methods might differ in terms of accuracy and might have several advantages or disadvantages over each other, but either yield the catheter position in the current fluoro frame which we denote as x_i . Here, x_i is the current catheter tip position x with respect to fluoro frame i . In a second step, we want to compensate the motion which is introduced by patient breathing (compare figure 5.1). If we are able to compute this breathing motion, represented by the displacement \vec{d}_i , we can derive position x_0 . A recent approach for real-time respiratory motion tracking has been proposed by Atasoy [2]. In this approach, the user selects an image region in which the catheter is visible and has a distinguishable shape. This region is used as a template and tracked over time using an intensity based similarity measure for evaluation. The difference between the position of the matching region and the position of the selected template gives us the respiratory motion displacement. In

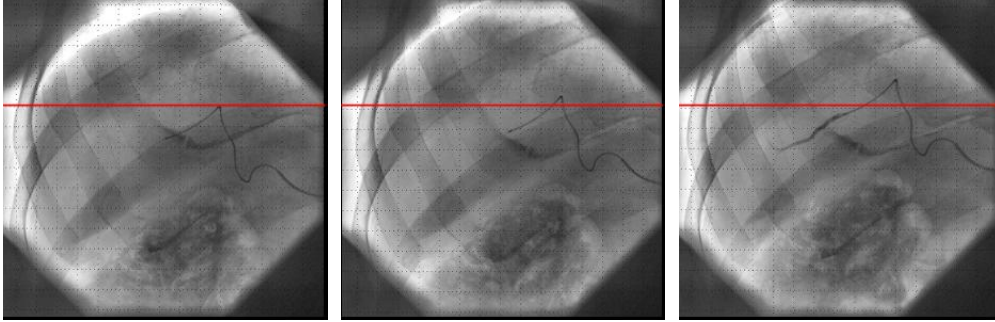


Figure 5.1.: Catheter moving over time due to patient's respiratory motion.

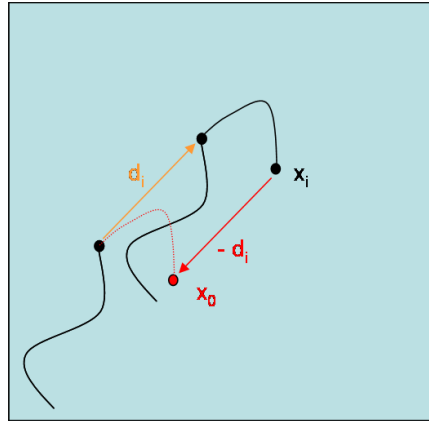


Figure 5.2.: Respiratory motion compensation of frame i .

order to avoid relative error accumulation and to deal with possible catheter deformations, template updates are performed if the similarity value drops below a given threshold. For each fluoro frame, we obtain the current catheter tip position x_0 by subtracting displacement \vec{d}_i from position x_i . This is visualized in figure 5.2.

5.2. Evaluation of 3D Catheter Tip Position

If we assume a conformity in spatial position and orientation between the DSA image and the first fluoro frame, we gather a correct relationship between the fluoro frame and the 3D vessel model by performing a 2D/3D registration between the vessel model and the DSA. Since the catheter tip position we derived in the previous section is computed with respect to the first frame's coordinate system, projection matrix P is also valid for all frames of the fluoro sequence. This is essential for a search for the 3D correspondence of the computed 2D catheter tip. The most trivial

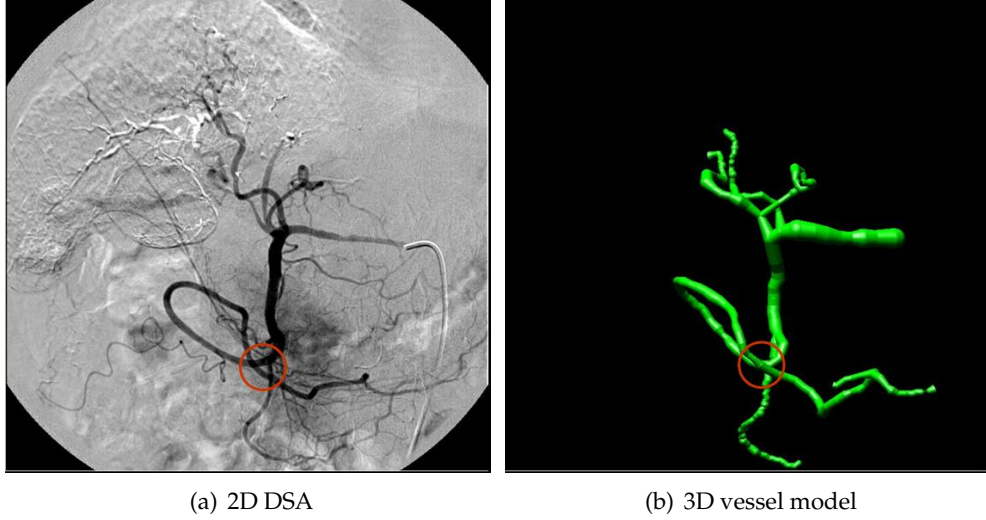


Figure 5.3.: Multiple 3D points map to the same 2D point (red circle).

way to obtain a 3D correspondence would be as follows: We use P to back-project our 2D position x_0 into 3D-space. While the mapping PX of a 3D point X into 2D is unique, its inverse $P^{-1}x$ from 2D into 3D is not. More precisely, the inverse of P does not exist at all since P is a 3-by-4 matrix. We rather need to compute the pseudo-inverse of P which is defined as

$$P^+ = P^T(PP^T)^{-1} \quad (5.1)$$

Still though, P^+x is not a unique mapping because multiple 3D points map to the exact same point in 2D (see encircled region in figure 5.3). Therefore, back-projecting x_0 using P^+ results in a 3D line rather than a 3D point. In the trivial approach, we would try to minimize the line-to-model distance, i.e. the 3D correspondence is defined as the solution to the equation

$$X = \underset{Y}{\operatorname{argmin}} d(P^+x, Y) \quad (5.2)$$

where Y is a point of M_{3D} .

Unfortunately, the solution to this equation does not have to be unique, and it is not even sure that the correct correspondence is a solution at all, due to deformation.

We therefore introduce a statistical approach that takes the uncertainty of registration parameters and motion compensation into account.

5.2.1. Uncertainty Propagation

The result of the motion compensation of section 5.1 is an estimation of the displacement \vec{d}_i . Since it is only an estimate, we cannot guarantee its correctness. More

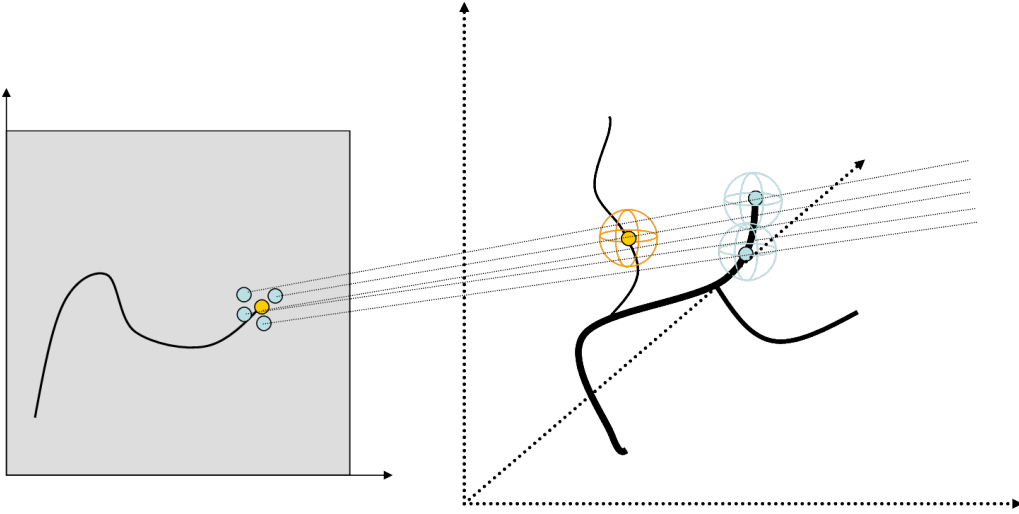


Figure 5.4.: Uncertainty propagation of motion compensation and registration.

precisely, there is always a given uncertainty connected to our result. Depending on the template we choose, this uncertainty might be smaller or larger in one direction compared to the other. For example, imagine a template which contains a horizontal line running from the very left border to the very right one. If we match this template to an image that contains the same line but longer, we will get a good result with a low uncertainty for the y -direction, but a result with a high uncertainty in the x -direction. This is because several estimates for parameter x yield a similar response from the similarity function. We receive a good estimate of this uncertainty if we compare the value of the similarity function for multiple sampling points around the template-matching result.

In our approach, we do not compute this uncertainty. Still, by modifying the parameters of the computed displacement \vec{d}_i within a small neighborhood, we generate a set of n sampling points $\{s_1, \dots, s_n\}$. We evaluate the similarity function for each modified displacement \vec{d}_{i_k} , normalize it to the interval $[0, 1]$, and use it as a weight w_k^{2d} for the generated sampling point s_k . We use the trivial approach described above to compute a 3D correspondence for each sample s_k , denoted S_k . This is visualized in figure 5.4.

Since registration is also just an estimation of registration parameters, it involves an uncertainty as well. This uncertainty is propagated to the vessel point locations that are translated by pose Θ . In a next step we want to compute this uncertainty of point locations for all 3D samples S_k . At first, by modifying the registered pose Θ randomly within a small neighborhood, we generate a set of m poses Θ_j . If we translate each sample S_k using the slightly modified poses Θ_j , we get n sets of sampling points $\{S_{k_j}\}$, where $k = 1..n$ and $j = 1..m$. We assign a weight w_j to each point

S_{k_j} , where w_j is computed as the reciprocal of the closest point cost function evaluated as described in section 4.4.2 using pose Θ_j . These weights are again normalized to the interval $[0, 1]$. The weighted samples are then used to compute the desired uncertainty, represented by the weighted covariance matrix of the sample sets. We compute the weighted mean and covariance matrix as described in [8] as

$$X_k = \bar{S}_k = \frac{1}{\sum_{j=1}^m w_j} \sum_{j=1}^m w_j S_{k_j} \quad (5.3)$$

and

$$C_k = \frac{1}{\sum_{j=1}^m w_j} \sum_{j=1}^m w_j (S_{k_j} - \bar{S}_k)(S_{k_j} - \bar{S}_k)^T \quad (5.4)$$

The computed covariance matrices with the computed means as their center points are visualized as ellipsoids in figure 5.4.

5.2.2. Non-Parametric Information Fusion

In a final step we fuse the information about mean and covariance to obtain one single point which we define to be the best estimate for the catheter tip's 3D location. We use an approach proposed by Comaniciu [10]. This approach takes the uncertainty of an estimate into account and, at the same time, is robust to outliers. Since we may have measurements that are located on different vessel branches with very different covariances, these are important properties to solve our problem. Comaniciu introduces an approach called Variable-Bandwidth Density-Based Fusion (VBDF). The idea is to estimate a density function for all measurements and define the fusion estimator, i.e. an estimate for the location of our 3D catheter tip position, as its most significant mode. To detect this mode, we use a method called variable-bandwidth mean shift. For gaussian kernels, mean shift is an iterative, hill-climbing approach, that moves a given point X towards a location with an equal or higher density in a local neighborhood, and converges to a location with zero gradient. Therefore, it is also called a mode-seeking algorithm [7]. We use variable-bandwidth to keep the algorithm nonparametric and improve the kernel estimator by adapting the scale of the kernel. The bandwidth we choose is based on the covariance matrix of each sample point, and on a scaling factor α . This scaling factor is initialized with a large value compared to the point spread of the measurements, and decreased iteratively over time. The resulting bandwidth matrix that is used to specify our density function is defined as

$$H_k = H(X_k) = C_k + \alpha^2 I \quad (5.5)$$

where C_k is the covariance matrix computed for sampling point X_k , and I is the identity matrix. Based on this, we define our sample point density estimator, i.e. the density function, as

$$f_v(X) = \frac{1}{n(2\pi)^{d/2}} \sum_{k=1}^n \frac{1}{|H_k|^{1/2}} \exp\left(-\frac{1}{2}D^2(X, X_k, H_k)\right) \quad (5.6)$$

where $D^2(X, X_k, H_k)$ is the Mahalanobis distance between sample point X_k and point X which is defined as

$$D^2(X, X_k, H_k) \equiv (X - X_k)^T H_k^{-1} (X - X_k) \quad (5.7)$$

The variable-bandwidth mean shift vector that is used to detect the most significant mode of f_v for each scale is defined as

$$m_v(X) \equiv \left(H_h(X) \sum_{k=1}^n w_k(X) H_k^{-1} X_k \right) - X \quad (5.8)$$

H_h is the data-weighted harmonic mean of the bandwidth matrix and is computed as

$$H_h(X) = \left(\sum_{k=1}^n w_k(X) H_k^{-1} \right)^{-1} \quad (5.9)$$

Each sample point X_k is associated with a weight w_k which is recomputed for each scale and is defined as

$$w_k(X) = \frac{\frac{1}{|H_k|^{1/2}} \exp\left(-\frac{1}{2}D^2(X, X_k, H_k)\right)}{\sum_{k=1}^n \frac{1}{|H_k|^{1/2}} \exp\left(-\frac{1}{2}D^2(X, X_k, H_k)\right)} \quad (5.10)$$

where the weights satisfy the condition $\sum_{k=1}^n w_k(X) = 1$.

If we choose the initial bandwidth large enough, i.e. we assign a high value for α , we receive a density function with only one mode. This is important, since we do not have any information about a good initial value for X . If we have a density function with only one mode, we can pick any random value for X and the mean shift algorithm will still converge to the desired mode. This position can then be used as an initial value for X at the next iteration with a decreased scale factor. These steps are repeated until a scale factor of zero. The detected mode of this iteration is declared to be the most significant mode of the density function, and therefore, the final result for the fusion estimator. An example for a 2D case is illustrated in figure 5.5. Input for our algorithm is a set of measurements visualized as blue crosses in figure 5.5(a), and an associated uncertainty visualized as ellipses around the measurements. Figure 5.5(b) shows a plot of density function f_v for an initial scale factor $\alpha = 30$. As you can see, the resulting density function contains only a single mode which is marked with a blue cross. This mode is tracked over scales until we detect the most significant mode at scale $\alpha = 0$. This is not necessarily the highest mode of the function as you can infer from figure 5.5(f). The highest mode is located at position (20, 120), but it is not the most significant mode if tracked over scales. This behavior describes the VBDF's robustness to outliers.

To make this algorithm even more reliable, we will add an extension to the original VBDF which includes information about the 3D catheter tip position of the previous

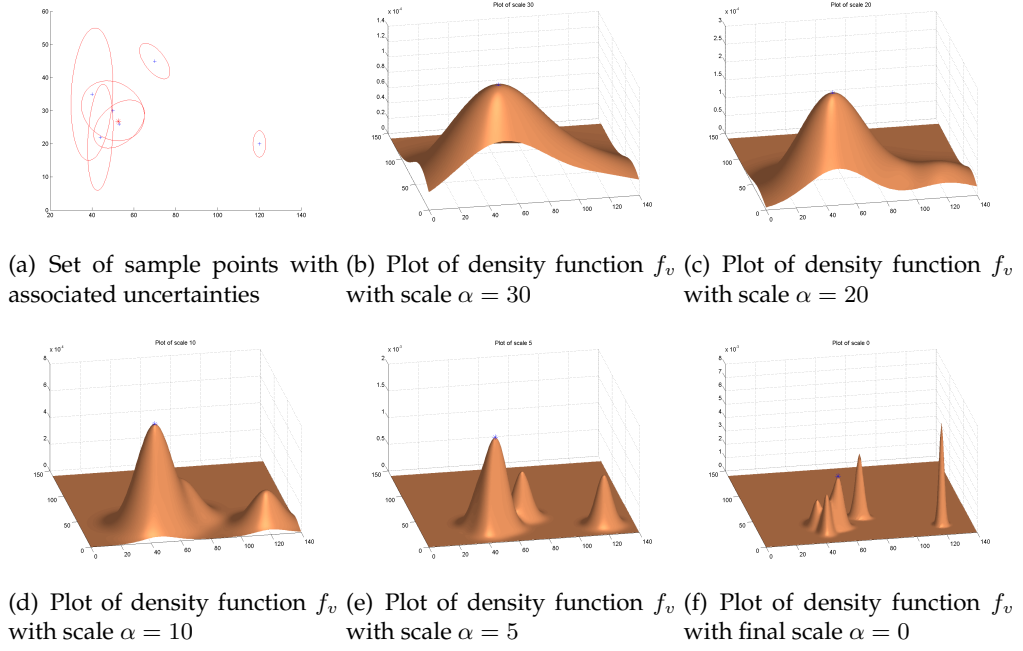


Figure 5.5.: Plot of density function f_v for different scaling factors α .

frame, and the weight of the 2D point that was used to compute S_k and accordingly X_k . This extension involves a modification of the weights w_k . The idea is to give a higher rating to sample points X_k which lie on the same or a connecting vessel branch as the catheter tip of the previous frame. This is justified with the fact, that the catheter tip cannot jump between totally different vessel branches in-between two successive image frames. To combine all three weights to one single weight, we normalize them to the interval $[0, 1]$. These values can be interpreted as probabilities, and thus joined by a simple multiplication, assuming the weights to be independent. This results in the following weights:

1. $w_k^{vb} = \begin{cases} 1.0, & \text{if } X_k \text{ on same branch as last } X \\ 0.2, & \text{otherwise} \end{cases}$
2. w_k^{2d} of section 5.2.1
3. Equation 5.10

The final weight w_k is computed as

$$w_k(X) = w_k^{vb} * w_k^{2d} * \frac{\frac{1}{|H_k|^{1/2}} \exp\left(-\frac{1}{2}D^2(X, X_k, H_k)\right)}{\sum_{k=1}^n \frac{1}{|H_k|^{1/2}} \exp\left(-\frac{1}{2}D^2(X, X_k, H_k)\right)} \quad (5.11)$$

For a better understanding a summary of the algorithm is given in figure 5.6.

- | |
|---|
| <ul style="list-style-type: none"> • input point set X_k with associated uncertainties C_k • output location X of most significant mode • initialize <ol style="list-style-type: none"> 1. Initialize X with random value. 2. Set α depending on sample set distribution. • iterate Repeat until $\alpha = 0$ <ol style="list-style-type: none"> 1. Compute H_k, w_k, and H_h for current scale 2. Detect mode for current scale, i.e. move X using m_v iteratively until $m_v(X) = 0$ 3. Decrease scale factor α |
|---|

Figure 5.6.: Variable-Bandwidth Density-Based Fusion.

Last but not least, we add a final constraint. This requires the catheter tip to be inside a vessel. Since this is not guaranteed after performing the statistical evaluation described above, we pick the closest point to our computed X and define this point to be the final 3D catheter tip position.

The complete process of dynamic roadmapping, including the initial 2D/3D registration, is summarized in figure 5.7.

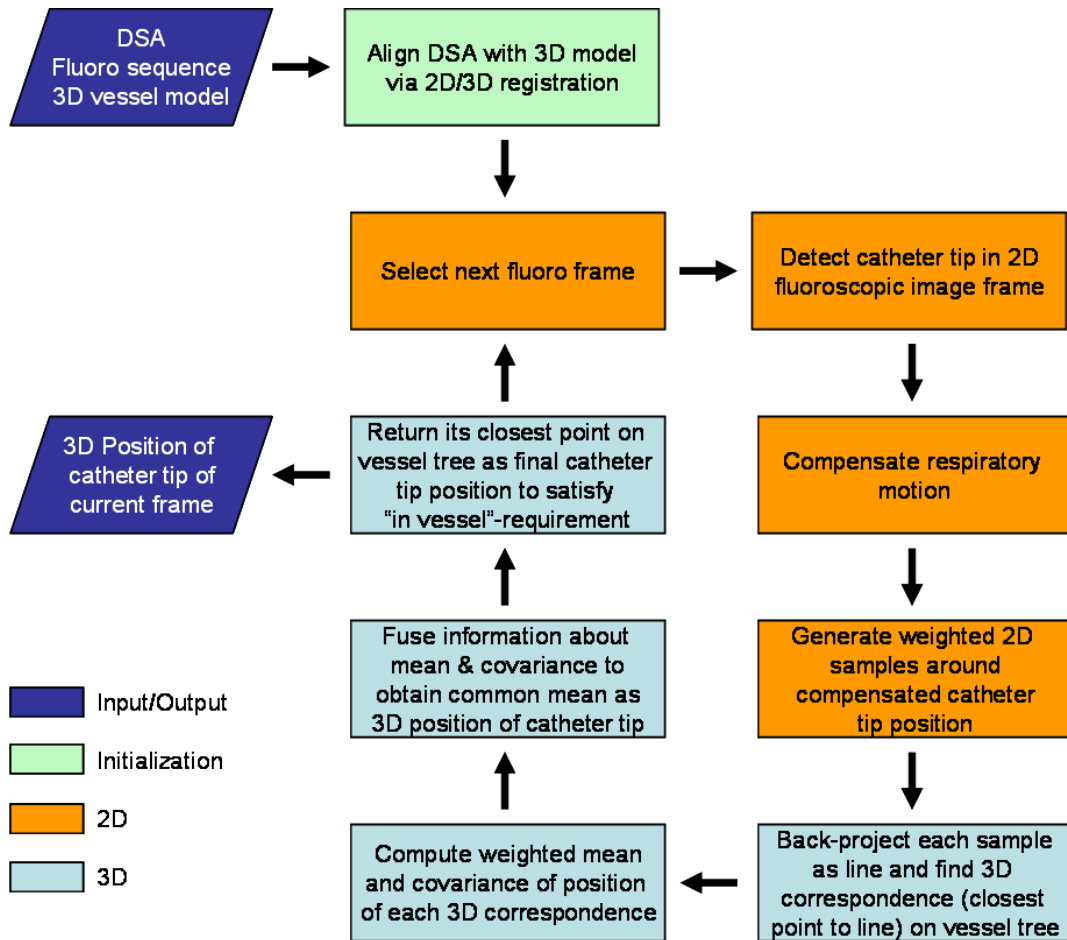


Figure 5.7.: Dynamic roadmapping process.

6. Evaluation and Results

This chapter deals with the evaluation of the previously described algorithms. Especially for medical applications, this is an important step. In the first part we evaluate our 2D/3D registration algorithm and measure its accuracy, robustness, and quality of convergence. The second part evaluates the correctness of detected 3D correspondences based on simulated data sets.

6.1. 2D/3D Registration

Our registration method is tested on five different data sets. The data sets are acquired from patients that suffer from hepatocellular carcinoma (HCC) and are treated with transarterial chemoembolization (TACE) (see chapter 1).

The result of the registration procedure is visualized in figure 6.1. The upper row shows the input DSA images, the lower row shows the registered 3D vessel models overlaid onto the DSAs. In figures 6.1(d) and 6.1(e), you can see large deformations between the vessels in the 2D image and the vessels of the 3D model. Figure 6.1(f) shows vessels in 3D that are not visible in 2D (e.g. the large vessel at the lower left). While simple registration methods would fail due to those issues, our algorithm is still able to cope with them and provides a good registration.

6.1.1. Evaluation

To evaluate the accuracy and robustness of our method, we asked experienced radiologists to define reasonable registration parameters, and create a proper vessel segmentation. These results were defined to be our ground truth for the evaluation process. In a random study, we modified the six registration parameters in an interval of 5 degrees for the rotational, and 5 mm for the translational part, around the ground truth. The modified poses were taken as input values and the registration was invoked. This process was repeated 200 times, and the standard deviation (σ) and the root-mean-squared error (ϵ), compared to the ground truth, computed. For a quality measurement, the methods of Jomier et al. [21] and Groher et al. [17] were implemented and compared to our algorithm. The result is summarized in table 6.1 which was taken from [16], the publication of our registration method.

If you take a look at the average values at the bottom row of the table, you can see that our algorithm achieves the best result for all parameters but the z-translation. For algorithms whose input is a single 2D image, the accuracy issue of the z-translation is very common. If high accuracy is required in z-translation as well, the algorithm could be improved by taking a second DSA as input image.

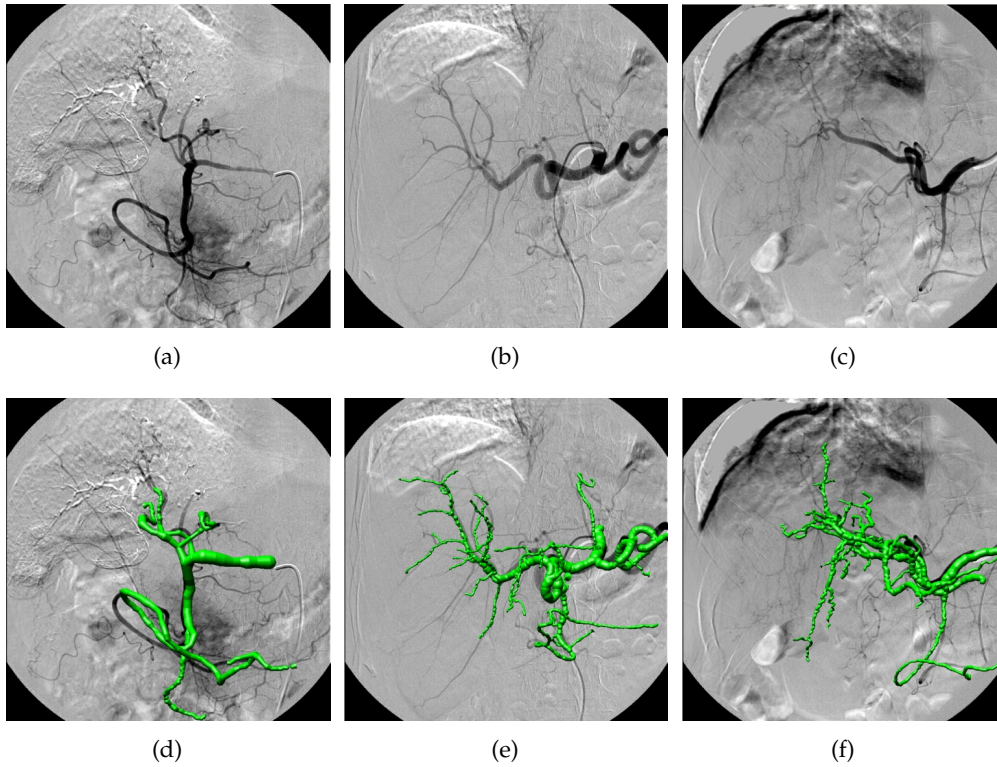


Figure 6.1.: Five tested data sets. Input images from top row, are overlaid with registered 3D vessel model in lower row.

From the last two data sets you can infer that the methods of Jomier and Groher have problems with scenarios as shown in figure 6.1(f), where the structure of the 3D model is significantly different to the 2D data. The ability to deal with such difficulties is one of the key properties of our approach, and we therefore still obtain a valid registration.

The runtime of our algorithm, however, shows the need for improvement. The total runtime (analyzed on an Intel Core2Duo 2.6 GHz machine) for a complete registration lies between 1.5 - 2 min and requires between two and five iteration to converge (both dependent on the data set). If averaged over all five data sets, we can split the total runtime as follows: (a) (bothat-) filtering (28.5 sec), (b) seed point extraction (1.0 sec), (c) region growing (0.3 sec), (d) centerline extraction (2.7 sec), (e) exhaustive initialization (34.9 sec), (f) average of each iteration¹ (12.9 sec).

The critical stages are (a), (e), and (f). Large improvements can probably be achieved if the methods are numerically optimized, and executed at the GPU (graphics card) instead of the CPU.

¹region growing, centerline extraction, and pose optimization

Patients	σ_{t_x}	σ_{t_y}	σ_{t_z}	σ_α	σ_β	σ_γ	ϵ_{t_x}	ϵ_{t_y}	ϵ_{t_z}	ϵ_α	ϵ_β	ϵ_γ
Jomier	0.76	1.80	4.66	3.48	3.17	1.60	0.78	2.21	5.30	4.75	3.60	1.66
Groher	0.20	0.06	3.43	0.08	1.89	0.11	0.34	0.23	6.58	0.22	4.41	0.38
New Method	0.54	2.68	34.63	3.79	1.88	1.25	0.54	2.70	36.12	3.79	1.97	1.25
Jomier	0.74	0.93	5.06	4.84	3.18	0.58	0.78	3.46	5.38	8.33	3.18	0.78
Groher	0.16	1.33	4.52	4.80	2.59	1.27	1.24	1.49	28.48	5.28	3.85	2.33
New Method	0.04	0.16	1.99	0.45	0.18	0.04	0.07	0.25	3.73	0.72	0.35	0.04
Jomier	3.34	2.03	7.39	4.43	4.37	2.04	4.68	2.96	7.43	5.62	4.39	2.07
Groher	2.32	4.56	43.72	3.84	3.98	4.78	6.72	6.21	62.07	5.19	11.58	4.87
New Method	3.55	1.53	24.11	7.47	3.33	4.62	3.54	1.54	24.36	7.49	3.33	4.62
Jomier	22.45	6.65	13.73	7.74	24.64	9.15	23.18	6.84	13.81	8.40	25.41	9.16
Groher	6.30	0.89	55.10	4.55	6.82	4.09	9.05	2.02	70.69	7.54	14.73	4.99
New Method	0.95	0.18	9.41	0.29	1.96	0.10	1.09	0.19	13.04	0.36	2.62	0.10
Jomier	26.35	16.37	10.34	7.15	13.60	5.81	26.66	16.37	10.87	7.25	18.63	6.07
Groher	20.65	1.82	148.41	14.88	12.29	15.10	52.74	3.39	422.62	28.76	16.95	24.40
New Method	1.22	0.74	16.83	3.05	2.07	0.67	1.41	0.74	20.19	3.05	2.44	0.76
Average												
Jomier	10.73	5.60	8.23	5.53	9.79	3.83	11.21	6.37	8.56	6.87	11.04	3.95
Groher	5.93	1.70	51.04	5.63	5.5	5.07	14.02	2.67	118.09	9.39	10.30	7.39
New Method	1.26	1.06	17.39	3.01	1.88	1.34	1.33	1.08	19.49	3.08	2.14	1.35

Table 6.1.: Results of standard deviations σ , and RMS errors ϵ of translations (in mm) and rotations (in deg). Deviation and error in z -translation are not as significant as those in-plane, or in rotations.

6.2. Dynamic Roadmapping

To test the correctness of our dynamic roadmapping method, we need a data set which includes three components: (1) CTA, (2) DSA, (3) fluoroscopic sequence. All three must stem from the same patient. We implemented a prototype software that takes this input data and allows us to register the CTA to the DSA, and resolves the 3D position of the catheter tip while scrolling through the fluoroscopic sequence. To verify a correct 3D position, we checked the conformity of catheter tip ground truth in 2D and projected 3D catheter tip position, and tested if the 3D location stays in the same, or a connected vessel branch over time.

6.2.1. Generating Data Sets

Since we do not possess data sets with all three required components so far, we generated our own synthetic data. We used real data to generate background noise for DSA and fluoroscopic images. To simulate the vessel structures of a DSA image, we projected a 3D vessel graph onto the background image. The catheter was drawn by adding nodes and connecting them using B-Splines. In a final step, we added gaussian noise to the vessel tree and the catheter, which resulted in a more realistic DSA fake. To generate a fluoroscopic sequence, we skipped the projection of the 3D vessel graph to retrieve an image showing a catheter only. We simulated catheter movement by extending the B-Splines, and translated them in-between successive frames of the sequence. Figure 6.2 shows components of three synthetic data sets. The upper row shows the generated DSA images, the lower row shows frames of

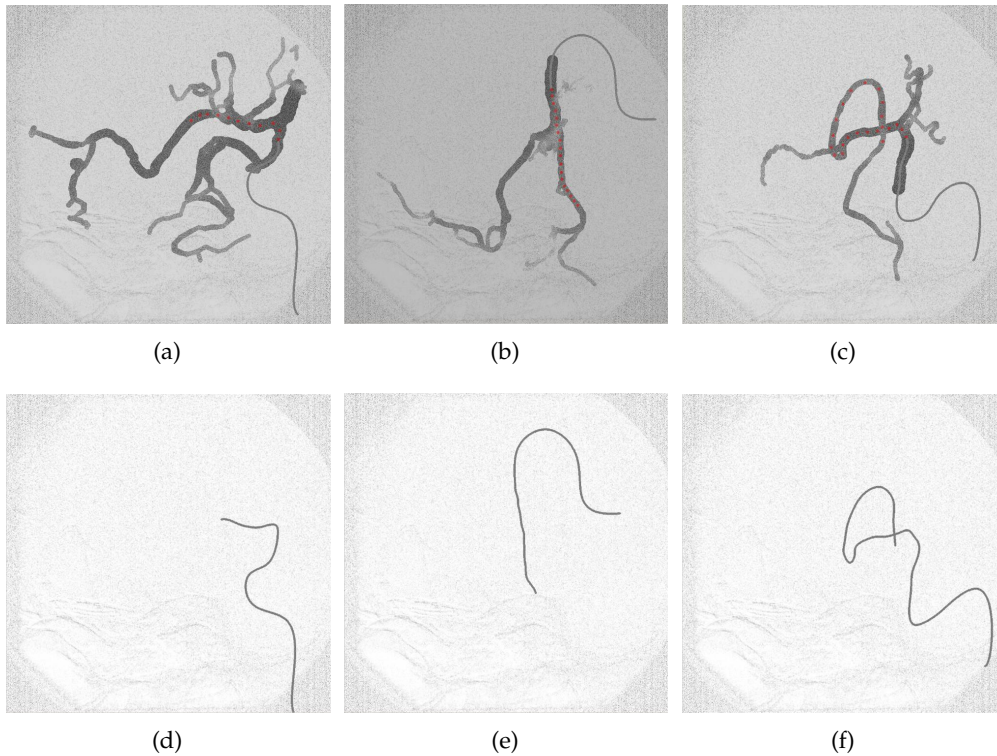


Figure 6.2.: DSA (upper row) and fluoroscopy (lower row) of three data sets.

the corresponding fluoroscopic sequences. The catheter tip positions at each frame are marked as red dots within the DSA image.

6.2.2. Evaluation

The testing started with a registration of the DSA and the 3D vessel model using the approach from chapter 4. After this initialization phase, we stepped through the fluoroscopic sequence and validated the 3D position which was computed in real-time for each frame using the method described in section 5. If the resolved 3D catheter tip position did not jump between different vessel branches, and if we had a conformity between the projected 3D position and a 2D ground truth (which we obtained by clicking on the catheter tip in the fluoro frame), we said the detected 3D position was correct. To show the advantage of our method, we also computed the closest point to a back-projected line (as described in section 5.2) and compared it to our non-trivial result. This is summarized in table 6.2.

As you can see, the statistical approach yields better results than the trivial approach. This is especially true for regions, where multiple vessel branches overlap. While our approach is able to detect the correct position, the trivial solution fails and jumps between several vessel branches in-between successive frames. For the

	Inliers	Accuracy	Inliers	Accuracy	Inliers	Accuracy	Inliers	Accuracy
Trivial	7/11	63%	16/18	88%	15/18	83%	21/22	95%
New Method	11/11	100%	18/18	100%	18/18	100%	22/22	100%

Table 6.2.: Results of resolved 3D catheter tip positions. 'Inliers' specifies the ratio of correct positions to the number of frames, 'Accuracy' specifies this ratio in percentage.

synthetic data we achieved an accuracy of 100%. This result can probably not be retained for real data, but we will still get an improvement compared to the trivial approach.

For accurate results, we have to find proper values for the number of sampling points and the sampling range for the 2D sampling of the motion compensation, and for the 3D sampling of the registration parameters. Since more sampling points result in a longer run-time, we have to find a trade-off. Good results were achieved with 50 sampling points within a range of 10 pixels in 2D, and 500 samplings in a range of 5 mm / 5 degrees for the 3D transformations.

7. Conclusion

We developed a robust method for feature-based 2D/3D registration of a DSA image and a 3D vessel model which we extract from a CTA volume. Compared to other approaches, ours keeps user-input low, and is able to deal with deformations and variability of vessel structures between 2D and 3D. This is achieved by a combination of segmentation and registration. In an iterative approach we use the segmentation to register both data sets, and use the information, obtained from the registration process, to further improve the segmentation via a probability map. The approach is also characterized by a high capture range and the freedom to choose other segmentation or registration algorithms.

We use the registration to initialize a method for dynamic roadmapping. Statistical analysis and uncertainty propagation allows us to resolve the 3D position of a catheter tip which is tracked in a fluoroscopic sequence. Compared to a trivial approach, our approach leads to a significant improvement in terms of accuracy.

Before we can integrate our approach in a clinical workflow, we have to combine the algorithms to a single application, merge it with a method for automatic catheter-tip tracking, and match it to the hardware used in the hospital. In a prototype software, we already combined the registration and the statistical approach to test our method for dynamic roadmapping. Result of this prototyping was also an innovative method for visualizing the medical data sets. On two distinct computer screens, we display 2D and 3D information simultaneously. One screen shows the on-line images acquired with the fluoroscopic imaging device, and the other one shows the 3D vessel model which is pre-operatively extracted from a CT volume. The second display also shows an arrow which points at the current 3D location of the catheter tip. Since each display defines its own coordinate system, the 3D model can be translated, rotated, and scaled as desired without interfering the second display device. The arrow thereby remains at its correct position. So far, we did not solve the problem of automatic catheter tip tracking in the fluoroscopic sequence. However, this is an essential part of the whole system. If the 2D catheter tip detection fails, the whole system fails. Therefore, we have to attach great importance to the task of finding a reliable tracking method. This does not have to be an image based tracking approach, it can also be acoustic or magnetic based.

Another issue, as already mentioned in chapter 6, is the speed of the algorithm. It has to be optimized such that it can keep up with the frame rate of the fluoroscopy. A hardware-based implementation is certainly preferable to the current software-based solution. Last but not least, everything has to be tested extensively. The previous tests must be further refined, and extended to suit a complete system test.

At the end, we have a clinical application which yields a considerable simplification of the navigation process of a catheter. It is easier to find the correct path to the region of interest, since the road map can be extracted in advanced, and used in its

original 3D form during the intervention. The result is a saving of time and effort, and therefore, a more convenient intervention for patient and physician.

Appendix

A. List of Symbols

\bullet	Morphological closing operator
\cap	Set intersection
\circ	Morphological opening operator
\cup	Set union
\ominus	Morphological erosion operator
\oplus	Morphological dilation operator
\bar{s}	Mean of sample set $\{s\}$
σ	Standard deviation
Θ	Pose vector (3 Euler angles, 3 translations)
\emptyset	Empty set
H	Homography
I	2D DSA input image
I_B	Bothat filtered image
I_P	Preprocessed DSA image
K	Camera calibration matrix
M_{2D}	2D vessel model
M_{3D}	3D vessel model
M_Θ	Projection of M_{3D} by P_Θ
O	Output image
P_Θ	Projection matrix with extrinsic parameters from Θ
P_v	Probability map of vessel structures
R	Rotation matrix
S	Morphological structuring element
t	Translation vector

B. List of Abbreviations

CP	Closest Point
CPU	Central Processing Unit
CT	Computed Tomography
CTA	Computed Tomography Angiography
DRR	Digitally Reconstructed Radiograph
DSA	Digital Subtraction Angiography
GPU	Graphics Processing Unit
HCC	Hepatocellular Carcinoma
ICP	Iterative Closest Point
MI	Mutual Information
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imaging
NCC	Normalized Cross-Correlation
PET	Positron Emission Tomography
RG	Region Growing
RMS	Root-Mean-Squared
SAD	Sum of Absolute Differences
SOD	Source-to-Object Distance
SSD	Sum of Squared Differences
TACE	Transcatheter Arterial Chemoembolization
VBDF	Variable-Bandwidth Density-Based Fusion,

Bibliography

- [1] ALPERIN, N., LEVIN, D. N., AND PELIZZARI, C. A. Retrospective registration of X-ray angiograms with MR images by using vessels as intrinsic landmarks. *Journal of Magnetic Resonance Imaging (JMRI)* 4 (1994), 139–144.
- [2] ATASOY, C. S. Real-time respiratory motion tracking: Roadmap correction for endovascular interventions. Master’s thesis, Technische Universität München, 2007.
- [3] BESL, P. J., AND MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), 239–256.
- [4] BORGEFORS, G. Distance transformations in arbitrary dimensions. In *Computer vision, graphics, and image processing* (1984).
- [5] CAN, A., SHEN, H., TURNER, J. N., TANENBAUM, H. L., AND ROYSAM, B. Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. *IEEE Transactions on Information Technology in Biomedicine* 3 (1999), 125–138.
- [6] CHAN, H., CHUNG, A., YU, S., AND WELLS, W. 2D-3D vascular registration between digital subtraction angiographic (DSA) and magnetic resonance angiographic (MRA) images. In *Proc. of the IEEE International Symposium on Biomedical Imaging* (2004), vol. 1205, IEEE, pp. 708–711.
- [7] CHENG, Y. Mean shift, mode seeking, and clustering. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1995).
- [8] CHUANG, Y.-Y., CURLESS, B., SALESIN, D., AND SZELISKI, R. A bayesian approach to digital matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001).
- [9] COLLOREC, R., AND COATRIEUX, J. L. Vectorial tracking and directed contour finder for vascular network in digital subtraction angiography. *Pattern Recognition Letters* 8 (1988), 353–358.
- [10] COMANICIU, D. Nonparametric information fusion for motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2003).
- [11] CONDURACHE, A.-P., AND AACH, T. Vessel segmentation in angiograms using hysteresis thresholding. In *Ninth IAPR conference on Machine Vision Applications* (2005).

- [12] DUONG, T. *Bandwidth selectors for multivariate kernel density estimation*. PhD thesis, University of Western Australia, Australia, School of Mathematics and Statistics, 2004.
- [13] FELDMAR, J., MALANDAIN, G., AYACHE, N., FERNÁNDEZ-VIDAL, S., MAURINCOMME, E., AND TROUSSET, Y. Matching 3D MR angiography data and 2D X-ray angiograms. In *Joint Conf., Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS)* (1997), vol. 1205, Springer, pp. 129–138.
- [14] FRANGI, A. F., NIESEN, W. J., VINCKEN, K. L., AND VIERGEVER, M. A. Multi-scale vessel enhancement filtering. *Lecture Notes in Computer Science 1496* (1998), 130–137.
- [15] GONZALEZ, R., AND WOODS, R. *Digital Image Processing*. Prentice Hall, 2002.
- [16] GROHER, M., BENDER, F., HOFFMANN, R.-T., AND NAVAB, N. Segmentation-driven 2d-3d registration for abdominal catheter interventions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (Brisbane, Australia, October 2007).
- [17] GROHER, M., PADOY, N., JAKOBS, T. F., AND NAVAB, N. New cta protocol and 2d-3d registration method for liver catheterization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2006).
- [18] HAJNAL, J. V., HILL, D. L., AND HAWKES, D. J. *Medical Image Registration*. CRC Press, 2001.
- [19] HARTLEY, R., AND ZISSERMAN, A. *Multi View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [20] HIPWELL, J., PENNEY, G., MCLAUGHLIN, R., RHODE, K., SUMMERS, P., COX, T., BYRNE, J., NOBLE, J., AND HAWKES, D. Intensity based 2D-3D registration of cerebral angiograms. *TMI 22* (2003), 1417–1426.
- [21] JOMIER, J., BULLITT, E., VAN HORN, M., PATHAK, C., AND AYLWARD, S. 3D/2D model-to-image registration applied to TIPS surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2006), pp. 662–670.
- [22] KITA, Y., WILSON, D., AND NOBLE, J. Real-time registration of 3D cerebral vessels to X-ray angiograms. In *MICCAI* (1998), vol. 1496, Springer, pp. 1125–1133.
- [23] LACROUTE, P., AND LEVOY, M. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994).
- [24] LAROSE, D. *Iterative X-ray/CT Registration Using Accelerated Volume Rendering*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2001.

- [25] MEIJERING, E. H. *Image Enhancement in Digital X-Ray Angiography*. PhD thesis, Utrecht University, 2000.
- [26] MEIJERING, E. H., ZUIDERVELD, K. J., AND VIERGEVER, M. A. Image registration for digital subtraction angiography. *International Journal of Computer Vision* 31 (1999), 227–246.
- [27] MOORE, A. An introductory tutorial on kd-trees. Tech. rep., Robotics Institute, Carnegie Mellon University, 1991.
- [28] NAIN, D., YEZZI, A., AND TURK, G. Vessel segmentation using a shape driven flow. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2004).
- [29] NELDER, J. A., AND MEAD, R. A simplex method for function minimization. In *Computer Journal* (1965).
- [30] OSHER, S., AND FEDKIW, R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [31] OTSU, N. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1979), 62–66.
- [32] PALÁGYI, K., SORANTIN, E., BALOGH, E., KUBA, A., HALMAI, C., ERDOHELYI, B., AND HAUSEGGER, K. A sequential 3d thinning algorithm and its medical applications. In *IPMI '01: Proceedings of the 17th International Conference on Information Processing in Medical Imaging* (London, UK, 2001), Springer, pp. 409–415.
- [33] PENNEY, G. P., WEESE, J., LITTLE, J. A., DESMEDT, P., HILL, D. L. G., AND HAWKES, D. J. A comparison of similarity measures for use in 2-d-3-d medical-image registration. *IEEE Transactions on Medical Imaging* 17 (1998), 586–595.
- [34] PHAM, D. L., XU, C., AND PRINCE, J. L. A survey of current methods in medical image segmentation. Tech. rep., The Johns Hopkins University, 1998.
- [35] PREPARATA, F. P., AND SHAMOS, M. I. *Computational Geometry: An Introduction*. Springer, 1985.
- [36] PRESS, W. H., VETTERLING, W. T., TEUKOLSKY, S. A., AND FLANNERY, B. P. *Numerical Recipes in C++*. Cambridge University Press, 2002.
- [37] QUEK, F. K. H., AND KIRBAS, C. Vessel extraction in medical images by wave-propagation and traceback. In *IEEE Transactions on Medical Imaging* (2001).
- [38] SILVERMAN, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall / CRC, 1986.
- [39] STEGER, C. An unbiased detector of curvilinear structures. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1998).

-
- [40] STEGER, C. *Unbiased Extraction of Curvilinear Structures from 2D and 3D Images*. PhD thesis, Technische Universität München, 1998.
- [41] SUNDAR, H., KHAMENE, A., XU, C., SAUER, F., AND DAVATZIKOS, C. A novel 2d-3d registration algorithm for aligning fluoro images with 3d pre-op ct/mr images. In *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*. *Proceedings of the SPIE*. (2006).
- [42] TOMAZEVIC, D., LIKAR, B., SLIVNIK, T., AND PERNUS, F. 3-d/2-d registration of ct and mr to x-ray images. *IEEE Transactions on Medical Imaging* 22 (2003), 1407–1416.
- [43] VASILEVSKIY, A., AND SIDDIQI, K. Flux maximizing geometric flows. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002).
- [44] WEIN, W., ROEPER, B., AND NAVAB, N. 2d/3d registration based on volume gradients. In *SPIE Medical Imaging 2005, San Diego* (2005).
- [45] ZAHLTEN, C., JÜRGENS, H., AND PEITGEN, H. Reconstruction of branching blood vessels from ct-data. In *Eurographics Workshop of Visualization in Scientific Computing* (1994), Springer.
- [46] ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision* 13 (1994), 119–152.
- [47] ZIKIC, D. Variational deformable registration for medical applications. Master's thesis, Technische Universität München, 2006.