Computer Aided Medical Procedures
(CAMP)
Prof. Dr. Nassir Navab

Dissertation

Medical Tool Tracking in Fluoroscopic Interventions

New Insights in Detection and Tracking of Tubular Tools

Tim Hauke Heibel

Fakultät für Informatik
Technische Universität München

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik
Computer-Aided Medical Procedures & Augmented Reality / I16

Medical Tool Tracking in Fluoroscopic Interventions
New Insights in Detection and Tracking of Tubular Tools

Tim Hauke Heibel

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. D. Burschka
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr. N. Navab
	2.	Prof. Dr. A. Frangi, Universität Pompeu Fabra, Barcelona, Spain.

Die Dissertation wurde am 23.09.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 11.03.2011 angenommen.

Abstract

These days, fluoroscopic imaging is the modality used most widely to guide physicians during minimally invasive abdominal interventions. They involve transarterial chemoembolization for hepatocellular carcinoma, the placement of transjugular intrahepatic portosystemic shunts or needle aspiration of solitary lung nodules, to name a few. These kinds of interventions share the task of navigating a surgical tool to a specific anatomical site. Difficulties for the physicians arise from the fact that fluoroscopy, a two-dimensional modality, is used to support an actually three-dimensional navigation task. Thus, enhanced navigation procedures incorporating three-dimensional data are desired for fluoroscopic interventions. Not only should they improve the navigation but also offer the potential to reduce treatment time and radiation exposure for the patient as well as the physician. Due to the complexity of such a system and despite previous efforts, an integrated solution does not yet exist.

Localization of medical tools and the estimation of their motion are core components of such a navigation system and this work focuses on related methods to improve the detection and tracking during fluoroscopic interventions.

To this end, different image enhancement algorithms required for the tool localization are reviewed and analyzed in a unified framework. Based on the analysis a fast and robust localization scheme is developed. Furthermore, a novel method for tracking the tools in a non-rigid environment is introduced. The tracking algorithm is based on discrete optimization and thus it is derivative free and can deal with complex models. It is fast and robust and hence complying with application specific constraints. Moreover, the presented tracking method is generic and can be extended to closed curves, which makes it applicable to other problems such as segmentation.

Based on the results, a novel respiratory motion estimation procedure is developed. It is the basis for the creation of a navigation system and is essential in order to establish the relation between the location of medical tools and three-dimensional data used for the actual navigation.

Keywords:
Biomedical Image Processing, Image-Guided Interventions, Optical Tracking,
Computer-Aided Navigation

Zusammenfassung

Durchleuchtung ist heutzutage die Modalität der Wahl, um Ärzte bei der Navigation während minimal-invasiven Eingriffen zu unterstützen. Abdominelle, minimal-invasive Eingriffe sind vielfältig und umfassen transarterielle Chemo-Embolisation (TACE) bei Hepatocellular carcinoma (HCC), die Platzierung von transjugulären intrahepatischen portosystemischen Shunts (TIPS) oder auch die Punktion einzelner Knoten der Lunge, um nur einige zu nennen. Eine Gemeinsamkeit dieser Art von Interventionen ist es, dass der behandelnde Arzt während des Eingriffs ein interventionelles Werkzeug zu einer bestimmten anatomischen Stelle, dem sogenannten Situs, navigieren muss. Schwierigkeiten bei der Navigation entstehen dadurch, dass zur Unterstützung bei dieser tatsächlich dreidimensionalen Navigationsleistung Durchleuchtung, also eine zweidimensionale Modalität verwendet wird. Demzufolge ist die Entwicklung einer verbesserten Navigationsunterstützung erstrebenswert, die im besten Fall etwaige vorhandene, dreidimensionale Daten nutzt. Diese Verbesserungen sollen nicht nur die Navigation vereinfachen, sondern möglichst auch Interventionszeiten verkürzen und damit die Belastung durch Röntgenstrahlung und dies sowohl für den behandelnden Arzt als auch den Patienten. Aufgrund der Komplexität solch eines Systems gibt es trotz verschiedenster Bemühungen bis heute noch keine in den klinischen Alltag integrierte Lösung.

Die Lokalisierung medizinischer Instrumente und die Bestimmung ihrer Bewegung sind zentrale Komponenten eines solchen Navigationssystems. Diese Arbeit beschäftigt sich mit Möglichkeiten zur Verbesserung von Methoden zur Detektion und dem sogenannten Tracking medizinischer Instrumente in Durchleuchtungssequenzen.

Zu diesem Zweck werden verschiedene Algorithmen zur Hervorhebung und Detektion von medizinischen Werkzeugen untersucht und in einem einheitlichen Framework analysiert. Basierend auf der Analyse, wird ein robustes Verfahren zur Detektion erarbeitet. Desweiteren wird in dieser Arbeit ein neuartiges Verfahren zum Tracking nicht-linearer Bewegungen von interventionellen Geräten beschrieben. Das Tracking-Verfahren ist nicht gradientenbasiert und erlaubt deshalb die Verwendung komplexer Modelle. Desweiteren ist das Verfahren effizient und entspricht somit strengen den Anforderungen der interventionellen Radiologie.

Basierend auf den Ergebnissen, wird ein neues Verfahren zur Bestimmung der Atembewegung entwickelt. Es bildet die Basis für zukünftige Navigationssysteme und ist notwendig, um eine Beziehung zwischen zweidimensionalen Positionen von Werkzeugen und den dreidimensionalen Datensätzen für die eigentliche Navigation herzustellen.

Schlagwörter:
Biomedizinische Bildverarbeitung, Bildgestützte Interventionen, Optisches Tracking, Computerunterstützte Navigation

Acknowledgments

First of all, I would like to thank Nassir Navab for his continuous support. His highly motivating spirit, his interest in the field of medical computer science and his attitude towards scientific research inspired and helped me a lot. Without his feedback and inspiration this work would not have been possible.

I also owe thanks to the Siemens Healthcare Angiography/X-Ray division (Forchheim, Germany) for their financial support. In particular I would like to thank Markus Pfister and Klaus Klingenbeck for their continuous support during my work.

I would also like to thank the people from CAMP for their part in creating the research friendly environment which I had the pleasure to work within. In particular, I would like to thank Ben Glocker for the many hours in which we discussed various problems on discrete optimization, in which we worked jointly on tracking, registration and segmentation problems and for the fun we had in our spare time. I also want to thank Darko Zikic, Andreas Keil, Olivier Pauly and Max Baust for the many fruitful discussions we had and the joint work we did together. Many thanks got to Marco Feuerstein and especially Martin Groher. Not only did they take vast amounts of work from my shoulders during the time in which I was writing my thesis but they always supported me with valuable feedback and their joint scientific experience. To all of you and those I did not mention, I would like to say thank you once more. You have made writing my thesis a great experience and have become more than mere co-workers.

I am much obliged to my family, my brother and especially my mother who supported me throughout my whole studies making this thesis at all possible. Last but not least, I would like to thank my beloved wife Elena. You have endured most of all and stood by my side the whole time, encouraging me when I lacked courage, cheering my up when I was not cheerful and listening to me when I needed to talk. You are the cornerstone of this work.

Without any of you this thesis would not have been possible.

This work is dedicated to my father.
Dr. med. Jörg-H. Heibel
(January 18, 1949 – May 28, 1998)

CONTENTS

Thesis Outline 1

1 Interventional Radiology and Navigation Procedures 5
1.1 Introduction . . . 5
1.2 Clinical Routine . . . 5
1.2.1 Imaging Modalities . . . 6
1.3 Problem Statement . . . 10
1.4 A Navigation System for Fluoroscopic Interventions . . . 11
1.4.1 Motion Compensation Module Details . . . 14

2 Previous Work 19
2.1 Enhancement of Tubular Structures . . . 19
2.1.1 Scale-Space Theory . . . 21
2.2 Detection of Guide-Wires and Needles . . . 22
2.3 Tracking of Guide-Wires and Needles . . . 28

3 Tool Detection 33
3.1 Modeling Tubular Tools . . . 33
3.1.1 X-ray Attenuation . . . 33
3.1.2 An Idealized Tube Model . . . 34
3.1.3 Typical Tube Models . . . 36
3.2 Enhancement of Tubular Structures . . . 39
3.2.1 Differential Geometry in Image Processing . . . 40
3.2.2 The Hessian and its Relation to Tube Models . . . 43
3.2.3 Properties of Tube Enhancement Filters . . . 44
3.3 Semi-Automatic Detection . . . 53
3.4 Fully Automatic Detection . . . 58
3.4.1 Canny Ridge Detection . . . 58
3.4.2 Steger Ridge Detection . . . 60
3.5 Discussion . . . 62

4 Tool Tracking 65
4.1 Modeling of Guide-Wires and Needles . . . 66
4.2 Modeling of Objective Functions . . . 69
4.3 Markov Random Fields for Tracking . . . 74
4.3.1 Optimization . . . 77
4.4 Experiments and Evaluation . . . 78
4.4.1 Error Assessment . . . 78
4.4.2 Approximation Error . . . 79
4.4.3 Deformation Space Discretization . . . 80
4.4.4 Experiments on clinical data . . . 82

5 A Simplified Motion Compensation System 87
5.1 Plugin Design . . . 87
5.2 Visualization Component . . . 89

6 Conclusion 93
6.1 Discussion . . . 94
6.2 Future Work . . . 95

A Authored and Co-Authored Publications 99

B Mathematical Extensions 101
B.1 Enhancement of Tubular Structures . . . 101
B.1.1 Sato's and Frangi's Feature Enhancement . . . 101
B.1.2 Koller's Feature Enhancement . . . 102

C Code Samples 103

List of Figures 105

References 107

Abbreviations 115

Thesis Outline

Minimally invasive procedures, in particular endovascular interventions have become clinical routine. The common task of these procedures is to navigate a medical tool like a needle or a guide-wire to an intervention situs. During this navigation task, physicians have access to instant visual feedback in form of an X-ray image stream. Besides the image data the interventionalists are solely relying on their knowledge of anatomical structures as well as haptic feedback. The actual three-dimensional pose of the tools needs to be currently reconstructed mentally. In case of endovascular interventions the mental reconstruction has to be regularly verified by injecting contrast agent into the X-ray transparent vasculature.

A possible way to aid physicians in this task is an improved three-dimensional navigation system. Two core components such a system requires are the localization and motion estimation of medical tools. Localization refers to the problem of detecting a medical tool in the images without prior information regarding its location. During the motion estimation or tracking task on the other hand side the system is initialized with the previous position of the medical tool. This prior knowledge simplifies the tracking problem such that automatic methods that require no user interaction can be designed though this is not yet possible for the tool detection. A complete navigation system requires more components which are described in detail in Section 1.4.

Difficulties with these techniques arise from the fact that as a consequence of the low energy X-rays, the images usually exhibit a low signal to noise ratio (SNR). Furthermore, since needles and guide-wires are thin one-dimensional structures, they are oftentimes hard to distinguish from the cluttered anatomical background. For tracking guide-wires and catheters another problem that arises is the highly non-linear motion the tools undergo. This is caused by patient breathing, different motions at organ interfaces, interactions of the physician, and perspective projection of transparent structures. Together, these issues make the motion estimation problem in X-ray fluoroscopy extremely difficult.

To be accepted by physicians, ease of use is an important requirement for a navigation system. In the optimal case it would not be interfering or changing existing work-flows and it would not rely on user input from the physicians. So far, no such system exists which is robust enough for every day clinical use. Especially the detection still requires user interactions. Another requirement is the real-time capability of a navigation system. Fast processing is required to keep the time intervals between observations small and because other modules belonging to a navigation system require computation time as well. In summary, a navigation system should be capable of robustly handling low-quality image data, requiring minimal user interaction and operating near real-time.

This thesis presents novel concepts to support physicians in their navigation task during fluoroscopic interventions. The work evaluates and revisits methods required to build a three-dimensional navigation system. The focus of this work lies on the detection and tracking components. In particular a novel scheme for deformable tracking of curvilinear tools such as guide-wires and needles in fluoroscopic X-ray sequences is derived and evaluated.

In the following, we give a brief outline and summary of the thesis and its chapters.

Chapter 1: Fluoroscopic Interventions and Navigation Procedures The first chapter provides an overview of fluoroscopy guided interventions. Problems physicians are facing and which can potentially be overcome or minimized by computer aided systems are discussed in detail. Furthermore, different imaging modalities used during such procedures are introduced. These image modalities are of significant importance. First, their understanding gives insights to many problems involved in the design and development of a navigation system. Second, imaging modalities are not always imposing problems but sometimes enable new ways of visualizations which are nowadays not used to their full potential. In the end, a navigation system for fluoroscopic interventions is presented. The required components of such a system are discussed and an example of the working system is given.

Chapter 2: Previous Work After introducing the problem and a potential solution, Chapter 2 gives an overview of the related work. It is divided in three sections where at first image enhancement methods for tubular or curvilinear structures are discussed. These methods are required because of the degraded image quality caused by low energy X-rays. A summary of existing tool detection methods follows this section. The section presents methods from various fields, not necessarily interventional imaging. A lot of research has focused on the detection of curvilinear structures which can be streets from aerial images and vessels or medical tools from clinical image data. The last section presents the related work on tracking methods. The approaches to solve the tracking problem can be categorized in two main branches. One branch is tackling the problem with methods driven by image features and a second branch uses approaches which are based on learned data.

Chapter 3: Tool Detection Chapter 3 on tool detection presents a novel semi-automatic strategy for medical tool delineation. The presented method is based on structural information contained in the second-order derivatives of image intensities and shortest-paths in a graph induced by the structural information. At first the mathematical properties of tools and their imaging process are presented in Section 3.1. This follows a detailed discussion and comparison of enhancement techniques of tubular structures and their mathematical formulations. Section 3.3 presents the actual method for tool delineation. An overview of fully automatic methods is given in Section 3.4. It presents two methods that can be used for the automatic generation of point sets representing line-like structures and discusses the limitations of these methods. The last Section 3.5 summarizes issues related to semi- and fully automatic methods and presents early ideas of what is required for the successful implementation of fully automatic detection schemes which are clinically applicable.

Chapter 4: Tool Tracking The main focus of the thesis is a novel scheme for the tracking of medical tools. The method is presented in Chapter 4 which is divided in several sub-sections. The first one, Section 4.1 presents how to model medical tools and their shapes mathematically. It also presents how tool motions are modeled and provides a scheme for their efficient evaluation. In Section 4.2 the objective function that defines the optimization process of the motion estimation is presented. Objective functions typically contain two components – a so called external energy which drives the optimization process and a second one, called internal energy which imposes soft constraints on the solutions. Choices for modeling internal energies are various and discussed separately. After the energy function is defined, Section 4.3 presents how to model the problem in a discrete framework in which the result is computed as a maximum a posteriori (MAP) estimate of a Markov random field (MRF) formulation. Experiments are part of the last section of this chapter. Section 4.4 contains experiments on various synthetic as well as clinical sequences. Different experiments are used to evaluate different aspects of the proposed solution and serve to support the design of the final system.

Chapter 6: Conclusion The presented methods are not solving all issues occurring during the detection and tracking of medical tools. The last chapter summarizes the work and discusses unsolved problems and parts that require further improvements in Section 6.1. The final part of the thesis, Section 6.2 provides novel ideas which can lead to further improvements during the detection and tracking. In particular it will discuss the feasibility of the integration of learning based methods into a MRF framework and provide an example of such a method.

CHAPTER

ONE

INTERVENTIONAL RADIOLOGY AND NAVIGATION PROCEDURES

1.1 Introduction

Interventional radiology is a medical branch in which diseases are treated minimally invasive using image guidance. To this end, physicians introduce small catheters, catheter-based devices or needles percutaneously without the need for surgery or general anesthesia. Application fields for minimally invasive procedures and in particular endovascular interventions are various and range from diagnostic (e.g, needle biopsies) over treatment (e.g., percutaneous transluminal angioplasty) to palliative (e.g., transarterial chemoembolization) purposes.
The common task in these procedures is to navigate a medical tool like a needle or a guide-wire to an intervention situs. This guidance task is mainly supported through different imaging modalities which provide visual feedback for the interventionalist. Along with the visual information physicians rely on their three-dimensional knowledge of anatomical structures as well as haptic feedback while tools are steered towards the desired area of interest.
Many cases which required surgical treatment in earlier days can today be treated non-surgically. The minimally invasive treatment results in reduced risk for the patient. It causes less physical trauma and leads to reduced infection rates, improved recovery time and shortened hospital stays.

1.2 Clinical Routine

Image-guided and in particular endovascular interventions have become clinical routine. Various procedures exist and they serve diagnostic, therapeutic and palliative purposes. Some common treatments are the transjugular intrahepatic portosystemic shunt (TIPS) placement, endovascular stent placement in case of abdominal aortic aneurysms (AAAs), needle biopsies and transarterial chemoembolization (TACE) to deal with hepatocellular carcinoma (HCC). The common difficulty during these procedures is the placement of a medical tool in the human vascular system. In interventional radiology this task is image guided and the physicians can be supported by various imaging modalities. Diagnosis and planning of interventions is generally supported by images of the patient. Since these images are acquired before the intervention, they are referred to as pre-interventionally acquired images. This kind of images is usually characterized by high quality and consequently high X-ray dose exposure. The dimensionality of images varies between 2D (e.g., chest X-ray) and 3D (e.g., magnetic resonance tomography (MRT) or computed tomography (CT)).

During interventions, physicians have access to so called intra-interventional images and again 2D and 3D images are used. Because it is important to reduce the radiation exposure during procedures, lower doses are common which results in degraded image quality (i.e. a low SNR). In interventional radiology fluoroscopic image sequences are most common. Their purpose is to monitor patient anatomy and the assessment of medical tool positions during the exam.

Not all image data used during interventions is of low quality. Two exceptions are digital subtraction angiographys (DSAs) and 3D rotational angiographys (3D-RAs). DSAs are commonly used and they mostly serve as roadmaps because fluoroscopic images show vasculature only when contrast agent (typically iodine based) is injected into the vessel system. Three dimensional image data such as 3D-RA on the other hand is more seldom used. The acquisition requires relatively high X-ray doses and also much contrast dye application to visualize the patient's vessel anatomy. 3D angiograms can be used to diagnose diseases (e.g., stenosis), for the verification of interventions (e.g., after balloon angioplasties) and also as roadmaps. Both modalities, DSA and 3D-RA, do not capture patient motion nor temporal changes of instrument locations which means they are not directly[1] usable for navigation purposes.

The imaging modalities mentioned above are the most common during endovascular interventions. Especially for needle interventions other modalities exist as well, in particular ultrasound is used oftentimes to guide the physician. Image properties and their corresponding acquisition processes, which are of most interest in this work, will be described in more detail in the next section.

1.2.1 Imaging Modalities

Fluoroscopy

One of the most common imaging techniques used in interventional radiology is fluoroscopy. The method provides physicians with a real-time X-ray "movie" of the patient's internal structures. Typical acquisition rates for fluoroscopy are 3.5, 7.5, 15 and 30 (continuous fluoroscopy) frames per second. Fluoroscopic imaging is used to monitor instruments like catheters, organ movement, and vessel structures if combined with contrast injection.

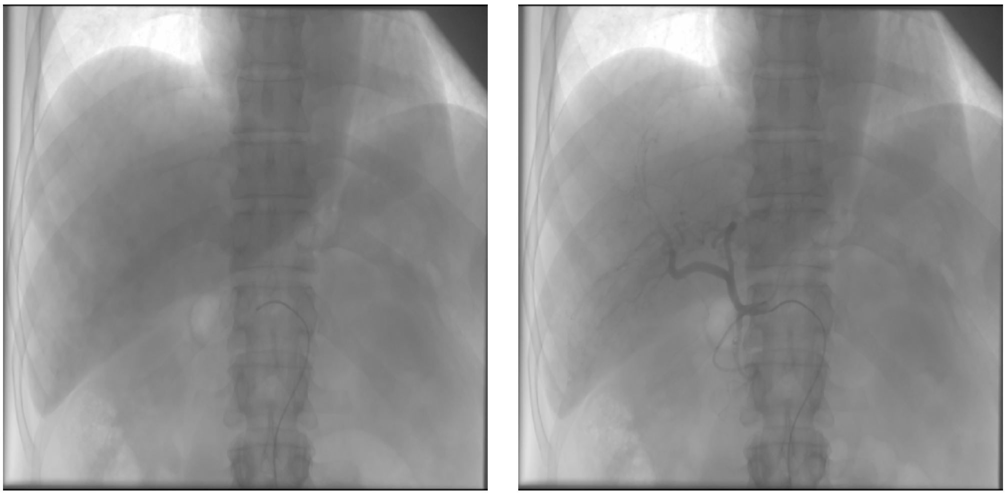
Even though, compared to classic radiography, physicians use low dose rates during fluoroscopies, the overall absorbed dose can become relatively high due to the large

---

[1] Indirectly, because they are oftentimes used as roadmaps.

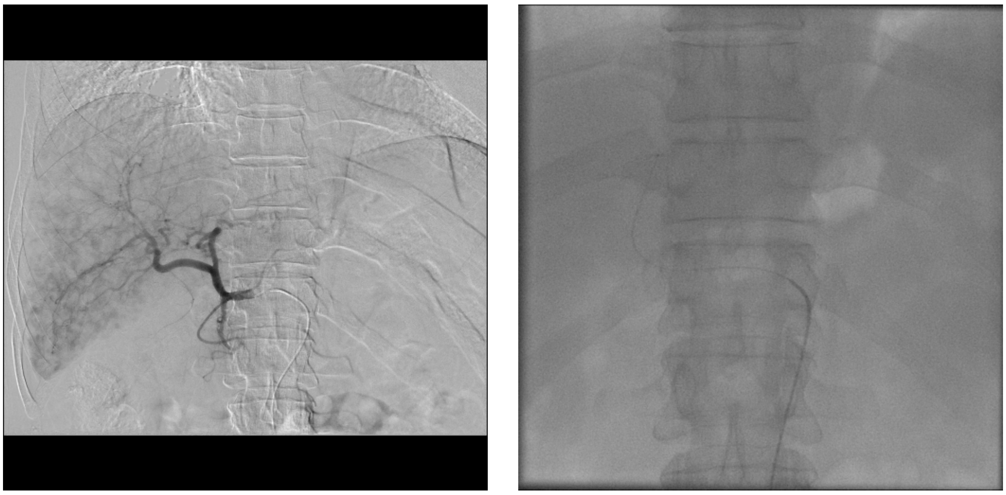
amount of images being acquired and the length of the procedure. The introduction of flat-panel detectors instead of formerly common image intensifiers lessens the problem but the procedure still requires careful balancing between risk and potential gain.

Images acquired during fluoroscopy oftentimes suffer from a low SNR. This appears as graining on the image and is caused by the low dose used for the image acquisition. The high noise level in fluoroscopy is one of the major limiting factors hampering object detection and it is directly affecting image processing in interventional radiology.



(a) Standard chest X-ray showing a catheter and no vasculature.

(b) Angriogram of liver vessels.



(c) Digital Subtraction Angiography.

(d) Fluoroscopy showing a magnification of a catheter and guide-wire.

Figure 1.1: Most common 2D image modalities in interventional radiology.

Angiography

Angiography is used for the visualization of hollow organs such as vessels, the heart or the gastrointestinal tract. This imaging technique was developed in 1927 by the Nobel Laureate António Egas Moniz. Till then no technique existed to visualize the inside of hollow organs since in general they are radio transparent.
To overcome this problem, radio opaque contrast is used. For the visualization of vessels the typically iodine based contrast dye is injected into the vessel system or administered through ingestion (e.g., barium). Shortly after the application of the contrast agent images (see Figure 1.1(b)) of the patient are acquired (computed tomography angiography (CTA), intra-interventional X-ray angiography). Alternatively, special acquisition sequences can be used to visualize hollow sequences (magnetic resonance angiography (MRA), ultrasound angiography).

Computed Tomography Angiography CT is an imaging technique where a three-dimensional image is generated from a series of two-dimensional X-ray sections. A CT system comprises several core components; the scanning unit consists of the gantry with a tube (X-ray source) and (multi-row) detector system. Other components are the patient table, the image processor and the command console.
During sequential CT acquisition, cross-sectional images are produced by scanning a transverse slice (or more; on modern systems up to 320) of the body from different angles while the X-ray source and the detector are rotating 360° around the patient. Important parameters of a CT system are the number of images that can be acquired during one rotation (number of detector rows), the rotation speed and the maximum spatial resolution at which images can be acquired. The number of detector rows affects the scanning times and the image resolution. A high rotation speed has several advantages as improved image quality, patient comfort, and reduced amount of required contrast agent during CTAs. The spatial resolution is directly affecting the size of objects that can be visualized and thus improving CT based diagnostic.
State-of-the-art scanners offer up to 64-slice detectors and achieve rotation times as low as 0.33 seconds. With these machines it is possible to acquire CT volumes with a spatial resolution of down to $0.24\,\mathrm{mm}^3$. CT scanners can visualize the typical range of Hounsfield units (HU), i.e. from -1000 HU (air, white) to ≥ 400 HU (bone, black).
CTA is a method to visualize blood vessels by contrast injection during CT scanning. The iodinated contrast material is typically injected in a peripheral vein and is rapidly distributed throughout the arterial and venous system. A region in the aorta is constantly scanned until a certain level of attenuation (≈ 150 HU) is reached, which triggers the actual image acquisition (bolus tracking). Additional scans can be done after a certain delay time. Contrast injection rates and injection duration as well as delay times used for bolus tracking and successive image acquisition, follow certain acquisition protocols that are fine-tuned not only to anatomy, patient, and disease but also to new hardware [21].

Magnetic Resonance Angiography Magnetic resonance imaging (MRI) is a primarily non-invasive imaging technique based manipulation and successive measurements of magnetic fields in the body. Because the method is not relying on ionizing radiation the imaging technique is not harmful for the patient in terms of X-ray exposure. Opposed to CT, MRI offers better contrast between different types of soft tissue which makes it useful for various clinical problems and in particular vascular and cardio-vascular imaging.

During the imaging process a strong magnetic field is used to align protons in the body. Once the protons are aligned, radio frequency (RF) fields (i.e., electromagnetic fields) are used to systematically change this alignment. When the fields are deactivated, the protons fall back into their original spin-state and the energy difference between the two states causes the release of photons with a specific frequency and energy. These photons are then measured by the magnetic resonance (MR) detector and the physical relationship between the field-strength and photon's frequency allows to use the process for imaging. Different tissues can be distinguished since protons return to their equilibrium state at different rates, which can be measured too. In order to determine the spatial location of the signals, magnetic field gradients are applied for slice selection, row selection (phase encoding), and column selection (frequency encoding).

There are several MRA techniques, some of them involving a contrast injection as CTA. Here, however, the contrast has to be paramagnetic extracellular, e.g. gaudolinium. There are also fully non-invasive techniques for angiographic MR imaging by using the flow property of blood (Time-of-Flight MRA, Phase-Contrast MRA). See Figure 1.2 for an example of contrasted CTA and MRA slices.

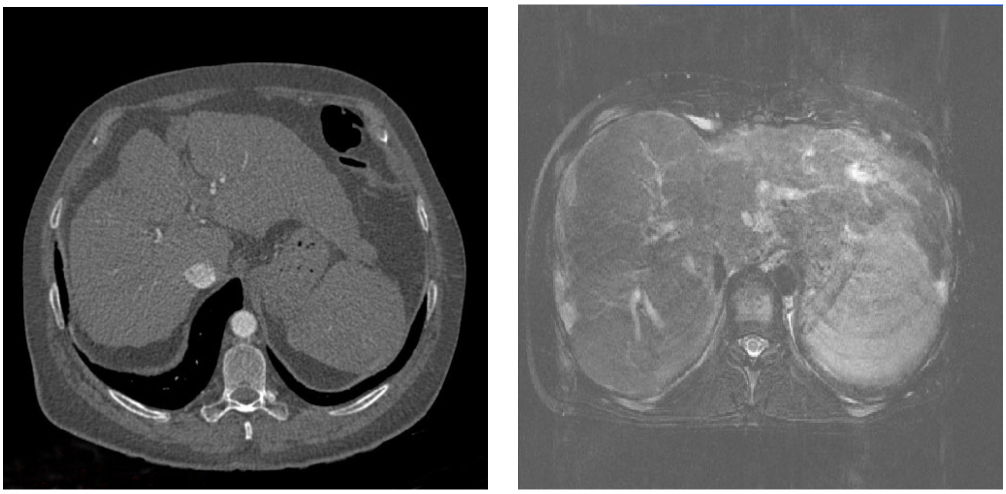


Figure 1.2: Computed Tomography Angiography (left) and Magnetic Resonance Angiography (right).

Cone-Beam Computed Tomography Angiography Cone-Beam CT which is also referred to as helical (or spiral) cone-beam CT is a technique in which an X-ray source rotates around the patient while acquiring a set of radiographies. Mathematical methods such as the filtered back projection (FBP) or algebraic reconstruction technique (ART) [37] can be used to reconstruct a three-dimensional representation of the patient. The image acquisition can e.g. be performed with a modern C-arm system in which the gantry is attached to a robot controlled arm which allows positioning with six degrees of freedom. Alternatively, smaller and mobile C-arm systems exist which allow to bring this technique into any intervention room or the operation theater.

The advantages of cone-beam CT include a reduced scan-time, dose reduction as well as improved image quality and reduced cost compared to conventional CT. From an interventionalist's point of view another improvement is that it is possible to perform the image acquisition in the intervention room without being required to move the patient.

Cone-Beam CTAs (sometimes also called 3D-RA or 3D X-ray angiography (3DXA)) are acquired by performing the image generation after the application of iodine based contrast agent. As it is common for other angiographic exams, the contrast agent is either injected or ingested.

1.3 Problem Statement

This section presents which problems physicians and patients are facing during endovascular interventions. Transarterial chemoembolizations (TACEs) serves as an exemplary procedure performed by interventionalists on a daily basis.

Hepatocellular carcinomas (HCCs) are a common disease (cancer) of the liver. TACE is usually performed on unresectable tumors. It is done as a safe palliative treatment or while patients are waiting for a liver transplant. During the procedure a mixture of an antineoplastic drug, an embolic agent and radio opaque contrast agent is inserted into the vessels feeding the liver tumor.

While the physician is performing the intervention, he inserts a guide-wire into the femoral artery in the groin. From there, the guide-wire is advanced through the aorta via the truncus coeliacus into the arteria hepatica communis and from there into the vessels feeding the actual tumor where the embolic mixture is released. Steering the

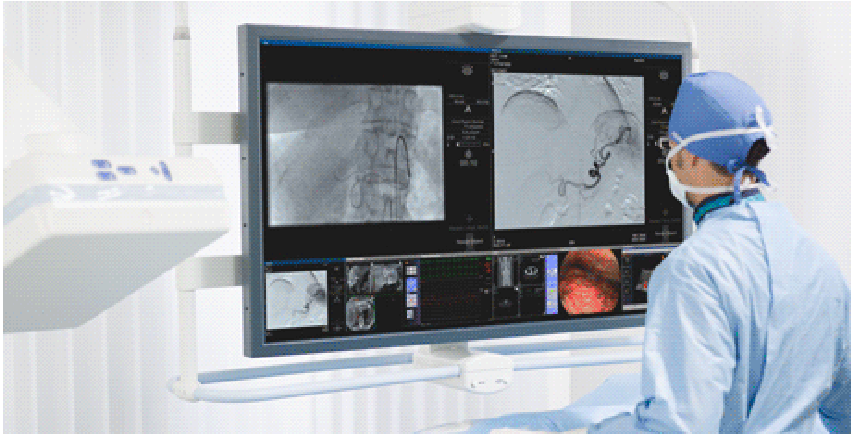


Figure 1.3: Angiographic intervention setup. The monitor shows a live X-ray video stream (left) next to a DSA (right) which serves as a roadmap during the intervention. The object on the left-hand side of the image is a flat-panel detector from a modern C-arm system.

guide-wire involves passing through several vessel branches and the main difficulty is the fact that blood vessels are radio transparent. The immediate question arising is how interventionalists deal with this difficult situation.

TACE is performed under angiographic image guidance. It is common that at first, when the patient is entering the intervention room, a cone-beam CTA is generated to assess the exact location at which the embolic mixture should be released. Next, a DSA (see Figure 1.3, right image on the monitor) is generated which serves as a roadmap during the navigation task. The actual procedure is performed under constant angiographic surveillance (see Figure 1.3, left image on the monitor). Whenever the physician assumes to be close to a branching point contrast dye is release to visualize the vessel structures in the vicinity of the catheter tip and in order to determine the next steps. The catheter itself is advanced along the guide-wire. When the intervention situs is reached the chemicals are released and the guide-wire and catheter are extracted. A standard system setup can be seen in Figure 1.3.

The current procedure involves several problems. First, it requires much experience of the interventionalist to perform the procedure in a relatively short time – and time is crucial due to the constant expose to ionizing radiation. Second, the physician needs to mentally transfer the observed flow information as perceived when injecting contrast agent to the roadmap to be able to correctly assess the current guide-wire and catheter tip location. There exist also problems related to the 2D imaging modalities used to support the navigation task. When a vessel branch is aligned with the projection direction spacial information is lost, because all 3D locations along that branch project onto the same 2D image location. There is no way of assessing the tool depth and in such a scenario, the physician must change the whole imaging setup and acquire a new DSA since it is only valid for a specific C-arm configuration.

The intention of this work is to provide solutions and new insights required to create a novel navigation system aiding physicians during endovascular procedures. Such a system should aim at improving the overall navigation. It should prevent the need for successive image acquisitions, the need for system reconfigurations (e.g., C-arm angulation) and thus reduce the intervention time and the involved exposure to ionizing radiation.

1.4 A Navigation System for Fluoroscopic Interventions

In order to support physicians during endovascular procedures, various researchers have come up with a variety of proposals. These can generally be classified into two different approaches – those using a roadmap while trying to overlay it onto a fluoroscopic image sequence [73, 52, 53, 38, 25, 1] and those trying to utilize a 3D-RA to directly annotate the tool locations in 3D [77, 8, 27]. Both approaches rely on establishing a spatial relationship between the fluoroscopic sequence and either a roadmap or a 3D-RA. One of the main problems aggravating the development of navigation systems is the dynamic nature of this relationship.

The methods aiming at blending the fluoroscopic sequence with a roadmap require the C-arm projection geometry to be known. The imaging parameters consists of intrinsic parameters as source to image distance (SID), pixel size and field of view (FOV) as well as extrinsic parameters, i.e. the C-arm orientation. Given the projection parameters, it is now possible to create the roadmap image by either creating a projection of a 3D-RA or by generating a DSA. Assuming furthermore a static scenario where the patient does not move (breathing, heart-beat) these roadmaps can directly be mixed with the fluoroscopic sequence because all data is generated under the same projection. Former research has focused primarily on precision analysis [73, 52] as well as the effects of image distortion [55, 68].

The second method is based on annotating guide-wire or catheter positions directly in 3D. Here, one needs to establish a spatial relationship between the two-dimensional fluoroscopic sequence and a three-dimensional data set such a CTA or a 3D-RA. Two ways to achieve this have been pursued so far – one approach based on biplane C-arm systems [2] and a second method which tries to compute the 3D locations from only a single view [77, 8, 27].

A common problem during the realization of such navigation systems is the dynamic nature of the observed scenes where patient breathing and heart-beat are two exemplary causes. The motion is problematic because it is not reflected in the static, pre-operatively acquired image data (e.g. DSA, 3D-RA) that is used by the interventionalists during the procedures. Dealing with the motion requires so called tracking techniques for both approaches, those using roadmaps and those annotating

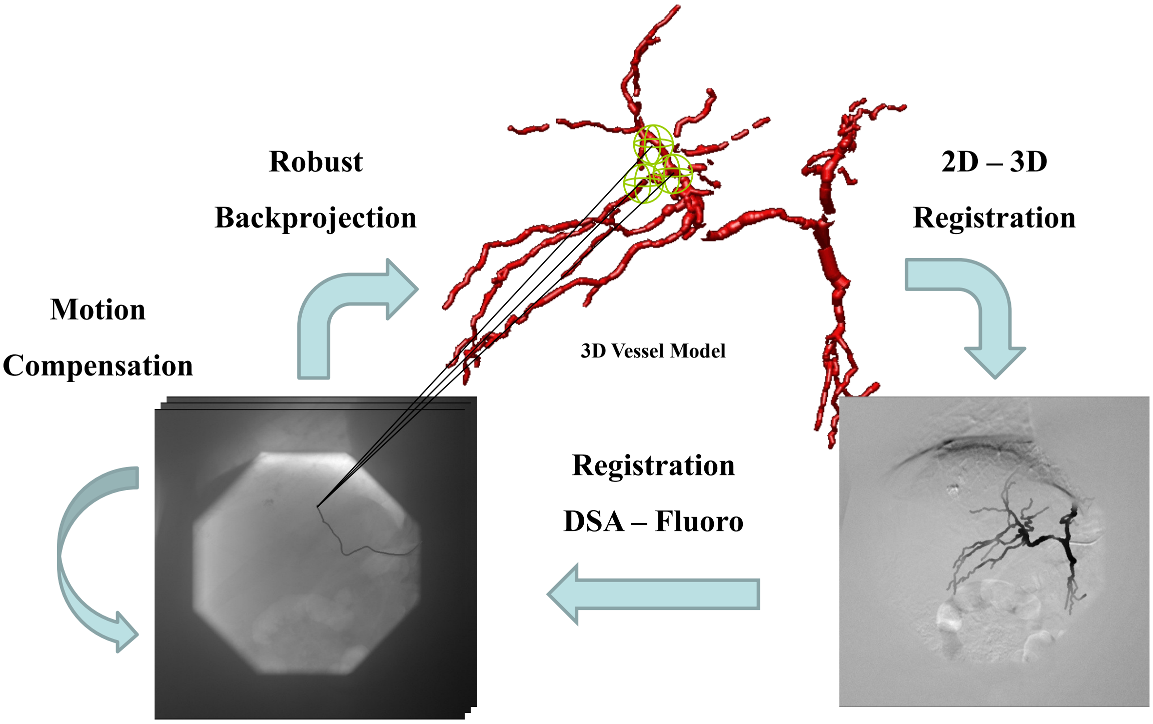


Figure 1.4: Components of interventional navigation.

tool positions in three-dimensional image data.
In the remainder of this section we will focus on the approach offering three-dimensional tool annotations and provide a detailed overview of the required components for such a system. In general, endovascular scenarios cannot be assume to be static – tools are moving and bending and patients are breathing, exhibit cardiac activity and are oftentimes moving. Therefore, it is desired that a navigation system is capable of dealing with such situations. It should be designed such that dynamic and non-linear processes can be modeled and captured. Ignoring the image acquisition, such a flexible navigation system requires 4 main components.

1. **Deformable 2D-3D Registration Module:** This module is required to establish a spatial relationship between the two-dimensional roadmap (e.g. DSA) and the three-dimensional image data (e.g. 3D-RA). Literature dealing with this problem (e.g. [26, 29]) is still limited because the task is severely ill-posed. Given only a single view, it is not possible to assess displacements in the projection direction, i.e. in z-direction of the camera coordinate system. Theoretically, this leads to an infinite number of possible solutions for such a registration and thus such a system requires (soft) constrained optimizations processes. Length preservation and deformation smoothness are two common ways of constraining such a process. This module is depicted on the right in Figure 1.4 and labeled as "*2D-3D Registration*".

2. **Deformable 2D-2D Registration Module:** The deformable registration between a key-frame from the two-dimensional fluoroscopic sequence and the two-dimensional DSA allows to relate pixel positions in the key-frame to positions in the DSA. This is an important step because so far, only the relation between the DSA and the 3D-RA has been established but not the relation between the fluoroscopic sequence and the 3D-RA. To our best knowledge, this particular problem has not yet been addressed by any researcher but it is likely that existing 2D-2D registration methods such as [24] could be used to perform this task. It is not even clear, whether a deformable registration is required at this step when the DSA is acquired under the same projection as the fluoroscopic image sequence. Given that the apparent motion observed in fluoroscopic videos is cyclic, there exists a key-frame in the image sequence which was acquired in a similar cardiac phase as the DSA and therefore offers maximal similarity. Furthermore, DSAs are generated from two X-ray images – one with and another one without contrast agent. The contrast free X-ray image is of the same modality as the fluoroscopic key-frame which is another indicator that such a registration should be feasible. The module is depicted at the bottom of Figure 1.4 and labeled as "*Registration DSA-Fluoro*".

3. **Motion Compensation Module:** The motion compensation is the component encapsulating the main contribution of this work. Compensation of apparent motion is required because the fluoroscopic video sequence is not static. Patient breathing and heart-beat as well as tool interactions performed by the physicians

are just two examples causing this behavior. The term apparent motion is used because only projections of the actual three-dimensional motion are observed. As a consequence, the observed motion is non-linear and requires deformable tracking techniques in order to properly recover tool motions. Exemplary difficulties arising during the development of tracking solutions are a low SNR and thus degraded image quality, the presence of other anatomical structures such as bone interfaces resulting in cluttered background information and the fact that medical tools are oftentimes only visible as tiny structures in the X-ray images which complicates their image-based enhancement and detection. Having this module in place, it is now possible to track medical tools in time which allows to relate any tool position to a specific key-frame in the fluoroscopic sequence. If this key-frame is identical to the one chosen in the *Deformable 2D-2D Registration Module*, each tool position in the fluoroscopic image sequence can now be related to the DSA. The missing step is to *back-project* the tool positions from the DSA to the 3D-RA which is covered by the fourth module. The motion compensation module is depicted on the left of Figure 1.4 and labeled as “*Motion Compensation*”.

4. **Backprojection Module:** The back-projection module is required to close the final gap, i.e. to be able to relate tool positions in the fluoroscopic sequence to the 3D-RA. The problem with the back-projection is similar to the one mentioned for the *Deformable 2D-3D Registration Module.* For each point in the image, there exist infinitely many corresponding potential points in 3D. This first problem can be overcome by segmenting the vessel tree $\mathcal{V}$ in the 3D-RA. When *back-projecting* any point of the DSA being located on the tool and in a vessel, the back-projection ray must intersect with a branch from the vessel tree $\mathcal{V}$. The practical issue of this solution is that the given scenario is idealized. It does not account for errors occurring in the parameter estimation of other modules and it does not cover the issue where a back-projected ray intersects $\mathcal{V}$ in multiple branches. Both issues have been addressed by Bender et al. [9, 8, 28]. The back projection module is depicted on the top-left of Figure 1.4 and labeled as “*Robust Backprojection*”.

1.4.1 Motion Compensation Module Details

This section provides a detailed overview of the motion compensation module and its required sub-modules. The main task of the motion compensation module can be formulated as follows. Given initial tool locations in some key-frame $\mathbf{p}^i_{t_{key}}$, determine the tool positions $\mathbf{p}^i_t$ in the current image $\mathcal{I}_t$ observed at time instant t. Formulated differently, we seek to find a transformation $\mathcal{T}_t$ (linear or non-linear) such that

$$\mathcal{P}_t = \mathcal{T}_t \circ \mathcal{P}_{t_{key}} \tag{1.1}$$

where $\mathcal{P}_{t_{key}}, \mathcal{P}_t = \left\{\mathbf{p}^1_t, \ldots, \mathbf{p}^M_t\right\}$ are the sets of tool positions at time instant t_{key} and t. The transformation $\mathcal{T}_t$ is mapping points from time instant t_{key} to t. Typically, $\mathcal{T}_t$ is determined through energy minimization as

$$\mathcal{T}_t = \arg\min_{\mathcal{T}} E\left(\mathcal{T} \circ \mathcal{P}_{t_{key}}, \mathcal{I}_t\right) \ . \tag{1.2}$$

Through application of the inverse transformation $\mathcal{T}_t^{-1}$, it is possible to relate each tool position $\mathbf{p}_t^i$ from time instant t to its corresponding position $\mathbf{p}_{t_{key}}^i$ in the key-frame. In order to be able to solve Equation (1.2), one needs access to the tool positions $\mathcal{P}_{t_{key}}$ from the key-frame as well as the currently observed image $\mathcal{I}_t$. The image access is not a problem since modern C-arm machines allow direct access to the live image data including all their acquisition parameters. A bigger problem are the tool positions – they are not readily available. It will be shown that an overall of three sub-modules are required in order to build a complete motion estimation.

Tool Detection: The first module deals with the initial tool detection. Optimally, the tool detection should be fully automatic without any user interactions. This work makes a first step in this direction and presents a semi-automatic initialization method in Section 3.3. The approach aims at minimizing user interactions and relies only on the selection of the two end-points of the medical tool. Starting from the two end-points, the system traces a path through the corresponding image in order to inter-connect the start- and end-point. This process results in a set of pixels $\mathcal{P}_{t_{key}} = \left\{\mathbf{p}_{t_{key}}^1, \ldots, \mathbf{p}_{t_{key}}^M\right\}$ representing the medical tool in the key-frame. The methods required during the detection as well as their mathematical models and the steps required for building a fully automatic approach are discussed in Chapter 3.

Tracking: The second module provides a way of solving the minimization problem of Equation (1.2). A common approach to this problem is to formulate the energy function E such that it takes its minimum, when the tool locations $\mathcal{P}_t$ at the current time instant correspond best to the image data $\mathcal{I}_t$. The quality of the fit can be measured in different ways. A common approach is to generate some sort of likelihood data from $\mathcal{I}_t$ by applying a function that turns pixel intensities into likelihood values being highest for pixels belonging most likely to the medical tool and small otherwise. Alternatively, it is possible to learn mappings from image intensities to cost function responses. In any case, the optimization process requires regularization techniques to guide the optimization process to tool configurations being physically meaningful. The underlying module and details of different optimization and regularization procedures are described in Chapter 4.

End Point Detection: Practice has shown that following these strategies is not enough to lead to satisfactory results. In fact, any commercial realization will require a third module dealing with the detection of the full tool, or at least tool tip. This problem originates from the fact that during the intervention the appearance and in particular the length of the medical tool changes dramatically and that none of the existing approaches attacking the tracking problem can detect those changes. Shirley A.M. Baert proposes a method [6, 4] in which the guide-wire or catheter positions are iteratively refined. The tool length is increased in tangential direction until the endpoint is advanced beyond the endpoint of the guide-wire or catheter.

Together, the three modules described above build the motion compensation module. Applying a two-step procedure of tracking and tool extension after the system is initialized results in the desired transformation $\mathcal{T}_t$ which allows the system to detect and transform tool positions computed at time instant t to the key-frame and from there to the pre-interventionally computed DSA. In applications, where an overlay of the DSA with the current fluoroscopic view is required, the computed transformation could be used to transform the DSA image. This allows a blending in which the apparent motion would be compensated.

CHAPTER

TWO

PREVIOUS WORK

Chapter two covers previous research performed in different areas which are relevant to this thesis. In Section 2.1 methods on enhancement of tubular structures are discussed and summarized. These are of particular importance, because they are directly affecting all components required in the detection and tracking pipeline. The next part, Section 2.2 provides an overview of work related to the detection of medical tools. Finally, the recently growing interest of work in the field of tracking is covered in Section 2.3.

2.1 Enhancement of Tubular Structures

Tubular structures are elongated line-like structures appearing at a specific width. These objects are also referred to as curvilinear structures which are important for a multitude of applications. In aerial images e.g., roads, railroads and rivers appear as curvilinear structures. Other examples come from medical imaging. Here, tubular structures appear oftentimes in angiograms, where vessels are visualized in 2D or 3D. In ophthalmology fundus images are of special interest because they contain indications about diseases of the eye and the primary diagnosis indicator are changes and abnormalities of vascularization. Performing measurements on vascularization is based on techniques for the analysis of tubular structures. Of most importance for this thesis is the enhancement of medical tools and in particular the enhancement of needles, guide-wires and catheters.

Because the enhancement of tubular structures is of such importance in various applications the topic received a considerable amount of interest in the past. The main focus of this section are mathematical methods for enhancing the visibility of curvilinear structures. Such an enhancement is a common pre-processing step for segmentation methods and the enhancement can be performed such that the output image intensities can be interpreted as likelihood values. In such images high intensities correspond to pixels having a high probability of belonging to a line-like structure and low intensities represent background pixels. Oftentimes, methods for extracting centerlines [44] of tubular structures have some overlap with those enhancing the structures. For this thesis image processing methods enhancing those structures are of primary interest and those are the methods which are revisited in this section.

In his work on “Finding Edges and Lines in Images” [14], Canny discussed not only edge detection methods but also the enhancement of ridges and valleys. When interpreting images as height fields, where the intensity value of a pixel corresponds to a height value, curvilinear structures appear in this landscape of image intensities either as mountain ridges or a valleys. According to Canny, “the only reason for using a ridge detector is that there are ridges in images that are too small to be dealt with effectively by the narrowest edge operator”. The scenario he is referring to is exactly what can be observed for medical tools since they appear only as tiny and thin structures within the medical images. Canny furthermore observed, that accurate estimates of ridge directions are much harder to obtain than edge-directions. He states that the principal curvature, while being normal to the ridge direction, is a much less reliable quantity to measure from even in smoothed images. As a consequence Canny concludes that the detection or enhancement of curvilinear structures requires several oriented filer masks at each point. Since this early work, several strategies for the enhancement of ridges (or valleys) have been investigated.

Koller et al. [39] propose to combine first and second order image filters for the enhancement of ridges. The authors build a line enhancement function by using the first derivative of a Gaussian $G'_\sigma(x)$ which is a well known edge detector [13]. Given a bright bar profile of width w which serves as an approximation of an intensity ridge, the Gaussian derivative $G'_\sigma(x)$ enhances the ridge edges on the left- and right-hand side of the profile. Following this observation, one can create shifted filters $E_l = G'_\sigma(x - \frac{w}{2})$ and $E_r = -G'_\sigma(x + \frac{w}{2})$ which, when convolved with the image, detect the left and right profile edges at $x = 0$. The authors combine the convolution responses in a non-linear way to overcome multiple line responses and sensitivity to edges, and the final one-dimensional filter becomes

$$\mathcal{R}(x) = \min\left(\mathrm{pos}(E_l * f(x)), \mathrm{pos}(E_r * f(x))\right) \ . \tag{2.1}$$

Switching from bright (ridges) to dark (valleys) profiles, affects the shifted filters in their sign and the function $\mathrm{pos}(x)$ maps all negative values of x to zero and acts as an identify operator on all other values. In 2D, the filter slightly changes because only the gradients orthogonal to the ridges must be evaluated. To achieve this, the responses of the filters E_l and E_r have to be projected on the ridge-normals which are the directions of principal curvature of the ridge. These direction correspond to an eigenvector of the Hessian of image intensities in each pixel which is covered in more detail in Section 3.2. Another topic which will be covered in this section is the choice of the smoothing parameter σ.

A different approach is proposed by Palti-Wasserman et al. in [58]. They suggest to use a modified Laplacian as well as a modified Marr-Hildreth [36] operator in order to enhance guide-wires while reducing the influence of noise. The authors achieve this behavior by adapting the original filters such that they act more globally. Each point is compared not to its immediate 8-neighborhood (in case of 2D data), but to more distant points in the image. The optimal filter size is depending on the width of the guide-wire though the authors suggest an empirical value of 11 pixels. In addition to modifying the

filters, the method is based on a *fuzzy* zero-crossing detection. The detection method is fuzzy in the sense that a pixel is marked as a zero crossing as soon as it has an intensity value smaller or equal than a specific threshold. The authors' second filter, the modified Marr-Hildreth filter is a smoothed version of the modified Laplacian. It requires an additional parameter in order to control the amount of smoothing.
Frangi et al. [22] proposed another measure, which is based on the eigenvalues of the Hessian matrix. The Hessian matrix $\mathcal{H}$ describes the second-order structure of local intensity variations around points of an image. The eigenvalues of $\mathcal{H}$ correspond to the principal curvatures and for line-like structures the eigenvector corresponding to the smallest absolute eigenvalue points along the vessel. The approach is an extension of the works of Sato et al. [66] and Lorenz at al. [50] who also use the eigenvalues of the Hessian to locally determine the likelihood that a vessel is present. Frangi et al. modified these works by considering all eigenvalues and by giving the vesselness (or line-likeliness) measure an intuitive, geometric interpretation. Their measure consists of three components in 3D and two in 2D respectively. Assuming sorted eigenvalues $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, the 3D specific component is based on the ratio of the largest absolute eigenvalues

$$\mathcal{R}_A = \frac{|\lambda_2|}{|\lambda_3|} \,. \tag{2.2}$$

The measure is used to distinguish between plate-like and line-like structures and will be zero for the latter. The second component applies to the 2D and 3D case and measures the deviation of the observed structure from a blob-like or circular structure. It is computed as

$$\mathcal{R}_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}} \,. \tag{2.3}$$

A last component is used to evaluate the amount of information contained in the Hessian by computing the Frobenius norm of $\mathcal{H}$. For a real and symmetric matrix, the measure is computed as $\mathcal{S} = \sqrt{\sum \lambda_i^2}$. This value will be low in background areas where no structure is present and it becomes larger in areas of high contrast. The final filter response is built up by combining the values through Gaussian distributions resulting in a scalar filter response which can directly be interpreted as a likelihood value. The problem this filter poses for the user is the choice of three parameters (two in the 2D case) for the Gaussians. Further details of the choice of these parameters are discussed in Section 3.2.

2.1.1 Scale-Space Theory

These days, most of the research dealing with the detection or enhancement of curvilinear structures in 2D or 3D utilizes in some way differential geometry. Of special interest is the Hessian matrix (here exemplary for the 2D case)

$$\mathcal{H} = \begin{bmatrix} \mathcal{I}_{xx} & \mathcal{I}_{xy} \\ \mathcal{I}_{xy} & \mathcal{I}_{yy} \end{bmatrix} \tag{2.4}$$

which describes the second order structure of intensities of an image $\mathcal{I}$. A first problem is that the computation of the Hessian requires second-order differentiation of the image intensities and this differentiation process is ill-posed. The problem is ill-posed because of noise introduced by the acquisition process which is unavoidable. Consequently, differentiation of images requires regularization and a detailed analysis of this observation has been presented by Torre and Poggio [74]. The authors conclude, that Gaussian derivative filters are nearly optimal filters for computing image derivatives. A problem that arises in this context is how to choose the parameter σ – the *scale* – of the Gaussian.

Where is now the connection between scale-space and the enhancement or detection of line-like structures? An intuitive idea, of why different scales are required here is because line-like objects are viewed from different distances or have different physical sizes. Both cases result in the images of the objects to be of different size. Hence, the detection of tubular objects requires a way to define the scale, i.e. the size of the line-like structure in the image. This scale is typically unknown and Witkin [82] proposed to obtain a multi-scale representation of the image. This multi-scale representation is generated by the convolution of the image with different Gaussian filters, where the parameter σ defines the scale. This pencil of filtering functions and their responses define the *scale-space*.

When performing analysis in scale-space, in particular when trying to select "optimal" scales, additional modifications are required. Lindeberg [47, 48] analyzed the problem of optimal scale selection and introduced so called γ-*normalized* filters. In Lindeberg's framework a one-dimensional γ-*normalized* Gaussian filter is written as

$$t^{\gamma/2} \frac{1}{\sqrt{2\pi t}} \exp\left(\frac{-x^2}{2t}\right) . \tag{2.5}$$

Here, the parameter $t = \sigma^2$ is referred to as the scale space parameter. The normalization constant γ is used, to ensure that a Gaussian based filter response yields only a single maximum within a potentially limited scale support. In some scenarios, the parameter can also be used to modify the response function's shape such that the maximum is reached exactly at a specific scale. Specific examples and mathematical details will be presented in Section 3.1.

2.2 Detection of Guide-Wires and Needles

Tracking algorithms estimate the motion of an object between two points in time. Given an object location at time t_{i-1}, the algorithms estimate the position of the same object at another point int time t_i. When such an algorithm is initially executed, the spatial information at some time t_0 is needed. This phase, when no a priori knowledge is available is oftentimes referred to as the initialization phase. During this phase, some way is required to detect the object, i.e. to determine its initial position in space. In the scenario of medical tool tracking, this means to be able to detect the object in the image only based on limited constraints regarding its position. Due to the limited constraints, this problem is much more difficult than the actual tracking step.

If a fully automatic method existed to solve this problem without a priori knowledge, tracking and other post-processing steps would not be required anymore. Unfortunately, the problem is so ill-posed, that currently no fully automatic methods exist which perform this task so reliably, that it could be applied in clinical practice. But why is the problem ill-posed? And why is the problem at all underlying any constraints? The reasons for the problem to be ill-posed are manifold. The list below gives a brief overview of some of the reasons:

- **Multiple Tools:** A first problem is that during minimal invasive procedures oftentimes multiple tools are used. Using for instance several catheters is not uncommon during cardiac interventions. All of them might be visible and without any user interaction it is not clear, which of these tools is the object of interest. A possible approach to handle such a situation is to use all potential candidates. This, however, would lead not only to a more complicated tracking algorithm but also to reduced performance regarding the run-time. Considering the fact, that the tracking is only one of many components required for a navigation application, it is important to minimize computational requirements.

- **Background Clutter:** For a human observer, background clutter is usually not a problem. This is especially then true, when observing dynamic scenes like for instance a video stream as it is the case during fluoroscopy guided interventions. For a computer on the other hand side the situation is different. In fluoroscopic images, anatomical structures like bone-tissue interfaces can have a quite similar appearance as guide-wires or needles. Therefore, background clutter ultimately causes similar problems as *Multiple Tools* do. The initialization algorithm requires guidance to be able to select the correct tool of interest.

- **Noise:** The detection process is usually based on an early preprocessing stage in which candidate pixels potentially belonging to tools are highlighted. As has been discussed before in Section 2.1 many methods for the enhancement of tubular structures rely on second-order image filters. These filters get unstable in the presence of noise which was also discussed by Canny [14] and generate spurious responses when the SNR is low. Additional feature responses created due to noise are more difficult to deal with than those created by multiple tools. Simply adding all false positive responses to the tracking algorithm does not only increase the run-time but it also leads to noise in the motion estimation. This is the case because noisy background cannot be assumed to exhibit motion which is free of noise.

- **Tool Size:** Guide-wires and interventional needles have small diameters resulting in bad visibility within fluoroscopic sequences. The physical diameters of guide-wires are typically in the range of 0.5 to 1.0 mm and a common biopsy needle size is 14-gauge (i.e. 2.108 mm). The resulting projections are often just around pixel wide and even result in multiple disconnected lines. This has implications on image filtering techniques because scale-space filter responses or for many methods designed such that they take their optimal response at a scale that corresponds to the ridge radius, i.e. half the tool diameter and thus the scale can be as 0.5 pixels. When using such small scale values for image filtering, the influence of discretization effects is strongly degrading the responses and results in noisy responses.

- **Temporal Averaging:** In order to improve image quality hardware-vendors of medical imaging devices such as C-arms integrate signal processing algorithms in their machines. One common method to reduce the speckle-noise observed in low-dose X-ray videos is the use of temporal averaging. A straight forward approach of this methods takes a number of images in a fixed size time-window and computes the average of these images. Indeed, this method reduces the speckle-noise but is not without drawbacks. The main problem behind this method is that in dynamic scenes with apparent motion, temporal averaging will cause phantom responses of objects resulting from different spatial locations of these objects. Such phantom locations are disturbing detection systems because again, it is not clear which responses are the correct ones belonging to the actual tool position.

All these issues make the detection task a difficult problem. The remainder of this section gives a brief overview of how researchers tried to solve the aforementioned issues or at least how to reduce their influence. The approaches are quite different ranging from semi-automatic methods to attempts at fully-automatic procedures.
Palti-Wassermann et al. [58] propose to use a semi-automatic method for the detection of guide-wires which is based on fitting second-degree polynomials in “limited active windows”. Initially, the authors pre-process the input image with modified Laplacian or Marr-Hildreth filters. The filters are modified such that they enhance intensity ridges instead of corners. This is achieved by altering the influence range of the original filters in combination with an adapted method for the labeling of zero crossings. A fuzzy labeling process of zero-crossings allows to turn the image into a binary form where only potential ridge pixels are colored white and all other pixels are marked as black. In a next step, the authors approximate the guide-wire shaped by a second-degree polynomial $y = ax^2 + bc + c$. To calculate the parameters a, b and c, three points belonging to the guide-wire need to be known. This requirement can be reduced by assuming two points, e.g. the end-points of the guide-wire are provided by the user. Once this condition is met, guide-wire locations can be represented by a single parameter and the authors chose to use b as their unknown variable. The parameter estimation is performed in sub-windows along the curve by applying a Hough transformation to the local window. After this transformation, the parameter b corresponding to the guide-wire is selected as the value corresponding to the maximum of the histogram of b-values in the sub-window. Given this parameter, the second-order polynomial representing the guide-wire part passing through the sub-window is completely known. In a next step, the sub-window is advanced further along the curve and starting from the known two overlap points, the procedure is applied iteratively until the local window is finally placed on the guide-wire’s end point. The authors use further techniques such as dynamic histogram size, b-value thresholding and

pixel-grouping to enhance the detection method. Regarding future work, the authors argue that "time must be added" to increase the level of automation.
Carsten Steger developed a robust and fast method in his research on the detection of curvilinear structures [70, 71]. The method results in multiple sets of points, where each set represents one line. Multiple properties are assigned to each point. The properties include the point's sub-pixel location, the line normal, line width, its strength and its type (end-point, center-point or junction-point). The algorithms starts by computing the line's normal directions for each point. This is done in the same way as during the enhancement of tubular structures. The direction (n_x, n_y) is computed as the eigenvector of the Hessian of image intensities which corresponds to the eigenvalue of maximum absolute value. In a next step, the point's sub-pixel location is computed by inserting $(d_x, d_y) = (t\, n_x, t\, n_y)$ into the Taylor polynomial

$$\mathcal{I}(x + d_x, y + d_y) = \mathcal{I}(x, y) + (d_x, d_y) \begin{pmatrix} \mathcal{I}_x \\ \mathcal{I}_y \end{pmatrix} + \frac{1}{2}(d_x, d_y) \begin{pmatrix} \mathcal{I}_{xx} & \mathcal{I}_{xy} \\ \mathcal{I}_{xy} & \mathcal{I}_{yy} \end{pmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{2.6}$$

and setting the derivative along t to zero. It turns out, t can be compactly represented as a function of the line normal and first- and second-order derivatives. To ensure, that the final pixel location is within the unit-circle centered at (x, y), only points with $t \in \left[-\frac{1}{2}; \frac{1}{2}\right]$ are accepted as line points. This first step results in a sub-set of the initial image pixels and associated sub-pixel locations $(x + d_x, y + d_y)$.
In a next step, pixels are linked together to form line segments. The system stores for each pixel, its sub-pixel location, its direction and its ridge strength. These values are then used to perform the actual linking. First, depending on the edge direction, the three candidate neighbors are selected for each pixel. This procedure is depicted in Figure 3.12. From the potential neighbors, the one that minimizes $d + c\,\beta$ is chosen. Here, $d = \|p_1 - p_2\|_2$ is the euclidean distance between two sub-pixel points and $\beta = |\alpha_1 - \alpha_2| \in [0, \pi/2]$, is the angle difference between the two points. In his work, Carsten Steger used $c = 1$ throughout his experiments. Similar to hysteresis thresholding, the linking process is controlled by two thresholds. A first, high threshold is used to determine whether a new line segment is created. As long, as the system detects a pixel with a second directional derivative being bigger or equal to the high threshold. A second, lower threshold is used to determine when the tracing of a line segment is terminated. In case all potential neighbors have second directional derivative being lower than the second threshold, the current line segment tracing is stopped. The edges and the width of line segments are determined similarly to what has been done for the sub-pixel locations. The paper proposes additional methods in order to deal with lines that having asymmetrical profiles. The evaluation has been primarily performed on aerial images though the work also contains examples from a coronary angiogram. Once correct parameters are chosen, the method is fully automatic. As it is often the case, good parameters can be estimated from properties of the expected line profiles. The problem on the other-hand side is that the method is detecting many points and segments resulting from background clutter. Currently, the correct lines must be chosen manually, or it must be accepted that only parts of the medical tools are extracted fully-automatically. Similar to the previous approach, a stable fully-automatic system

needs to incorporate time information during the detection process.
The algorithm presented by Can et al. [11] focuses on vessel segmentation in fundus images. The authors propose a tracing algorithm which works recursively given a set of initial seed points. Seed points are detected by sampling the image along a grid of one-pixel-wide lines and by detecting local gray-level minima along these lines. This approach will choose points lying most probably on vessels which appear as dark tubular structures in fundus images. Once the seed points p_0 are computed, the algorithm continues by recursively adding new neighbors p_k to the seeds. Here, k is representing the time-step or iteration. Vessel directions s_k are discretized into 16 fixed orientations, one every 22.5°. To determine the direction s_k for a point p_k, directional filters are applied on the left- and right-hand side of the vessel, orthogonal to the currently probed direction s. These filters are not only applied for different potential orientations s but also at different distances to the current centerline location p_k. The left- and right-hand side filter searches result in responses $R_{\max}(s)$ and $L_{\max}(s)$. The optimal orientation is chosen as [1]

$$s_{k+1} = \arg\max_{s}\Big(\max\left(R_{\max}(s), L_{\max}(s)\right)\Big) \tag{2.7}$$

and the tracing follows the strongest ridge by setting the new position vector $p_{k+1} = p_k + \alpha u_{k+1}$. The vector u_{k+1} corresponds here to the unit vector pointing in the direction defined by s_{k+1}. The iterative procedure is stopped when a new point p_{k+1} is located outside the image or when a previously detect vessel intersects the current one, or when the sum of the filter responses falls below a certain threshold T, $|R_{\max}(s_{k+1}) + L_{\max}(s_{k+1})| < T$. Additionally, the authors took care in their method to develop approaches to extract branching and cross-over points while aiming at a high level of automation and robustness. To summarize, the method proposed by Can et al. uses vessel tracing which does not require global image filtering but only local filter application. The method is capable of extracting the whole vessel tree from fundus images while offering good performance both in terms of robustness and run-time. Nonetheless, the method can not be seen as a fully automatic implementation for vessel or tool detection. The robustness is depending on the choice of several parameters such as grid size, neighborhood size, and amongst others the step size used to select the next neighbor. Despite the parameter choice, the algorithm is potentially detecting or segmenting more candidates as are of interest for the application of guide-wire tracking. No improvement has yet been made on detecting and removing false positive segmentation results.
Another algorithm based on tracing vessels is proposed by Schneider et al. [67]. The authors use a Hessian based algorithm for the segmentation of vessels. The method is generic and can also be applied to the detection of medical tools such as guide-wires or needles. The idea is based on using not only the eigenvalues of the Hessian matrix of image intensities but also the eigenvectors as well as a likelihood image (e.g. Frangi's [22] vesselness). Given initial seed points, the algorithm propagates a wave front in the direction of the eigenvector corresponding to the smallest absolute eigenvalue. The

---

[1] The formula has been slightly rewritten. Its orignal can be found in [11], in Equation (7).

segmentation result keeps growing iteratively until the following boolean criteria is fulfilled:

$$\left(\left|\mathbf{v}_1^\top \mathbf{v}_1'\right| \leq \theta_\alpha\right) \vee \left((V < \theta_V) \wedge (\lambda_1 \geq \theta_{\lambda_1} \vee \lambda_2 \leq \theta_{\lambda_2})\right) \tag{2.8}$$

The first term is corresponding to a smoothness criterion and ensures that neighboring points exhibit similar directions. The second, two-component term ensures, that either the likelihood of the currently selected point is high enough, or that the eigenvalues exhibit characteristics corresponding to a bifurcation. The last term, the bifurcation check is required because it is common that eigenvalue based vesselness measures are degraded on bifurcations. The authors coined the resulting vessel or tool representation as *streamline representation* and argue that it provides additional information about the global object shape. One example are length and density maps. The length map associates to each pixel the length of the longest fiber crossing its location. Similarly, the density maps represents for each pixel a counter of how many fibers passed through its location. So far, the procedure does not result in a single pixel wide segmentation and thus rely on additional thinning techniques to create a representation of the center line only. The thinning is performed by iteratively removing pixels that do not fulfill certain criteria depending on the current pixel's density and the expected object length and uniqueness. Finally, the authors propose a method that allows the detection of catheters and their removal from vessel images. The idea is based on heuristics regarding the length of the catheter and its shape. The authors claim that catheters typically pass through the whole image and "have a rather straight shape with small constant curvature". The authors assess the curvature by fitting B-splines to streamline candidates and by computing curvature responses analytically on the estimated B-spline curves. Angular changes and the segment length are then computed against empirically estimated values in order to classify a segment as being a guide-wire or not. Altogether, the method is promising but still requires user interactions. In case seed points were detected automatically, one would be facing the same problem as before – segmentation results from the anatomical background without obvious ways to detect or remove them.
A different, learning-based approach for the detection of guide-wires has been proposed by Mazouer et al. [54]. As opposed to their previous work [7], the authors extended the new method such that it allows user-interactions to constrain the detection process. The method allows to follow two distinct paths for the guide-wire detection. A first path allows to perform a fully-automatic detection which can be refined by interactively adding additional constraints. The second path performs already the initial detection based on constraints provided by the user.

The overall system consists of several modules. In the beginning constant length segment candidates are extracted by a low-level segment detector. To each segment three parameters (x, y, θ) are associated, where (x, y) is the segment location and $\theta \in [-90°; 90°]$ the segment orientation. The segment detector is based on Haar features and uses a probabilistic boosting tree (PBT) for the detection process. The second component is a curve classifier, again based on a PBT. The curve classifier is used to model the shape and appearance of an n-segment curve. In the initial work by Barbu et al. [7], the probability of a guide-wire consisting of segments s_1 to s_n and end-points x_B

and x_E is computed as

$$P(G(s_1,\ldots,s_n,x_B,x_E)) = P(C(s_1,\ldots,s_n))P_E(x_B)P_E(x_E)\ . \tag{2.9}$$

The guide-wire with the highest probability is reported as the final localization result when the fully automatic procedure is being used. For the path that allows user interactions, Mazour et al. introduced a novel scheme. So far, the method is based on segments s_k and probabilities $P(s_i, s_j)$ of connecting two segments. User inputs are provided as two dimensional point coordinates (x, y) on the guide-wire. In order to be now able to incorporate these coordinates into the probability framework, probabilities of connecting a given point with already known segments is required. These probabilities are computed based on a so called extension field (EF) introduced by Guy et al. [30]. Once the missing probabilities are available, dynamic programming (DP) is used to connect the segments as well as the user inputs.

In their evaluation the authors report false detection rates of below 5% as soon as five user interactively selected points are available. Given the same number of user interaction, the missed detection rate reaches similar values. As an average annotation time for full guide-wires including loops the paper reports about 6.23 seconds.

The methods above represent more than 10 years of research and constitute only a fraction of what has been done in the field of ridge detection. The applications are various ranging from aerial image analysis over the detection of vasculature to the detection of medical tools. The approaches range from early image processing and image filtering techniques over approached using ridge tracing to learning based methods including hybrid solutions which aim at utilizing the benefits from all worlds. There is still a trend towards semi-automatic methods that allow user interactions because for many applications the fully-automatic approaches do not yet meet the strong real-world requirements. In particular the robustness, i.e. being able to distinguish spurious line-like structures from the desired objects is often-times difficult. Since so far, all methods are based on static images, future work is probably going to investigate the integration of temporal information which might give additional clues about how to detect and remove noisy image structures.

2.3 Tracking of Guide-Wires and Needles

In this section existing approaches are discussed which mainly deal with the problem of tracking guide-wires during fluoroscopic interventions. The problem is yet unsolved and in no commercially available system integrated. First publications working on the idea of enhancing navigation during fluoroscopic interventions appeared in the mid 80s – that means we are already looking back at a quarter century of research. Since then, a variety of work has been performed in the direction of navigation aids and on the enhancement and detection of medical tools. Surprisingly, image-based (deformable) tracking of medical tools has only relative recently received growing interest and it seems as if it still keeps growing.

Baert et al. [5, 3] propose a guide-wire tracking procedure based on energy minimization of a B-spline parameterization. The method is stratified, i.e. the motion estimation is

performed in two steps. At first a rigid transformation of the guide-wire is estimated which is followed by computing the remaining deformable component.
The rigid transformation, here translation only, is determined by creating a binary template from the currently known guide-wire positions. Once the template is given, the optimal location in the new frame is determined by maximizing the template's cross correlation within a certain search region in the new image. The cross correlation can be computed on the original image intensities or some feature image highlighting line-like structures.
The deformable motion component is estimated by energy minimization using Powell's direction set method. To this end, the guide-wire shape is approximated by a third-degree B-spline curve defined as

$$C(u) = \sum_{i=0}^{p} N_{i,3}(u) P_i \ , \quad 0 \leq u \leq 1 \ . \tag{2.10}$$

In the papers of Baert et al. [5, 3, 4] 3 or 4 control points were used to model different guide-wire shapes. The optimization is now carried out, by finding displacements d_i for each control point P_i such that the displaced curve corresponds in an optimal way to the current fluoroscopic frame. Because mere image based constraints lead to an ill-posed formulation, the deformed curve must adhere to additional geometric constraints. The optimization is formulated as an energy minimization and the energy function models the aforementioned components by external and internal forces.
The external forces are based on feature images F. These are constructed by enhancing line-like structures with and without using directional information of the initial spline position. The structures are enhanced in unprocessed images, in background-subtracted images as well as in images which are preprocessed by coherence enhancing diffusion (CED). As the internal energy the authors have chosen to penalize the curvature κ of the curve at a given point. The overall energy is formulated as

$$E = \sum_{i=0}^{n} \Big(\alpha \kappa^2(i) + F(i) \Big) \, l(i) \ . \tag{2.11}$$

Here, $n = 100$ is the number of samples used along the spline curve. The weighting parameter $\alpha = 6.25$ is used to control the influence of the internal energy. $F(i)$ and $\kappa(i)$ are the values of the external and internal energy for the i-th sample point on the curve and $l(i)$ can be interpreted as the integration step.
For this work it is of most interest to note that the authors conclude that Hessian feature images allow to achieve results with a precision "similar to the inter-observer variability". This indicates that it is not essential to use advanced preprocessing methods such as CED which is advantageous because such methods would lead to longer processing times. The authors report running-times of about 5 seconds per frame and suppose that a reduction by an order of magnitude is possible after optimizing the method.
Slabaugh et al. [69] present a similar approach. The authors are also using a stratified approach, i.e. perform a rigid as well as a deformable registration and they are also modeling guide-wires as B-splines. Besides these similarities, the work presents various novelties. As before, the curve locations are determined by energy minimization. Here,

the authors propose to use two regularization terms. The first regularization term minimizes the curve's length while the second one tries to enforce that the curve's length stays as close as possible to some initial length L_0. The data term is integrated in the same way as the one presented by Baert et al. though in this work, a different type of image enhancement is suggested. The overall energy in this work can be expressed as

$$E = w_1 \int\limits_C F ds + w_2 \int\limits_C ds + w_3 \left(\int\limits_C ds - L_0 \right)^2 , \quad s \in [0, L] \ . \tag{2.12}$$

In this equation, C is representing the curve, $s \in [0, L]$ is the arc-length parameter and L the contour's length. The first addend corresponds to the external energy and ensures that splines stick to features detected in the X-ray image. The second term is by the authors referred to as *smoothness*, though it is effectively representing the curve length. Only in combination with the third term, the length preservation, the latter ensures some level of smoothness of the curve. To summarize, the energy term tries to guide curves towards image features while preserving an initial curve's length and possessing a certain smoothness.

After derivation in a differential framework and after re-parameterizing the curves as B-splines and discretization, the energy function can be minimized with any gradient based optimization as e.g. gradient descent. As opposed to Frangi's vesselness measure or CED enhanced images as use by Baert et al., this work proposes to use phase congruency images for the data term because they are reported to be less sensitive to contrast than differential techniques. However, a direct comparison between different feature images as in the previous work is not provided. The method has been evaluated on 158 frames with a success rate of 93% and an approximate processing time of 175 milli-seconds.

A quite different, learning-based approach has recently been presented by Peng Wang et al. [79]. Here, the authors incorporate learning-based measurements for solving the tracking problem. Again, guide-wires are modeled by B-splines and also in this work the guide-wire motion is decomposed into rigid and non-rigid motions which are estimated separately.

Based on a Bayesian framework, the authors propose two energy functions which are used for the rigid tracking and the deformable tracking, respectively. The posterior probability is re-written as

$$P(\Gamma_t(x)|Z_t) = P(\Gamma_t(x)) \left(w_1 P(Z_t|x_1) + w_N P(Z_t|x_N) + x_{\text{mid}} \sum_{i=2}^{N} P(Z_t|x_i) \right) \ . \tag{2.13}$$

In this equation, the prior $P(\Gamma_t(x))$ is a function that penalizes the distance between the current curve $\Gamma_t(x)$ and the previously observed curve $\Gamma_{t-1}(x)$. The second term accumulates the likelihoods $P(Z_t|x_i)$ of individual points x_i on the guide-wire. In addition, it introduces separate weights for the end-points of the curve. The latter is useful to deal effectively with open curves.

Two different likelihood formulations are introduced for $P(Z_t|x_i)$, a learned measurement $P^d(Z_t|x_i)$ and an appearance based measurement $P^a(Z_t|x_i)$. The learned measurement is computed from the numerical output $f(z, x_i)$ of an AdaBoost classifier for any given point x_i. It takes its maximum, when the classification value of $f(z, x_i)$ is maximal. The AdaBoost classifier is trained with a set of offline collected data. Appearance based measurements are evaluated by the likelihood $P^a(Z_t|x_i)$. Here, the likelihood becomes maximal, when the image patch centered at x_i in the current frame is most similar to the patch centered at the corresponding location in the previous (or alternatively initial) frame. The general likelihood term is computed as

$$P(Z_t|x_i) = P^d(Z_t|x_i)P^d + P^a(Z_t|x_i)P^a \ , \tag{2.14}$$

where P^d and P^a are priors for the two types of measurement models. Wang et al. propose so called kernel-base measurement smoothing for the likelihood functions. This additional modification allows them to perform a hierarchical and multi-resolution tracking and the details can be found in [79].

During the solution of the rigid-tracking, the authors maximize the posteriori based on Equations (2.13) and (2.14). This is not the case during the estimation of the deformable component. In the latter case, additional terms for the regularization of the problem are introduced. These terms are integrals of first- and second-order derivatives of guide-wire curves, and act as guide-wire smoothness priors.

The algorithm has been evaluated on 47 clinical sequences. It achieves an overall guide-wire tracking precision of 1.8 pixels and is currently processing data at 2 frames per second (i.e. 500 ms per frame). The authors claim that near real-time speed can be achieved with an optimized implementation based on multi-threading and GPU accelerations. Learning appearance models or likely deformations of guide-wires seems like a promising direction though currently they impose the additional difficulty to meet required real-time constraints.

Table 2.1: Overview of different guide-wire tracking methods.

method	parameter estimation	likelihood	prior	stratified	fps [a]
Baert [3]	continuous	feature image (Hessian, CED, ...)	curvature	✓	0.2
Slabaugh [69]	continuous	feature image (phase congruency based)	curvature & length changes	✓	5.7
Wang [79]	coarse-to-fine, exhaustive	AdaBoost classifier & intensity differences	displacements & derivative norms	✓	2
Method A	discrete	feature image (Hessian & gradient based)	length changes	–	20
Method B	discrete	SVR of orthogonal intensity profiles	length changes	–	1.5

[a] The given run-times cannot directly be compared since they are assessed on different systems. They serve mainly to get an idea of the magnitude of the algorithm run-times.

To summarize, all methods dealing with guide-wire tracking are using B-splines for modeling curvilinear medical tools (guide-wires, catheters and even needles since they may bend). Furthermore, it is important that the methods presented in this work as well as the aforementioned methods are tracking a segment of a medical tool while trying to prevent that this segment is vanishing. This implies, that all methods may require an additional procedure to perform the full tool detection or at least the tool-tip detection. A possible approach of dealing with this end-point detection is presented by Baert et al. [6].

The most important differences between the individual methods are summarized in Table 2.1. Here, the *parameter estimation* column is intended to provide a rough classification of the underlying optimization process. Entries of the *likelihood* (external energy, data term) column are meant to be read as "is based on" or as "is a function of" and the entries of the *prior* (internal energy, regularization) column list which geometrical properties the authors are minimizing in order to constrain the set of possible solutions. Some approaches (so called stratified ones) decompose the problem into a rigid and a non-rigid motion and solve both problems individually whereas others represent the whole curve motion by a single displacement field – the second last column *stratified* denotes to which of these two groups the method belongs. The last entry of the table is intended to give the reader a rough idea of the magnitude of the individual algorithm's run-times. Since the experiments were performed on separate systems with different processors and program environments, these values cannot be directly compared. Furthermore, the different methods offer different levels of potential regarding optimizations which follows that faster run-times may be achieved.

The key innovations of the method presented in this work can be summarized as follows. The algorithm introduced here is using discrete optimization to solve the parameter estimation. Discrete optimization has the advantage of being able to provide optimal solutions within the discretized parameter space. This property reduces the likelihood of the optimization process to get stuck in local minima.

Method A introduces a novel motion model and an adapted regularization scheme. Related tracking algorithms try to minimize geometric properties as e.g. curvature or the first- and second-order derivative norms of the curve in order to impose smoothness on the resulting curves and in order to prevent over-deformation. Slabaugh et al. [69] started to leave this path by penalizing absolute length changes, thus enforcing some level of self-similarity between the tracked curves and the curves as they are appearing during the initialization. This work is adapting this regularization approach and introducing a novel deformation model which is required to properly address discrete optimization inherent approximation errors.

A second learning based algorithm is briefly introduced in Section 6.2 and referred to as *Method B*. The idea is to learn a mapping between orthogonal intensity profiles of curves and the associated likelihood of belonging to a medical tool. This has the advantage of being able to adapt to the tool appearance which is typical for fluoroscopic image sequences including the influence of noise. Another advantage is that by learning an appearance model, one overcomes the problems which are inherent to image filtering because the learning can be performed on the raw and unprocessed image data.

CHAPTER

THREE

TOOL DETECTION

3.1 Modeling Tubular Tools

Tubular tools, or those that appear as tubular or line-like after projection by an X-ray system are of critical importance during the development of a tracking system. In this chapter the mathematical properties of line-like structures are revised. Only, when a mathematical model for line-like structures is given, it is possible to understand, analyze and optimize enhancement, detection and tracking algorithms. Already the modeling is not an easy task. The fact that till today, researchers have no common agreement upon a single mathematical model is one indicator that supports this statement.

3.1.1 X-ray Attenuation

The way in which tubular objects appear in images depends on various factors. Primarily, the appearance is defined by the physical rules underlying the imaging process of conventional X-ray. On a modern C-arm system, two-dimensional radiographs are acquired by an X-ray source and a planar, flat-panel detector. Image intensities are depending on the amount by which X-rays are attenuated while passing through the body. The attenuation of X-rays passing through a single object is an exponential function which depends on the object's attenuation coefficient. X-rays passing through an object with a diameter d and with constant attenuation coefficient μ are measured on the detector plane with an intensity of

$$I = I_0 \exp(-\mu d) \ . \tag{3.1}$$

For objects with locally varying attenuation properties, this equation generalizes to

$$I = I_0 \exp\left(-\int \mu(u) \mathrm{d}u\right) \ , \tag{3.2}$$

where the parameter of the exponential term is the line integral through $\mu(u)$. In both Equations (3.1) and (3.2), I_0 is the intensity of the incident X-ray beam. Ideal X-ray detectors are producing images with intensities being proportional to the negative

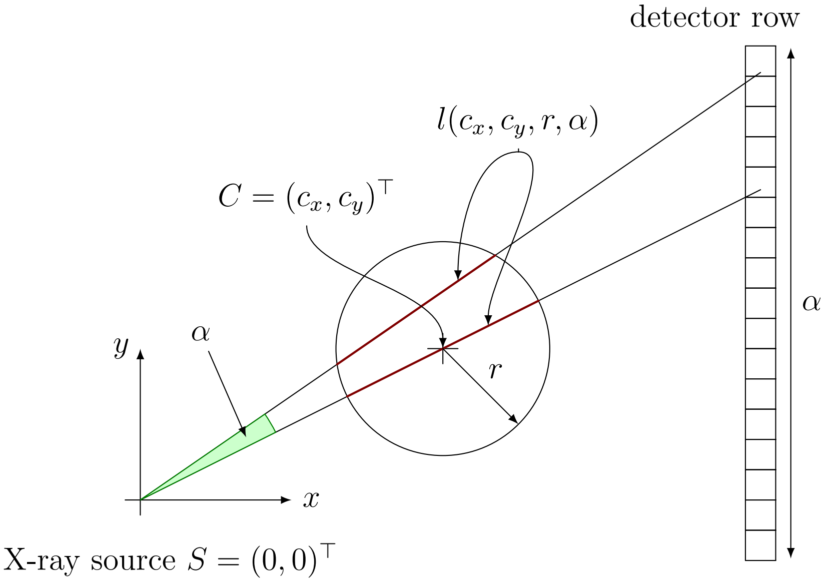


Figure 3.1: Idealized X-ray projection of a tubular object. The object is centered at C. The path lengths at which X-rays are attenuated in the tubular object vary with the projection direction α. The path length is denoted as $l(c_x, c_y, r, \alpha)$ and two examples for $\alpha = 0$ and $\alpha = 8°$ are shown in red.

logarithm of the attenuation intensities. An optimal projection image would have image intensities corresponding to

$$\mathcal{I}(x, y) = -\log(I_0) + \int \mu(u) \mathrm{d}u \ . \tag{3.3}$$

Image intensities are depending on the incident X-ray beam intensity I_0, the attenuation coefficients and the distance at which X-rays are traversing objects while being attenuated. These observations allow the formulation and analysis of a simplified projection model for tubular objects.

3.1.2 An Idealized Tube Model

For the analysis of image intensities as generated by tubular objects, a simplified two dimensional projection model is considered. The setup is sketched in Figure 3.1. In this model, a coordinate system is centered in the X-ray source S. A single scan-line in the image is generated by the attenuation responses measured on the detector row. The measurement are assumed to be corresponding to the ideal function of Equation (3.3). X-rays are emanating from the origin towards the detector row and the detector excitation is depending on the projection direction α. The model is parameterized such that for $\alpha = 0$, the projection ray is passing through the center of a tubular structure. In Figure 3.1, this ray corresponds to the bottom ray.

To be able to evaluate Equation (3.3), one needs to know for each projection direction α the path length l at which the rays travel through the tubular structure. In this example, the tubular structure is representing a cross-section of a three-dimensional object. For a two-dimensional circular object C with a radius r, its equation is given implicitly by

$$\sqrt{(c_x - x)^2 + (c_y - y)^2} = r \; . \tag{3.4}$$

The general line equation of a projection ray l_p deviating by the ray passing through the object center at an angle α (e.g. the upper ray from Figure 3.1) can be written as

$$y = \frac{-\sin(\alpha)c_x + \cos(\alpha)c_y}{\cos(\alpha)c_x + \sin(\alpha)c_y} x \; . \tag{3.5}$$

If a projection ray l_p and the object C intersect, two real-valued intersection points p_{in} and p_{out} can be computed from Equations (3.4) and (3.5). The length l as a function of c_x, c_y, s and α, at which X-rays are traversing the circular object is then

$$\begin{aligned} l(c_x, c_y, r, \alpha) &= \|p_{\text{in}} - p_{\text{out}}\| & (3.6) \\ &= 2\sqrt{r^2 + (\cos(\alpha)^2 - 1)(c_x^2 + c_y^2)} \; . & (3.7) \end{aligned}$$

Combining now Equations (3.7) and (3.3) one can picture the synthetic intensity profile generated by a perfect C-arm system when imaging a tubular structure. Before assembling the final equation another simplification needs to be introduced. In Equation (3.3), the general case of varying attenuation along the X-ray path is assumed though in this example the object C is assumed to have a constant attenuation coefficient of μ. This assumption leads to the simplified equation [1]

$$\begin{aligned} \mathcal{I}(\alpha) &= -\log(I_0) + \mu l(c_x, c_y, r, \alpha) & (3.8) \\ &= -\log(I_0) + 2\mu\sqrt{r^2 + (\cos(\alpha)^2 - 1)(c_x^2 + c_y^2)} \; . & (3.9) \end{aligned}$$

The intensity profile from Equation (3.9) is not corresponding to reality for several reasons – this is why it is referred to as idealized. The model depicted in Figure 3.1 makes several strong assumptions which are not true in clinical practice. First, tubular objects are in general not perfectly tubular; they are not even elliptic because many patients undergoing procedures from interventional radiology have calcified vessels and suffer from stenoses, aneurisms or other diseases of the vascular system. The second reason why the model does not hold is because it assumes that the vessel is running parallel to the detector plane. Again, this is not true because it can be arbitrarily oriented with respect to the detector plane. This results in Equation (3.7) being only a rough approximation which is typically underestimating the path length. The effect, which one observes in this case is foreshortening. A similar effect is caused by the fact, that the typical ray direction is not depending on a single angle α as in the given example. In order to describe a line direction passing through the origin, two angles for the azimuth and elevation are required. There are more reasons, why the given model is not general. They include background clutter, X-ray scattering, noise effects during the image acquisition, just to name a few.

---

[1] The model and X-ray parameters are omitted for the sake of brevity.

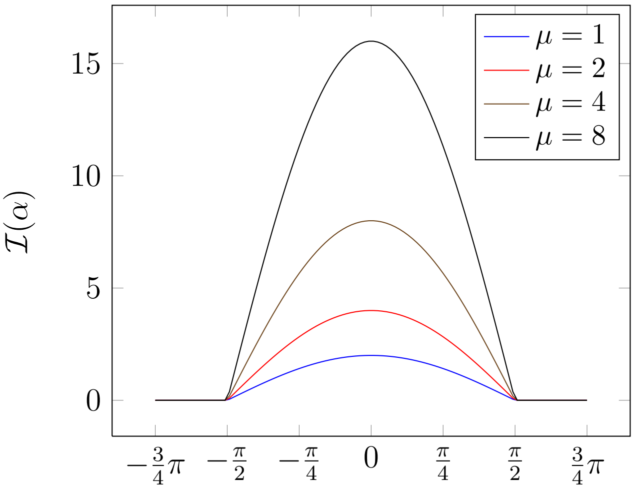


Figure 3.2: Idealized intensity responses of a tubular object with a radius $r = 1$ which is located at a distance of 1 (i.e. $c_x^2 + c_y^2 = 1$) to the X-ray source. The plot shows the responses for a constant incident intensity of $I_0 = 1$ for varying attenuation coefficients μ. The functions correspond to the output of Equation (3.9).

The question may arise, why the modeling is then done this way if there are so many reason rendering it inaccurate. One of the reasons is that incorporating all phenomena presented before is close to impossible. Let us assume for a moment a simple formula were available that modeled actual three-dimensional projection rays by incorporating a second angle β. What would one gain in the analysis? In this section, our intention is to give a rough idea about the image generation process and how close different models proposed by different authors are to reality. It is not the intention to create a model that perfectly resembles reality.

But there is more to ask since it could well be that a model which is closer to reality would allow for a better modeling of the application. So again, let us assume we incorporated the second angle β. How could this help in a real application? There is no obvious way how that could help because in practice we have no a priori knowledge regarding the vessel orientation within the patient. The orientation is even subject to change during the intervention – one of the aspects this thesis tries to recover.

To summarize, it is true, that the model is approximative but it is likewise as close to reality as it is useful for the following comparison. Some of the properties, like e.g. the limited support of the function will not change when moving to more realistic models. In the next section, the curve model given by Equation (3.9) will be compared to models proposed by different authors.

3.1.3 Typical Tube Models

During the analysis of enhancement filters which is following in the next section, mathematical models of contour intensities are required. This is especially then true, when the analysis is of analytical nature. Thus it is not a surprise, that already many different models exists in the literature. It is also no more a surprise that there is not a single model. The previous section has given some indications why the derivation of a generic model is hardly possible.

The different models which are used in today's literature have different origins. Some of them are likely to be evidence based. The investigation of image data, be it of medical nature or not, in combination with the knowledge about what is actually being images allows researchers to follow such a path. Other reasons may be that some models allow for a simplified analytical investigation of model properties. Since in image processing the convolution of image data with image filters is a fundamental procedure and because it has been already shown that especially for the analysis of edges and ridges Gaussian image filters are most appropriate, having Gaussian image profiles is beneficial. This is the case because in such a scenario, the convolution can easily be expressed analytically as it is just another Gaussian with a different mean and variance.

A commonly used (e.g. [70, 71, 39]) model is the block model or bar shaped model. Besides its spatial location x, the model is only parameterized by a radius r and the ridge height h. It can be defined as a piecewise function as

$$\mathcal{I}_{\mathrm{b}}(x, r, h) = \begin{cases} h, & |x| \leq r \\ 0, & |x| > r \end{cases} . \tag{3.10}$$

An extension to this model has been proposed by Carsten Steger [71]. In his works on the detection of curvilinear structures he wanted to be able to model different contrast levels on the two ridge sides and thus introduced what he called the "asymmetrical bar shape" model

$$\mathcal{I}_{\mathrm{ab}}(x, r, a, h) = \begin{cases} 0, & x < -r \\ h, & |x| \leq r \\ h\,a, & x > r \end{cases} . \tag{3.11}$$

Another model of which we will show that it is most close to the previously derived *idealized* model is a parabolic shape. As for the block model, it is only parameterized by the object radius r and height h.

$$\mathcal{I}_{\mathrm{p}}(x, r, h) = \begin{cases} h\left(1 - \frac{x^2}{r^2}\right), & |x| \leq r \\ 0, & |x| > r \end{cases} . \tag{3.12}$$

As opposed to the asymmetrical bar shape, the parabolic function in the form presented here, does not support different contrast levels on the two vessel sides. The last, commonly used model is a Gaussian model. Here, the radius is controlled by the Gaussian's variance, effectively controlling the radius of the function. A second parameter is required to model the ridge height and it is incorporated as a simple scaling. Sato et al. [66] used a similar model in their work.

$$\mathcal{I}_{\mathrm{G}}(x, \sigma, h) = h\,\exp(-\frac{x^2}{2\sigma^2}) \tag{3.13}$$

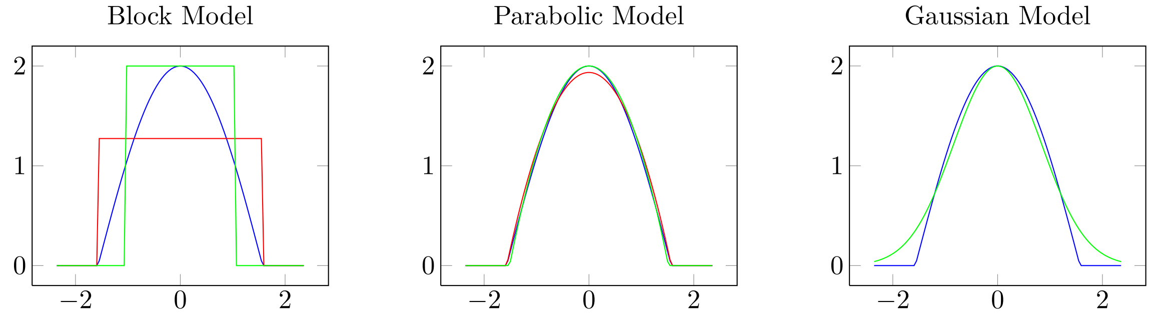


Figure 3.3: The figure shows optimal fits between three different ridge models and the *idealized* block model derived in Section 3.1.2. The models are (from left to right), a symmetric block model, an elliptical model and a Gaussian model. The red curves represent the optimal models after fixing the width to the expected value while the green curves show the optimal models after fixing their height to the expected value.

What is of interest for the remainder of this work is how well each model can approximate the *idealized* one. This question can be analyzed by finding for each model $\mathcal{I}_*$ the optimal parameters $\mathbf{p}$ such that the following quantity gets minimal

$$\int_{-\infty}^{\infty} \Big(\mathcal{I}(x) - \mathcal{I}_*(x, \mathbf{p})\Big)^2 \mathrm{d}x \ . \tag{3.14}$$

In order to be able to perform this derivation analytically, one of the parameters of each model will be fixed (e.g. the radius r) and the remaining parameter p will be computed by solving

$$\frac{\partial}{\partial p} \int_{-\infty}^{\infty} \Big(\mathcal{I}(x) - \mathcal{I}_*(x, p)\Big)^2 \mathrm{d}x = 0 \ . \tag{3.15}$$

The analysis has been carried out for the symmetrical profiles only. Including the additional degree of freedom the asymmetrical bar profile is offering does not bring any advantages in terms of error reduction – for the block model, the minimal error that can be achieved is identical to the asymmetrical bar shape. The three models that were analyzed are the block model (Equation (3.10)), the parabolic model (Equation (3.12)) and the Gaussian model (Equation (3.13)).

The function $\mathcal{I}(x) = 2\sqrt{\cos(x)^2}$ serves as a reference model. It is corresponding to the idealized model response for a vessel at a 1 unit distance to the projection center, with incident intensity of $I_0 = 1$ and a constant attenuation of $\mu = 1$. For this function both parameters are known, its exact radius $r = \frac{\pi}{2}$ and its height $h = 2$. The “optimal” model parameters are determined by fixing one out of these two parameters. This approach does not necessarily reveal the global optimum of Equation (3.14) but it is most closely related to the intended interpretation of the individual parameters and serves as an acceptable approximation.

For the block and parabolic model, both parameters can be easily fixed and the remaining parameter can be computed by solving Equation (3.15). Following this approach leads to analytical solutions for the optimal height parameters

$$h_\mathrm{b} = \frac{4}{\pi} \qquad h_\mathrm{p} = \frac{60}{\pi^3}\ . \tag{3.16}$$

The resulting curves corresponding to these parameters are the red curves in Figure 3.3. Because the Gaussian model has infinite support, it is not easily possible to fix the model's radius and thus the optimal height has not been computed in this case. Fixing the height in all three cases to $h = 2$ leads to a different set of curves and these are corresponding to the green curves in Figure 3.3. Here, the optimal parameters could only be determined numerically and they resulted in

$$w_\mathrm{b} \approx 1.04720 \qquad w_\mathrm{p} \approx 1.51513 \qquad \sigma \approx 0.84173\ . \tag{3.17}$$

In Figure 3.3, the three optimal models are plot together with the reference function. Already a visual inspection of the three plots reveals, that the parabolic model represents the most optimal fit. The block model is used primarily because of its simple mathematical properties but does not reflect well the physical properties. The Guassian model on the other-hand side results in a better approximation of physical properties but again, it is primarily chosen because of its nice mathematical properties and in particular its infinite support does not agree with reality.

3.2 Enhancement of Tubular Structures

In this section several methods for the enhancement and detection of tubular structures are reviewed and analyzed in a unified framework. Besides this review a novel view on the requirement for γ-normalized Gaussian derivatives is provided. So far, Lindeberg [49] has shown that γ-normalized derivatives are necessary for scale selection, when "the scale selection mechanism should commute with re-scalings of the image pattern". In other places γ-normalized derivatives are said to be required to ensure a "fair comparison between different scales". Here, we will provide an intuitive reason why and when γ-normalization is required in the context of the enhancement of tubular structures. Furthermore, this section will provide the required steps of an attempt at deriving formulas to choose detection parameters in an optimal way. This is not possible in all situations and it will be shown that a priori knowledge of the model is required. Ridge detection and especially the scale space analysis are oftentimes used to answer two questions.

- **Where are ridges located in the image?** This question is a placeholder for the highlighting step. Each pixel in the image gets some intensity assigned which is in the optimal case corresponding to the likelihood of that pixel to be a part of the ridge. An optimal detector would be highlighting ridge pixels only and should not create spurious responses as e.g. on edges.

- **What are the geometric properties of the ridge?** The second question is dealing with the model parameters. As shown in Section 3.1, different models exhibit different parameters like the ridge height, its radius and the contrast difference on the two sides of the ridge. The capabilities of a method to recover the correct parameters for a specific model is what is referred to by this question.

Depending on the actual application it may be the case, that both questions are subject of the enhancement or detection method. In some cases only this first question is of interest and that the second question is known beforehand. It will be shown, that in such a situation it is possible to determine some of the algorithms parameters analytically. Most methods dealing with the topic of ridge enhancement are in some way based on derivative filters. Even those methods that are working with steerable filters are usually working with approximations of special directional derivative filters. Differential geometry deals with the shape description through derivatives and thus the basic principles of differential geometry will be shortly summarized before continuing with the individual methods.

3.2.1 Differential Geometry in Image Processing

Differential geometry is based on the assumption that structures observed in an image exhibit smooth, i.e. differentiable intensity profiles. The typical coordinate system used when applying methods from differential geometry to images is depicted Figure 3.4. Here, the special case of a tubular structure is visualized but what is in general important is that images as those shown on the right side of Figure 3.4 are interpreted as three-dimensional surfaces.

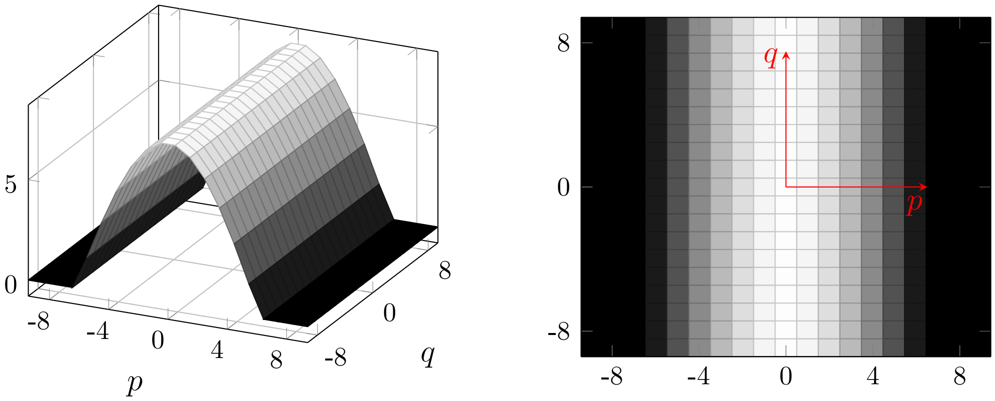


Figure 3.4: The two graphs in this figure depict a ridge model as it is generated by Equation (3.9). The left-hand side figure displays the model as a three-dimensional surface and the right-hand side figure show the corresponding two-dimensional gray-scale image. The red coordinate system with axes p and q represents the *second-order gauge coordinates* which are intrinsic to the image surface.

First and second-order derivatives are of main interest for many vision tasks and in particular the enhancement of ridges. The entities behind these derivatives are curve or surface tangents as well as normals. For the formalization of derivatives the notion of an image as a function is required. When $\mathcal{I}(x,y)$ is the image intensity of a pixel located at (x,y), derivatives with respect to the spatial coordinates x and y are written as

$$\frac{\partial \mathcal{I}(x,y)}{\partial x} = \mathcal{I}_x(x,y) \qquad \frac{\partial \mathcal{I}(x,y)}{\partial y} = \mathcal{I}_y(x,y) \ . \tag{3.18}$$

For the sake of brevity, the spatial coordinates are usually omitted and the generalized derivative in an arbitrary direction v is denoted as $\mathcal{I}_v$. The same notation can be extended for second-order derivatives resulting in $\mathcal{I}_{xx}$, $\mathcal{I}_{xy}$ and $\mathcal{I}_{yy}$, respectively. Generalizing once more, the derivative with respect to a vector u as it changes in the direction of v is expressed as $\mathcal{I}_{uv}$.

Image Gradient: The first-order derivative of an image is quantified by the image gradient and defined by

$$\nabla \mathcal{I} = \begin{bmatrix} \mathcal{I}_x \\ \mathcal{I}_y \end{bmatrix} \ . \tag{3.19}$$

The general form of the first derivative, the derivative in an arbitrary direction v can be written as

$$\mathcal{I}_v = v^\top \nabla \mathcal{I} \ . \tag{3.20}$$

Geometrically, the gradient direction at a point (x,y) is the direction of steepest ascent in that point. The steepness of the slope in the same point is given by the gradient magnitude. Furthermore, the gradient direction is equal to the normal of the level set curve passing through the point (x,y).

The Hessian Matrix: The equivalent to the gradient for second-order derivatives is the *Hessian* matrix $\mathcal{H}$. The Hessian is a symmetric real square matrix of the form

$$\mathcal{H} = \begin{bmatrix} \mathcal{I}_{xx} & \mathcal{I}_{xy} \\ \mathcal{I}_{xy} & \mathcal{I}_{yy} \end{bmatrix} \tag{3.21}$$

which has been used earlier in Equation (2.4). Similar to the computation of first-oder derivatives, second-order derivatives can be computed as

$$\mathcal{I}_{uv} = u^\top \mathcal{H} v \ . \tag{3.22}$$

An important property of the Hessian $\mathcal{H}$ is that its eigenvalues λ_1 and λ_2 are invariant to rotations.

Theorem 1. *Given a Hessian matrix $\mathcal{H}$, its eigenvalues λ_1 and λ_2 are invariant to any rotation $\mathbf{R}$ of the underlying coordinate system.*

Proof. Since the derivative of a surface is independent of the coordinate system in which the surface is expressed the following equation must hold.

$$u^\top \mathcal{H} v = (\mathbf{R}u)^\top \mathcal{H}'(\mathbf{R}v) \tag{3.23}$$
$$= u^\top \mathbf{R}^\top \mathcal{H}' \mathbf{R} v \tag{3.24}$$

From this equation follows, that the Hessian matrix $\mathcal{H}'$ in the transformed coordinate system can be expressed as

$$\mathcal{H}' = \mathbf{R}\mathcal{H}\mathbf{R}^\top \; . \tag{3.25}$$

Assuming the eigendecomposition of the real symmetric Hessian $\mathcal{H}$ is given by

$$\mathcal{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top \; , \tag{3.26}$$

the eigendecomposition corresponding to $\mathcal{H}'$ computes to

$$\mathcal{H}' = \mathbf{R}\mathbf{Q}\Lambda\mathbf{Q}^\top\mathbf{R}^\top = \mathbf{Q}'\Lambda\mathbf{Q}'^\top \; . \tag{3.27}$$

Because both matrices $\mathbf{R}$ and $\mathbf{Q}$ are orthogonal, their product $\mathbf{Q}'$ is orthogonal too. For any eigenvector v of $\mathcal{H}$ holds $\mathcal{H}v = \lambda v$ and one can easily verify that the corresponding equation for $\mathcal{H}'$ is holding as well:

$$\mathcal{H}'\mathbf{R}v = \mathbf{R}\mathbf{Q}\Lambda\mathbf{Q}^\top\mathbf{R}^\top\mathbf{R}v \tag{3.28}$$
$$= \mathbf{R}\mathcal{H}v = \lambda\mathbf{R}v \; . \tag{3.29}$$

□

This theorem, even though well known, is of particular importance for the analysis of functions dealing with the enhancement of line-like structures. A direct implication of this theorem is that there exists a coordinate system transformation $\mathbf{R}$ for every Hessian matrix which turns it into a diagonal matrix. Setting $\mathbf{R} = \mathbf{Q}^\top$ in Equation (3.27) will result in the Hessian $\mathcal{H}'$ being a diagonal matrix where the diagonal elements are equal to the eigenvalues.

To summarize, each Hessian $\mathcal{H}$ has the following mathematical properties. First, its determinant is equal to the product of its eigenvalues and as seen before, it is invariant to rigid transformations of the underlying coordinate system. It can similarly be shown that its trace is invariant under rotations and equal to the sum of its eigenvalues. Because $\mathcal{H}$ is real and symmetric, the matrix has real valued eigenvectors which are mutually orthogonal and thus provide a basis of $\mathbb{R}^N$ where N is the dimensionality – for image processing applications common values are $N = 2$ or $N = 3$.

For the analysis of ridges, geometrical interpretations of the Hessian are useful too. Assuming, the eigendecomposition of a N-dimensional Hessian is performed such that the eigenvalues are ordered by magnitude, i.e. $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_N|$, one can use the notion of first, second, and N-th eigenvector. Geometrically, the first eigenvector, i.e. the one corresponding to the eigenvalue of largest absolute value, is the direction of largest curvature. From an image processing point of view, this can be interpreted such that this eigenvector is pointing in the direction in the image in which the intensities have maximal variability. The N-th eigenvector, the one corresponding to the eigenvalue being closest to zero, is the direction of least curvature. Again, in terms of image processing this means the N-th eigenvector is pointing in the direction with least intensity variations.

This means for a two-dimensional ridge, that when located directly on top of the ridge, there is in the optimal and noise free case, one eigenvector pointing along the ridge. Its corresponding eigenvalue should be equal to zero. A second eigenvector is pointing orthogonal to the ridge, in its normal direction and the associated eigenvector $|\lambda_1|$ is expected to be greater than zero.

Gauge Coordinates: A general concept of differential geometry is that surface properties like the aforementioned curvature, can be decoupled from a global coordinate system. These properties can be analyzed with respect to a local coordinate system induced by the local surface of image intensities. Such coordinate systems exist for both, first- and second-order derivatives.

The gradient direction and its normal build the basis of the first-order gauge coordinate system. Geometrically, the normal of the gradient is corresponding to the tangent of the level set of image intensities in the current pixel location and the gradient itself is corresponding to the normal. The gauge coordinates can be expressed as directions $w = \nabla\mathcal{I} / \left\|\nabla\mathcal{I}\right\|$ and $v = w_{\perp}$.

The second-order gauge coordinates are defined by the principle direction, i.e. the eigenvectors of the Hessian matrix $\mathcal{H}$. Like the first-order gauge basis, the direction of the eigenvectors is defined by the image intensities and they are oftentimes denoted as p and q. Figure 3.4, is representing a synthetic ridge and its corresponding gauge coordinate system.

3.2.2 The Hessian and its Relation to Tube Models

In the next section this work tries to find analytical answers to questions related to the Hessian based enhancement of line-like structures. Because of the simplified shape (a diagonal matrix), the Hessian will be expressed in the gauge coordinate system. All findings in this coordinate system are true for all other coordinate systems because of the invariance of the eigenvalues to rotations.

For the analysis, an analytic form of the Hessian and therefore its eigenvalues will be presented. Its form is depending on the line models which were introduced in Section 3.1.2 and 3.1.3 as well as the model parameters. Given such a representation it is then possible to perform an analysis of the vesselness responses proposed by different authors and this analysis can be related to the model parameters.

In order to be able to derive the analytical expression of the Hessian, Gaussian derivatives are required. In fact, because the investigation is solely carried out in the gauge coordinate system only the second-order derivatives of that system will be needed.

The γ-normalized, one-dimensional Gaussian is defined as

$$g(x,t,\gamma) = t^{\frac{\gamma}{2}} \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{x^2}{2t}\right) , \tag{3.30}$$

and its derivatives have the form

$$g_x(x,t,\gamma) = \frac{\partial g(x,t,\gamma)}{\partial x} \tag{3.31}$$

$$g_{xx}(x,t,\gamma) = \frac{\partial g(x,t,\gamma)}{\partial x^2} . \tag{3.32}$$

Because of the separability of multi-dimensional Gaussian filters, the second-order two-dimensional Gaussian G_{xx} can be written as

$$G_{xx}(x,y,t,\gamma) = g_{xx}(x,t,\gamma)g(y,t,\gamma) \tag{3.33}$$

$$= -\frac{t^{\gamma-3}(t-x^2)}{2\pi} \exp\left(-\frac{x^2+y^2}{2t}\right) . \tag{3.34}$$

In all equations $t = \sigma^2$ represents the scale-space parameter as it was introduced by Lindeberg [46]. For the remainder of this discussion, the derivative subscripts will be changed to p and q respectively, in order to emphasize that the remaining discussion is carried out in the pq gauge coordinate system. The system is described in Section 3.2.1 on differential geometry and it is visualized in Figure 3.4.

Given an arbitrary function $\mathcal{I}_*$ representing a tubular shape's intensity profile being aligned with the q-coordinate axis as it is depicted in Figure 3.4, the two-dimensional Hessian has the following form

$$\mathcal{H} = \begin{bmatrix} G_{pp} * \mathcal{I}_* & 0 \\ 0 & G_{qq} * \mathcal{I}_* \end{bmatrix} = \begin{bmatrix} G_{pp} * \mathcal{I}_* & 0 \\ 0 & 0 \end{bmatrix} . \tag{3.35}$$

Without loss of generality, we assume the following ordering on eigenvalues, $|\lambda_1| \leq |\lambda_2|$ and additionally for three-dimensional cases $|\lambda_2| \leq |\lambda_3|$. Given such an ordering, the diagonalized Hessian matrix from Equation (3.35) has a single zero eigenvalue $\lambda_1 = 0$ and a second eigenvalue $\lambda_2 = G_{pp} * \mathcal{I}_*$.

3.2.3 Properties of Tube Enhancement Filters

The filters and functions which are discussed in this section are well known and widely applied – especially in the field of medical image processing. Historically, the filter and works introduced here were developed between 1995 and 1999. Earlier works in line enhancement exist but investigating more but a few common filters is out of the scope of this work. The different methods are discussed in no particular order.

Sato's Feature Enhancement

Sato et al. [66] propose a filter for the enhancement of three-dimensional tubular structures which utilizes all three eigenvalues. Assuming eigenvalues are ordered as

$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, one expects the following properties for the eigenvalues when filtering tubular objects. First, λ_1 is expected to be an eigenvalue close to zero. Second, the other two eigenvalues are expected to exhibit similar values, i.e. $|\lambda_2| \approx |\lambda_3|$ when circular structures are processed.

Following these observations, Sato et al. formulated the following filter for bright lines

$$\lambda_c = \min(-\lambda_2, -\lambda_3) \qquad \mathcal{V}_{\mathrm{S}} = \begin{cases} \lambda_c \exp(-\frac{\lambda_1^2}{\alpha_1^2 \lambda_c^2}) & \lambda_1 \leq 0 \\ \lambda_c \exp(-\frac{\lambda_1^2}{\alpha_2^2 \lambda_c^2}) & \lambda_1 > 0 \\ 0 & \lambda_c \leq 0 \end{cases} , \tag{3.36}$$

where they chose $\alpha_1 = 0.5$ and $\alpha_2 = 2.0$ in their experiments. In the two-dimensional case, and in the pq-coordinate frame, the function reduces to

$$\mathcal{V}_{\mathrm{S}} = \begin{cases} -\lambda_2 & \lambda_2 < 0 \\ 0 & \text{otherwise} \end{cases} . \tag{3.37}$$

We have seen before, that in the pq-coordinate frame $H_{pp} = \lambda_2 = G_{pp} * \mathcal{I}_*$. Applying this equation to the block-model from Equation (3.10) results in the following convolution

$$H_{pp}(x, t, \gamma, r, h) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G_{pp}(u, v, t, \gamma)\, \mathcal{I}_b(x - u, y - v, r, h)\, \mathrm{d}u\, \mathrm{d}v . \tag{3.38}$$

In this equation, an additional parameter y has been added to the function $\mathcal{I}_b$ from Equation (3.10) without modifying the function itself, i.e. the response is actually independent of y – exactly what we expect to happen in the pq-coordinate frame. The convolution result is no more depending on y and thus it has been omitted on the right-hand side. The introduction of the second dimension is nonetheless important because it affects the influence of γ. Because $H_{pp} = \lambda_2$, substituting Equation (3.38) and (3.37) results in the following function

$$\mathcal{V}_{\mathrm{S}}(x, t, \gamma, r, h) = \begin{cases} -H_{pp}(x, t, \gamma, r, h) & H_{pp}(x, t, \gamma, r, h) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.39}$$

which can now be analyzed for its extremal properties.

First, it is desired that the function takes a maximum when $x = 0$. Given the derivative, it can easily be shown that the function is extremal at $x = 0$ by verifying that

$$\left. \frac{\partial \mathcal{V}_{\mathrm{S}}(x, t, \gamma, r, h)}{\partial x} \right|_{x=0} = 0 \tag{3.40}$$

holds. In order to ensure a maximum we need to ensure in addition that the second-derivative with respect to x is negative and it turns out that this is only the case when $-2r^3 + 6rt > 0$. Given r is positive, which it is because curves do not have negative radius, this is exactly true when

$$t > \frac{r^2}{3} . \tag{3.41}$$

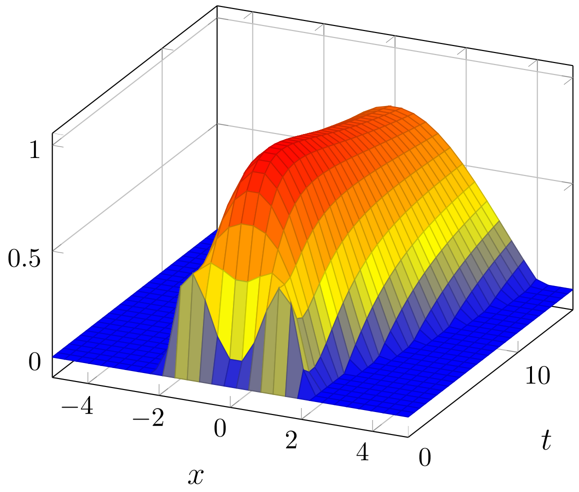


Figure 3.5: The figure shows the vesselness response of the filter proposed by Sato et al. for a block model $\mathcal{I}_b(x, 2, 2)$ with width and height equal to 2. The ditch for small scale-space values of t and $x = 0$ indicates, that the responses are not meaningful for all scale-space parameters.

This fact can be visually observed in Figure 3.5. Only for scale-space parameters $t > \frac{r^2}{2}$ the energy function exhibits a maximum at $x = 0$.
The second desired property of the energy $\mathcal{V}_S(x, t, \gamma, r, h)$ is that it takes its maximum not only for $x = 0$ but also at a specific scale t. A popular choice in the past was to choose $t = r^2$ which is corresponding to $\sigma = r$. Being able to shape the function such that its extremum in the scale-space is a function of the curve radius enables users to determine the radius by searching for the optimum in scale-space. It turns out, that exactly this property can be controlled with the scale-space normalization as introduced by Lindeberg. Formally, we seek γ such that the following equation holds

$$\left.\frac{\partial \mathcal{V}_S(x, t, \gamma, c, r, h)}{\partial t}\right|_{\substack{x=0\\t=r^2}} = 0\ . \tag{3.42}$$

For the block model $\mathcal{I}_b$ it turns out, that the optimal value is $\gamma = 1$. A problem is, that both of these values are model dependent. Using a different model, as e.g. the parabolic model I_p results in different values. Table 3.1 contains exemplary γ values for the block and parabolic model. As mentioned earlier, these value are oftentimes chosen such that $t = r^2$. Analytic expressions for Equations (3.38) and (3.39) as well as analytic

Table 3.1: Required parameter choices for Sato's line enhancement.

Model Type	Block Model $\mathcal{I}_b$	Parabolic Model $\mathcal{I}_p$
Maximum for $x = 0$, $\mathcal{V}_S$	$t > \frac{r^2}{3}$	none
Maximum for $t = r^2$ $(\sigma = r)$, $\mathcal{V}_S$	1	1.217474755

expressions of the optimal γ parameter for the parabolic model from Table 3.1 can be found in the Appendix B.1.1.

Frangi's Feature Enhancement

A different feature enhancement filter has been proposed by Frangi et al. [22]. The authors propose a filter which utilizes all eigenvalues. In the two-dimensional case, the filter consists of a *blobness* measure and a *structuredness* measure. The blobness term is similar to the function proposed by Sato et al. and is based on

$$\mathcal{R}_{\mathcal{B}} = \frac{\lambda_1}{\lambda_2} \,. \tag{3.43}$$

An object's *structuredness* is measure by the Frobenius norm of the Hessian and it is computed in 2D as

$$\mathcal{S} = \|\mathcal{H}\|_{\mathrm{F}} = \sqrt{\lambda_1^2 + \lambda_2^2} \,. \tag{3.44}$$

In fact, this is the only term controlling the vesselness in the pq-coordinate frame because $\mathcal{R}_{\mathcal{B}}$ is vanishing when $\lambda_1 = 0$. The general vesselness is measured as

$$\mathcal{V}_{\mathrm{F}} = \begin{cases} \exp\left(-\frac{\mathcal{R}_{\mathcal{B}}^2}{2\beta^2}\right)\left(1 - \exp\left(-\frac{\mathcal{S}^2}{2c^2}\right)\right) & \lambda_2 \leq 0 \\ 0 & \text{otherwise} \end{cases} \,. \tag{3.45}$$

Utilizing again that in the pq-coordinate frame $H_{pp} = \lambda_2$ holds, this term reduces to

$$\mathcal{V}_{\mathrm{F}}(x, t, \gamma, c, r, h) = \begin{cases} 1 - \exp\left(-\frac{H_{pp}(x,t,\gamma,r,h)^2}{2c^2}\right) & \lambda_2 \leq 0 \\ 0 & \text{otherwise} \end{cases} \,. \tag{3.46}$$

During the analysis of Frangi's feature enhancement, it turns out that the scale-space parameter t and the scale-space normalization value γ are subject to the same constraints as for Sato's filter. Intuitively, this is the case because in the pq-coordinate frame, the *structuredness* measure is very similar to the measure used by Sato et al. The only difference is that Frangi formulated the problem probabilistic by modeling the expected Frobenius norm of the Hessian by a Gaussian. Contrary to Sato's filter, the Gaussian modeling requires to choose an additional parameter c, the variance. The additional degree of freedom offered by Frangi's parameter leads to the problem of proper parameter choice. The authors propose a heuristic value equal to half of the

Table 3.2: Required parameter choices for Frangi's line enhancement.

Model Type	Block Model $\mathcal{I}_b$	Parabolic Model $\mathcal{I}_p$
Minimal scale parameter, $\mathcal{V}_{\mathrm{F}}$	$t > \frac{r^2}{3}$	none
Optimal γ value ($t = r^2$, $\sigma = r$), $\mathcal{V}_{\mathrm{F}}$	1	1.217474755

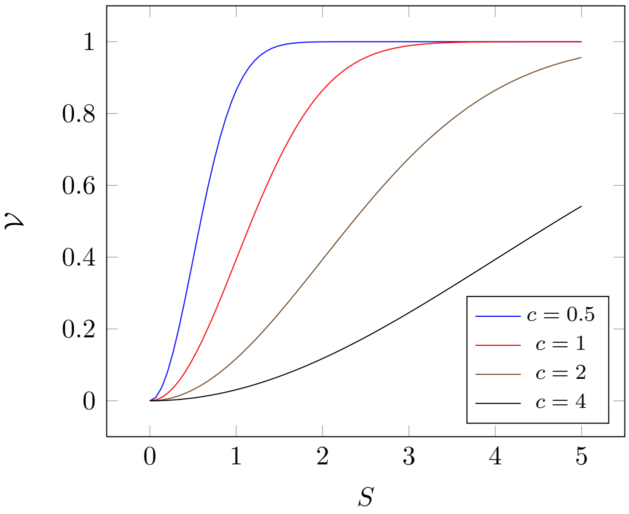


Figure 3.6: This plot shows the effect of the choice of the parameter c for Frangi's *structuredness* measure $1 - \exp\left(-\frac{S^2}{2c^2}\right)$. When c is chosen too small, the filter responses of neighboring but large values of S are hard to distinguish.

maximum Hessian norm $S_{\max}$ observed in an image. Following this advice and setting $c = \frac{S_{\max}}{2}$ leads to the following upper bound for the *structuredness* response

$$1 - \exp\left(\frac{S^2}{2c^2}\right) = 1 - \exp\left(\frac{S^2}{2c^2}\right) \leq 1 - \exp\left(-2\right) \approx 0.8646647168 \; . \tag{3.47}$$

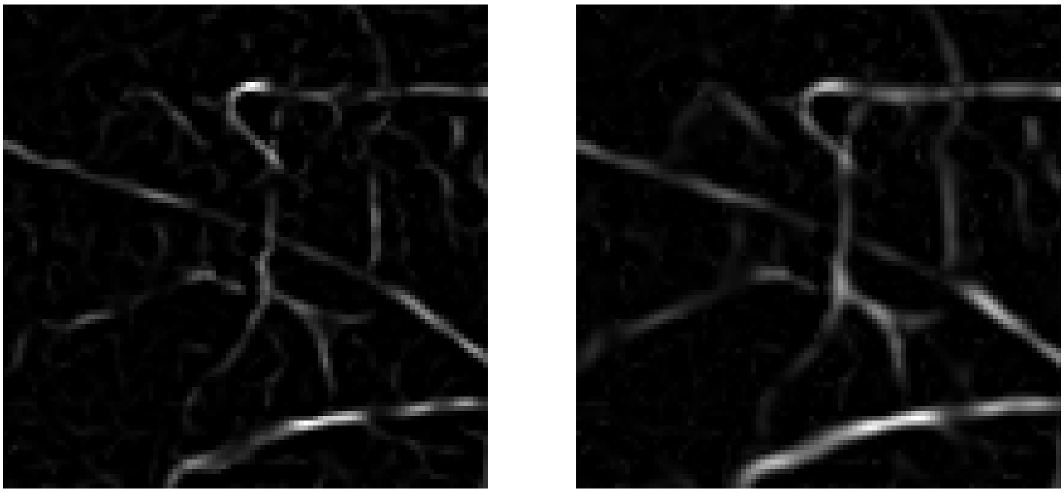
Following this idea, a more generalized approach is to find a factor n such that for $c = nS_{\max}$ the upper bound is defined by $1 - \epsilon$. It turns out, that n can be computed as

$$n = \frac{1}{\sqrt{-2 \ln \epsilon}} \; . \tag{3.48}$$

From a practical point of view, this choice is similar to computing the vesselness response for every pixel and rescaling the responses such that the maximal value is mapped to $1 - \epsilon$. What is important for the choice of c is that c is not chosen too small. Figure 3.6 shows this problem. Assuming the Hessian norms are uniformly distributed between 0 and 5 and setting $c = 0.5$, more than 60% of the response values are mapped so close to 1 that they can hardly be distinguished.

In both vessel-enhancement methods discussed until now, the scale-space normalization constant γ has been determined by isolating γ in Equation (3.42). It is interesting to note that this equation is depending on the side-constraint $t = r^2$ and this choice is effecting at which scale, i.e. for which σ the function takes its maximum. If the constraint were generalized to $t = (nr)^2$, it were possible to design a function taking its maximum at a different scale which were resulting in additional smoothing. Such a modification could be beneficial in cases where the radii are small and when the input profiles were corrupted by noise.

Changing Equation (3.42) to the side-constraint $t = (nr)^2$ instead of $t = r^2$ and isolating



(a) Frangi Features – $t = r^2$ (b) Frangi Features – $t = (1.5r)^2$

Figure 3.7: Comparison between Frangi's vesselness enhancement with different scale-space normalization constants. The images were created with scales $\sigma_i \in \{0.5, 1.5, 2.0\}$ and $1.5\sigma_i$. The scale-space normalization constants are computed according to Equation (3.49).

γ leads to

$$\gamma_f(n) = \frac{\exp\left(-\frac{1}{2n^2}\right)\sqrt{\pi}}{n^2\left(-2\exp\left(-\frac{1}{2n^2}\right)\sqrt{\pi} + \sqrt{2}\,\mathrm{erf}\left(\frac{1}{n\sqrt{2}}\right)n\pi\right)} \; . \tag{3.49}$$

An example of the additional smoothing effect is provided in Figure 3.7. The latter equation is exactly then of importance, when the user desires to determine the actual vessel width from the optimal scale.

Koller's Feature Enhancement

The filter that has been proposed by Koller et al. [39] is following a slightly different approach as compared to the two previously discussed filters. In this case, the idea is based on the assumption that ridge profiles exhibit two symmetric edges located at $-r$ and r. The authors propose to use shifted first-derivative Gaussian filters for the computation. In the one-dimensional case, the profile edges can be detected by a non-linear combination of the following filters

$$\mathcal{R}_l(x,t,\gamma,r,h) = \quad G_x(x-\sqrt{t},t,\gamma) * \mathcal{I}_*(x,r,h) \tag{3.50}$$

$$\mathcal{R}_r(x,t,\gamma,r,h) = -G_x(x+\sqrt{t},t,\gamma) * \mathcal{I}_*(x,r,h) \; . \tag{3.51}$$

A scan-line of the filter responses is depicted in Figure 3.8. Koller et al. propose several possibilities for the non-linear combination of the responses $\mathcal{R}_l$ and $\mathcal{R}_r$. The two combinations which are of most interest are

$$\mathcal{V}_{\mathrm{K_a}} = \begin{cases} \sqrt{\mathcal{R}_l \mathcal{R}_r} & \mathcal{R}_l > 0 \text{ and } \mathcal{R}_r > 0 \\ 0 & \text{otherwise} \end{cases} , \tag{3.52}$$

corresponding to the geometrical mean and

$$\mathcal{V}_{\mathrm{K_b}} = \begin{cases} \min\left(\mathcal{R}_l, \mathcal{R}_r\right) & \mathcal{R}_l > 0 \text{ and } \mathcal{R}_r > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{3.53}$$

The latter form is equivalent to the filter $\mathcal{R}_s$ presented by Koller et al. [39] though Equation (3.53) is considered to be more clear. So far, there are a few differences compared with the original work. In order to stay within the same scale-space framework, the variable σ has been substituted by $\sqrt{t}$, where t is corresponding to the scale-space parameter as introduced by Lindeberg [46]. The second difference is that the shifted Gaussian derivatives are written as $\pm G_x(x \mp \sqrt{t}, t, \gamma)$, i.e. with a single scale-space parameter t.

For the extension of the filter to two dimensions, one needs to compute the first-order derivative responses in the direction of the ridge normal. In Section 3.2.1 is explained, that the eigenvector corresponding to the largest absolute eigenvalue is pointing in the direction of highest curvature. For a ridge this direction is equivalent to the curve normal and it will be denoted as $\mathbf{n} = (n_x, n_y)$. Given this normal direction, the formulas of Equations (3.50) and (3.51) can be compactly written as

$$\mathcal{R}_l = \mathbf{n}^\top \nabla \Big(G(t,\gamma) * \mathcal{I} \Big)(x - tn_x, y - tn_y) \tag{3.54}$$

$$\mathcal{R}_r = -\mathbf{n}^\top \nabla \Big(G(t,\gamma) * \mathcal{I} \Big)(x + tn_x, y + tn_y) \; . \tag{3.55}$$

which is corresponding to Equation (3.20) presented in Section 3.2.1. Because of the separability of the Gaussian G, the same equation can be expressed more verbosely as

$$\mathcal{R}_l = G_x(x - tn_x, y - tn_y, t, \gamma)n_x + G_y(x - tn_x, y - tn_y, t, \gamma)n_y \tag{3.56}$$

$$\mathcal{R}_r = -G_x(x + tn_x, y + tn_y, t, \gamma)n_x - G_y(x + tn_x, y + tn_y, t, \gamma)n_y \; . \tag{3.57}$$

The remaining discussion of extremal points of the filter is based on the Equations (3.50) and (3.51). These equations are corresponding to the two-dimensional filters when being applied in gauge coordinates. Similar to what has been done earlier in this work, one needs to investigate the following functions

$$\left. \frac{\partial \mathcal{V}_{\mathrm{K}}(x,t,\gamma,r,h)}{\partial x} \right|_{x=0} = 0 \qquad \left. \frac{\partial \mathcal{V}_{\mathrm{K}}(x,t,\gamma,r,h)}{\partial t} \right|_{\substack{x=0 \\ t=r^2}} = 0 \; . \tag{3.58}$$

These equations are analyzed for both possible linear combinations $\mathcal{V}_{\mathrm{K_a}}$ as well as $\mathcal{V}_{\mathrm{K_b}}$. Furthermore, the optimal parameters are computed for the block model as well as the parabolic mode. What is initially surprising is that the values computed in this work

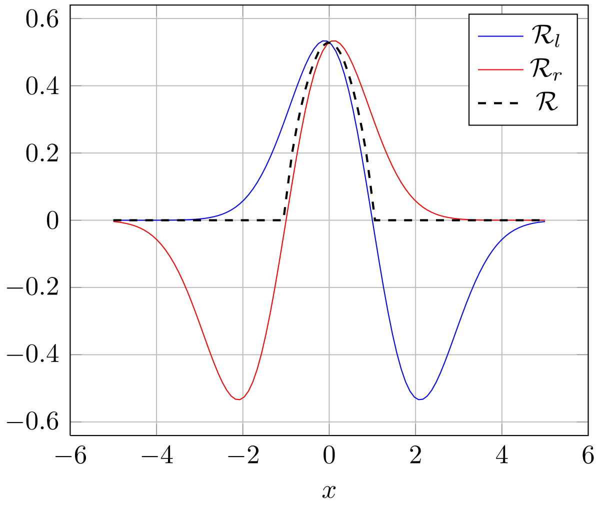


Figure 3.8: Shifted first-derivative filter responses of a parabolic profile $\mathcal{I}_p$ with radius $r = 1$ and height $h = 2$. The filter response $\mathcal{R}_l$ is depicted in blue. The red line is corresponding to the filter $\mathcal{R}_l$. Finally, the dashed line represents the non-linear combination $\mathcal{R}_s = \sqrt{\mathcal{R}_l \mathcal{R}_r}$ in cases where $|x| \leq r$ and zero otherwise.

differ from those presented originally by Koller et al. [39]. In the initial work, the authors analyzed the function

$$\mathcal{R}_{\mathrm{Line}_\sigma} = c_\sigma \left(G'_\sigma \left(x - \frac{w}{2} \right) - G'_\sigma \left(x + \frac{w}{2} \right) \right) \tag{3.59}$$

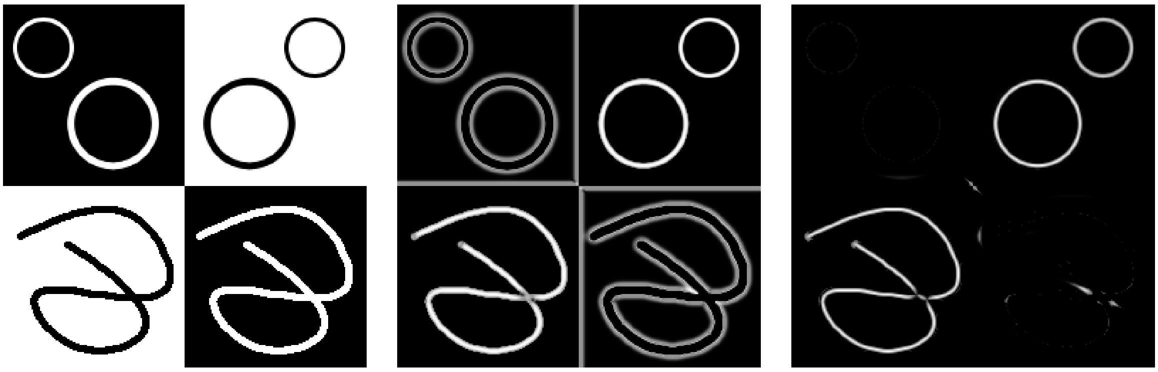
which is fundamentally different form analyzing the function $\mathcal{V}_{\mathrm{K_a}}$ or $\mathcal{V}_{\mathrm{K_b}}$. Furthermore, Koller et al. used a unnormalized Gaussian function which influences the choice of the scale-space parameter γ and which is unified in this work. The results of this analysis are summarized in Table 3.3.

Table 3.3: Required parameter choices for Koller's line enhancement.

Model Type	Block Model $\mathcal{I}_b$	Parabolic Model $\mathcal{I}_p$
Maximum for $x = 0$, $\mathcal{V}_{\mathrm{Ka}}$	none	none
Maximum for $t = r^2$ $(\sigma = r)$, $\mathcal{V}_{\mathrm{Ka}}$	0.6565176425	0.6565176425
Maximum for $x = 0$, $\mathcal{V}_{\mathrm{Kb}}$	none	none
Maximum for $t = r^2$ $(\sigma = r)$, $\mathcal{V}_{\mathrm{Kb}}$	0.8036851245	0.8036851145

Tube Enhancement Filter Experiments

An important criterion for algorithms enhancing line-like structures is to which extend they respond to other image features such as step edges or image noise. It can be shown analytically, that the filter proposed by Frangi et al. exhibits three extrema for step



(a) Toy Example (b) Frangi et al. (c) Koller et al.

Figure 3.9: Filter responses on a toy example. The scale-space responses are computed with the optimal σ values for the parabolic model. Scale values are ranging from 1 to 4 and an overall of 21 scales were probed, i.e. $t \in \{1, 1.15, 1.3, \dots, 4.0\}$.

edges. For a step edge $\mathcal{I}_s$ defined as

$$\mathcal{I}_s(x,h) = \begin{cases} 0 & x \leq 0 \\ h & \text{otherwise} \end{cases} \tag{3.60}$$

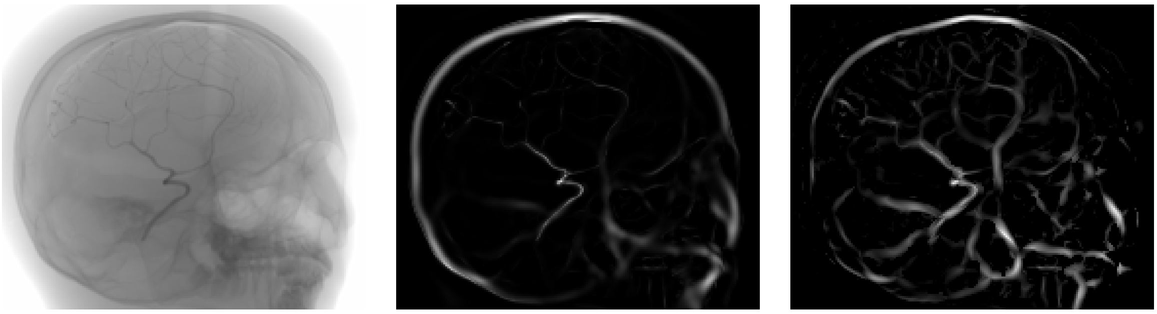
the filter response becomes

$$\mathcal{V}(x,t,\gamma,c,h) = 1 - \exp\left(-\frac{\exp\left(-\frac{x^2}{t}\right) h^2 x^2 t^{-3+2\gamma}}{4\pi c^2}\right) . \tag{3.61}$$

Computing the extrema of $\mathcal{V}(x,t,\gamma,c,h)$ depending on the variable x leads to three solutions. These are

$$x \in \left\{0, +\sqrt{t}, -\sqrt{t}\right\} . \tag{3.62}$$

These additional responses on step edges can be observed as expected in the toy example presented in Figure 3.9(b). The second filter discussed in this section does not exhibit such behavior on step edges. In fact, the nonlinear combination of the filter is effectively suppressing these kind of edges. The toy example of Figure 3.9(a) contains four edges on the horizontal and vertical image centers and these are suppressed in the response Figure 3.9(c). The remaining noise responses in the Figure 3.9(c) are hardly possible to remove. In the bottom right square of the toy example which show a white curve on black background, Koller et al.'s filter produces spurious responses. However, these responses are according to what is modeled by the filter. Close to the crossing of the white lines, two of the path are approaching each other in a steep angle and thus start acting as the borders of a dark line structure which is correctly detected by the filter. A real world example of different filter responses is provided in Figure 3.10.
The above analysis provides a simple derivation for properly configuring and using two of the most popular feature enhancement algorithms. In particular, it was derived how



(a) Contrasted Brain Image (b) Frangi et al. (c) Koller et al.

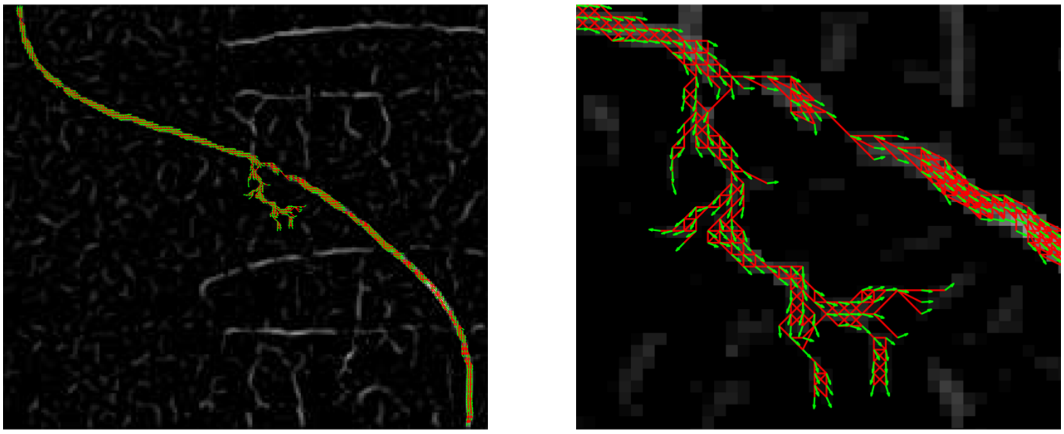
Figure 3.10: Filter responses of a medical example. The input image on the left shows a head in which the vessels supporting the brain are enhanced by contrast agent.

to properly normalize filter responses within the scale-space when using normalized Gaussian derivative filters base on Euqations 3.30 to 3.32. Especially Sato's and Frangi's vessel enhancement filters are commonly used in medical applications. Both algorithms are integrated in the well known and widely used Insight Toolkit [34]. From a mathematical point of view, Frangi's filter is an extension to Sato's filter because it does not only include the *structuredness* but it is also modeling the blobness as the ratio between the smallest and largest eigenvalue. This additional term acts as an additional discrimination factor to separate line-like from blob-like structures.

Koller's line enhancement algorithm on the other hand side seems still being less accepted as other alternatives. Here, it has been shown that the filter has several advantageous properties. First, the filter does not require any constraints on the minimal scale selection in order to ensure a maximal response on ridge tops and this independently of the ridge models analyzed in this work. Second, Koller et al.'s filter does not create any spurious responses on step-edges which means the method is causing less spurious responses. A third advantage is the fact that the filter does not require the choice of an additional parameter and thus no individual tuning. However, the method is not without problems though they are primarily of practical nature. Because the filter requires the gradient projection onto ridge normals, the method requires to calculate the Hessian and its eigendecomposition as well as the image gradient which is not needed for Frangi et al.'s filter. The result is a slightly worse run-time behavior. Overall, it has been decided for this work, that the advantageous outweigh the run-time impact and it is suggested to favor the filter proposed by Koller et al.

3.3 Semi-Automatic Detection

In Section 1.4.1 two modules for motion compensation are introduced. The first model deals with the initial tool detection and is designed to create a set of pixels $\mathcal{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^M\}$ from a given key- or reference frame acquired at time t_{key}. In the optimal case, this detection process would be fully automatic. However, currently no method for the fully automatic detecting of guide-wires in fluoroscopy exists which



(a) Overview (b) Close-up

Figure 3.11: Closeup of minimal initialization-graph with branching points.

produces results of sufficient accuracy [54]. In this section we present an approach for the semi-automatic detection of guide-wires. The method is based on the second-order structural information contained in the images which has already been used for the enhancement of ridges. It utilizes the directional information provided by the Hessian analysis to establish neighborhood relationships within the image and form a graph with pixels being the nodes where the graph's edges connect neighboring pixels. Guide-wires are then detected by manual selection of start- and end-nodes within this graph and by computing a shortest path [20] between these terminal nodes. The approach is similar to the work recently presented by [67]. Other algorithms which motivated the development this method are e.g. Livewire (also known as Intelligent Scissors) by Mortensen et al. [56] or the point linking method Steger [71, 70] proposes.

In contrast to Livewire, the proposed method creates only a sparse graph of the pixel neighborhoods which results in faster processing. However, the presented approach may in some cases result in segmentation failures because it may happen that the generated graph contains disconnected components. A prerequisite of Steger's method is the existence of multiple pre-detected sets of pixels, where it is assumed that all pixels represent objects of interest. This is not the case for the method at hand. Here, it is required that two pixels on the curve are manually selected and the algorithm tries to establish a connection between the selected points while being guided by image features.

Graph Creation

Here, an initialization procedure is proposed that requires the user to select two points close to the extremities or end-points of a single medical tool. The basic idea is that the algorithm tries to connect these two points by traversing pixels supposedly belonging to the medical tool. In fact, the algorithm traverses a graph $G = (V, E)$ with nodes V corresponding to individual pixels. The edges E of the graph are connecting neighboring pixels and to each edge $e_i \in E$ is a weight w_i associated which is small for those edges belonging most likely to the medical tool.

Given two user-selected seed points $\mathbf{s}_1$ and $\mathbf{s}_2$, the system initializes a graph $G_1 = (V_1, E_1)$ with the seed point $\mathbf{s}_1$ as its first node, i.e. $V_1 = \{\mathbf{s}_1\}$. Additionally to the graph initialization, the node initializes a stack $U = \{\mathbf{s}_1\}$ of unprocessed nodes. From here, the algorithm proceeds as follows. As long as the list of unprocessed nodes is not empty, i.e. $U \neq \oslash$, a node $\mathbf{v}_c$ is popped from the stack U. For this node, up-to three neighbors $\mathbf{n}_1, \mathbf{n}_2$ and $\mathbf{n}_3$ are selectively added to the graph G_1. In case a node $\mathbf{n}_i$ is not yet part of the graph, i.e. $\mathbf{n}_i \notin V_1$, it is added to the stack of unprocessed nodes and the stack becomes $U = U \cup \{\mathbf{n}_i\}$. The weighted graph itself is extended by each neighbor $\mathbf{n}_i$ as

$$V_1 = V_1 \cup \{\mathbf{n}_i\} \tag{3.63}$$
$$E_1 = E_1 \cup \{(\mathbf{v}_c, \mathbf{n}_i, w_i)\}\ . \tag{3.64}$$

This growing process is continued until the stack U is empty and results in a first graph G_1. In case the second seed point is already a part of the graph G_1, one sets $G = G_1$ and continues with a single pair shortest-path computation. If $\mathbf{s}_1 \notin V_1$, the process is repeated in the same way with the second, yet unprocessed seed point $\mathbf{s}_2$ which will result in a graph G_2 and the final graph G becomes $G = G_1 \cup G_2$. The final set of pixels $\mathcal{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^M\}$ representing the medical tool is now computed as the solution of the shortest path problem on G for the node pair $\mathbf{s}_1$ and $\mathbf{s}_2$.

Several question are now remaining. First, it needs to be defined how the neighbor selection is working. Next, it needs to be discussed what happens in the aforementioned case, when $G_1 \cap G_2 = \oslash$. This means, G consists of two disconnected components and in this case a special fallback routine is required. Finally, computing shortest paths requires edge weights w_i and it needs to be defined how to compute these weights.

Neighborhood Selection

During the graph creation process, a neighbor selection is required. It is claimed in the introduction, that the method is utilizing directional information contained in the image data. As described in Section 3.2.1, the Hessian $\mathcal{H}$ of image intensities is providing the required directional information. For any point $\mathbf{v}_c$ being currently processed by the growing routine, the local Hessian matrix is analyzed. Its eigenvector $\mathbf{t}_c$ corresponding to the smallest absolute eigenvalue is equivalent to the tangential direction of the curve in the point $\mathbf{v}_c$. Examples of such eigenvectors are represented by the green arrows in Figure 3.11(b).

The directional information of the eigenvector or tangent is then used for the neighbor selection in the following way. Assuming the tangent $\mathbf{t}_c$ in the current point $\mathbf{v}_c$ is defined as

$$\mathbf{t}_c = \begin{pmatrix} t_c^x \\ t_c^y \end{pmatrix}\ , \tag{3.65}$$

where $\|\mathbf{t}_c\| = 1$, the corresponding direction α can be computed as $\alpha = \operatorname{atan2}(t_c^y, t_c^x)$. This angle will fall into one of eight quadrants each spanning 45° and each defining a single triplet of neighbors. The quadrants and the selected neighbors $\mathbf{n}_1, \mathbf{n}_2$ and $\mathbf{n}_3$ for an exemplary tangent direction are sketched in Figure 3.12.

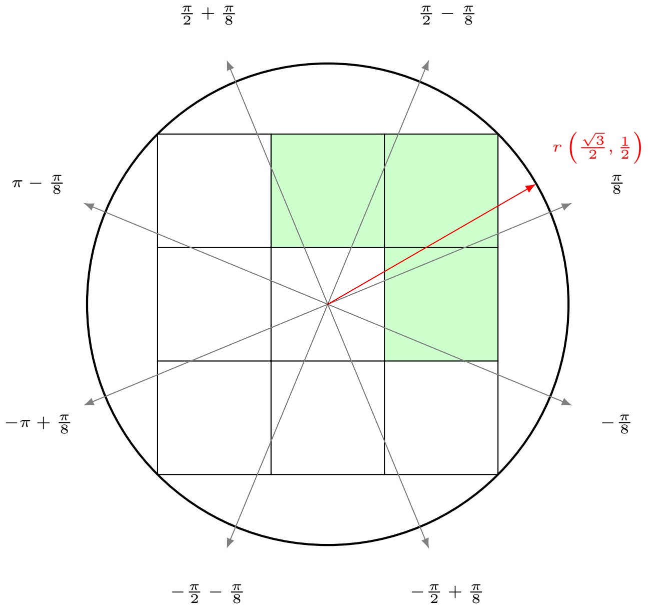

$\frac{\pi}{2}+\frac{\pi}{8}$
$\frac{\pi}{2}-\frac{\pi}{8}$
$r\left(\frac{\sqrt{3}}{2},\frac{1}{2}\right)$
$\frac{\pi}{8}$
$\pi-\frac{\pi}{8}$
$-\pi+\frac{\pi}{8}$
$-\frac{\pi}{8}$
$-\frac{\pi}{2}-\frac{\pi}{8}$
$-\frac{\pi}{2}+\frac{\pi}{8}$

Figure 3.12: Neighbor selection based on tangent direction $\mathbf{t}$ (corresponding to the red vector). The green squares indicate the selected neighbors.

Neighbor Saliency

As mentioned before, not all three candidate neighbors $\mathbf{n}_1, \mathbf{n}_2$ and $\mathbf{n}_3$ are added to the graph but only salient ones. In order to accept all of the three following conditions must hold:

1. $\mathbf{n}_i \in \Omega_{\mathcal{I}}$
2. $\mathcal{V}(\mathbf{n}_i) \neq 0$
3. $\gamma = \arccos\Big(\|\langle \mathbf{t}_c, \mathbf{t}_i \rangle\|\Big) \leq \frac{1}{4}\pi$

The first condition is trivial and simply preventing the selection of invalid points by ensuring the no points outside the image domain are chosen. The second condition is based on any vesselness measure $\mathcal{V}$ introduced in Section 3.2.3. In case of Frangi's vesselness measure, $\mathcal{V}(n_i^x, n_i^y)$ is non-zero if and only if the largest eigenvalue of the Hessian at point $\mathbf{n}_i$ is positive which ensures that we are only tracing dark and line-like structures. This can be seen from Equation (3.45) though it is defined for bright lines as opposed to dark lines which explains the negation of the relation. For Koller's vesselness, the feature value is non-zero when both side responses $\mathcal{R}_l$ and $\mathcal{R}_r$ are positive (see also Equations (3.52) and (3.53)). The last condition ensures that the eigenvectors located at $\mathbf{v}_c$ and $\mathbf{n}_i$ are pointing in similar directions and do not deviate by more than 45°. This is motivated by the assumption that the object we want to segment exhibits certain smoothness.

Graph Dilation

The *graph dilation* is the aforementioned fallback routine which is used to deal with cases in which the created graph G consists of two disjunct components. How can this happen? During the selection of salient neighbors, some neighbors are rejected. In particular problematic is the second saliency criterion which requires that $\mathcal{V}(n_i^x, n_i^y) \neq 0$ holds for a candidate neighbor $\mathbf{n}_i$. Due to the noisy nature of interventional images and due to the small size of the desired structures and partial occlusion by background objects such as ribs or the spine, guide-wires and other medical tools sometimes appear as disconnected objects. In cases where the object appearance is corrupted by noise, the corruption are is most often caused by single pixels. This means, that beyond this single corrupted pixel, the object is again visible and leads to the idea to *dilate* the pixel graph. So far, the neighborhood selection considers only direct eight-neighborhood of a pixel $\mathbf{v}_c$. The dilation process relaxes this neighborhood selection by extending the neighbor search range.
The method is performed by reconfiguring the graph growing such that the eighteen neighbor pixels around $\mathbf{v}_c$ are investigated. Each of the leafs of graph G is pushed into an empty stack U and the graph growing is run a third time which results in a new graph G which may contain edges (v_c, n_i) of length up to $2\sqrt{2}$. This method is primarily a heuristic approach giving the system a *second chance* to recover from initial failures. This indicates, that even after the *graph dilation* it is possible to face the situation where either $\mathbf{s}_1 \notin V$ or $\mathbf{s}_2 \notin V$ which prevent the computation of a shortest path between the nodes $\mathbf{s}_1$ and $\mathbf{s}_2$. In such cases, the algorithm failed and the user is required to choose a different set of seed points $\mathbf{s}_1$ and $\mathbf{s}_2$ or a different input image $\mathcal{I}$.

Edge Weights

Assuming, everything went fine and the graph G allows the computation of a shortest path between the nodes $\mathbf{s}_1$ and $\mathbf{s}_2$ it is required to provide edge weights w_i for every edge $e_i = (\mathbf{n}_j, \mathbf{n}_k, w_i)$. Given the tangents $\mathbf{t}_j$ and $\mathbf{t}_k$ of the nodes $\mathbf{n}_j = (n_j^x, n_j^y)$ and $\mathbf{n}_k = (n_k^x, n_k^y)$ and the feature image $\mathcal{V}$, the edge weights w_i are computed as

$$w_i = \left(1 - \sqrt{\mathcal{V}(\mathbf{n}_j)\mathcal{V}(\mathbf{n}_k)}\right) \arccos\left(\|\langle \mathbf{t}_j, \mathbf{t}_k \rangle\|\right) . \tag{3.66}$$

This measure leads to small values when the geometric mean of the vesselness measures at locations $\mathbf{n}_j$ and $\mathbf{n}_k$ is high, i.e. when both pixel locations are likely to belonging to a line-like structure. Furthermore, the measure gets the smaller the smoother the curve is, which is modeled by the angle between the tangents $\mathbf{t}_j$ and $\mathbf{t}_k$.
Using these edge weights w_i as defined above, it is possible to Dijkstra's [20] shortest path algorithm and to compute an ordered set

$$\mathcal{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^M\} \tag{3.67}$$

resembling the discrete points belonging to the desired medical tool.
An overview of the basic algorithm without the second chance or *graph dilation*, respectively is provided in the pseudo-code algorithm 22.

3.4 Fully Automatic Detection

In this section, two methods for the fully automatic detection of ridge like structures are discussed. The first method is the result of the work of Carsten Steger [71, 70] and it is not only detecting and linking edge points but also providing a classification of the pixel type, i.e. whether the pixel is a center curve pixel, whether it is an end-point or belonging to a junction. Furthermore, the method computes sub-pixel locations of ridge centers and local tube widths. The second method that is presented in this section is a modification of Canny's well known edge detection algorithm. The adaption replaces the gradient magnitude by a ridge indicator and adapts the non-maximum suppression in a way that it is performed along ridge normals.

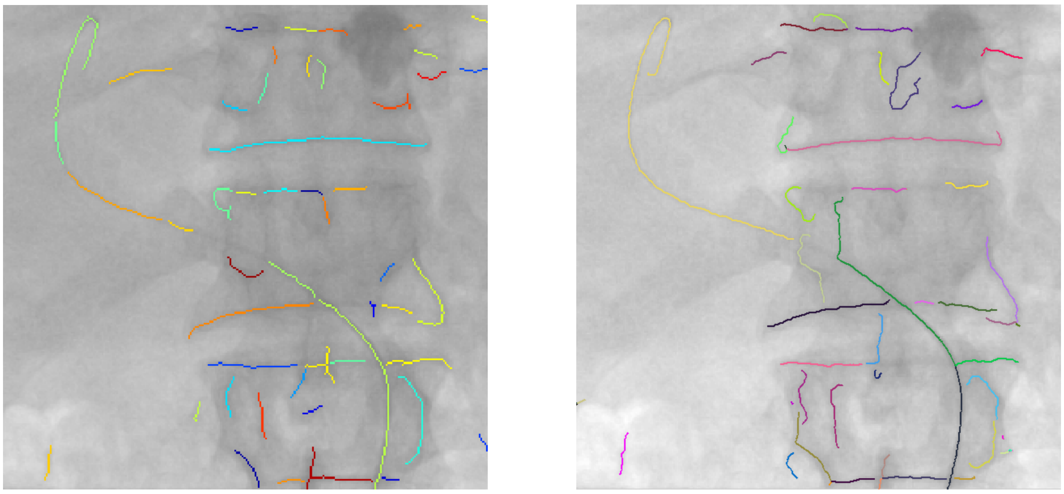
3.4.1 Canny Ridge Detection

The well known edge detection method proposed by Canny [13] is originally based on the computation of the image gradient $\nabla\mathcal{I}$. It uses the gradient magnitude $\|\nabla\mathcal{I}\|$ as an

Algorithm 1: Semi-Automatic Tool Detection

```
input  : Image $\mathcal{I}$, Vesselness $\mathcal{V}$, Tangents $\mathcal{T}$
output: Tool Pixels $\mathcal{P}$

$\{\mathbf{s}_1, \mathbf{s}_2\} \leftarrow$ initializeSeedPoints();
$G \leftarrow (V, E)$;
for $i \leftarrow 1$ to $2$ do
    $U \leftarrow \{\mathbf{s}_i\}$;
    while $U \neq \emptyset$ do
        $\mathbf{v}_c \leftarrow$ pop($U$);
        $\mathbf{t}_c \leftarrow \mathcal{T}(\mathbf{v}_c)$;
        $\mathcal{N} \leftarrow$ computeNeighbors();
        foreach $\mathbf{n}_i \in \mathcal{N}$ do
            if $\mathbf{n}_i \in V$ then continue;
            $U \leftarrow$ push($\mathbf{n}_i, U$);
            $\mathbf{t}_i \leftarrow \mathcal{T}(\mathbf{n}_i)$;
            $\gamma \leftarrow \arccos\Big(\|\langle \mathbf{t}_c, \mathbf{t}_i \rangle\|\Big)$;
            if $\mathbf{n}_i \in \Omega_{\mathcal{I}} \wedge \mathcal{V}(\mathbf{n}_i) = 0 \wedge \gamma \leq \pi/2$ then
                $w_i \leftarrow$ computeEdgeWeight($\mathcal{V}(\mathbf{n}_c), \mathcal{V}(\mathbf{n}_i), \gamma$);
                $V \leftarrow V \cup \{\mathbf{n}_i\}$;
                $E \leftarrow E \cup \{(\mathbf{n}_c, \mathbf{n}_i, w_i)\}$;
            end
        end
    end
end
$\mathcal{P} \leftarrow$ dijkstra($G, \mathbf{s}_1, \mathbf{s}_2$);
```



(a) Canny Ridges (b) Steger Ridges

Figure 3.13: The figure shows two exemplary results of a fully automatic ridge detection method. On the left-hand side, an approach derived from Canny's edge detection algorithm has been used and on the right-hand side is the method of Carsten Steger. Both methods generate multiple responses and require additional post-processing to filter out spurious results.

indicator of edge strength. After the computation of the gradients and gradient magnitude, the method starts by a non-maximum suppression in the direction of the gradient, i.e. orthogonal to edges. Gradients at a pixel location $\mathbf{p}$ are only then retained when neighboring gradient magnitudes in the gradient direction $\pm\nabla\mathcal{I}$ are both smaller than the gradient magnitude located in the current pixel. This method is then followed by a hysteresis threshold based on two threshold values t_{high} and t_{low}, where the high threshold value is used to select salient pixels and the low threshold is used to perform a region growing starting from this pixel and including neighbors until their gradient magnitude is greater or equal to the lower threshold. This method ensures, that the output of the detection algorithm are only a single pixel wide and in the original method, these pixel sets are corresponding to edge locations.

As opposed to Steger's method (see below), Canny's edge detection does not provide any ordering on the pixels nor any additional geometric information besides the fact that pixels are located on edges. Additional methods for linking pixels to line segments are required to generate this additional kind of information.

The adaption to the detection of ridges is straight forward and requires only two modifications of the original algorithm. First, the gradient magnitude needs to be replaced with a measure which indicates high likelihood for pixels belonging to ridges as opposed to edges. Here, again any vesselness measure $\mathcal{V}$ of the previously mentioned (see Section 3.2.3) tube or ridge enhancement methods suffices. After the adaption of the salience measure, the gradient direction need to be adapted. Non-maxima need to be suppressed orthogonal to the object of interest. For edges, this direction is corresponding to the gradient

$$\nabla\mathcal{I} = \begin{bmatrix} G_x * \mathcal{I} \\ G_y * \mathcal{I} \end{bmatrix} \ , \tag{3.68}$$

though for ridges this choice is not correct. Following the findings of Section 2.2, it is clear that again the directional information provided by the Hessian $\mathcal{H}$ of image intensities can be used to determine the normal of the ridges – it is equal to the eigenvector $\mathbf{p}$ corresponding to the largest absolute eigenvalue. From a practical, run-time point of view it is no problem that now second-order derivatives of the image are required since the data is already computed for during the generation of the vesselness $\mathcal{V}$.
Exemplary results of this detection algorithm are shown in Figure 3.13(a). The results indicate that additional post-processing methods are required to filter out spurious responses. But not only does this method create false positives but it is also not providing a segmentation of the whole guide-wire but only parts of it. Even though it may theoretically be possible to improve the results by careful tuning of the choice of the hysteresis parameters as well as the parameters which are required for the generation of the feature image $\mathcal{V}$ the results will most likely not be generally applicable. Consequently, so far the method requires additional user interaction in order to select the correct line-segments or at least for the verification of the segmentation results.

3.4.2 Steger Ridge Detection

In his work on “An Unbiased Detector of Curvilinear Structures” [70], Carten Steger proposes a method which is based on the saliency or vesselness measure introduced by Sato et al. The algorithms consists of two phases, a first one which selects points forming ridges and a second phase in which these points are linked to lines.
As in the “Canny Ridge Detection”, points are detected by combining a saliency measure in combination with a hysteresis thresholding. The saliency measure Steger proposes is identical to the measure proposed by Sato et al. presented in Equation (3.39). Therefore, the algorithm is also bound to the same constraints as Sato’s method and requires the same scale-space normalization and constraints on t. Table 3.1 summarizes these results and they are corresponding in general to the findings of C. Steger. A minor difference is that the choice of σ (see Equation (10) in [70]) which corresponds to $\sqrt{t}$ should be tightened to[2]

$$\sigma > \frac{r}{\sqrt{3}} \ . \tag{3.69}$$

The basic idea of the saliency measure is derived from the commonly known second derivative test. Given a stationary point, here the ridge top, the ridge profile $\mathcal{I}_*(x, r, h)$ is a local maximum (belongs to a bright line) when

$$\frac{\partial\mathcal{I}_*(x, r, h)}{\partial^2 x} < 0 \ . \tag{3.70}$$

---

[2] In the work of Carsten Steger, the tube radius is defined by w as opposed to r which is used in this work.

For dark lines, $\mathcal{I}_*$ must exhibit a local minimum. In 2D, second derivatives are represented by the Hessian matrix $\mathcal{H}$ and the eigenvalue of largest absolute value is corresponding to the second derivative in the curve's normal direction.
Steger's method is not based on integral pixel coordinates but the author performs the actual ridge detection at sub-pixel accuracy. Sub-pixel coordinates of an arbitrary one-dimensional ridge profile $\mathcal{I}_*$ can be found by computing a solution for δx such that

$$\frac{\partial \mathcal{I}_*(x+d)}{\partial d} = 0 \tag{3.71}$$

holds. For a two-dimensional ridge profile, the offset needs to be determined in the direction of the ridge normal $\mathbf{n} = (n_x, n_y)^\top$. As usual, this direction is corresponding to the eigenvector corresponding to the largest absolute eigenvalue and Equation (3.71) becomes

$$\frac{\partial \mathcal{I}_*(x+dn_x, y+dn_y)}{\partial d} = 0 \; . \tag{3.72}$$

The solution to this equation can be found by using a second order Taylor approximation of $\mathcal{I}_*$. The second order Taylor approximation is given by

$$\mathcal{I}_*(x+dn_x, y+dn_y) = \mathcal{I}(x,y) + d\mathbf{n}^\top \nabla \mathcal{I}_*(x,y) + \frac{1}{2} d^2 \mathbf{n}^\top \mathcal{H}_{\mathcal{I}_*}(x,y)\mathbf{n} \; . \tag{3.73}$$

Deriving Equation (3.73) for d, setting it to zero and solving for d results in

$$d = \frac{\mathbf{n} \nabla \mathcal{I}_*(x,y)}{\mathbf{n}^\top \mathcal{H}_{\mathcal{I}_*}(x,y)\mathbf{n}} \tag{3.74}$$

and leads to sub-pixel precise ridge centers as $\mathbf{p} = (x+dn_x, y+dn_y)$. Solutions to this equation are only accepted it $d \in [-\frac{1}{2}, \frac{1}{2}]$ which leads to a second criterion that may cause the rejection of points as ridges.
To summarize, the algorithms processes all pixels in the image and accepts only points where the directional derivative in the normal direction has the correct sign and it accepts only those points, where the optimum is located within the unit circle centered in the current point, i.e. $|d| \leq \frac{1}{2}$. All other points are assumed to belong to the background and immediately discarded by the method.
The next step is the hysteresis thresholding. Identically to the previously introduced method, seed points are selected as pixels having a saliency measure being higher than some strong threshold. Neighbors are then added to the seed points by looking at potential candidates and verifying that these candidates exhibit a salience measure being higher than some weak threshold. In fact, the method used by Carsten Steger is slightly more difficult because it does not only perform a simple hysteresis thresholding but in addition the algorithm is linking neighboring pixels and thus introducing a spatial relationship rather than a mere labeling of fore- and background pixels.
Exemplary results of Steger's detection algorithm are shown in Figure 3.13(b). Similarly to the "Canny Ridge Detection" the results indicate that additional post-processing methods are required. A difference is that this method does provide additional

information as geometric properties like the ridge width and ridge orientation as well as semantic properties where ridge pixels are classified as center pixels, end pixels or pixels belonging to junctions. It is not really an advantage because when looking at the algorithm from a modular point of view, the same information can be added to the "Canny Ridge Detection". This is not only true for the geometric and semantic properties but also for the sub-pixel positioning of individual pixels. For future research it might be beneficial to replace the salience measure used by Carsten Steger by the vesselness measure of Koller et al. because of its advantageous properties discussed at the end of Section 3.2.3.

3.5 Discussion

In this chapter, different methods for the enhancement and the semi-automatic as well as fully automatic detection of medical tools were discussed. It turns out, that these days no method for the fully automatic detection of medical tools exists which performs at a level of reliability being required in clinical practice. Actually, it is most likely a valid assumption that simplified methods which operate only on single frames without the consideration of temporal changes will not even in the future be capable of performing a segmentation at the desired level of reliability. From a practical point of view this statement is strongly supported by the fact that even human observers have sometimes difficulties in detecting small structures in fluoroscopic image sequences when these do not exhibit any motion.

Two ideas to tackle the problem in the future are presented in this section. In some sense both potential approaches considered here solve a classification problem. Recently, some authors [54, 33] proposed learning based algorithms for guide-wire detection and in fact, these methods are already solving a fuzzy classification problem but one could claim that even hysteresis thresholding is a crude classification method. More interestingly, both methods mentioned before do not yet utilize the time domain which is in this work considered crucial for the development of fully automatic methods. Similarly to the image based methods presented in this chapter, the aforementioned learning based methods mainly focus on appearance and intensity models.

The incorporation of the time domain is considered important for future work and what most obviously changes in time is the spatial location of medical tools. For these objects their location is determined by the patient motion, dynamic changes of the projection geometry and tool interactions performed by the physician. One can observe that background objects typically follow a different motion pattern than medical tools. Responses caused by image noise are unlikely to exhibit regular motion but can be assumed to follow random motion patterns. Other responses as e.g. the spinal discs as it is observed in abdominal interventions are hardly moving at all but remaining stationary. One possibility for the detection of medical tools within noisy background is spectral clustering [16, 83]. The method is based on the design of an affinity function which glues together objects belonging most likely to the same cluster. Given curves C_i^t $(i = 1, \ldots, n)$ and a method for tracking curves over time, it is possible to collect spatial information for each curve C_i^t resulting in a set of curves

$\mathcal{C}_i = \{C_i^t, C_i^{t+1}, \ldots, C_i^{t+m}\}$. A symmetric and positive function mapping pairs of objects, here sets of curves, to $\mathbb{R}$ as

$$f : (\mathcal{C}_i, \mathcal{C}_j) \mapsto \mathbb{R} \ , \tag{3.75}$$

is called an affinity function and large values correspond to a high likelihood that the curve sets $\mathcal{C}_i$ and $\mathcal{C}_j$ belong to the same group, e.g. noisy background. Besides the curve locations, the function f can be designed based on additional information such as the image data $\mathcal{I}^t$ given at each time instant t. From this function an affinity matrix A is created where

$$A(i, j) = f(\mathcal{C}_i, \mathcal{C}_j) \ . \tag{3.76}$$

Without going into too much detail on spectral clustering, it is possible to create the so called graph Laplacian L and a degree matrix D from the affinity matrix A. And the first d eigenvectors of the generalized eigenvalue problem $Lu = \lambda Du$ are then used to map the objects one intends to cluster into a d-dimensional space. The first d eigenvectors can be assembled into a matrix

$$U = [u_1, \ldots, u_k] \in \mathbb{R}^{m \times d} \ . \tag{3.77}$$

Here, each of the m data points $\mathcal{C}_i$ is corresponding to a d-dimensional point $\mathcal{S}_i$ represented by the i-th row of the matrix U (i.e. U(i,:) in Matlab notation). In the optimal case the function f is designed such that the points $\mathcal{S}_i$ form clouds and thus enable a linear separation of individual clusters. The actual separation step is carried out by performing a k-Means clustering and the outcome of the grouping process are $\mathcal{K}_i$ with $i \in \{1, ..., k\}$.

Before discussing problems associated to such a clustering approach, a simple affinity function will be presented. Considering that curve motions can be approximated by two-dimensional rotations and translations, it is possible to generate for each curve set $\mathcal{C}_i$ a set of angles $\mathcal{R}_i = \{\alpha_i^t, \alpha_i^{t+1}, \ldots, \alpha_i^{t+m}\}$ and translations $\mathcal{T}_i = \{\mathbf{t}_i^t, \mathbf{t}_i^{t+1}, \ldots, \mathbf{t}_i^{t+m}\}$. Assuming further, that corresponding curves $\mathcal{C}_i$ exhibit similar motion pattern over time, a possible affinity function could be written as

$$f\Big((\mathcal{R}, \mathcal{T})_i, (\mathcal{R}, \mathcal{T})_j\Big) = \exp\left(-\frac{\sum_{t=1}^{m} \left|\alpha_i^t - \alpha_j^t\right|}{\sigma_\alpha^2}\right) \exp\left(-\frac{\sum_{t=1}^{m} \left\|\mathbf{t}_i^t - \mathbf{t}_j^t\right\|}{\sigma_{\mathbf{t}}^2}\right) \ . \tag{3.78}$$

This function becomes 1, when the two curves have identical motions and results in smaller values for different motions.

Initially, this clustering approach seems to be feasible but it turns out that naïvely pursuing this recipe does not lead to convincing results – and this independently of the choice of the affinity function. The problem is that the cluster $\mathcal{K}_{\text{tool}}$ representing the object of interest is most often too small to justify the creation of an individual class. Assuming the detection algorithm presented in the previous section right away detect the medical tool, the cluster's cardinality can become as small as 1. A possible way to deal with this problem is to ensure that the detection algorithm splits the object of

interest into a reasonable amount of sub-segments. This can be achieved by separating the image $\mathcal{I}$ in the initial detection step into small sub-patches and by applying the detection method on each of these sub-patches separately.
Since the method proposed here is just a suggestion in which direction future research might continue, no evaluation has been performed. The recipe presented above is a simple way to incorporate information provided by the time domain. State of the art methods aiming at the fully automatic detection of medical tools do not yet perform at an acceptable precision and stability suitable for clinical practice which leads to the conclusion that additional information is essential.

CHAPTER

FOUR

TOOL TRACKING

This chapter presents a novel method for the tracking of curves modeled by B-splines. In principle the tracking can be applied to open as well as closed curves thus enabling the application of the method to segmentation problems where the B-spline is describing a contour as opposed to an open line-like structure. However, in this work the focus lies on medical tools and thus open curves.

The tracking algorithm is based on discrete optimization methods and does not rely on gradient based formulations. This approach enables the utilization of complex models for which derivatives can most often only be approximated by numerical methods. Such numerical differentiation methods are sometimes difficult to handle because they are sensitive to the discretization schemes, e.g. the step size at which derivatives are approximated. This problem gets even more difficult for functions having a high dimensional parameter space and parameters which live in different sub-spaces as e.g. angles and translations which require different step sizes for a proper differentiation. One example where discrete optimization is beneficial and where derivatives can hardly be estimated are energy terms of cost functions which are learned with offline data.

Local minima, oftentimes caused by noise represent another problem which can be potentially overcome by discrete optimization strategies. Because discrete optimization methods do not rely on gradient information, they lack strong forces keeping them from leaving local minima and have the potential to discard such configurations and converge towards a *global optimum*. Global optima in discrete optimization are slightly different from those in continuous optimization because *global* refers to global within the discrete set of solutions. Especially for open curves, there exist discrete optimization strategies that are guaranteed to find the global optimum within the discrete set of solutions, i.e. that do not rely on any approximations.

Nonetheless, also discrete optimization techniques are not without disadvantages. Gradient-based method intrinsically model interactions of different parameters. For discrete optimization techniques which are based on graphs in which nodes represent random variables and edges dependencies of these variables (or parameters) this would be only guaranteed in cases where a node were connected to all nodes on which it is depending. Such a procedure is in practical applications not feasible because it results in an explosion of the computational complexity and therefore these interactions require approximative solutions. The efficient design of approximation schemes is an important topic in this work and will be discussed in this chapter.

The first section of this chapter focuses on the modeling of curves and the modeling of tool motions. Afterwards, several possibilities for the modeling of objective functions are discussed and defined as continuous functions which could potentially be used in any optimization framework. It is common that such cost function consist of at least two components. A first component which drives the optimization and which is usually referred to as *data term* or *internal energy*. These terms are usually based on some saliency measures sometimes also called feature images or vesselness images in case of line-like structures. The second term, the regularization is required to deal with the ill-posedness of the optimization problem. Regularization terms are used to convexify objective functions and are often based on geometric observations. Several methods for the regularization are reviewed in this chapter and a group of regularization schemes is presented.

After the introduction of the energy terms, MRFs are introduced and it is described how to use these optimization techniques for tracking. Section 4.3 goes into detail on how to discretize the continuous functions presented in Section 4.2. Additionally, the section will provide details on approximation errors related to an incomplete modeling of parameter interdependency. Various approximation schemes are introduced and evaluated.

The last part, Section 4.4 presents a variety of synthetic experiments which are in particular used to support the validity of the approximation schemes. In addition, the experimental section contains results on hand-labeled clinical data. The results show the applicability of the method and its potential to correctly estimate the apparent tool motion. Besides synthetic experiments and those on clinical data, the last section contains a quantitative comparison of the proposed tracking algorithm with a recently introduced learning based method.

4.1 Modeling of Guide-Wires and Needles

In this work, curves are modeled by non-uniform B-splines. The general form of a p-th degree B-spline is

$$C(u) = \sum_{i=0}^{m-1} N_{i,p}(u)\mathbf{P}_i \ , \tag{4.1}$$

where u is a parameter defining the location on the curve. The curves which are used in this work are not arc-length parameterized. In Equation (4.1), $u = 0$ and $u = 1$ are corresponding to the curve's end-points. The functions $N_i(u)$ are different B-spline basis functions, each associated to a control point $\mathbf{P}_i$. In this work, the objects of interest are observed in two-dimensional images and thus the B-spline's control points are two-dimensional as well, i.e. $\mathbf{P}_i \in \mathbb{R}^2$. The i-th B-spline basis function of degree $N_{i,p}(u)$

is defined by

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \ , \tag{4.3}$$

which is usually referred to as the Cox-de Boor recursion formula [19]. This function is defined by so called knots $u_i \in \mathbb{R}$ and the augmented knot-vector

$$U = \{u_0, u_1, \ldots, u_{m+p}\} \ . \tag{4.4}$$

Augmented is here used because the first and last $p + 1$ knots are equal to 0 and 1, respectively. For an augmented knot vector with $n = m + p + 1$ knots, the knots u_i with $i = p + 1, p + 2, \ldots, n - p - 1$ are referred to as internal knots.
The utilization of B-splines is most common for the tracking of medical tools. In fact, all other works presented in Section 2.3 are using B-splines too. In general researchers tend to use uniform B-splines because they can be efficiently implemented since in this special case, the basis functions are just shifted copies of each other. The uniformity of a B-spline curve is reflected in the distribution of the knots on the interval $[0; 1]$ since in the uniform case, the knots are uniformly distributed on that interval. From a practical point of view, non-uniform distributions enable better approximation of curves which exhibit strong local variations of curvature.

Fitting of Smoothing Splines: For the task of tracking, one is initially given a list of points

$$\mathcal{P} = \{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^M\} \tag{4.5}$$

by the detection module for which a spline function $C(u)$ with a maximum of m control points is sought after which obeys certain optimality criteria. The spline $C(u)$ should approximate the data by finding a compromise between a *close fit* and a *smooth solution* in the sense that discontinuities in the curve's p-th derivative should be as small as possible. Dierckx [17, 19] proposes to formulate this problem in terms of the following minimization:

$$\text{Minimize} \quad \eta := \sum_{i=p+1}^{n-p-1} \Big(C^{(p)}(u)(u_i+) - C(u)^{(p)}(u_i-) \Big)^2 \tag{4.6}$$

$$\text{s.t.} \quad \delta := \sum_{j=1}^{M} w_j \Big(\mathbf{p}_j - C(u_j) \Big)^2 \leq S \tag{4.7}$$

In this equation $C^{(p)}$ is the p-th derivative with respect to u. An implementation of this method called `parcur` is publicly available in the library FITPACK [18]. It does not only compute the optimal control points $\mathbf{P}_i$ but also allows for the computation of a non-uniformly distributed knot vector U. In this work, the weights for individual data points are set to $w_j = 1$.

Deforming Spline Curves: Once a spline fit for the point set $\mathcal{P}$ and thus the medical tool is given one needs a model of curve deformations. In fact, these deformation can be linear and non-linear. Since for B-spline basis functions holds

$$\sum_{i=0}^{m-1} N_{i,p}(u) = 1 \tag{4.8}$$

and because of the B-spline definition from Equation (4.1), a transformation by a matrix $\mathbf{R}$ and a translation $\mathbf{t}$ can be written as

$$\begin{aligned}
\widetilde{C}(u) &= \mathbf{R}C(u) + \mathbf{t} &\quad (4.9)\\
&= \mathbf{R}\sum_{i=0}^{m-1} N_{i,p}(u)\mathbf{P}_i + \sum_{i=0}^{m-1} N_{i,p}(u)\mathbf{t} &\quad (4.10)\\
&= \sum_{i=0}^{m-1} N_{i,p}(u)\Big(\mathbf{R}\mathbf{P}_i + \mathbf{t}\Big)\ . &\quad (4.11)
\end{aligned}$$

This shows that applying a linear transformation and even a translation to the control points of a spline is identical to transforming the whole curve. Non-linear transformations are modeled slightly differently by displacing single control points. Once the basis functions $N_{i,p}(u)$ are fixed, the curve shape is solely defined by the positions of the control points $\mathbf{P}_i$, and arbitrary deformations can be modeled by displacing individual control points as

$$\widetilde{C}(u) = \sum_{i=0}^{m-1} N_{i,p}(u)(\mathbf{P}_i + \mathbf{d}_i)\ , \tag{4.12}$$

and with this representation it is possible to model arbitrary deformations.
This equation indicates that each curve point is depending on all variables $\mathbf{P}_i$ and $\mathbf{d}_i$ which in an optimization framework would imply that each variable is depending on every other one and in case of discrete optimization this would lead to a combinatorial explosion. Thus, a question which is of particular interest in this work is how many control points define the spatial location of a point $C(u)$. From Equation (4.1) one can see that each $C(u)$ is affected by every control point for which the B-spline basis function is non-zero, i.e. $N_{i,p}(u) \neq 0$. It can be shown that for a curve of degree p this is true for exactly $p + 1$ basis functions and consequently a single point $C(u)$ is depending on $p + 1$ or for cubic splines, four control points. This observation is important because it implies that for each curve point $C(u)$, $p + 1$ variables subject to the upcoming optimization are depending on each other as opposed to all m. As mentioned in the introduction of this chapter, such inter-dependency of variables must either be modeled correctly in the optimization process or, in cases where this is computationally impractical, the error introduced by an incomplete model must be carefully handled.
In the optimization scheme which will be presented in the next sections, only the inter-dependency of two variables will be modeled. Therefore, a deformation model of a curve where only two variables are displaced is required. Assuming, a curve $C(u)$ is

displaced in the j-th and k-th control points, the deformed curve can be written as

$$\begin{aligned}\widetilde{C}(u, \mathbf{d}_j, \mathbf{d}_k) &= N_{j,p}(u)(\mathbf{P}_j + \mathbf{d}_j) + N_{k,p}(u)(\mathbf{P}_k + \mathbf{d}_k) + \sum_{\substack{i=0 \\ i \neq j, i \neq k}}^{m-1} N_{i,p}(u)\mathbf{P}_i \\ &= C(u) + N_{j,p}(u)\mathbf{d}_j + N_{k,p}(u)\mathbf{d}_k \ . \end{aligned} \tag{4.13}$$

In the same way, it is possible to derive the displacement in three control points i, j and k as

$$\widetilde{C}(u, \mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_k) = C(u) + N_{i,p}(u)\mathbf{d}_i + N_{j,p}(u)\mathbf{d}_j + N_{k,p}(u)\mathbf{d}_k \ . \tag{4.14}$$

The deformation model which is used in this work is slightly different than the one presented in Equation (4.13). A problem caused by Equation (4.13) is the assumption that only two control points are moving while no other control point is changing its position. This assumption leads to intermediate curves $\widetilde{C}(u, \mathbf{d}_j, \mathbf{d}_k)$ which are only roughly approximating the reality. The fact, that in reality neighboring curve points exhibit a similar motion lead to the development of a new deformation model. In the new model, the smoothness of the motion is incorporated by applying the displacements of the control points $\mathbf{P}_j$ and $\mathbf{P}_k$ also to their neighbors. The resulting curve being displaced by $\mathbf{d}_j$ and $\mathbf{d}_k$ is represented by

$$\begin{aligned}\widetilde{C}(u, \mathbf{d}_j, \mathbf{d}_k) &= \sum_{i=0}^{j} N_{i,p}(u)\Big(\mathbf{P}_i + \mathbf{d}_j\Big) + \sum_{i=k}^{m-1} N_{i,p}(u)\Big(\mathbf{P}_i + \mathbf{d}_k\Big) \\ &= C(u) + \sum_{i=0}^{j} N_{i,p}(u)\mathbf{d}_j + \sum_{i=k}^{m-1} N_{i,p}(u)\mathbf{d}_k \ . \end{aligned} \tag{4.15}$$

To summarize, the most important point of this section is that medical tools are modeled with non-uniform B-splines. The non-uniformity improves the curve evaluation because it allows a linear sampling of the domain $[0; 1]$ while preventing linearly spaced points on the curve – i.e. even though the domain is linearly sampled, curve points may cluster in areas of high curvature. A second important result is the way in which tools are deformed which is described by Equations (4.14) and in particular (4.15). Regarding the remaining details as e.g. the efficient evaluation of B-splines the reader is kindly asked to refer to secondary literature [61, 19].

4.2 Modeling of Objective Functions

In the following, the general continuous formulation of the curve tracking problem is reviewed. Given an initial curve C, one wants to estimate transformation parameters leading to a new curve C^* which is conforming with observed image data $\mathcal{I}$. Mathematically, this means one seeks to maximize the posterior probability

$$C^* = \arg\max_{C \in \mathbb{F}} P(C \,|\, \mathcal{I}) \ , \tag{4.16}$$

and in this equation C^* is referred to as the MAP estimate. Based on Bayes' theorem, which states that

$$P(C \,|\, \mathcal{I}) = \frac{P(\mathcal{I} \,|\, C) P(C)}{P(\mathcal{I})} \;, \tag{4.17}$$

Equation (4.16) can be rewritten. Assuming a constant $P(\mathcal{I})$ is generally considered safe and since a constant factor does not affect the result of the maximization, the problem can be rewritten as

$$C^* = \arg\max_{C \in \mathbb{F}} P(\mathcal{I} \,|\, C) P(C) \;. \tag{4.18}$$

In this equation the term $P(\mathcal{I} \,|\, C)$ is referred to as the *likelihood* distribution and it evaluates how well a given curve C fits the observed image $\mathcal{I}$. The second term $P(C)$ is called the *prior* distribution and models potentially available *a priori* knowledge. In the case of curve tracking this term can for instance be used to encode constraints on the geometric appearance of a curve or prior knowledge regarding the type of transformation one expects the curve to undergo. If no *a priori* knowledge is available, $P(C)$ is assumed to be a uniform distribution where each curve configuration has the same prior probability.
Assuming furthermore, the probability distributions of Equation (4.18) follow a Gibbs distribution with underlying energy functions, one can rewrite the MAP estimation as an energy minimization of the form

$$C^* = \arg\min_{C \in \mathbb{F}} E(\mathcal{I} \,|\, C) + E(C) \;. \tag{4.19}$$

In this case, one can interpret the *likelihood* energy $E(\mathcal{I} \,|\, C)$, often referred to as cost function or data term, as measuring the quality of a certain model configuration. Likewise the *prior* energy $E(C)$ can be interpreted as a regularization (or smoothness) term on the parameter space which allows to impose constraints on the solution.

External Energy When formulating the tracking problem based on B-spline curves which are defined on the interval $[0; 1]$, the likelihood term (4.20) or *external energy*, driving the curve to its actual position can be formulated as

$$E_{\text{ext}}(\mathcal{I} \,|\, C) = \int_0^1 \psi\Big(\mathcal{I}\left(C(u)\right)\Big) \,\mathrm{d}u \;. \tag{4.20}$$

Given that the image $\mathcal{I}$ is equal to one of the feature images $\mathcal{V}$ presented in Section 3.2, i.e. $\mathcal{I} = \mathcal{V}$, the function ψ is used to make the feature image's values suitable for minimization. For a feature image with intensity values normalized to the interval $[0; 1]$, this function can be defined as

$$\psi(\mathcal{I}) = 1 - \mathcal{I} \;. \tag{4.21}$$

When using such a function definition, the tracking problem is said to be modeled as an *image based* problem as compared to alternative, e.g. learning based models. In general, image based methods differ only in the type of image $\mathcal{I}$ being used in Equation (4.20). To be more precise, the methods differ in the sort of pre-processing applied to the raw image data. For the tracking of curvilinear objects, the most important image features are typically chosen as one of the earlier introduced ridgeness or vesselness measures [66, 22, 39].

Several alternatives are possible too. In addition to Frangi's vesselness filter [22], Baert et al. evaluated vesselness images on preprocessed input images in [3, 4]. Two types of images were evaluated. One with input images being pre-processed by some sort of background removal and a second method where the input image was preprocessed with coherence enhancing diffusion (CED) as proposed by Weickert [80]. According to the experimental results of Baert et al. [3], Heassian feature images computed on unprocessed and pre-processed images lead both to a spline localization being similar to the inter-observer variability. These results lead to the decision to ignore minimal potential gains of these methods while preventing their inherent run-time penalties. For the same reason, other diffusion based filters such as the works of Canero et al. [12] and Manniesing et al. [51] have not been adopted in this work.

Internal Energy The prior, which is also referred to as *internal energy*, constrains the motion of the curve. Regularization is well known and many regularization functions are universally applicable for a multitude of problems. Constraints on B-splines which penalize geometric properties such as the curvature, length, or relative changes of these values, are generically applicable to curve evolution algorithms.

Another option for regularization is regarding the curve motions. In Equation (4.12), the curve motion is parameterized by displacements $\mathbf{d}_i$ for each control point. Similarly to Equations (4.14) and (4.15), the deformation can be decomposed into the original curve C plus some deformation field D as

$$\widetilde{C}(u) = C(u) + D(u) \tag{4.22}$$

$$= \sum_{i=0}^{m-1} N_{i,p}(u)\mathbf{P}_i + \sum_{i=0}^{m-1} N_{i,p}(u)\mathbf{d}_i \ . \tag{4.23}$$

These two curves have identical knots and differ only in their control points. A common approach is to ensure smoothness of the displacement field or to penalize deviations of neighboring displacements. In fact, it is a common assumption that both approaches lead to the same results though this is not generally true. First, the displacement field one observes in this application is intrinsically smooth because it is a spline of degree p and thus C^{p-1} continuous. Minimizing the distances of neighboring displacements leads to a different effect causing the transformation to tend towards a pure translation. In the latter case, the spline D exhibits multiple identical control points $\mathbf{d}_i$ which leads to a degree reduction of the spline curve and thus to D being a straight line and finally an infinitesimal point which is the same effect one achieves by minimizing the length of $D(u)$.

Because of the natural smoothness of the displacement field, regularization approaches trying to explicitly enforce such a smoothness are considered to be needless for the task of tracking and the focus will be laid on constraints of the curve's geometry and its changes over time.

In this work, relative changes to initially observed data are considered in order to formulate the regularization. This is advantageous because it does not rely on fixed values which may be subject to changes caused by a different acquisition protocol or because of varying patient anatomy.

Investigating the curve properties being acquired during the semi-automatic detection phase results in reference data v_{init} which can be used to realistically constrain curve deformations. Deviations of observed data values v from these values can then be penalized by the squared relative error

$$\rho(v, v_{\text{init}}) = \left(1 - \frac{v}{v_{\text{init}}}\right)^2 . \tag{4.24}$$

During the development of the tracking algorithm, two possible ways of regularizing the curve geometry were investigated. The first measure is based on length changes compared to a reference curve C_{init} observed during the detection phase and the second measure is penalizing deviations of the curvature between the currently observed curve C and the reference curve.

In order to evaluate length changes, a mathematical formulation of a curve's length is required and from differential geometry of curves it is known that for any given curve C, its length can be computed as

$$l = \int_0^1 \|C'(u)\| \, \mathrm{d}u , \tag{4.25}$$

where C' is the first derivative of the curve with respect to u. This follows, that length changes can be penalized by

$$E_{\text{int}}^{\text{length}} = \rho\left(\int_0^1 \|C'_{\text{init}}(u)\| \, \mathrm{d}u, \int_0^1 \|C'(u)\| \, \mathrm{d}u\right) . \tag{4.26}$$

For reasons that will be justified later, it is desired to formulate energies as integrals of some error function and not like in Equation (4.26) as error functions taking integrals as their parameters. Therefore, Equation (4.26) is reformulated and leads to the modified penalty term

$$E_{\text{int}}^{\text{length}} = \int_0^1 \rho\Big(\|C'(u)\| , \|C'_{\text{init}}(u)\|\Big) \, \mathrm{d}u , \tag{4.27}$$

with ρ being defined as in Equation (4.24). This modification does not lead to the exact same results but it is an acceptable approximation. In fact, it is even beneficial for the tracking because it helps to prevent a drift of the tracked curve along the observed one. The drift is oftentimes prevented because the measure forces tangent norms to be *locally* equal.

The second penalty term measures differences of the curvature between the initial curve C_{init} and the currently observed one C. For a single point on the curve, the curvature is computed as

$$\kappa(C,u) = \frac{\Big|C'_x(u)C''_y(u) - C'_y(u)C''_x(u)\Big|}{\|C'(u)\|^3} \; . \tag{4.28}$$

Here, the indices indicate which parameter is used (x for coordinates of the x-axis and vice versa y) and the primes indicate first- and second-derivatives of the curve C with respect to u. Based on this equation, the penalty term can be written as

$$E_{\text{int}}^{\text{curvature}} = \int_0^1 \rho\Big(\kappa(C(u)), \kappa(C_{\text{init}}(u))\Big)\,\mathrm{d}u \; , \tag{4.29}$$

This approach is not following the common methodology of minimizing the curvature to enforce smoothness. In case of B-splines, the inherent smoothness is often sufficient and terms such as the absolute value of the curvature can be discarded [15]. We are more interested in modeling the fact that the curve as seen at time instant t is similar to the curve observed at a later point in time $t+n$ which is based on the assumption that the vessel-anatomy is not changing too much during the intervention.

Experimental validation has revealed that the most important regularization component during the tracking is the length preserving term from Equation (4.27). Using the curvature preservation alone is not sufficient because the measure is not scale variant and allows the curve to shrink or grow. Consequently, the only alternative to using length preservation only, is to combine both measures. Such an approach would lead to the following energy function

$$E_{\text{total}} = (1-\lambda)\,E_{\text{ext}} + \lambda\,\Big((1-\gamma)\,E_{\text{int}}^{\text{length}} + \gamma\,E_{\text{int}}^{\text{curvature}}\Big) \tag{4.30}$$

where λ and γ act as weighting factors from $[0;1]$ controlling the influence of external versus internal energy and length versus curvature regularization, respectively.

Exhaustive experiments have shown that introducing the curvature preservation does not lead to greatly improved tracking results. The main problem which arises when using Equation (4.30) is the proper choice of the regularization weights λ and γ. To be more precise, even though it is possible to find specific parameters which work on a single sequence, it is not necessarily true that the same parameters could be applied to another sequence. These experiences lead to the choice of the follwoing energy formulation

$$E_{\text{total}} = (1-\lambda)\,E_{\text{ext}} + \lambda\,E_{\text{int}}^{\text{length}} \; . \tag{4.31}$$

In continuous optimization, minimizing the above energies is commonly done in a gradient descent approach. The initial contour is updated iteratively by computing the derivative of the energy function with respect to the model parameters. The algorithm stops if no further improvement on the energy can be achieved, meaning the method converges to a local minimum. Even if sometimes convenient to use, such an approach

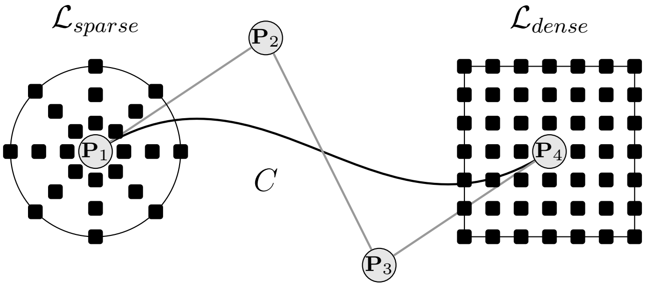


Figure 4.1: MRF Model of an open cubic B-spline curve C with control points $\mathbf{P}_i$. A sparse and a dense version of the discrete set of labels $\mathcal{L}$ capturing the deformation space are illustrated (each black square corresponds to a displacement plus the zero-displacement at the control point position).

has two major drawbacks. First of all, the algorithm requires the derivation of the energy term, which is oftentimes complex to calculate analytically and has to be done specifically for every function. Second, a convergence to a good solution relies on the fact that the initial contour is sufficiently close. If the gradient descent starts far away from the structure to be tracked, chances are high for obtaining bad solutions. This may easily happen in sequences with larger motion. Multi-resolution approaches (e.g. Gaussian scale space) might help in certain scenarios but do not overcome this general limitation. Discrete optimization improves these issues and a strong indication is that in this work, as opposed to previous works, no stratified approach is used (see also Table 2.1).

4.3 Markov Random Fields for Tracking

So far, this chapter provides a formulation of continuous energies which could potentially be minimized by using gradient based optimization. As mentioned earlier, this approach is difficult and can lead to problems. In order to overcome the limitations of continuous optimization or at least in order to be able to diminish their influence, it is proposed in this work to render the tracking problem in a discrete framework.

Discrete problems can be represented by so called probabilistic graphical models (PGMs). The entities required to build a PGM similar to the one presented in Figure 4.1 are nodes $v \in \mathcal{V}$ and edges $e \in \mathcal{E}$ which define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes of a PGM represent random variables of the problem being modeled which are linked by edges representing the variable interdependency. To be more general, $\mathcal{G}$ can even be a hyper-graph and in that case the edges $e \in \mathcal{E}$ are referred to as hyper-edges and may contain more than just two nodes.

Markov random fields (MRFs) are a special form of a PGM and belong to the class of so called undirected graphical models (UGMs). Over the last years, MRFs have become increasingly popular and the reasons for this raising popularity are two fold. First, MRFs are well suited for typical vision problems because the MRF graph structure is directly corresponding to the image structure which makes the modeling process intuitive. Second, new methods for the efficient handling and solution of MRFs emerged recently which allow to utilize these methods in real-world applications with demanding run-time requirements.

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of discrete variables or nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting neighboring pairs of variables in order to represent their dependencies. Additionally, we introduce a discrete set of labels $\mathcal{L}$ capturing the search space of the model parameters. Each label $x_i \in \mathcal{L}$ is associated with a two-dimensional displacement vector $\mathbf{d}^{x_i}$ from the deformation space $\mathcal{D} \subset \mathbb{R}^2$. Two possibilities for the discretization of the deformation space are illustrated in Figure 4.1. If we associate each control point $\mathbf{P}_i$ of our B-spline model with one node of our graph, the task is to assign an optimal label to each of the nodes or a displacement to each of the control points, respectively. A common approach for modeling the labeling task in terms of energy minimization is the usage of first-order MRFs [45]

$$E_{\text{mrf}} = \sum_{i \in \mathcal{V}} \theta^i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta^{ij}(x_i, x_j) \tag{4.32}$$

where θ^i are the *unary potentials* and θ^{ij} are the *pairwise potentials*.

In most applications, the unary terms play the role of the likelihood energy. Independently from all other nodes, the cost for an assignment of a certain label x_i is evaluated. Then, the pairwise interaction terms play the role of regularization between neighboring nodes. However, the assumption that the likelihood of a labeling can be computed from a sum of independent unary terms is actually not valid in our scenario. Considering B-splines with higher-order basis functions, the effect of a single control point onto the deformation of the curve cannot be modeled independently from its neighbors because the basis functions overlap. Therefore, a novel MRF model is proposed for the case of curve tracking using B-splines. First, the basis functions will be used as weighting coefficients within the energy terms. This leads to a higher influence on the energy for curve points close to a certain control point as compared to the influence of points being farther away. A similar approach is used by Glocker et al. in [24] where the authors are using MRFs for non-rigid image registration based on cubic B-splines. Such a weighting allows a suitable approximation of the energy terms with respect to the control points.

In conventional modeling, external energies are represented by the unary terms of Equation (4.32). This procedure imposes difficulties because it is based on the assumption that the energy is locally affected by a single node or control point which were only true when also the curve location were affected by a single control point. It has already been explained in Section 4.1 that this is not true for B-splines of degree p which are depending on $p + 1$ control points.

In order to improve this approximation, it is proposed to reformulate the external energy from the continuous domain as pairwise interaction terms and not as it is commonly done as unary terms. Modeling the external energy as pairwise terms has big advantages. The non-vanishing interval of basis functions along the curve domain for control point tuples is bigger than the corresponding interval of a single control point. Compared to unary potentials, the energy computation for the simultaneous movement

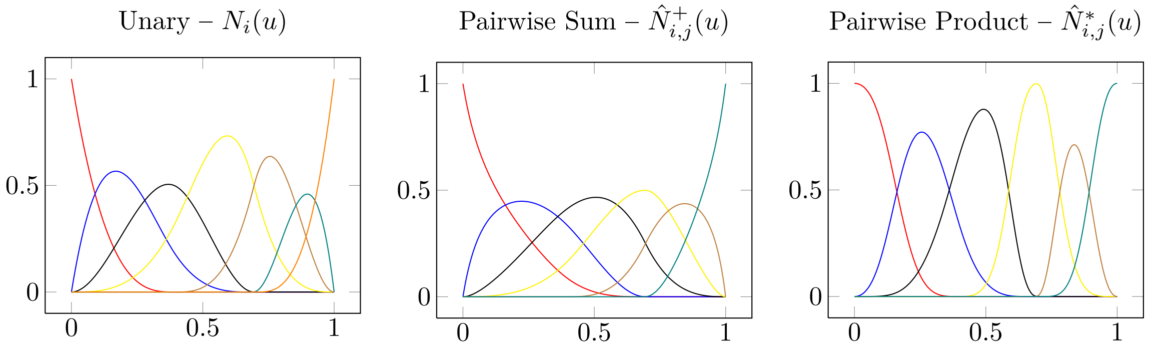


Figure 4.2: Influence functions originating from an open cubic B-spline with seven control points.

of a pair of control points yields a more accurate approximation of the continuous energy. In the proposed approach both energies, the external and internal are defined as pairwise interaction terms as

$$E_{\mathrm{mrf}} = \sum_{(i,j)\in\mathcal{E}} \Big((1-\lambda)\, \theta^{ij}_{\mathrm{ext}}(x_i, x_j) + \lambda\, \theta^{ij}_{\mathrm{int}}(x_i, x_j) \Big) \tag{4.33}$$

where the discrete version of Equation (4.20) is build up by the pairwise terms

$$\theta^{ij}_{\mathrm{ext}}(x_i, x_j) = \int_0^1 \hat{N}_{ij}(u)\, \psi(\mathcal{I}\,(C_{ij}(x_i, x_j, u)))\, \mathrm{d}u \tag{4.34}$$

and the internal energy from Equation (4.27) by all pairwise terms

$$\theta^{ij}_{\mathrm{int}}(x_i, x_j) = \int_0^1 \hat{N}_{ij}(u)\, \rho\Big(\left\| C'_{ij}(x_i, x_j, u) \right\|, \left\| C'_{\mathrm{init}}(u) \right\| \Big)\, \mathrm{d}u\ . \tag{4.35}$$

Here, the weighting coefficient $\hat{N}_{ij}(u)$ evaluates the influence of a curve point of the site u to the energy of a control point pair (i, j) and the curve function $C_{ij}(x_i, x_j, u)$ describes the *potential deformation* of a curve when two control points i and j are displaced simultaneously by $\mathbf{d}^{x_i}$ and $\mathbf{d}^{x_j}$, respectively. The potential deformation can be computed efficiently by evaluating

$$C_{ij}(x_i, x_j, u) = C(u) + \sum_{k=0}^{i} N_k(u)\, \mathbf{d}^{x_i} + \sum_{k=j}^{m-1} N_k(u)\, \mathbf{d}^{x_j}\ , \tag{4.36}$$

according to Equation (4.15).
Regarding the influence functions $\hat{N}_{ij}$, two different versions are considered in this work. Either through the addition of basis functions which is called the *sum model*

$$\hat{N}^{+}_{ij}(u) = \frac{N_i(u) + N_j(u)}{\sum_{l=k}^{k+p-1} N_l(u) + N_{l+1}(u)} \tag{4.37}$$

or through multiplication which is referred to as the *product model*

$$\hat{N}^*_{ij}(u) = \frac{N_i(u)N_j(u)}{\sum_{l=k}^{k+p-1} N_l(u)N_{l+1}(u)} \ . \tag{4.38}$$

Here, k is the index of the first non-zero basis function of a point $C(u)$ and p is the degree of the curve. The number of elements in the normalization terms (denominator) is solely depending on the curve's degree because every point $C(u)$ on a curve of degree d is defined by exactly $p+1$ control points, i.e. for every point $p+1$ basis functions $N_l(u)$ are non-zero. The normalization is required in order to ensure that the discretized version (4.33) of the energy term is correctly approximating the continuous version from Equation (4.31).

Exemplary plots of these functions can be found in Figure 4.2. The curves in the middle and right-hand side plot can be interpreted as weighting functions for points belonging to curve segments defined by two control points. As can be seen in the section on experiments in Table 4.1, the product model performs best. The reason is that the functions are steeper than those of the sum model. In other words one can say that the locality of the product model is better than the one of the sum model.

The performance of the two weighting functions is evaluated in Section 4.4.2 and the results which include a comparison to the naïve approach where external energies are modeled by unary potentials, i.e.

$$\theta^i_{\text{ext}}(x_i) = \int_0^1 N_i(u)\, \psi(\mathcal{I}\left(C_i(x_i,u)\right))\, \mathrm{d}u \ , \tag{4.39}$$

are summarized in Table 4.1.

It is shown that the presented modeling of the tracking represents a good compromise between model accuracy and complexity. One could claim that the approximation error could be reduced (or even completely removed) if more complex models were used. However, the consideration of higher-order cliques as proposed by Ramalingam et al. [64] is even despite recent developments of Ishikawa [35] computationally too time-consuming in order to be applicable in real-time environments.

4.3.1 Optimization

Once the problem is formulated in a discrete setting, one needs to choose an MRF optimization strategy. Fortunately, recent advances in discrete optimization brought a couple of very powerful techniques, mainly based on iterative graph-cuts or efficient message passing. Regarding the proposed model, there are two properties which should be considered when using one of the existing techniques. First, for the special case of open curves the MRF graph from Figure 4.1 is a tree which allows the exact computation of the global optimal labeling when using max-product algorithms [60, 81] (e.g. Belief Propagation, TRW-S [40]). Second, the energy is nonsubmodular which is a (theoretical) problem for some methods using graph-cuts [42]. Using certain truncation techniques [65] on the energy terms make it still possible to use graph-cut based techniques (e.g. Expansion Move [10]). Another possibility for minimizing nonsubmodular functions is described in [41] but this technique might result in unlabeled nodes which is not appropriate in the given setting. There are also methods based on iterative graph-cuts which can handle a wider class of MRF energies (e.g. Swap Move [10], FastPD [43]). Especially the FastPD algorithm is interesting in this case since a good performance with strong optimality properties is reported. In Section 4.4 the performance of the TRW-S and the FastPD algorithms are compared as representatives for the message-passing and graph-cut approaches. As it turns out, as usual there is a compromise between speed and accuracy.

4.4 Experiments and Evaluation

In this section the performance analysis of the proposed tracking approach is presented. The evaluation consists of three parts. First, it is shown based on synthetic tests that the optimal influence function is the *product model* from Equation (4.38). Next, the influence of sparse as well as dense label sets is evaluated. In parallel to the label sets the performance of fast but approximative optimization strategies is compared to exact though slower strategies. For this analysis, TRW-S (exact), FastPD (approximative) and QPBO (approximative) were chosen as representatives for the two possible optimization strategies. The last part presents evaluations of the tracking algorithm on hand-labeled clinical data and a comparisons to the guide-wire tracking approach presented recently by Wang et al. [79].

4.4.1 Error Assessment

Before continuing with the experiments the assessed error measures needs to be defined. A typically used error term for measuring the similarity of curves is the Hausdorff distance

$$\begin{aligned} d_H(C, C_G) = \max\Big(& \max_s \min_t d\big(C(s), C_G(t)\big)\,, \\ & \max_s \min_t d\big(C_G(s), C(t)\big)\Big). \end{aligned} \tag{4.40}$$

It is defined between a tracked curve C and the expected ground truth curve C_G and informally computes the maximal shortest distance between the two curves. One problem is that the whole curve C could be displaced orthogonal by some vector of length l and the Hausdorff distance would report the same error as for the case in which both curves were perfectly aligned with only a single pixel being off by a distance of l. In other words the distance measure does not handle outliers.

Another issue is that the proposed method tracks guide-wire segments over time while the guide-wire itself is usually advanced further through the vessel system during interventions. As a consequence such a system may be initialized with curves being substantially shorter than the ones observed later in the sequences. Since they are nonetheless labeled by experts during the generation of the ground-truth data, the

Hausdorff distance would report the length difference of the curves – even when the tracking were perfectly working.

These constraints lead to the choice of a simple discretized and asymmetric error measure

$$d(C, C_G, s) = \min_t \left(\left\| C(s) - C_G(t) \right\|_2 \right) \tag{4.41}$$

where one computes the minimal distance between the tracked and annotated curve. Evaluating Equation (4.41) at different sites s allows to compute statistics such as the mean tracking error, its standard deviation or the median error.

Similarly to what has been done in the work of Wang et al. [79] additional statistics are computed. These are the percentage of *missed* annotated pixels and the percentage of *false* tracked pixels. The *missed* pixels are pixels of the annotation having a distance to the tracked curve being higher than a specific threshold. Similarly, the *false* pixels are those on the tracked curve being further away from the annotation as defined by the same threshold. Conforming with previous works, the threshold is chosen to be equal to 3 pixels.

In the first case, regarding the *missed* pixels, the tracking failed to annotate ground-truth data. It basically *missed* to highlight given data. An example when this occurs is exactly the case described before where the initial curve or tracked curve is shorter than the corresponding annotated curve. The second case, the *false* pixels are those where the tracking computes a curve in locations where it actually does not belong to. The algorithms is literally creating *false* or wrong results.

4.4.2 Approximation Error

The first experiment is evaluating different ways of diminishing our model inherent approximation errors. In order to measure the influence of the approximation error several tests are performed on synthetic data. The data is generated by randomly deforming an initial open B-spline curve C_{init} consisting of six control points. After the deformation, an image frame is generated by careful rasterization of the deformed curve C_t. Different sequences with different amount of deformations are generated and each frame in every sequence is generated from the same initial curve C_{init}.

In this experiment the TRW-S [40] is used as the optimization method since it can recover the global optimal solution for tree-like MRF topologies as they occur in the proposed model. By knowing that the exact ground truth spline C_{init} is within the discrete label space, it is possible to estimate the error induced by the energy approximation only. Since furthermore each frame from every sequence is generated from the exact same initial curve C_{init} and since in this experiment the system is reinitialized with that specific curve in every frame, it is possible to discard the regularization term (i.e. set $\lambda = 0$) and thus an additional source which could bias the results because regularization is mainly of interest in longer sequences where tracking errors can accumulate over time.

As mentioned before, different amounts of maximum deformations are used in each sequence, ranging from 6 to 20 pixels control point displacements. Both proposed

max. def.	model unary	 pairwise sum	 pairwise product
6	1.00(±0.43)	0.34(±0.19)	0.28(±0.17)
8	1.15(±0.52)	0.36(±0.23)	0.30(±0.19)
10	1.33(±0.60)	0.42(±0.25)	0.36(±0.22)
12	1.41(±0.70)	0.43(±0.27)	0.36(±0.23)
14	1.74(±0.82)	0.48(±0.30)	0.43(±0.28)
16	1.79(±0.93)	0.49(±0.32)	0.43(±0.29)
18	2.00(±1.04)	0.52(±0.34)	0.47(±0.32)
20	2.19(±1.15)	0.57(±0.40)	0.52(±0.34)

Table 4.1: Synthetic experiment which assesses the energy approximation error under different amounts of deformation and three different likelihood models. Reported are the average curve distances and standard deviations in pixels over one hundred frames per sequence.

pairwise weighting functions, the *sum model* and the *product model*, are evaluated as well as the naïve approach with unary potentials for the external energy. For a quantitative evaluation of the synthetic results, the mean of the distances (and their standard deviations) from Equation (4.41) at different, uniformly sampled sites s are evaluated. The results are summarized in Table 4.1. The *product model* performs best on all sequences. Especially, when considering larger deformations the approximation error has a strong influence in case of unary potentials while both pairwise models still yield good results of always less than one pixel.

4.4.3 Deformation Space Discretization

The second experiment is intended to evaluate different discretization strategies as well as the effect of approximative optimizers like the FastPD or QPBO. Since the number of labels is an important parameter for the runtime of MRF optimization techniques, it is desired to determine a reasonable compromise between speed and tracking accuracy. Two different strategies for discretization, a *sparse* one and a *dense* one, are proposed. They are visualized in Figure 4.1. Both versions are parametrized by two values, the number of sampling steps along a certain displacement direction and the range which defines the allowed maximum displacement. In case of sparse discretization, the deformation space is sampled along eight directions, namely horizontal, vertical and diagonal each in positive and negative direction. In case of dense sets, the complete square space at a control point is sampled. Given the number of sample steps S, one gets $|\mathcal{L}_{\text{sparse}}| = 8S + 1$, i.e. eight directions with S steps and in addition the zero-displacement. For the dense version one gets $|\mathcal{L}_{\text{dense}}| = (S + 1)^2$. Similar to the first experiment, eight synthetic sequences are generated by assigning uniformly distributed random displacements on the six control points of a given initial curve C_{init}. The difference is that this time the ground truth is not covered by the label space. The range of the label space is set to the maximum random deformation and different values for the number of sampling steps are tested, namely 5, 10, and 20.

		sparse			dense		
max. def.	steps	TRW-S	FastPD	QPBO	TRW-S	FastPD	QPBO
6	5	0.32(±0.11)	0.44(±0.14)	0.35(±0.11)	0.30(±0.10)	0.41(±0.14)	0.33(±0.12)
	10	0.32(±0.11)	0.43(±0.13)	0.33(±0.11)	0.29(±0.10)	0.40(±0.12)	0.33(±0.12)
	20	0.31(±0.10)	0.43(±0.13)	0.33(±0.12)	0.29(±0.10)	0.39(±0.13)	0.32(±0.11)
8	5	0.42(±0.13)	0.59(±0.28)	0.48(±0.20)	0.37(±0.12)	0.53(±0.18)	0.43(±0.16)
	10	0.41(±0.14)	0.56(±0.22)	0.45(±0.18)	0.35(±0.12)	0.48(±0.18)	0.40(±0.15)
	20	0.40(±0.14)	0.55(±0.22)	0.44(±0.17)	0.35(±0.12)	0.48(±0.17)	0.40(±0.14)
10	5	0.52(±0.16)	0.76(±0.29)	0.61(±0.23)	0.48(±0.12)	0.69(±0.25)	0.57(±0.24)
	10	0.51(±0.15)	0.74(±0.28)	0.59(±0.23)	0.43(±0.12)	0.66(±0.25)	0.54(±0.20)
	20	0.50(±0.15)	0.73(±0.29)	0.57(±0.18)	0.44(±0.12)	0.64(±0.26)	0.54(±0.22)
12	5	0.62(±0.24)	0.87(±0.39)	0.72(±0.34)	0.53(±0.20)	0.75(±0.33)	0.65(±0.31)
	10	0.60(±0.23)	0.84(±0.33)	0.69(±0.29)	0.50(±0.17)	0.70(±0.26)	0.64(±0.38)
	20	0.59(±0.22)	0.82(±0.33)	0.70(±0.33)	0.48(±0.16)	0.73(±0.38)	0.61(±0.36)
14	5	0.75(±0.27)	0.98(±0.39)	0.85(±0.34)	0.59(±0.22)	0.86(±0.39)	0.75(±0.36)
	10	0.71(±0.29)	0.93(±0.42)	0.84(±0.39)	0.56(±0.20)	0.82(±0.42)	0.71(±0.41)
	20	0.71(±0.28)	0.93(±0.44)	0.82(±0.38)	0.56(±0.21)	0.79(±0.46)	0.71(±0.39)
16	5	0.91(±0.49)	1.30(±0.75)	1.13(±0.75)	0.69(±0.33)	1.17(±0.89)	0.95(±0.70)
	10	0.88(±0.47)	1.34(±0.96)	1.04(±0.65)	0.67(±0.28)	1.24(±0.97)	0.98(±0.79)
	20	0.87(±0.48)	1.33(±0.89)	1.09(±0.68)	0.66(±0.29)	1.14(±0.89)	0.94(±0.77)
18	5	1.11(±0.56)	1.47(±0.90)	1.26(±0.76)	0.89(±0.54)	1.34(±0.96)	1.22(±0.87)
	10	1.07(±0.56)	1.42(±0.84)	1.20(±0.70)	0.84(±0.49)	1.29(±0.95)	1.18(±0.94)
	20	1.07(±0.59)	1.48(±0.90)	1.25(±0.77)	0.84(±0.51)	1.28(±0.93)	1.13(±0.84)
20	5	1.16(±0.64)	1.54(±0.81)	1.38(±0.76)	0.96(±0.46)	1.37(±0.86)	1.20(±0.74)
	10	1.09(±0.68)	1.52(±0.87)	1.27(±0.70)	0.85(±0.45)	1.40(±0.98)	1.16(±0.78)
	20	1.08(±0.63)	1.51(±0.84)	1.31(±0.69)	0.83(±0.38)	1.33(±0.95)	1.14(±0.69)
run-time	5	119.49	25.80	38.15	356.00	30.55	33.61
	10	493.13	39.53	51.42	$> 4.5 \cdot 10^4$	71.22	70.07
	20	1573.80	68.36	81.49	$> 1.1 \cdot 10^5$	303.56	284.03

Table 4.2: Synthetic experiment for comparing the sparse and dense deformation space discretization. Run-times are reported in milliseconds and assessed on a 1.6 GHz Intel Core i7. All errors are reported in pixels. Throughout the experiment the FastPD was configured with a maximum of 100 iterations and the QPBO with 2 outer sweeps.

Another difference is that the initial curve used during the tracking is not parameterized in the same way as the curve used to generate the synthetic data. Therefore, the experiment does not only reflect errors resulting from deformations which are not covered by the label space but also those errors resulting from a sub-optimal curve parameterization.

Again, we use the mean distance (and its standard deviation) as a measure of tracking quality. The TRW-S optimizer is chosen since it represents a state-of-the-art algorithm computing the optimal labeling for our energy term (recall that our MRF is a chain). On the approximative side, we chose the FastPD and QPBO because of their speed and good optimality properties.

sequence name	frames	spacing	deformation		curve length	
			mean	max	mean	max
cardiac 01	55	0.184	17	28	413	438
cardiac 02	72	0.184	13	25	440	457
cardiac 03	30	0.184	16	43	505	543
cardiac 04	10	0.278	20	51	432	437
cardiac 05	11	0.278	13	49	645	665
abdominal 01	143	0.432	8	20	170	175
abdominal 02	228	0.308	8	19	78	83

Table 4.3: Sequence Properties. The pixel spacing is given in *mm* and the deformation and length parameters are denoted in *pixels*.

category	parameter	
	name	value
spline modeling	degree	3
	control points	12
discretization	search steps	40 (321 labels)
	search range	15 (mm)
regularization	λ	0.7

Table 4.4: Tracking Parameters.

Table 4.2 shows that the use of FastPD combined with sparse label sets is extremely efficient and reasonable tracking accuracy can be achieved. Furthermore, it shows that the QPBO achieves tracking results which are even closer to the optimal ones while introducing a minor runtime penalty. The difference between sparse and dense label sets is quite small regarding the tracking error while in case of sparse sets we achieve near real-time performance in all experiments. Expectedly, the TRW-S gives the better results in terms of accuracy but it is not suitable for real-time applications. In the initial work [31] these experiments lead to choosing the FastPD as the underlying optimizer which is now changed in favor of the QPBO.

4.4.4 Experiments on clinical data

In the final experiment, the tracking is evaluated on 7 clinical sequences out of which 5 are acquired during cardiac (diagnostic catheterizations) and 2 during abdominal (transarterial chemoembolization; TACE) interventions with an AXIOM-Artis [1] system at a cine rate of 15 frames per second. The sequences contain an overall of more than 500 frames and deformations of up to 50 pixels. More detailed properties of the individual sequences can be found in Table 4.3.
The cardiac data sets were made available to us by Wang et al. [79] and each of the

---

[1] Siemens Healthcare, Forchheim, Germany

sequence name	mean		standard deviation		median	
	MRF-T	RGWT	MRF-T	RGWT	MRF-T	RGWT
cardiac 01 (55 frames)	1.50812	1.66698	1.87401	1.63623	0.97893	1.17436
cardiac 02 (72 frames)	1.52339	1.57126	2.71825	1.80749	0.70623	0.89453
cardiac 03 (30 frames)	1.49126	1.29458	2.40836	1.50879	0.69463	0.730933
cardiac 04 (10 frames)	0.84560	0.82023	0.92164	0.732867	0.61347	0.686065
cardiac 05 (11 frames)	1.13857	1.33017	1.31548	1.4922	0.68615	0.817549
abdominal 01 (143 frames)	0.80728	n/a	0.63047	n/a	0.68212	n/a
abdominal 02 (228 frames)	1.18741	n/a	1.37192	n/a	0.78049	n/a

Table 4.5: Experimental validation between the robust guide-wire tracking (RGWT) method of [79] and the proposed MRF based tracking (MRF-T). The last two rows of the table show our method applied to two additional abdominal sequences where no comparison with the robust guide-wire tracking was possible.

data sets did not only contain image data but also ground-truth annotations as well as the tracking results of the author's algorithm. The abdominal sequences were provided by our clinical partners; they were acquired during TACE interventions and as the cardiac sequences, hand-labeled by an expert.

We use a single set of parameters (see Table 4.4) throughout all experiments. The regularization weight is set to $\lambda = 0.7$ though Figure 4.3 indicates that other choices are possible without severely decreasing the algorithm's performance. The figure shows the results of an experiment in which we evaluated the tracking error (the mean of false and missed rates) for 100 different parameters λ and for each medical sequence.

The labeling is configured with a search range of 15 mm and 40 search steps in each direction resulting in an overall of 321 labels per node. The sparse labeling is chosen accordingly to Section 4.4.3 and following the results of Section 4.4.2 the *product model* is used as our influence function.

Regarding the algorithm run-time, it is useful to split the overall processing into two steps. The first step, before the optimization is carried out, is the computation of the feature image $\mathcal{I}$ and the second step is the actual optimization process. We have measured a mean processing time of about 33.52 ms ($\sigma = 13.06$ ms) for the computation of the feature image $\mathcal{I}$ and a mean optimization time of 26.22 ms ($\sigma = 9.60$ ms). This results in an overall of about 60 ms processing time. With these processing rates this is to the best of our knowledge the first algorithm which is capable of processing fluoroscopic images at the same rate as they are acquired by the imaging system.

A summary of the comparison between the two methods can be found in Tables 4.5 and 4.6. The experiments show that our method is performing on average equally well as compared to the robust guide-wire tracking (RGWT) by Wang et al. [79] while it is not relying on any learning. Visual results of the tracking performance are depicted in Figure 4.4. The results show that the tracking may suffer from the introduction of local loops and foldings in areas of high curvature but it also shows corresponding images from the same sequences in which the algorithm recovered from such intermediate errors. The first row of Figure 4.4 shows in particular the robustness to poor features

sequence name	missed		false	
	MRF-T	RGWT	MRF-T	RGWT
cardiac 01 (55 frames)	10.7778	12.213	15.6204	13.5093
cardiac 02 (72 frames)	7.14789	11.3662	7.65493	9.94366
cardiac 03 (30 frames)	8.86207	9.18966	9.87931	11.1552
cardiac 04 (10 frames)	3.11111	1.44444	2	0.61111
cardiac 05 (11 frames)	9.1	10.3	8.8	11.85
abdominal 01 (143 frames)	1.86972	n/a	4.1338	n/a
abdominal 02 (228 frames)	6.14537	n/a	18.7489	n/a

Table 4.6: Experimental validation between the robust guide-wire tracking (RGWT) method of [79] and the proposed MRF based tracking (MRF-T). The last two rows of the table show our method applied to two additional abdominal sequences where no comparison with the robust guide-wire tracking was possible.

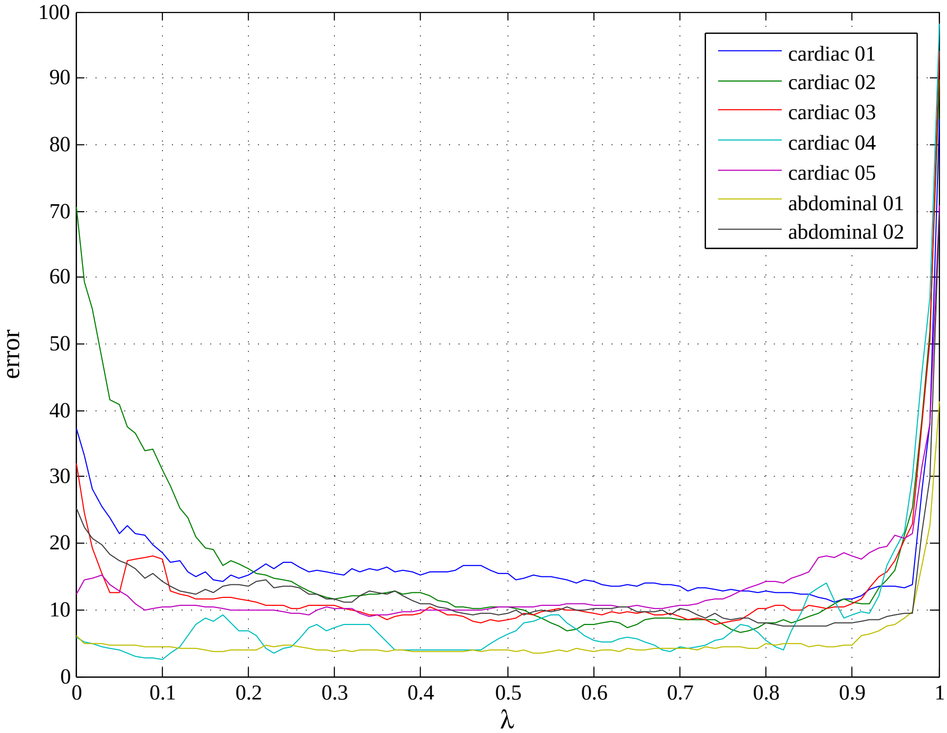
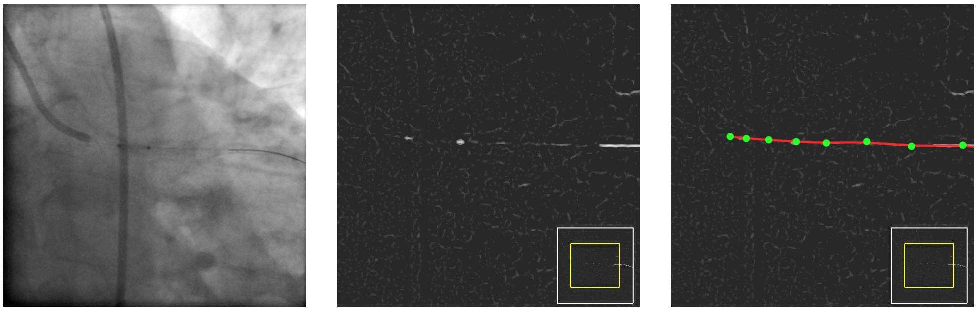
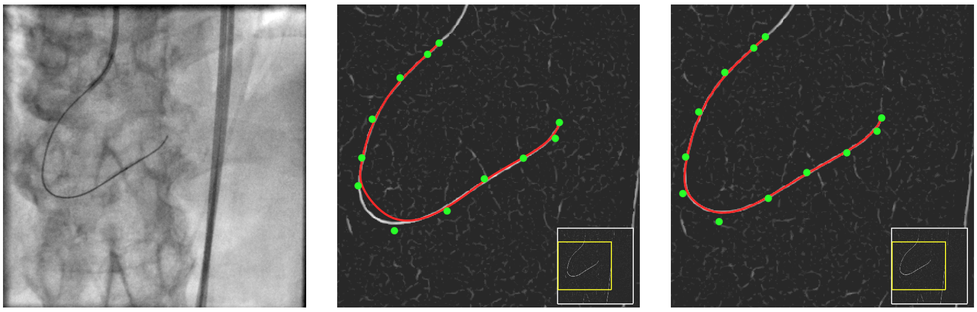


Figure 4.3: The plot shows the evaluation of the regularization parameter λ and its dependency on the mean of false and missed tracking rates. The plot indicates that a wide variety of parameter choices is possible without dramatically hampering the tracking quality.

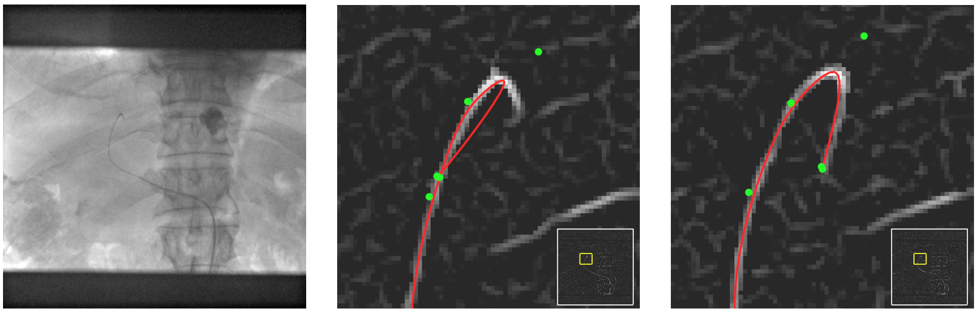
where the algorithm is capable of maintaining correct curve shapes despite the lack of image information.



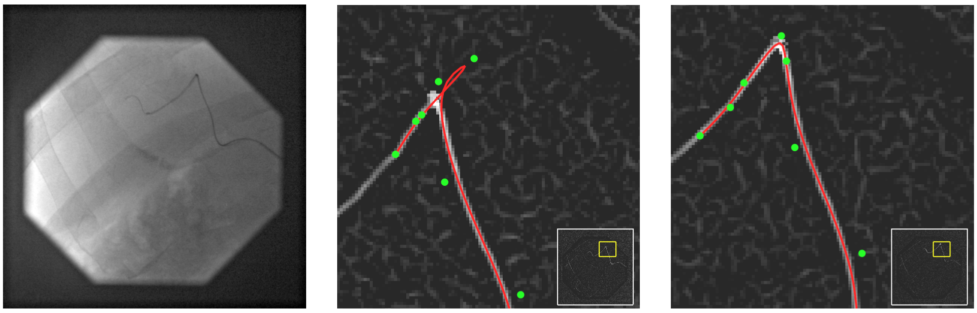
(a) fluoro - cardiac 01 (b) noisy feature image $\mathcal{I}$ (c) guide-wire annotation



(d) fluoro - cardiac 04 (e) too sparse label set (f) tracking recovered



(g) fluoro - abdominal 01 (h) folded curve) (i) tracking recovered



(j) fluoro - abdominal 02 (k) looped curve (l) tracking recovered

Figure 4.4: Tracking results from selected clinical sequences. The first row shows that the algorithm is able to deal with noisy feature images and the remaining rows depict the sources of intermediate tracking errors – a too sparse label set, foldings and the introduction of loops. In all three cases, the algorithm was capable of recovering from these errors.

CHAPTER

FIVE

A SIMPLIFIED MOTION COMPENSATION SYSTEM

In this chapter, the current status of an integrated motion compensation system is presented. The work which is described in this chapter has been carried out in close collaboration with the Siemens Healthcare Angiography/X-Ray division (AX) located in Forchheim, Germany and Siemens Corporate Research (SCR) in Princeton, New Jersey. The presented motion compensation system has been integrated into InSpace, a volume imaging application shipped with many radiologic hardware products of Siemens. The application allows the addition of new software modules in form of plug-ins which can be developed in C++ and added to any existing system.
The motion compensation module's structure can be described at best with three layers. A first layer is the actual core application hosting the plug-in, i.e. InSpace itself. InSpace provides the infrastructure for loading different plug-ins, for generating user interfaces and most importantly for accessing medical image and meta data. The second layer is the actual plug-in. It is used to leverage the infrastructure provided by InSpace in order to allow user interactions and for the generation of a user interface. Besides the user interface and interaction, the plug-in implements a control loop in which the data flow between InSpace and the underlying algorithmic component is handled. The third layer is build by the actual algorithms for guide-wire, catheter and needle detection as well as the tracking and motion estimation.

5.1 Plugin Design

Since the plug-in was designed as a research tool and not a clinical product it does not only contain dialogs which would be required in clinical routine. The additional interfaces allow to modify the visualization, e.g. the blending between the two-dimensional fluoroscopic image sequence and three-dimensional image data. It also allows the specification of offline data instead of being bound to online data which is important to be able to evaluate different methods on identical data sets. Regarding the online processing of data in principle only a single user input is required which allows the user to manually enter the detection phase where user interactions are required which are in general not available.

Another requirement during the design of the plug-in was the ability to interchange different tracking algorithms. To achieve this a generic programming interface was designed which allows different researchers to implement individual approaches to solve the detection, tracking and motion estimation. The implementation is provided in form of a so called dynamic library which is loaded during the run-time by the plug-in, i.e. from within the second layer. This approach allows a seamless exchange of the underlying library and thus the used algorithms and improves future maintainability because it separates the application interface from the underlying algorithmic components. The developed interface is provided in Listing C.1.

The interface has been designed based on the subsequent reasons. Starting with the requirements, it can be assumed that a motion compensation system needs to provide the following functionality

1. algorithm configuration
2. initialization
3. tracking
4. data exchange

and these items have been directly translated into the design of the interface. Another important factor is a strong decoupling of the second layer, i.e. the plug-in and the algorithmic component. Its definition must not impose dependencies on user specific data types but should be solely build upon fundamental data types or aggregates of these. This constrained is fulfilled on the one-hand side by creating an interface which is based on simple and well defined objects. On the other-hand side, a special implementation method, the Pimpl idiom [72] has been used. It allows to move all implementation details into a separate object which does not require to be exposed to the end-user. This procedure allows to provide an implementation of the above functionality without introducing additional dependencies. It reduces the interface to an absolute minimum of pre-defined objects.

The interface which is provided for the first point, the algorithm configuration assumes that all kinds of algorithms can be setup through a single configuration file. This should even be true for learning based algorithms which do not only need to provide general algorithmic parameters but also offline learned data and other prior information.

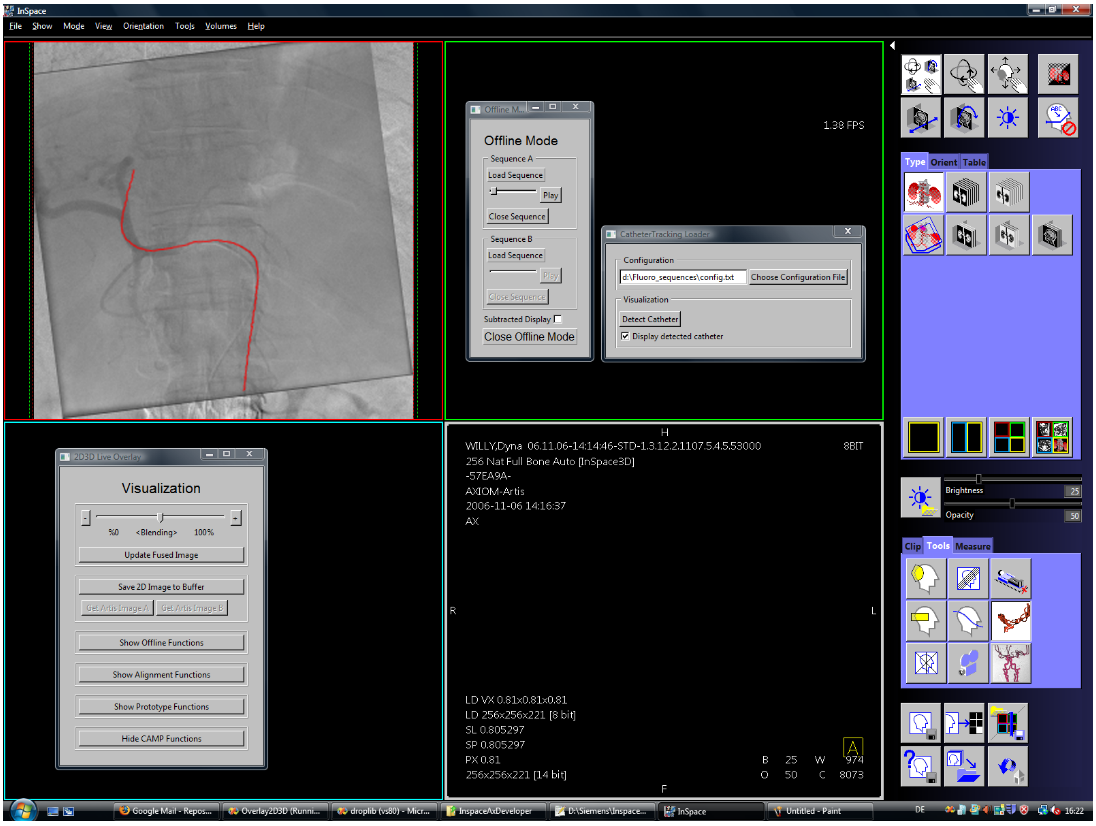
So far, the second point, the initialization is a semi-automatic method. Thus it requires not only images and the associated meta data but also the data generated by the user interaction. Currently, this data is simply a set of two user selected points but in order to be flexible in regard to potential extensions, the data is encapsulated in a separate class.

For the actual tracking a slightly different approach has been chosen in the sense that the interface does not allow to provide additional information besides the evidently required image data. The reasoning here is that even for complex algorithms, customizations are performed either internally or the required data is provided during the configuration phase. It is assume unlikely that a use case exists which explicitly requires the plug-in layer to provide dynamic information during the course of tracking.

The last point is regarding the data exchange. So far, only two use-cases are considered. First, it is of interest to visualize the tracking results and thus the interface provides functionality to retrieve the locations of tracked curves. Second, the plug-in layer requires information regarding the actual motion the curves performed. This information is provided in form of a two-dimensional transformation even though the presented algorithms actually compute dense deformation fields. The reason is that at the moment not all components of the system envisioned in Section 1.4 are available which lead to the development of an intermediate and simplified visualization system.

5.2 Visualization Component

Because the current work is only dealing with the third component presented in Section 1.4, which is the motion compensation within a fluoroscopic image sequence, a fully featured navigation system cannot yet be provided. Instead a simpler approach for aiding physicians during their navigation task is chosen. The approach chosen in this work is based on the rigid alignment of a roadmap image $\mathcal{I}_{\text{map}}$ and the currently observed fluoroscopic image $\mathcal{I}_t$. This is either possible by applying a rigid transformation $\mathcal{T}_t$ provided by the motion estimation to the roadmap image $\mathcal{I}_{\text{map}}$, or



d:\Fluoro_sequences\config.txt

```
WILLY,Dyna  06.11.06-14:14:46-STD-1.3.12.2.1107.5.4.5.53000          8BIT
256 Nat Full Bone Auto [InSpace3D]
-57EA9A-
AXIOM-Artis
2006-11-06 14:16:37
AX

LD VX 0.81x0.81x0.81
LD 256x256x221 [8 bit]
SL 0.805297
SP 0.805297
PX 0.81                                             B   25    W    974
256x256x221 [14 bit]                                O   50    C   8073
```

Brightness 25

Opacity 50

Figure 5.1: The image shows a screenshot of InSpace. The top-left view is used to display the actual fluoroscopic image stream including a DSA as its roadmap overlay.

alternatively the inverse transformation to the fluoroscopic image $\mathcal{I}_t$.
All choices presented here assume that the alignment between the roadmap image $\mathcal{I}_{\text{map}}$ and a key-frame $\mathcal{I}_{t_{key}}$ is known and without loss of generality this transformation is assumes to be the identity transformation. The required roadmap image can be created in one of the following ways:

1. **Based on a 3DRA:** Given a pre-interventionally acquired 3D-RA, it is possible to create a projection image $\mathcal{I}_{\text{map}}$ with the same projection geometry as the one of a fluoroscopic image sequence. 3D-RAs are reconstructed from native X-ray images generated by a calibrated C-arm system. The projection is possible because the 3D image as well as the 2D image sequences are generated by the same system in which all projections are performed relative to a common coordinate frame. Modern robot controlled system do not only allow high precision positioning of the X-ray source and the digital detector but also enable to read out the required pose information.

2. **Based on a DSA:** Alternatively to 3D-RAs, the generally acquired DSA can be used as a roadmap image. As opposed to the usage of a 3D-RA, the DSA must be generated with the same projection configuration as the fluoroscopic sequence in order to be able to overlay the images. Potential zooming which is oftentimes used during fluoroscopic runs can be compensated by an additional scaling and translation of the two-dimensional image data and the required transformation information is provided by the C-arm systems.

Both approaches suffer from the same problem – once the two-dimensional roadmap image is generated $\mathcal{I}_{\text{map}}$ the question remains how to choose the correct key-frame $\mathcal{I}_{t_{key}}$ from the fluoroscopic image sequence. The issue is that the choice is not arbitrary since one needs to find a reference image which is optimally corresponding to the roadmap, i.e. with an underlying physical patient geometry being as close as possible to the one from which the roadmap was generated. A first step towards a similar patient geometry is to find a fluoroscopic image which is acquired in the same phase of the breathing cycle as the roadmap image $\mathcal{I}_{\text{map}}$. In Section 1.4, this task is meant to be solved within the *Deformable 2D-2D Registration Module.* For the task of finding a fluoroscopic image $\mathcal{I}_{t_{key}}$ corresponding best to a given DSA a straight forward approach were to select the image which correlates best to the native X-ray image based on which the DSA was generated. The so chosen image $\mathcal{I}_{t_{key}}$ is the best candidate one can find to pursue the missing 2D-2D registration in order to compensate additional deformations.
The only component which is missing to finalize the plug-in visualization is the actual computation of the rigid transformation $\mathcal{T}_t$ which relates the guide-wire locations $\mathcal{P}_{t_{key}}$ of the key-frame to the currently observed image $\mathcal{I}_t$. To this end, a method for the least-squares estimation of transformation parameters between two points sets which is proposed by Umeyama [76] is used. In his work Umeyama shows that for two points sets $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$, the transformation that minimizes

$$\epsilon^2(\mathbf{R}, \mathbf{t}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{y}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|^2 \tag{5.1}$$

can be computed from the covariance matrix of the demeaned point sets

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \mu_y)(\mathbf{x}_i - \mu_x)^\top \ . \tag{5.2}$$

and the corresponding mean values μ_x and μ_y. Having computed the solution to this problem, for points sets $\mathcal{P}_{t_{key}}$ and $\mathcal{P}_t$, one gets

$$\mathcal{T}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix} \ . \tag{5.3}$$

Based on the above description a motion compensation system has been built and integrated into InSpace. Figure 5.1 shows a screenshot of the system including a transformed image $\mathcal{I}_t$ such that it optimally fits a DSA roadmap.

CHAPTER

SIX

CONCLUSION

This thesis introduces novel methods for the semi-automatic detection of medical tools such as guide-wires, catheters or biopsy needles. Furthermore, the work presents for the first time an approach which solves the tracking problem for B-spline curves based on discrete optimization. The focus of this work lies on clinical applications being guided by fluoroscopic imaging. In particular the method has been evaluated on abdominal data sets acquired during transarterial chemoembolization (TACE) and five clinical cardiac sequences.

The development of the semi-automatic detection method which is based on sparse graphs induced by the second-order structure of image intensities does not only allow a robust but also a fast segmentation of curvilinear structures. Since full featured applications usually consist of processing pipelines containing several algorithms at once, run-time reduction while retaining stable algorithms is an important topic.

A new tracking approach which is based on a MRF formulation represents the main contribution of this work. The problem is formulated as finding a solution to a discrete labeling problem which has recently received increasing attention in many early vision problems. In contrast to the common approach applied in vision problems, in which MRF nodes correspond to the image lattice this work proposes to model MRF nodes as control points of a B-spline curve. Such curves can be either open curves representing curvilinear structures such as e.g. medical tools or they can be close curves as parameterizations of object contours. This implies that the approach is not limited to tracking but maybe as well applied to segmentation problems.

One of the main advantages of discrete optimization is that it is currently considered as one of the most-powerful techniques to minimize non-convex functions. Especially in the scenario of tracking curves in fluoroscopic image sequences non-convexity is a potential problem. All image based data terms rely on feature images highlighting curvilinear structures in one or the other way and these methods have no way of distinguishing between bone interfaces and thin medical tools. As it can be seen in the examples in Figure 3.13, the images contain many line-like structures which can potentially attract the curves and cause local minima.

Another advantageous property of MRF based methods is that they do not rely on

gradients. This property makes it possible to use a large variety cost functions without being able to provide analytical derivatives and without worrying about performance or stability degradations that might be caused by numerical differentiation. This allows for instance to use learning based measures which offer the potential of further increasing tracking stability.
Experiments have shown that the proposed method is performing at high-precision. This is true for multiple sequences which are even acquired during different interventional procedures (abdominal and cardiac). Furthermore, experiments analyzing the behavior of the regularization revealed that the algorithm is relative stable within a certain parameter range. Last but not least, the procedure is running near real-time being faster than other tracking systems being presented so far without necessarily requiring a rigid pre-registration.

6.1 Discussion

Besides the just presented advantages the approach is not without difficulties. One problem is the incomplete modeling of the inter-dependency of individual random variables. This problem leads to intermediate curves $\widetilde{C}(u, \mathbf{d}_j, \mathbf{d}_k)$ which only approximate the expected curve as it is depicted in Figure 6.1. Careful handling of this approximation via weighting functions (Equations (4.37) and (4.38)) diminishes the issue but does not completely solve it.
In the configuration depicted in Figure 6.1, a source curve consisting of control points $\mathbf{S}_i$ being rotated by 15° to a target curve $\mathbf{T}_i$ is displaced in the control points S_1 to S_4 with the expected displacements of S_2 and S_3 repectively. This results in and intermediate curve where $I_1 = S_1 + (T_2 - S_2)$, $I_2 = T_2$, $I_3 = T_3$, $I_4 = S_4 + (T_3 - S_3)$ and $I_5 = S_5$. Such a configuration is exactly what were evaluated by θ^{23}_{ext} and θ^{23}_{int} from Equations (4.34) and (4.35). The problem in this case is evident in Figure 6.1. Despite the correct displacements being applied the control points S_2 and S_3 of the source curve, the intermediate approximation does not overlap with the target.

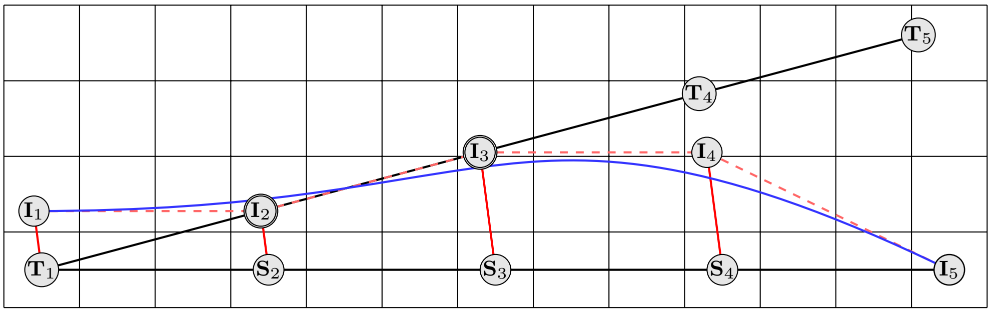


Figure 6.1: Approximation Errors of Intermediate Curves.

Even though a perfect alignment is not necessary it can still lead to difficulties and hamper the convergence properties. There are two possible reasons, why the perfect intermediate alignment is not required. First, even though thin, medical tools have a specific width, i.e. it exists a corridor around the target curve in which the external energy is non-zero and where the intermediate curve does result in an energy response which drives the optimization towards the correct solution. The second potential reason is that the optimization algorithm can solve the problem by initially assigning wrong intermediate results. In fact, such results will be tested and even assigned as long as they result in a reduced energy. Since the optimization is an iterative process, upcoming iterations can fix the initially wrong result.

Again, there may occur a potential problem with the latter solution of choosing actually wrong intermediate results. It arises from certain regularization terms. In theory it may happen that wrong intermediate labels are the only way to reduce the external energy but on the other hand side such labelings may vastly increase the internal energy. An example is a regularization term penalizing transformations varying from rigid transformations. In case such a cost function would be evaluated on the actual spline points the intermediate labeling would be penalized by the regularization. Unfortunately, the situation is rather difficult to evaluate because in this case the data term and regularization are acting against each other.

The issues above were raised for the sake of clarity and to increase awareness. The experimental section has proven the effectiveness of the method and should encourage future researchers to utilize the possibilities offered by discrete optimization frameworks.

6.2 Future Work

In this section an alternative learning based tracking method is provided. The work has been developed in collaboration with Olivier Pauly and Nassir Navab [59]. In principle this work proposes to replace the external energy function defined in Equation (4.20) by another function which is learned from hand-labeled offline data. The proposed modification also leads to a slight modification of the original discretization (see Equation (4.34)).

The main idea of the approach is that instead on relying on feature image intensities $\mathcal{V}$ along the curve profile $C(u)$, a curve descriptor is extracted from the unprocessed image $\mathcal{I}$. To this end, so called *mean orthogonal intensity profiles* are used. They represent a relatively simple descriptor for curves and the external energy can be seen as measuring the deviation from an observed descriptor to a learned and expected descriptor.

Figure 6.2 shows a spline curve and the close-up of a ground-truth spline segment (dashed red curve) as well as a displaced spline segment (solid red curve). In the close-up view, the top-right plot shows the mean orthogonal profile of the ground-truth curve (green) and the mean orthogonal profile of the displaced curve (red).

As it is expected from the derivation provided in Sections 3.1.2 and 3.1.3, the ground-truth curve's model is similar to a parabolic model. However, the model is not identical since it is not symmetric in the sense that the intensities on both sides of the medical tool are varying. For such cases, learning based methods are advantageous because they make no prior assumptions on the actual model though they do rely on a learning database which provides the whole variety of possible curve appearances.

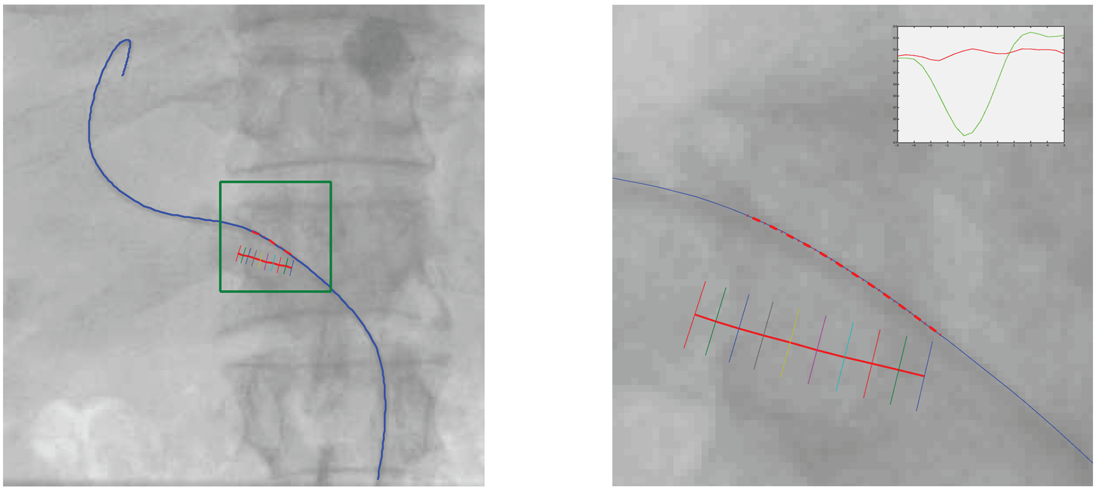


Figure 6.2: The left-hand side picture shows a guide-wire with a curve segment (dashed red line). The curve segment corresponds to the part of the guide-wire which is influenced by two neighboring control points. The right-hand side image shows a close-up with a transformed spline segment and equally sampled orthogonal profiles. The top left shows the mean profiles of the ground truth as well as the displaced curve segment.

For a curve segment $C_{ij}(x_i, x_j, u)$ orthogonal line segments of length $2r$ are defined as

$$\mathcal{N}_{ij}(x_i, x_j, u, r) = C_{ij}(x_i, x_j, u) + r\mathbf{n}(u) \tag{6.1}$$

where $\mathbf{n}(u)$ is corresponding to the normal of the curve segment C_{ij} at the site u. A mean intensity profile can then be defined as

$$\mathcal{M}_{ij}(x_i, x_j, r) = \frac{1}{|\mathrm{supp}(C_{ij})|} \int\limits_{\mathrm{supp}(C_{ij})} \int\limits_{-r}^{r} \mathcal{I}(\mathcal{N}_{ij}(x_i, x_j, u, r))\, \mathrm{d}r\, \mathrm{d}u \ . \tag{6.2}$$

The mean profiles are only parameterized by the control points i and j as well as the normal radius. Given a curve $C(u)$ with m control points one can compute exactly $m-1$ profiles with neighboring control points which build the input data set

$$\mathcal{M} = \left\{ \mathcal{M}_{i,i+1} | i = 0, 1, \ldots, m-2 \right\} \tag{6.3}$$

whose elements are later evaluated by the learned cost function.
The goal is now to learn a function ψ which maps local mean orthogonal profiles $\mathcal{M}_{ij} \in \mathbb{M}$ to a real values corresponding in some way to the expected tracking error, i.e. ψ is defined as

$$\psi \ : \ \mathbb{M} \mapsto \mathbb{R} \ , \tag{6.4}$$

where $\mathbb{M}$ is the set of mean orthogonal profiles. Since such a function should be convex and smooth, a thorough sampling of the space $\mathbb{M}$ is required. In [59] it is proposed to learn a probability distribution corresponding to typical guide-wire motions from hand-labeled sequences to ensure a proper sampling of $\mathbb{M}$. Once the probability distribution is available it is possible to create random curve displacements exhibiting typical guide-wire motions. Such a sampling allows the generation of pairs of mean intensity profiles and the associated errors

$$\Big\{(\mathcal{M}_{i,i+1}, \epsilon_i)\Big\}_k \tag{6.5}$$

on which a regression can be performed to learn the actual function ψ. Details on the learning of *typical guide-wire motions* and the regression learning can be found in [59]. The actual pairwise data term is then computed as

$$\theta_{ij}(x_i, x_j) = \psi\left(\mathcal{M}_{ij}(x_i, x_j, r)\right) \ . \tag{6.6}$$

Preliminary results have shown that the method is feasible for the tracking of guide-wires in fluoroscopic sequences. Sub-pixel precision was achieved in different experiments with different profile radii of $r = 5mm$ and $r = 10mm$ on two abdominal data sets. The experiments were carried out by learning on one abdominal sequence and performing the tracking on the other and vice versa. The advantage of the method is that no pre-processing of the images is required and no prior assumptions on the model are required since its appearance is learned.

Despite the integration of a learning based cost function into a discrete framework, the work also proposes a second novel idea. Initially, the high-dimensionality of the feature space $\mathbb{M}$ posed a problem since it severely complicates a meaningful sampling of the space in order to generate data for the learning process. This problem has been overcome by fitting a mixture of Gaussian model to hand-labeled deformations which in turn allows the random generation of deformations which are likely to occur in reality. Future improvements can be achieved by exploring the possibility of an adapted regularization term which could be e.g. derived from the motion distribution model. Another improvement can probably be achieved by integrating the weighting functions introduced in Equation (4.38). So far these are ignored and as it has been shown in this thesis, they may increase the performance of the algorithm even further.

APPENDIX

A

AUTHORED AND CO-AUTHORED PUBLICATIONS

1. Bender, F., Groher, M., Khamene, A., Wein, W., Heibel, T. H., and Navab, N. 3D dynamic roadmapping for abdominal catheterizations. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (New York, USA, September 2008)

2. Groher, M., Bender, F., Khamene, A., Wein, W., Heibel, T. H., and Navab, N. 3D guide wire navigation from single plane fluoroscopic images in abdominal catheterizations. In *Bildverarbeitung in der Medizin* (2009)

3. Heibel, T. H., Glocker, B., Groher, M., Paragios, N., Komodakis, N., and Navab, N. Discrete tracking of parametrized curves. In *IEEE Conference on Computer Vision and Pattern Recognition* (2009)

4. Heibel, T. H., Glocker, B., Paragios, N., and Navab, N. Needle tracking through higher-order MRF optimization. In *IEEE International Symposium on Biomedical Imaging* (2010)

5. Pauly, O., Heibel, T. H., and Navab, N. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2010)

6. Platonov, J., Heibel, T. H., Meier, P., and Grollmann, B. A mobile markerless AR system for maintenance and repair. In *International Symposium on Mixed and Augmented Reality* (2006)

7. Traub, J., Heibel, T. H., Dressel, P., Heining, S., Graumann, R., and Navab, N. A multi-view opto-xray imaging system: Development and first application in trauma surgery. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2007)

8. Platonov, J., Heibel, T. H., and Meier, P. Camera position and orientation determining method for use in e.g. production area, involves projecting

data model of virtual information with preset camera parameters and finding current parameters using external values of cost-function. Patent No. DE102005025470 (A1), 2006

9. Navab, N., Traub, J., and Heibel, T. H. Method for calibration of a camera augmented c-arm. Patent No. WO2009077480 (A1), 2009
10. Glocker, B., Heibel, T. H., Navab, N., Kohli, P., and Rother, C. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *European Conference on Computer Vision* (Crete, Greece, September 2010)
11. Wang, L., Springer, M., Heibel, T. H., and Navab, N. Floyd-Warshall All-Pair Shortest Path for Accurate Multi-Marker Calibration. In *International Symposium on Mixed and Augmented Reality* (October 2010)

APPENDIX

B

MATHEMATICAL EXTENSIONS

B.1 Enhancement of Tubular Structures

B.1.1 Sato's and Frangi's Feature Enhancement

The following equations represent the analytic expressions of Sato's and Frangi's feature enhancement in the *pq*-coordinate frame which are used in Equation (3.39) and Equation (3.46), respectively. Equation (B.1) is corresponding to the block model $\mathcal{I}_b$ which is defined in Equation (3.10) and Equation (B.2) corresponds to the parabolic model $\mathcal{I}_p$ defined by Equation (3.12).

$$
\begin{aligned}
H_{pp}(x,t,\gamma,r,h) = -\frac{1}{\sqrt{2}\sqrt{\pi}} \Bigg(& \\
& \exp\left(-\frac{(r+x)^2}{2t}\right) t^{\gamma-\frac{3}{2}} h \left(\exp\left(\frac{2rx}{t}\right)(r-x)+r+x\right) \Bigg)
\end{aligned}
\tag{B.1}
$$

$$
\begin{aligned}
H_{pp}(x,t,\gamma,r,h) = -\frac{\exp\left(-\frac{rx}{t}\right) t^{\gamma-1} h}{\pi r^2 \sqrt{2}} \Bigg(& \\
& \pi t \sqrt{2} \exp\left(\frac{rx}{t}\right) \left(\operatorname{erf}\left(\frac{r+x}{\sqrt{2}\sqrt{t}}\right) + \operatorname{erf}\left(\frac{r-x}{\sqrt{2}\sqrt{t}}\right)\right) - \\
& 2\sqrt{t}\sqrt{\pi} r \left(\exp\left(-\frac{r^2+x^2}{2t}\right) + \exp\left(-\frac{r^2-4rx+x^2}{2t}\right)\right) \Bigg) \ .
\end{aligned}
\tag{B.2}
$$

The γ value provided in Table 3.1 and Table 3.2 results form the following analytical expression

$$
\gamma = \frac{1}{\sqrt{2}\sqrt{\mathrm{e}}\sqrt{\pi} \operatorname{erf}\left(\frac{1}{2}\sqrt{2}\right) - 2} \ . \tag{B.3}
$$

B.1.2 Koller's Feature Enhancement

The following equations represent the analytic expressions of Koller's feature enhancement in the pq-coordinate frame which are used to compute Equations (3.50) and (3.51). The formulas are computed as $I_x = G_x * \mathcal{I}_*$.

$$
\begin{aligned}
I_x(x,t,\gamma,r,h) = -\frac{1}{\sqrt{2}\sqrt{\pi}} \Bigg(\\
\sqrt{2}\exp\left(-\frac{(r+x)^2}{2t}\right) t^{\gamma-\frac{1}{2}} h \left(\exp\left(\frac{2rx}{t}\right) - 1\right) \Bigg)
\end{aligned}
\tag{B.4}
$$

$$
\begin{aligned}
I_x(x,t,\gamma,r,h) = \frac{\exp\left(-\frac{rx}{t}\right) t^{\gamma} h}{\pi r^2 \sqrt{2}} \Bigg(\\
\sqrt{2}\pi x \exp\left(\frac{rx}{t}\right)\left(\operatorname{erf}\left(\frac{r+x}{\sqrt{2}\sqrt{t}}\right) - \operatorname{erf}\left(\frac{r-x}{\sqrt{2}\sqrt{t}}\right)\right) + \\
2\sqrt{t}\sqrt{\pi}\left(\exp\left(-\frac{r^2-4rx+x^2}{2t}\right) - \exp\left(-\frac{r^2+x^2}{2t}\right)\right) \Bigg)
\end{aligned}
\tag{B.5}
$$

The actual formulas $\mathcal{R}_l$ and $\mathcal{R}_r$ are computed as

$$
\mathcal{R}_l(x,t,\gamma,r,h) = I_x(x-\sqrt{t},t,\gamma,r,h) \tag{B.6}
$$

$$
\mathcal{R}_r(x,t,\gamma,r,h) = -I_x(x+\sqrt{t},t,\gamma,r,h) \tag{B.7}
$$

where I_x is chosen as Equation (B.4) for the block model $\mathcal{I}_b$ and as Equation (B.5) for the parabolic model $\mathcal{I}_p$.
The analytic equations for the γ values presented in Table 3.3 are defined below.

$$
\gamma = \frac{1}{2}\,\frac{\mathrm{e}^2+1}{\mathrm{e}^2-1} \tag{B.8}
$$

$$
\gamma = -\frac{1}{2}\,\frac{4\sqrt{\pi} + \sqrt{2}\operatorname{erf}\left(\sqrt{2}\right)\pi\mathrm{e}^2 - 4\sqrt{\pi}\mathrm{e}^2}{2\sqrt{\pi} + \sqrt{2}\operatorname{erf}\left(\sqrt{2}\right)\pi\mathrm{e}^2 - 2\sqrt{\pi}\mathrm{e}^2} \tag{B.9}
$$

$$
\gamma = -\frac{1}{2}\,\frac{\operatorname{erf}\left(\sqrt{2}\right)^2\pi\mathrm{e}^4 + 3\sqrt{2}\sqrt{\pi}\operatorname{erf}\left(\sqrt{2}\right)\mathrm{e}^2(1-\mathrm{e}^2) - 8\mathrm{e}^4 + 4\mathrm{e}^4 + 4}{\operatorname{erf}\left(\sqrt{2}\right)^2\pi\mathrm{e}^4 + 2\sqrt{2}\sqrt{\pi}\operatorname{erf}\left(\sqrt{2}\right)\mathrm{e}^2(1-\mathrm{e}^2) - 4\mathrm{e}^4 + 2\mathrm{e}^4 + 2} \tag{B.10}
$$

Above, Equation (B.8) is the optimal γ for the block model $\mathcal{I}_b$ for both non-linear functions defined in Equations (3.52) and (3.53). The last two Equations are the γ values of the parabolic model $\mathcal{I}_p$ belonging to Equations (3.52) and (3.53), respectively.

APPENDIX

C

CODE SAMPLES

Listing C.1: Minimalistic but generic interface for tracking which allows the implementation and evaluation of various approaches.

```
class MotionEstimation
{
public:
  /** Member variable initialization. */
  MotionEstimation();

  /** Implements memory cleanup. */
  ~MotionEstimation();

  /** System configuration via configuration file. */
  void ConfigureSystem(const char *configuration_file);

  /** Curve detection with an image and potential user interactions as its parameters. */
  void Detection(const ImageContainer& image, const DetectionParameter* detection_parameter = 0);

  /** Tracking based on the current image and meta data. */
  void Tracking(const ImageContainer& image);

  /** Retrieval of two-dimensional rigid motion data. */
  void GetMotionEstimate(float& x, float& y, float& angle) const;

  /** Retrieval of detected curves. */
  void GetGWCurves(GWCurve const*& curves, int& no_curves) const;

  /** Optional system reset to recover from errornous states. */
  void Reset();

private:
  /* Copying or cloning as well as assigning of the class are prohibited. */
  MotionEstimation(const MotionEstimation&) {}
  MotionEstimation& operator=(const MotionEstimation&) { return *this; }

  int m_numberOfCurve; // Holds the number of tracked/detected curves.
  GWCurve* m_curves; // Holds tracked curves in form of point sets.

  struct MotionEstimationImpl; // Forward declaration of the implementation.
  MotionEstimationImpl* m_pImpl; // A pointer to the implementation.
};
```

Listing C.2: Eigenvalue decomposition for symmetric 2-by-2 matrices. The code requires additional checks to prevent devision by zero.

```
template <typename T>
void eig(const T& xx, const T& xy, const T& yy, T evals[2], T evec[2])
{
  const T root = std::sqrt( std::pow(xx-yy,2) + 4*xy*xy );
  const T l1 = T(0.5)*(xx + yy - root);
  const T l2 = T(0.5)*(xx + yy + root);

  T ilen = 0;
  if (xx-yy>=0)
  {
    evec[0] = 2*xy / (yy - xx - root);
    evec[1] = 1;
    ilen = T(1) / std::sqrt(1 + evec[0]*evec[0]);
  }
  else
  {
    evec[0] = 2*xy / (yy - xx + root);
    evec[1] = 1;
    ilen = T(1) / std::sqrt(1 + evec[1]*evec[1]);
  }
  evec[0] *= ilen;
  evec[1] *= ilen;
}
```

LIST OF FIGURES

1.1 Most common 2D image modalities in interventional radiology. 7

1.2 Computed Tomography Angiography (left) and Magnetic Resonance Angiography (right). 9

1.3 Angiographic intervention setup. The monitor shows a live X-ray video stream (left) next to a DSA (right) which serves as a roadmap during the intervention. The object on the left-hand side of the image is a flat-panel detector from a modern C-arm system. 10

1.4 Components of interventional navigation. 12

3.1 Idealized X-ray projection of a tubular object. The object is centered at C. The path lengths at which X-rays are attenuated in the tubular object vary with the projection direction α. The path length is denoted as $l(c_x, c_y, r, \alpha)$ and two examples for $\alpha = 0$ and $\alpha = 8^\circ$ are shown in red. 34

3.2 Idealized intensity responses of a tubular object with a radius $r = 1$ which is located at a distance of 1 (i.e. $c_x^2 + c_y^2 = 1$) to the X-ray source. The plot shows the responses for a constant incident intensity of $I_0 = 1$ for varying attenuation coefficients μ. The functions correspond to the output of Equation (3.9). 36

3.3 The figure shows optimal fits between three different ridge models and the *idealized* block model derived in Section 3.1.2. The models are (from left to right), a symmetric block model, an elliptical model and a Gaussian model. The red curves represent the optimal models after fixing the width to the expected value while the green curves show the optimal models after fixing their height to the expected value. 38

3.4 The two graphs in this figure depict a ridge model as it is generated by Equation (3.9). The left-hand side figure displays the model as a three-dimensional surface and the right-hand side figure show the corresponding two-dimensional gray-scale image. The red coordinate system with axes p and q represents the *second-order gauge coordinates* which are intrinsic to the image surface. 40

3.5 The figure shows the vesselness response of the filter proposed by Sato et al. for a block model $\mathcal{I}_b(x, 2, 2)$ with width and height equal to 2. The ditch for small scale-space values of t and $x = 0$ indicates, that the responses are not meaningful for all scale-space parameters. 46

3.6 This plot shows the effect of the choice of the parameter c for Frangi's *structuredness* measure $1 - \exp\left(-\frac{S^2}{2c^2}\right)$. When c is chosen too small, the filter responses of neighboring but large values of S are hard to distinguish. 48

3.7 Comparison between Frangi's vesselness enhancement with different scale-space normalization constants. The images were created with scales $\sigma_i \in \{0.5, 1.5, 2.0\}$ and $1.5\sigma_i$. The scale-space normalization constants are computed according to Equation (3.49). 49

3.8 Shifted first-derivative filter responses of a parabolic profile $\mathcal{I}_p$ with radius $r = 1$ and height $h = 2$. The filter response $\mathcal{R}_l$ is depicted in blue. The red line is corresponding to the filter $\mathcal{R}_l$. Finally, the dashed line represents the non-linear combination $\mathcal{R}_s = \sqrt{\mathcal{R}_l \mathcal{R}_r}$ in cases where $|x| \leq r$ and zero otherwise. 51

3.9 Filter responses on a toy example. The scale-space responses are computed with the optimal σ values for the parabolic model. Scale values are ranging from 1 to 4 and an overall of 21 scales were probed, i.e. $t \in \{1, 1.15, 1.3, \ldots, 4.0\}$. 52

3.10 Filter responses of a medical example. The input image on the left shows a head in which the vessels supporting the brain are enhanced by contrast agent. 53

3.11 Closeup of minimal initialization-graph with branching points. 54

3.12 Neighbor selection based on tangent direction $\mathbf{t}$ (corresponding to the red vector). The green squares indicate the selected neighbors. 56

3.13 The figure shows two exemplary results of a fully automatic ridge detection method. On the left-hand side, an approach derived from Canny's edge detection algorithm has been used and on the right-hand side is the method of Carsten Steger. Both methods generate multiple responses and require additional post-processing to filter out spurious results. 59

4.1 MRF Model of an open cubic B-spline curve C with control points $\mathbf{P}_i$. A sparse and a dense version of the discrete set of labels $\mathcal{L}$ capturing the deformation space are illustrated (each black square corresponds to a displacement plus the zero-displacement at the control point position). . . 74

4.2 Influence functions originating from an open cubic B-spline with seven control points. 76

4.3 The plot shows the evaluation of the regularization parameter λ and its dependency on the mean of false and missed tracking rates. The plot indicates that a wide variety of parameter choices is possible without dramatically hampering the tracking quality. 84

4.4 Tracking results from selected clinical sequences. The first row shows that the algorithm is able to deal with noisy feature images and the remaining rows depict the sources of intermediate tracking errors – a too sparse label set, foldings and the introduction of loops. In all three cases, the algorithm was capable of recovering from these errors. 85

5.1 The image shows a screenshot of InSpace. The top-left view is used to display the actual fluoroscopic image stream including a DSA as its roadmap overlay. 89

6.1 Approximation Errors of Intermediate Curves. 94

6.2 The left-hand side picture shows a guide-wire with a curve segment (dashed red line). The curve segment corresponds to the part of the guide-wire which is influenced by two neighboring control points. The right-hand side image shows a close-up with a transformed spline segment and equally sampled orthogonal profiles. The top left shows the mean profiles of the ground truth as well as the displaced curve segment. 96

REFERENCES

[1] Atasoy, S., Groher, M., Zikic, D., Waggershauser, T., Marcus, P., and Navab, N. Real-time respiratory motion tracking: Roadmap correction for hepatic artery catheterizations. In *SPIE Medical Imaging* (2008).

[2] Baert, S., van de Kraats, E. B., van Walsum, T., Viergever, M. A., and Niessen, W. J. Three-dimensional guide-wire reconstruction from biplane image sequences for integrated display in 3-D vasculature. *IEEE Transactions on Medical Imaging 22* (2003), 1252 – 1258.

[3] Baert, S., Viergever, M. A., and Niessen, W. J. Guide-wire tracking during endovascular interventions. *IEEE Transactions on Medical Imaging 22*, 8 (Aug. 2003), 965–972.

[4] Baert, S. A. M. *Guide wire tracking and reconstruction in interventional radiology.* PhD thesis, University Medical Center Utrecht, 2003.

[5] Baert, S. A. M., Niessen, W. J., Meijering, E. H. W., Frangi, A. F., and Viergever, M. A. Guide wire tracking in interventional radiology. In *Computer Assisted Radiology and Surgery* (2000).

[6] Baert, S. A. M., van Walsum, T., and Niessen, W. J. Endpoint localization in guide wire tracking during endovascular interventions. *Academic Radiology 10* (2003), 1424–1432.

[7] Barbu, A., Athitsos, V., Georgescu, B., Böhm, S., Durlak, P., and Comaniciu, D. Hierarchical learning of curves application to guidewire localization in fluoroscopy. In *IEEE Conference on Computer Vision and Pattern Recognition* (2007), IEEE Computer Society.

[8] Bender, F., Groher, M., Khamene, A., Wein, W., Heibel, T. H., and Navab, N. 3D dynamic roadmapping for abdominal catheterizations. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (New York, USA, September 2008).

[9] Bender, F. T. Segmentation, 2D-3D registration and uncertainty propagation for dynamic roadmapping in angiographic interventions. Master's thesis, Technische Universitüt München, 2007.

[10] Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239.

[11] 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 Trans. Inf. Technol. Biomed. 3*, 2 (1999), 125–138.

[12] Canero, C., and Radeva, P. Vesselness enhancement diffusion. *Pattern Recogn. Lett. 24*, 16 (2003), 3141–3151.

[13] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8* (1986), 679–698.

[14] Canny, J. F. Finding edges and lines in images. Tech. rep., Artifical Intelligence Lab, MIT, 1983.

[15] Cipolla, R., and Blake, A. The dynamic analysis of apparent contours. In *IEEE International Conference on Computer Vision* (1990).

[16] Dhillon, I. S., Guan, Y., and Kulis, B. Kernel k-means: spectral clustering and normalized cuts. In *International ACM Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2004), pp. 551–556.

[17] Dierckx, P. An algorithm for smoothing, differentiation and integration of experimental data using spline functions. *Journal of Computational and Applied Mathematics 1* (1975), 165–184.

[18] Dierckx, P. FITPACK user guide part 1: curve fitting routines. Tech. Rep. 89, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1987.

[19] Dierckx, P. *Curve and Surface Fitting with Splines.* Oxford University Press, Inc., New York, NY, USA, 1993.

[20] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271.

[21] Fleischmann, D. Contrast medium injection protocols for CT angiography. Tech. rep., Department of Radiology, Standford University, 2009.

[22] Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. Multiscale vessel enhancement filtering. *Lecture Notes in Comput. Sci. 1496* (1998), 130–137.

[23] Glocker, B., Heibel, T. H., Navab, N., Kohli, P., and Rother, C. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *European Conference on Computer Vision* (Crete, Greece, September 2010).

[24] Glocker, B., Komodakis, N., Tziritas, G., Navab, N., and Paragios, N. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis 12*, 6 (2008), 731–741.

[25] Gorges, S., Kerrien, E., Berger, M.-O., Trousset, Y., Pescatore, J., Anxionnat, R., Picard, L., and Bracard, S. 3D Augmented Fluoroscopy in Interventional Neuroradiology: Precision Assessment and First Evaluation on Clinical Cases. In *Workshop on Augmented environments for Medical Imaging and Computer-aided Surgery* (2006).

[26] Groher, M. *2D-3D Registration of Vascular Images.* PhD thesis, Technische Universität München, 2007.

[27] Groher, M., Bender, F., Khamene, A., Wein, W., Heibel, T. H., and Navab, N. 3D guide wire navigation from single plane fluoroscopic images in abdominal catheterizations. In *Bildverarbeitung in der Medizin* (2009).

[28] Groher, M., Bender, F., Khamene, A., Wein, W., Heibel, T. H., and Navab, N. 3D guide wire navigation from single plane fluoroscopic images in abdominal catheterizations. In *Bildverarbeitung in der Medizin* (2009).

[29] Groher, M., Zikic, D., and Navab, N. Deformable 2D-3D registration of vascular structures in a one view scenario. *IEEE Trans. Med. Imag. 28*, 6 (2009), 847–860.

[30] Guy, G., and Medioni, G. Inferring global perceptual contours from local features. *International Journal of Computer Vision 20* (1996), 113–133.

[31] Heibel, T. H., Glocker, B., Groher, M., Paragios, N., Komodakis, N., and Navab, N. Discrete tracking of parametrized curves. In *IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[32] Heibel, T. H., Glocker, B., Paragios, N., and Navab, N. Needle tracking through higher-order MRF optimization. In *IEEE International Symposium on Biomedical Imaging* (2010).

[33] Honnorat, N., Vaillant, R., and Paragios, N. Robust guide-wire segmentation through boosting, clustering and linear programming. In *IEEE International Symposium on Biomedical Imaging* (2010).

[34] Ibanez, L., Schroeder, W., Ng, L., and Cates, J. *The ITK Software Guide*, second ed. Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, 2005.

[35] Ishikawa, H. Higher-order clique reduction in binary graph cut. In *IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[36] Jr., T. G. S., Marks, W. B., Lange, G. D., Jr., W. H. S., and Neale, E. A. Edge detection in images using Marr-Hildreth filtering techniques. *Journal of Neuroscience Methods 26* (1988), 75–81.

[37] Kak, A. C., and Slaney, M. *Principles of Computerized Tomographic Imaging.* Society of Industrial and Applied Mathematics, 1999.

[38] Klein, J. L., Hoff, J. G., Peifer, J. W., Folks, R., Cooke, C. D., King, S. B., and Garcia, E. V. A quantitative evaluation of the three dimensional reconstruction of patients' coronary arteries. *International Journal of Cardiac Imaging 14* (1998), 75–87.

[39] Koller, T., Gerig, G., Szekely, G., and Dettwiler, D. Multiscale detection of curvilinear structures in 2d and 3d image data. In *IEEE International Conference on Computer Vision* (1995), pp. 864–869.

[40] Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 10 (2006), 1568 – 1583.

[41] Kolmogorov, V., and Rother, C. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 7 (2007), 1274 – 1279.

[42] Kolmogorov, V., and Zabih, R. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (2004), 147–159.

[43] Komodakis, N., Tziritas, G., and Paragios, N. Fast, approximately optimal solutions for single and dynamic MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[44] Krissian, K., Kikinis, R., and Westin, C.-F. Algorithms for extracting vessel centerlines. Tech. rep., Laboratory of Mathematics in Imaging, 2004.

[45] Li, S. Z. *Markov random field modeling in image analysis.* Springer-Verlag New York, Inc., 2001.

[46] Lindeberg, T. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch.* PhD thesis, Royal Institute of Technology, 1991.

[47] Lindeberg, T. On Scale Selection for Differential Operators. In *Scandinavian Conference on Image Analysis* (1993).

[48] Lindeberg, T. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision 30* (1998), 465–470.

[49] Lindeberg, T. Principles for Automatic Scale Selection. Tech. rep., Computational Vision and Active Preception Laboratory, 1998.

[50] Lorenz, C., Carlsen, I.-C., Buzug, T. M., Fassnacht, C., and Weese, J. Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2d and 3d medical images. In *CVRMed-MRCAS* (1997).

[51] Manniesing, R., and Nissen, W. Multiscale vessel enhancing diffusion in ct angiography noise filtering. *Lecture Notes in Computer Science 3565/2005* (2005), 138–149.

[52] Masutani, Y., Dohi, T., Yamane, F., Iseki, H., and Takakura, K. Augmented reality visualization system for intravascular neurosurgery. *Computer 3* (1997), 239 – 247.

[53] Masutani, Y., Dohi, T., Yamane, F., Iseki, H., and Takakura, K. Interactive virtualized display system for intravascular neurosurgery. In *CVRMed-MRCAS* (1997).

[54] Mazouer, P., Chen, T., Zhu, Y., Wang, P., Durlak, P., Thiran, J.-P., and Comaniciu, D. User-constrained guidewire localization in fluoroscopy. In *SPIE Medical Imaging* (2009).

[55] Molina, M. C., Prause, G. P., Radeva, P., and Sonka, M. 3D catheter path reconstruction from biplane angiograms. In *SPIE Medical Imaging* (1998).

[56] Mortensen, E. N., and Barrett, W. A. Intelligent scissors for image composition. In *Proc. SIGGRAPH* (New York, NY, USA, 1995), ACM, pp. 191–198.

[57] Navab, N., Traub, J., and Heibel, T. H. Method for calibration of a camera augmented c-arm. Patent No. WO2009077480 (A1), 2009.

[58] Palti-Wasserman, D., Brukstein, A., and Beyar, R. Identifying and tracking a guide wire in the coronary arteries during angioplasty from x-ray images. *IEEE Trans. Biomed. Eng. 44*, 2 (1997), 152–164.

[59] Pauly, O., Heibel, T. H., and Navab, N. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2010).

[60] Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., 1988.

[61] Piegl, L., and Tiller, W. *The NURBS book (2nd ed.).* Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[62] Platonov, J., Heibel, T. H., and Meier, P. Camera position and orientation determining method for use in e.g. production area, involves projecting data model of virtual information with preset camera parameters and finding current parameters using external values of cost-function. Patent No. DE102005025470 (A1), 2006.

[63] Platonov, J., Heibel, T. H., Meier, P., and Grollmann, B. A mobile markerless AR system for maintenance and repair. In *International Symposium on Mixed and Augmented Reality* (2006).

[64] Ramalingam, S., Kohli, P., Alahari, K., and Torr, P. Exact inference in multi-label CRFs with higher order cliques. In *IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[65] Rother, C., Kumar, S., Kolmogorov, V., and Blake, A. Digital tapestry. In *IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[66] Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., and Kikinis, R. 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In *CVRMed-MRCAS* (1997).

[67] Schneider, M., and Sundar, H. Automatic global vessel segmentation and catheter removal using local geometry information and vector field integration. In *IEEE International Symposium on Biomedical Imaging* (2010).

[68] Shechter, G., Devernay, F., Coste-Maniere, E., and McVeigh, E. R. Temporal tracking of 3D coronary arteries in projection angiograms. In *SPIE Medical Imaging* (2002).

[69] Slabaugh, G. G., Kong, K., Unal, G. B., and Fang, T. Variational guidewire tracking using phase congruency. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2007).

[70] Steger, C. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 2 (1998), 113–125.

[71] Steger, C. *Unbiased Extraction of Curvilinear Structures from 2D and 3D Images.* PhD thesis, Technische Universität München, 1998.

[72] Sutter, H. *Exceptional C++: 47 engineering puzzles, programming problems, and solutions.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[73] Tobis, J., Johnston, W., Montelli, S., Henderson, E., Roeck, W., Bauer, B., Nalcioglu, O., and Henry, W. Digital coronary roadmapping as an aid for performing coronary angioplasty. *American Journal of Cardiology 56* (1985), 237–241.

[74] Torre, V., and Poggio, T. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8* (1984), 147–163.

[75] Traub, J., Heibel, T. H., Dressel, P., Heining, S., Graumann, R., and Navab, N. A multi-view opto-xray imaging system: Development and first application in trauma surgery. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2007).

[76] Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 4 (1991), 376–380.

[77] van Walsum, T., Baert, S. A. M., and Niessen, W. J. Guide wire reconstruction and visualization in 3DRA using monoplane fluoroscopic imaging. *IEEE Transactions on Medical Imaging 24*, 5 (May 2005), 612–623.

[78] Wang, L., Springer, M., Heibel, T. H., and Navab, N. Floyd-Warshall All-Pair Shortest Path for Accurate Multi-Marker Calibration. In *International Symposium on Mixed and Augmented Reality* (October 2010).

[79] Wang, P., Chen, T., Zhu, Y., Zhang, W., Zhou, S. K., and Comaniciu, D. Robust guidewire tracking in fluoroscopy. In *IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[80] Weickert, J. Coherence-Enhancing Diffusion Filtering. *International Journal of Computer Vision 31* (1999), 111–127.

[81] Weiss, Y., and Freeman, W. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. On Information Theory 47*, 2 (2001), 736 – 744.

[82] Witkin, A. Scale-Space Filtering. In *International Joint Conference on Artificial Intelligence* (1983).

[83] Zelnik-manor, L., and Perona, P. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17* (2004), MIT Press, pp. 1601–1608.