A Model-Based Approach to Intraoperative Guidance of Flexible Endoscopy

Dissertation
zur Erlangung des Grades des
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik
von

Markus Kukuk

Princeton
2002

Tag der mündlichen Prüfung:

24. März 2003

Dekan:

Prof. Dr. Bernhard Steffen

Gutachter:

Professor Dr. Heinrich Müller

Professor Dr. Wolfgang Banzhaf

Dedicated to my father Albert

Contents

Preface

Medical imaging is one of the most fascinating fields in modern science. Its development and applications continue to evolve and expand at an accelerated pace, fuelled by advances in various disciplines, such as medicine, mathematics, physics, electrical and mechanical engineering and last but not least computer science. Modern imaging technology continues to help us understand the basic processes of life and disease by exploiting the most powerful of our senses, vision.

This dissertation represents an expedition into the exciting field of medical imaging and computer science. Starting in summer of 1998 until the summer of 2002, I worked (amongst other things) on the problem of intraoperative guidance in endoscopic procedures, a post-processing step in the acquisition of volumetric data by a CT (computed tomography) or MRI (magnetic resonance imaging) scanner.

I was supervised by Prof. Dr. Heinrich Müller, head of the "Computer Graphics" institute at the computer science department, University of Dortmund, Germany. The research was carried out in the United States of America, at the Siemens research facility "Siemens Corporate Research (SCR)" in Princeton, New Jersey. As a member of technical staff and part of a Ph.D program in the "Imaging and Visualization" department, I was fortunate enough to work in an exiting and vibrant research and development environment.

I am grateful to the following people and institutions:

Prof. Dr. Heinrich Müller,

for establishing the contact with SCR, for his long distance supervision and for his visit to SCR in Spring of 1999.

Prof. Dr. Banzhaf (University of Dortmund),
for accepting to examine this work.

Bernhard Geiger (SCR),
for valuable discussions and support.

Jim Williams, Thomas O'Donnell, Steven Dilliplane, Sathykama Sandilya,
(Siemens Corporate Research, Inc)
for "yankeefying" smaller sections of this text in between their own work.

Deutscher Akademischer Austauschdienst (DAAD),
for one year of funding in the context of a Ph.D. scholarship (HSP III financed by the German Federal Ministry of Science and Research).

Doreen J. Addrizzo-Harris M.D.,
Timothy J. Harkin M.D.,
(NYU Medical Center, Department of Medicine (Pulmonary Disease), Manhattan),
for demonstrating to me the practical side of bronchoscopy and for the medical advice.

Prof. David P. Naidich M.D.
(NYU Medical Center, Department of Radiology),
for medical advice.

My family and friends,
for their support in good and in bad times.

Natasha Coetzee,
for her love.

Markus Kukuk
Princeton, NJ
Winter, 2002

... and the journey continues.

Abstract

Endoscopy is a medical procedure, where a physician uses an optical instrument called an “endoscope” to obtain a magnified view of the inner surface of hollow organs and to access the tissue through surgical tools. An endoscope is a flexible tube, which is inserted into the body through either natural body openings or small incisions. Flexible endoscopy often results in a better outcome for the patient as opposed to open surgery. However, this technique also presents increased challenges to the physician and therefore often necessitates intraoperative guidance.

This dissertation presents a new approach to the intraoperative guidance of flexible endoscopy. It proposes to calculate a patient specific “protocol” prior to the intervention to achieve a sensor-less guidance during the procedure. This protocol prescribes in detail how to handle the endoscope and tools in order to successfully perform an endoscopic procedure. During the intervention, the physician executes the protocol, by setting endoscope and tools to the prescribed configuration.

The calculation of the protocol is based on three components: (1) A 3D model of the target anatomy derived from a CT/MRI scan of the patient, (2) a deformable model representing the endoscope and (3) a virtual endoscopy system. These components are combined to simulate an endoscopic procedures and to estimate a set of endoscope parameters. This general approach is validated for an endoscopic procedure called “Transbronchial Needle Aspiration” (TBNA), which involves the “blind” placement of a needle into a target. Based on a set of candidate shapes for the real endoscope obtained from the endoscope model, an “optimal” needle placement strategy is presented that maximizes the probability of success for TBNAs.

Notation

Chapter 1

Scalars are denoted by cursive, lowercase letters:

l, l_1, l_2	Link length, length of first sleeve, length of second sleeve
d, d_s, d_b	Link diameter, shaft diameter, bending section diameter
n	Number of links
n'	Number of links, significantly smaller than n ($n' << n$)
e	Endoscope length
s	Shaft length
b	Bending section length
n_s	Number of shaft links
n_b	Number of bending section links
u	Number of rotation axis
v	Number of discrete rotation steps
θ	Step size for rotation angle
r	Radius of minimal circle (Flexibility)
α	Resistance to bending (Material property)
β	Resistance to twisting (Material property)
p	Filter selectivity
κ	Curvature
τ	Torsion

Vectors are denoted by boldface, lowercase letters:

$\mathbf{v}$	Vector in homogeneous coordinates
$\|\|\mathbf{v}\|\|$	Norm or length of vector $\mathbf{v}$
$\hat{\mathbf{x}}$	Unit Vector in the direction of the positive $x-$ axis
$\hat{\mathbf{y}}$	Unit Vector in the direction of the positive $y-$ axis
$\hat{\mathbf{z}}$	Unit Vector in the direction of the positive $z-$ axis
$\mathbf{0}$	Zero vector
$\mathbf{t}$	Unit tangent vector of smooth curves ("Frenet" frame)
$\mathbf{n}$	Principal unit normal vector of smooth curves ("Frenet" frame)
$\mathbf{b}$	Unit binormal vector for smooth curves ("Frenet" frame)
$\check{\mathbf{t}}$	Unit tangent vector for piecewise linear curves
$\check{\mathbf{n}}$	Principal unit normal vector for piecewise linear curves
$\check{\mathbf{b}}$	Unit binormal vector for piecewise linear curves

Matrices and **n-tuple** are denoted by boldface, uppercase letters. Additionally, **n-tuple** have a bar accent:

$\mathbf{F}$	Matrix representing a reference frame (homogeneous transform)
$\mathbf{T}$	Matrix representing a translation matrix
$\mathbf{R}$	Matrix representing a rotation matrix

$\bar{\mathbf{L}}$	3-tuple representing a "link"
$\bar{\mathbf{J}}$	5-tuple representing a "joint"
$\bar{\mathbf{C}}$	4-tuple representing a "chain"
$\bar{\mathbf{T}}$	7-tuple representing a "tube"
$\bar{\mathbf{S}}$	8-tuple representing a "shaft"
$\bar{\mathbf{S}}^{\mathrm{E}}$	10-tuple representing an "elastic shaft"
$\bar{\mathbf{B}}$	3-tuple representing a "bending section"
$\bar{\mathbf{E}}$	2-tuple representing an "endoscope"

Sets are denoted by cursive, uppercase letters:

L	Set of all links
A	Set of links
C	Set of links representing a “chain-like” structure
T	Set of links representing a “tube-like” structure
S	Set of links representing a “shaft-like” structure
S^E	Set of links representing a “elastic shaft-like” structure
$J_{u,v,\theta}$	Set of all joints
D	Set of rotation axis
$P_{l,d}$	Set of link surface points

A **class** (of sets) is denoted by calligraphic, uppercase letters:

$\mathcal{L}$	Set of all sets of links
$\mathcal{C}$	Set of all sets representing a “chain-like” structure
$\mathcal{T}$	Set of all sets representing a “tube-like” structure
$\mathcal{S}$	Set of all sets representing a “shaft-like” structure
$\mathcal{S}^E$	Set of all sets representing a “elastic shaft-like” structure

Functions are denoted by cursive letters, followed by parenthesis:

$f_{\mathsf{gen}}()$	Generator, enumerating all possibilities
$f_{\mathsf{filter}}()$	Filter
$f_{\mathsf{link}}()$	Link filter
$f_{\mathsf{joint}}()$	Joint filter
$f_{\mathsf{boundingTube}}()$	Bounding tube filter
$f_{\mathsf{geometry}}()$	Geometry filter
$f_{\mathsf{energy}}()$	Energy filter
$f_{\sim}()$	Flexibility function
$E()$	Deformation energy
$E_{\kappa}()$	Bending energy
$E_{\tau}()$	Torsion energy

Chapter 2

Scalars are denoted by cursive, lowercase letters:

a	Dimension of position domain P
b	Dimension of target domain T
c	Dimension of needle parameter domain N
k	Number of needles
Δ_T	Cell size in T^Δ
Δ_N	Cell size in N^Δ
l	Dimension of column vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$
i, j, k	Subscripts of a 3D array (Algorithm “kCP_Greedy()”)
p_i	Probability of success for needle i

Vectors are denoted by boldface, lowercase letters:

$\mathbf{p}$	Point in the position domain P
$\mathbf{q}$	Point in the position domain P
$\tilde{\mathbf{p}}$	Unknown position of the real endoscope
$\mathbf{t}$	Point in the target domain T
$\mathbf{t}_{center}$	Center point of target T
$\mathbf{n}$	Point in the needle parameter domain N
$\mathbf{n}_i^c$	Center of cell i in the discrete needle parameter domain N^Δ
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$	Input column vectors to the “kCP_Greedy()” algorithm.
$\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{z}_i$	$i-$th element of vector $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, respectively

n-tuple are denoted by boldface, uppercase letters:

$\mathbf{I}$	Input to Algorithm “kCP_Greedy()”

Sets are denoted by cursive, uppercase letters:

P	Initial position domain
P^{Δ}	Set of samples from P
T	Target domain
T^{Δ}	Discretized target domain
N	Needle parameter domain
N^{Δ}	Discretized needle parameter domain
C_i	Cluster of set P
N_{naive}	A set of needle parameters ("naive" solution)
P_{naive}	A set of initial positions ("naive" solution)
N_{better}	A set of needle parameters ("better" solution)
P_{better}	A set of initial positions ("better" solution)
N_{opt}	A set of needle parameters ("optimal" solution)
P_{opt}	A set of initial positions ("optimal" solution)
$S^{T}(\mathbf{p})$	A scan of T from viewpoint $\mathbf{p}$
$S^{T^{\Delta}}(\mathbf{p})$	A scan of T^{Δ} from viewpoint $\mathbf{p}$
V_i	Set of viewpoints of cell i
U	Set of elements (Set Covering Problem)

A **class** (of sets) is denoted by calligraphic, uppercase letters:

$\mathcal{S}$	A class of subsets of U
$\mathcal{W}$	A subset of $\mathcal{S}$

Functions are denoted by cursive letters, followed by parenthesis:

$f()$	A function $f : P \times T \rightarrow N$.
$\bar{f}()$	A function $\bar{f} : P \times N \rightarrow \mathbb{R}^b$
$d()$	Euclidian distance function

Introduction

“This . . . will take you where no one has ever been before; no eye witness has actually seen what you are about to see . . .”. This is an excerpt of the prologue to the 1966 science-fiction movie “Fantastic Voyage” (German title: “Die phantastische Reise”). The movie tells the story of a crew of physicians, who are shrunk to microscopic size and, after traditional medical methods have been exhausted, injected into a patient’s body to destroy a blood clot and save his life.

This storyline reflects the dream of probably all physicians in the history of medicine. Although dramatized, it covers the three most important aspects of any surgical intervention: Seeing inside the human body for diagnosis, getting to the target site without destroying healthy tissue and, finally, fixing the problem. However, until the end of the 19th century, open surgery was the only available technique to provide visual and tactile access to inner organs. Unfortunately, this often results in damage to healthy tissue from the surgery itself, post-operational discomfort and long recovery times for the patient. The credo of this medical era has been summarized by the phrase “Big doctors – big incisions”.

With the rise of “Minimally Invasive Surgery” since the late 1980s this credo has changed today to “Big doctors – small incisions”. Surgical endoscopy, also called “key-hole surgery” plays an important role in the field of minimally invasive surgery. It is a clinical procedure where a small tube is inserted into the body through either natural body openings or small incisions. The tube, generally called “endoscope”, has an optical system (fiberoptics or CCD camera) and a light source at its tip, which allows the surgeon a continuous and magnified view of the inner surfaces of hollow organs. Furthermore, instruments like knives, scissors or needles can be passed through a channel inside the endoscope, to perform an operation inside the body. Among others, endoscopic procedures have been used for treating diseases of the lungs (Bronchoscopy), abdomen (Laparoscopy), stomach (Gastroscopy) and colon (Colonoscopy). Each procedure has its own specialized endoscope, called "bronchoscope", "gastroscope" and "colonoscope". However, essentially they only differ by length, diameter and maneuverability of their bendable tip.

This new technology, which often results in a better outcome for the patient as opposed to open surgery, brings also new challenges to the surgeon. Because of the increased number and complexity of surgical skills required to perform endoscopic surgery, compared to open surgery the chance of surgical errors is magnified. For example, the lack of direct sight and tactile feedback, together with the complicated hand-eye coordination, requires excellent education and training for the endoscopist.

Without additional guidance, endoscopy is limited to clinical applications, where the site of operation is in close proximity to cavities and furthermore, where the target can be directly seen by the endoscope's visual system. Since endoscopes are only capable of displaying the inner surface of hollow organs, they yield no information about structures within or beyond the wall. This represents a general obstacle in applying endoscopy to a variety of clinical procedures. However, a parallel development called "Virtual Endoscopy" has the potential of guiding endoscopic procedures, where a direct line of sight to the target is not possible.

Virtual endoscopy is a technique in the field of digital medical imaging. It represents a post-processing step in the acquisition of volumetric data by a CT (computed tomography) or MRI (magnetic resonance imaging) scanner. As shown in Figure 1 (left), the scanner output is a stack of gray scale images, each representing a planar axial cross section through the body. The entire stack can be regarded as a 3D scalar field, where each (gray-) value represents the tissue property of the corresponding

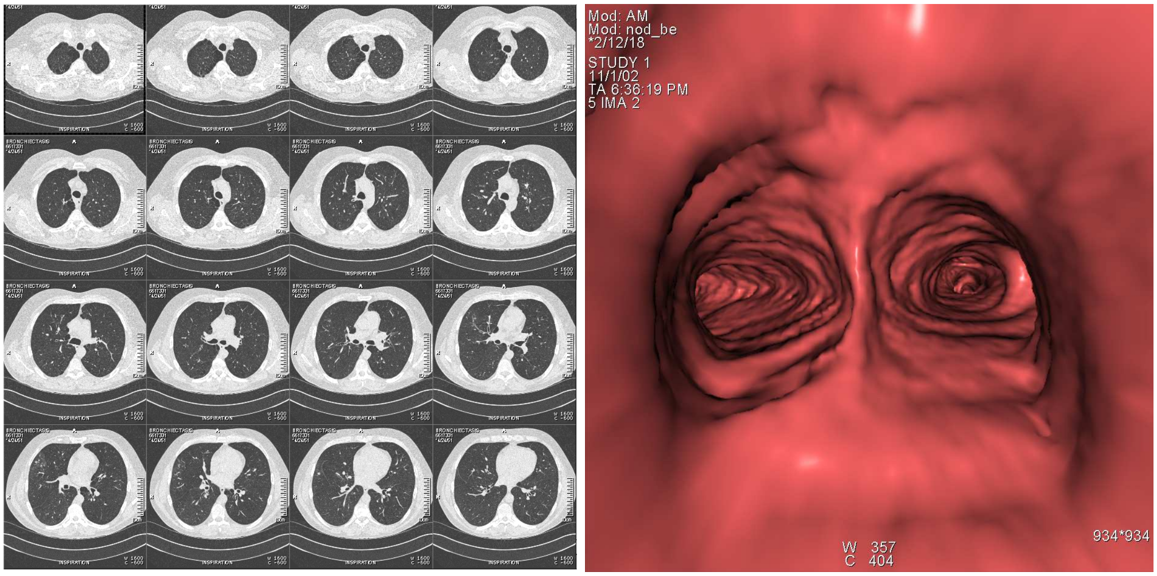

Figure 1: Chest CT scan of a 84 year old male patient. Left: 16 (out of 300) single axial cross sections. Right: Virtual endoscopic view (perspective volume rendering) from inside the lung, showing the main carina (bifurcation). *Courtesy: Siemens Corp. Research (SCR)*

small volume element of the body, called voxel. The entire dataset (or volume) is loaded into a computer and 2D perspective projections are calculated from the 3D dataset. Each 2D projection represents a synthetic or virtual view, typically displaying internal anatomical structures. Figure 1 (right) is an example of a virtual view from inside the lungs. The field was pioneered by work from Vining *et al.* '93 [1] on virtual bronchoscopy and '94 [2], [3] on virtual colonoscopy, Geiger and Kikinis '94 [4] on virtual bronchoscopy, Lorensen *et al.* '95 [5], Rubin *et al.* '96 [6] and Napel *et al.* '96 [7]. This frontier work was followed by a rapidly maturing development of virtual endoscopy, as indicated by numerous publications in the following years [8] – [18].

Today, two fundamentally different rendering techniques are used for generating 2D virtual images of volumetric datasets [19]: "shaded surface display" and "direct volume rendering". Shaded surface display [20] requires an explicit segmentation of the anatomical structure of interest. Based on the segmentation, the organ surface is approximated by a triangle mesh. This process is called surface fitting. Standard techniques for surface fitting are the "marching cubes" algorithm and "contour tracking/connecting". Then, for a given viewpoint and light source a local lightning model is applied to calculate the shading of all visible triangles, creating the impression of a 3D object. Rendering an image can usually be done in real-time (30 frames per second), due to the great availability of low-cost graphic boards, which realize the underlying rendering pipeline in hardware.

Direct volume rendering [21] is a technique for directly displaying a 3D scalar field without first fitting geometric primitives to the samples. In contrast to "shaded surface display", volume rendering is usually done in software or in some cases in hardware, found in expensive special purpose graphics boards. Ray-casting is a volume rendering technique based on a model by Blinn [22]/ Kajiya [23]. A ray is cast from the eye-point through each pixel of the 2D projection image into the volume. The value of each voxel, intersected by the ray on its way through the volume, contributes to the final color of the pixel.

In both cases, the resulting image can be regarded as the output of a virtual camera. However, due to the limited scanner resolution, the image quality is poor in comparison to an optical image. In particular, the absence of surface texture and color information represents a limitation to the diagnostic interpretation of virtual images.

Today, all major vendors of radiology equipment (Siemens, GE, Toshiba, Philips, etc.) offer a "virtual endoscopy" package as part of their product line. Virtual endoscopy is typically used by a radiologist as a diagnostic tool in addition to, or as a replacement for the standard procedure of "reading" 2D cross-sectional images (see Figure 1). Firstly, the radiologist places the virtual camera inside the anatomy. Then, he/she controls the movement of the camera, either freely, or along a precomputed smooth path. In both cases, the radiologist observes the virtual images as the camera "flies" through the anatomy / volume. The radiologist performs a visual inspection of the inner surface of hollow organs using a virtual camera, similar to an endoscopist performing an endoscopy using an endoscope, which explains the name “virtual endoscopy”.

There are several reasons for the success and widespread use of this new technology. Since the images of the virtual camera show perspective renderings of the 3D anatomy, virtual endoscopy can result in improved diagnostic yield over the reading of 2D cross-sectional images. For example, one of the most promising applications, is the detection of intestinal polyps using “virtual colonoscopy”. Polyps are malformations that are difficult to distinguish from other anatomical features on 2D cross-sectional images. The wide angle perspective projection used in virtual colonoscopy helps identifying suspicious regions. Then, the viewpoint can be placed near to an object of interest for a closer inspection.

An other advantage is that the examination time can be shorter using virtual endoscopy, especially for large volumes. Furthermore, virtual endoscopy is considered a non-invasive technique, since the patient is only involved in the scanning process, resulting in less discomfort. Besides that, its greatest advantage is its unlimited viewing control. Virtual views can be rendered from any position and with any viewing direction. Unlike real endoscopy, its movement and “optical system” is not limited to physical confines. Figure 2 shows an example of a virtual view that is impossible to achieve with conventional endoscopy. By using an appropriate rendering technique, virtual views can be created that make organ walls appear transparent and reveal the anatomy lying behind. This ability of virtual endoscopy can be used to overcome the limited viewing problem of conventional endoscopy. On the other hand, real endoscopy has an excellent image quality that includes surface texture and color and most of all, it is not only diagnostic, but also therapeutic. The property to be able to actually interact with the tissue, makes “real endoscopy” superior to virtual endoscopy.

For this reason, numerous attempts have been made in recent years, to combine

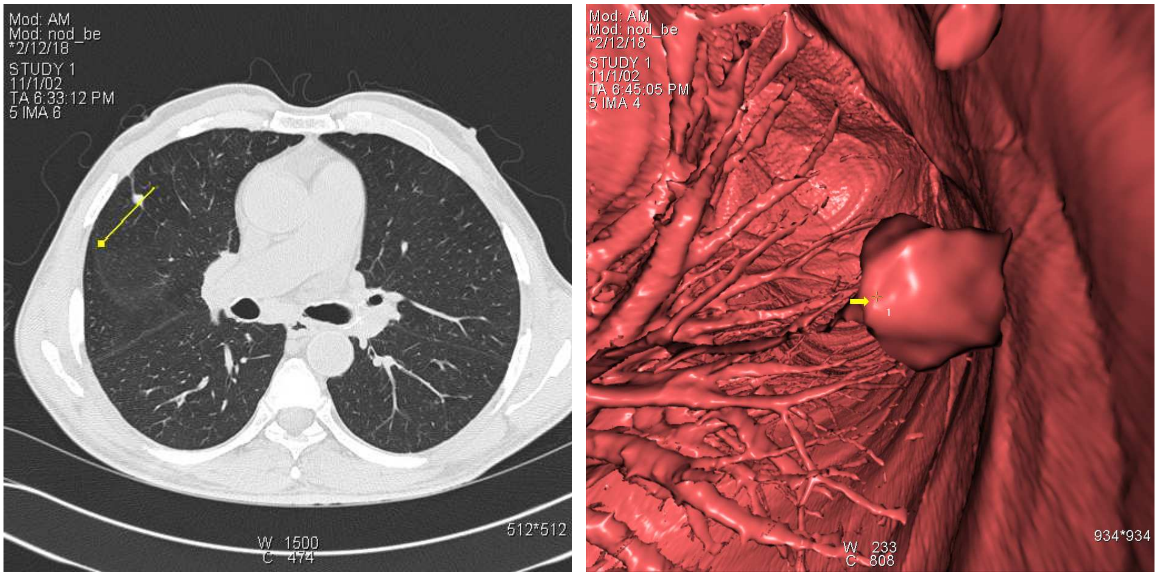

Figure 2: Same as Figure 1. Left: A cross-sectional image containing a lung nodule (top left). The yellow line indicates the position and viewing direction of the virtual camera shown on the right. Right: Virtual endoscopic view from outside the airway tree, showing a 3D rendering of the nodule. A smaller second nodule can be seen above, which is not (yet) visible in the current cross-sectional image on the left.

conventional endoscopy with the advantages of virtual endoscopy. Research interest in the field of virtual endoscopy has shifted from preoperative planing and rehearsal to intraoperative guidance. Fusion of conventional and virtual endoscopy appears to be a promising development in the field of minimally invasive surgery and might bring physicians closer to the dream of a “fantastic voyage”.

This dissertation [24] [25] [26] [27] [28] [29] [30] proposes a new method for combining standard virtual endoscopy techniques with conventional endoscopy, in order to make this minimally invasive procedure available to a wider range of clinical applications. One application that has drawn special attention in this regard is an endoscopic procedure called “Transbronchial Needle Aspiration” (TBNA). TBNA is a medical procedure that allows a physician to sample a target lesion that is located behind an organ wall and consequently not directly visible by the endoscopic camera. It is a valuable, minimally invasive procedure with an often positive outcome for the patient. As opposed to open surgery for this task, TBNA is a outpatient procedure. Unfortunately, only few bronchoscopists routinely perform TBNA, since it requires a maximum of skill and experience.

Several methods for guiding TBNA have been proposed. They can be classified into three groups: Imaging-based, vision-based and sensor-based approaches. However, most of those approaches regard the task of guiding TBNA as an application of a more general problem: the continuous tracking of the endoscope's tip. Knowing at any point in time the position and viewing direction of the endoscopic camera is used to register the camera with the corresponding virtual view. This allows one to superimpose structures from the 3D dataset onto the real endoscopic image, generating the impression of "x-ray vision". However, due to the complexity of this general task, none of the approaches found in the literature can be regarded as a mature technique to guide TBNA. Furthermore, even with "x-ray vision", the bronchoscopist still has the task of actually maneuvering the needle into the target.

This work presents a dedicated solution to the problem of guiding TBNA. It abandons the idea of a continuous tracking of the endoscope's tip and supports the bronchoscopist directly in maneuvering the biopsy needle into the target lesion. The basic idea is to calculate a "TBNA-protocol" prior to the intervention. The protocol describes in detail how to perform a number of tissue samples, by prescribing for each sample, how to handle the bronchoscope, in order to move the biopsy needle into the target lesion. Furthermore, the protocol gives probabilities of success for each tissue sample. This allows the bronchoscopist to decide whether or not the gain of an additional biopsy justifies the associated discomfort and risk of harm for the patient. During the operation, the bronchoscopist executes the protocol by setting the bronchoscope to the prescribed configuration. To gain control over the current configuration of the bronchoscope, a set of passive controls is used to monitor its degrees of freedom. The use of passive controls requires no computers or other devices in the operating room.

To calculate the protocol, the handling of the endoscope is parameterized. This set of parameters is called an “endoscope configuration”. The problem of guiding TBNA can then be formulated as the problem of estimating a set of configurations that together maximize the probability of a successful biopsy. The parameters are estimated, using a model of a flexible endoscope that takes the real endoscope’s physical and mechanical properties into account. The model calculates the real endoscope’s workspace, given a set of internal and external constraints. The workspace can be regarded as a set of candidate shapes for the (unknown) shape of the real endoscope. Having a set of candidate shapes can also be interpreted as knowing the position of the real endoscope with some error. To compensate for this error, an “optimal” strategy for placing several biopsy needles has been developed.

The outline of this thesis is as follows: Chapter 1 describes a general model for flexible instruments inserted into tubular structures. The model can be configured to represent flexible instruments, such as catheters or endoscopes. It is used to predict the shape of the real endoscope, inserted into a target branch of the tracheobronchial tree. To compensate for the model error, Chapter 2 presents an “optimal” strategy for placing several biopsy needles. Based on the results of Chapter 1 and 2, Chapter 3 describes a model-based approach of guiding transbronchial needle aspiration biopsies.

Chapter 1

A Real-Time Deformable Model for Flexible Instruments Inserted Into Tubular Structures

Flexible instruments, like endoscopes or catheters, play a central role in the field of minimally invasive surgery. They provide access to even remote operating sites within the human body through natural body openings or small incisions. However, performing endoscopic procedures or catheterizations presents a challenge to the physician. In contrast to rigid instruments, there is no direct connection between the handling of a flexible instrument and the location and orientation (pose) of its tip. Knowing approximately the pose of the instrument's tip during an intervention would greatly improve the accuracy and rate of success of minimally invasive procedures such as needle biopsies. In this chapter a model for flexible instruments is described that facilitates the estimation of the tip's pose from a given insertion depth.

The basic idea is to calculate the instrument's "workspace". In robotics, the term "workspace" describes that volume of space which the end-effector of a robot can reach. Similarly, the workspace of a flexible instrument is defined as the volume of space that can be occupied by the instrument's tip. From the instrument's perspective, the workspace is determined by its internal (mechanical) constraints, as well as by its external constraints, such as organ geometry.

If, during surgery, the current insertion depth of an instrument is known, calculating the instrument's workspace for that length provides information about possible locations of the tip. In the presence of bifurcations, for example in case of the tracheobronchial tree, the workspace describes in which branches the instrument could possibly be and which are inaccessible to the given instrument. Furthermore, if the target branch is known, for example from an insertion protocol obtained during a pre-operative planning phase, the workspace provides an estimation for the pose of the real instrument's tip.

The basic idea behind the instrument model is to exhaustively enumerate all possible shapes and to simultaneously filter them according to given mechanical and physical constraints. The instrument model is discretized and all possible combinations are recursively enumerated by creating a “Filtered Spatial Tree” (FST). During creation of the spatial tree, the shapes are filtered to impose constrains such as “minimizing the instrument's deformation energy” and “organ geometry”.

Although this brute-force approach has an exponential worst-case complexity, it is shown with a typical example that in case of tubular structures the empirical complexity is polynomial. Two approximation methods are presented that reduce this bound to an empirically linear complexity.

The instrument model presented in this chapter can be configured to represent catheters or endoscopes. For the first, the length, diameter and flexibility of the catheter are the crucial parameters. For the latter, these three parameters are needed to describe the endoscope's shaft, plus a set of parameters which describe the endoscope's bendable tip.

The outline of this chapter is as follows: In Section 1.1 the physical structure of catheters and endoscopes are described and requirements regarding their modelling are derived. In the following Section 1.2 models for flexible instruments as found in the literature are presented. Based on this description, their shortcomings regarding the requirements compiled in the previous section are pointed out. Section 1.3 gives an overview of our model for flexible instruments, which fulfills all requirements. A high-level description of the model is given to facilitate the understanding of its components and their interaction. In the following Section 1.4 the model is described mathematically and algorithmically in detail. Section 1.5 describes a technique to validate the model. The idea is to determine the centerline of a real endoscope by tracking a set of markers attached to the instrument. This technique is then used to determine the model accuracy by comparing the shape of a real endoscope inserted into a calibration phantom with the shape of the instrument model inserted into a model of the phantom. In the last Section 1.6, results are presented in form of several examples of virtual catheters and endoscopes inserted into various anatomies. The run-time of the algorithm was measured for three insertion simulations.

1.1 Model Requirements

In this section, requirements for a model for flexible instruments, such as endoscopes and catheters are compiled. As an example of such instruments, Figure 1.1 shows a flexible video endoscope (left) and a set of five catheters (right). For a detailed description of the design and internal structure of flexible endoscopes, see Chapter 5 “The Flexible Bronchoscope” in [31]. In comparison to catheters, endoscopes are more sophisticated structures. A catheter is a flexible tube with a circular cross section of usually less than one millimeter diameter. It is mainly inserted into the vascular system for the treatment of abnormalities. To maneuver them into remote places inside the body, catheters come equipped with specific, preshaped tips for specific anatomical configurations of blood vessels.

An endoscope is a more versatile surgical instrument. It uses either fiberoptics or a CCD camera together with a bendable tip, to allow the surgeon to navigate inside the human body. As shown in Figure 1.2 (left), an endoscope consists of four basic components: the control head, the flexible shaft of circular cross section (insertion tube), the biopsy (instrument) channel and the bending section.

The angle of tip deflection is controlled by pull-wires running all the way through the endoscope’s shaft to the bending wheel, located at the control head. In order to control the endoscope’s path, to “look-around” or to guide a biopsy needle into a target, the surgeon turns the bending wheel to actively bend the endoscope’s tip. To facilitate accurate navigation, endoscopes show virtually no lag between twisting the endoscope head and the rotation of the optical system in the tip. This is achieved by using a torsionally stiff hull for the endoscope shaft.

The endoscope shaft is an inhomogeneous structure. It contains cables for the CCD camera (or optical fibre bundles), optical fibre bundles for the light source, pull-wires for the tip deflection, a hollow instrument channel and an air/water channel.

As shown in Figure 1.2 (right), the transition of the shaft into the bending section is realized by a rigid sleeve, marked by ’1’. This is followed by a chain of links that

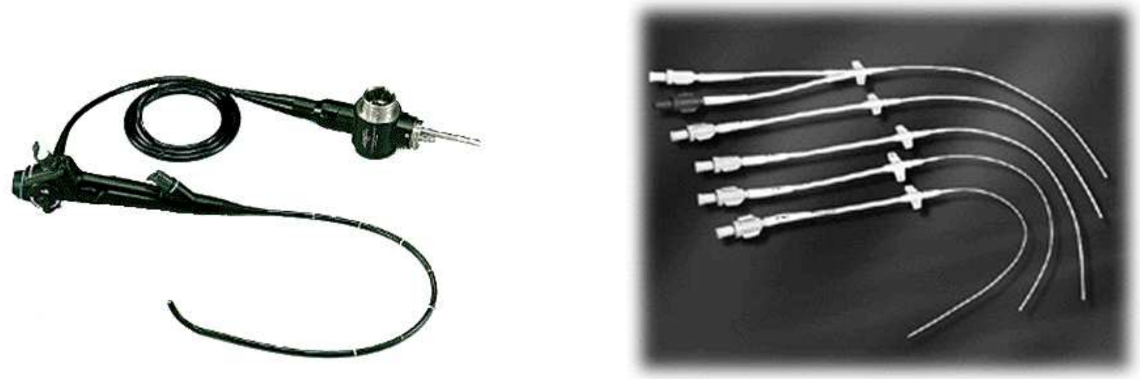

Figure 1.1: Left: A flexible video endoscope. Right: Five catheters.

allows the active bending of the tip ('2'). Finally, this chain connects to another rigid sleeve that contains the optical system and light guide ('3'). Figure 1.2 (right) also shows that the bending section has a greater diameter than the endoscope shaft.

In terms of modelling flexible instruments, catheters can be regarded as a subset of endoscopes. The catheter's rigidly shaped tip and its sub-millimeter cross-sectional diameter, which can be approximated by a infinitely thin line, facilitate its modelling. The goal of the remainder of this chapter, is to find a model that can be configured to represent either catheters or endoscopes. Consequently, the requirements for a flexible instrument model can be derived from a flexible endoscope.

A typical example of a flexible endoscope is the OLYMPUS[1] GIF-100, a gastrointestinal CCD videoscope. Its most important features and mechanical properties are listed in Table 1.1.

As can be seen from the table, the GIF-100 has 1.5 mm differential between shaft and bending section diameter. Only about one third of the overall bending section length is flexible, the rest consists of rigid sleeves. The material property of the shaft is elastic and its degree of elasticity can be described as " ductile". When bend out of

---

[1] Olympus Optical Co., LTD., San-Ei Building, 22-2, Nishi Shinjuku 1-chome, Shinjuku-ku, Tokyo, Japan; URL: http://www.olympus.co.jp/indexE.html

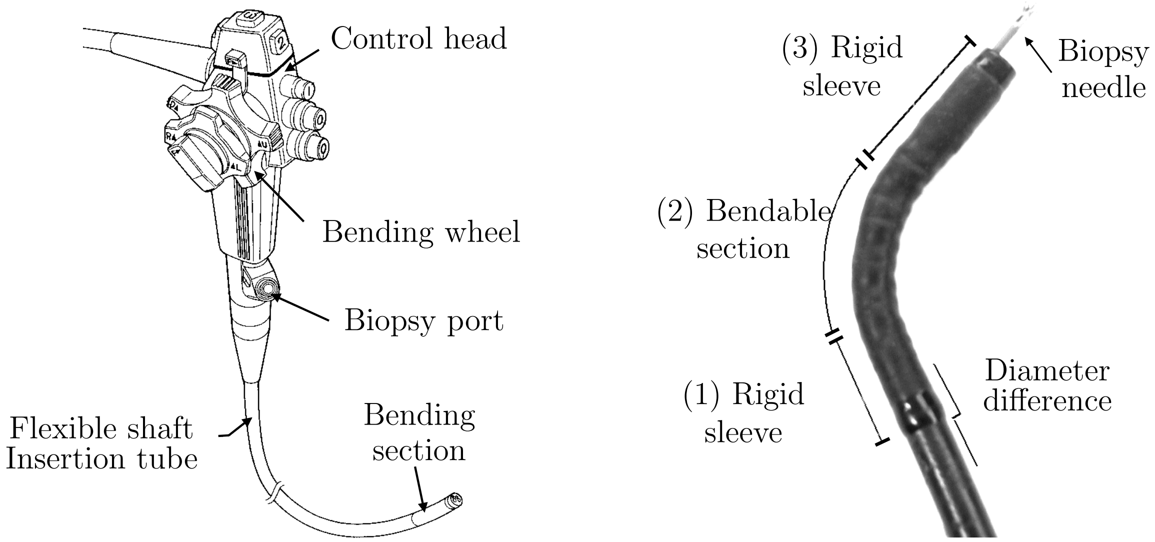

Figure 1.2: Left: Basic components of a flexible endoscope (from the “OLYMPYS GIF-100 Instruction Manual”). Right: Magnification of the endoscope’s bending section (photograph).

a fully straight position, it tries to recover its original shape but the restoring forces are not always strong enough to achieve a full recovery. A simple test confirms the high torsional stiffness of the endoscope’s shaft. By holding the endoscope tip in one hand and the control head in the other, the same amount of rotation (twist) applied to the control head can be felt at the tip. The maximum flexibility was determined by forming the smallest possible circle without damaging the shaft. The radius of this circle is regarded as a measure for the shaft’s maximum flexibility.

The measures and properties listed in Table 1.1 represent the model parameterization. Therewith, the model requirements are given by:

Input:

1. The mechanical properties of a real endoscope, as listed in Table 1.1.
2. A patient specific model of the target organ, derived from a preoperative scan.
3. A length l that describes the endoscope’s insertion depth into the target organ.

Shaft		Bending section	
Length:	100 cm	Length:	95 mm
Diameter:	9.5 mm	Diameter:	11 mm
Material property:	elastic	Sleeve length ’1’:	25 mm
Elasticity (stiffness)	“ductile”	Bendable part ’2’:	35 mm
Torsional rigidity	“high”	Sleeve length ’3’:	35 mm
max. flexibility:	ca. 50 mm	Bending range:	$-90°$ to $+90°$

Table 1.1: Relevant mechanical properties of the OLYMPUS GIF-100 Videoscope.

4. A region of interest (needed, if the insertion depth alone is ambiguous, for example in the existence of bifurcations).

Output: A set of all possible shapes the real endoscope can take (workspace) under the constraints that the shapes are of length l, reach the region of interest and remain inside the target organ.

1.2 Related Work

Modelling and simulating real world objects and their interaction with the environment leads often to a trade-off between physical correctness and computational complexity. In this chapter, three approaches for modelling flexible instruments are presented which build on physically-based modelling but introduce simplifications or assumption to reduce the computational complexity.

Deformable models for catheter simulation have been described by van Walsum *et al.* [32] and Anderson *et al.* [33], [34]. In both cases, the application is to build a simulator for the training of vascular catheterization procedures to facilitate improved education of medical students. A deformable model for endoscopes has been described by Ikuta [35] [36] as part of a virtual colonoscopy simulator with force sensation.

The catheter model described by van Walsum *et al.* [32] uses a snake based approach. It minimizes the sum of internal energy, the deformation of the catheter, and the external energy, caused by vessel deformation. The catheter is represented by a forth order B-spline. To find a minimum energy representation, the control points of the B-spline are moved during an optimization process. The optimization method used is Powells method, whose efficiency strongly depends on the initialization of the unknown parameters (control points). The authors propose to place the control points along the central lumen line of the vessel, as a good initial guess for the parameter initialization. The central lumen line of the vessel is believed to be close to the final shape of the catheter. To avoid clustering and spreading of the control points during the optimization, the authors restrict the motion of the control points to a plane which is locally orthogonal to the spline.

The advantage of this approach is its elegant and physically-based combination of instrument and organ deformation. However, the described approach is not suitable for modelling general flexible instruments, such as endoscopes for the following reason: The model approximates the instrument's diameter by an infinitely thin line. This might be justified for thin catheters but represents a problem for endoscopes with an up to 10 mm shaft diameter. Secondly, a "good initial guess" for the final shape of the instrument is not necessarily available for arbitrary target organs. Even in the case of tubular structures, finding the medial axis is a difficult and computationally expensive problem. Thirdly, it is not clear how to incorporate the rigid sleeves and the portions of higher flexibility of the bending section into the model. Finally, restricting the degrees of freedom of the control points to a plane, appears to be an artificial constraint.

The catheter model described by Anderson *et al.* [33], [34] is part of a real-time interactive simulator for vascular catheterization procedures called daVinci (**Vi**sual **N**avigation of **C**atheter **I**nsertion). The catheter is modelled using the Finite Element Method (FEM). It is discretized into a finite number of 3D beam elements (FEM nodes). In FEM analysis, computing the interaction between the model and its environment is the most time consuming step. Therefore the authors use a pre-computed potential field to speed up the computation of the contact forces between the catheter and the vessel walls. The blood vessels are modelled as rigid cylinders of varying diameter. The potential field is defined as a sparse grid, which embraces the vessel. For each grid point, a vector $\mathbf{v}$ pointing from the grid point to the vessel's center line is computed. The contact force f at a grid point is given by $f = c\mathbf{v}$, where c is a material coefficient. The contact force at an FEM node is then given by trilinear interpolation of the contact forces of the eight grid points surrounding the node. By storing the cylinder radius r for each grid point, collision of a FEM node with the vessel wall can be determined by evaluating if $r > ||\mathbf{v}||$.

The advantage of this approach is that it is build on a physically-based FEM foundation. The contact force model allows collision detection and collision response calculations by simple table look-up and thereby in real-time. However, the approach highly idealizes the interaction of the catheter with the vessel wall and the vessel geometry. The contact force simply pushes the FEM node towards the center of the

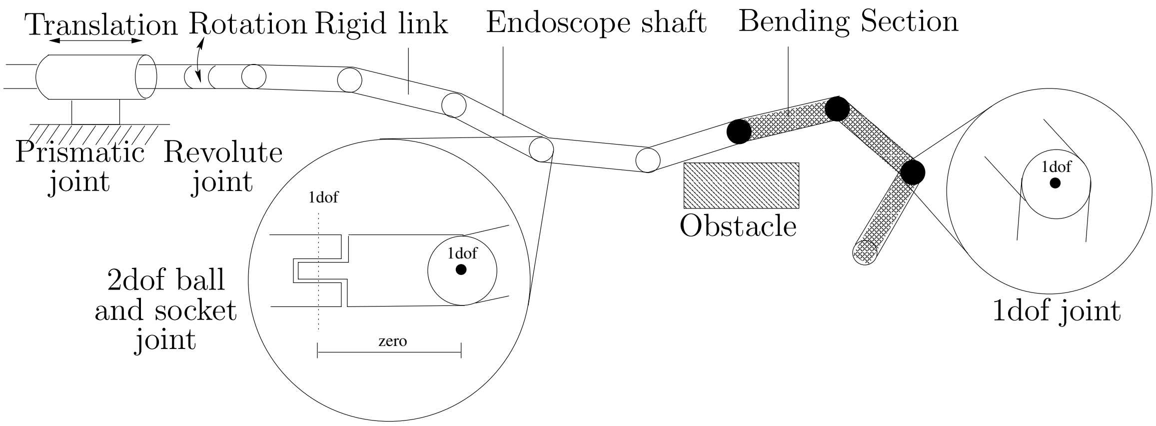

Figure 1.3: A multibody model for flexible instruments. A two degree of freedom (dof) ball and socket joint is modelled as two 1 dof joints.

vessel, which is a stylized (not physically based) model. Furthermore, approximating vasculature by cylinders is not suitable for an analysis of patient specific anatomy.

Another physically-based model for flexible instruments is based on "multibody systems". Multibodies are collections of rigid bodies interconnected with joints that are used in robotics to control industrial robots such as articulated mechanical manipulators [37]. Nowadays, dynamic multibody systems are used to simulate among others the locomotion of a human biped [38] or a ten-pin bowling throw [39]. In both cases complex interactions of the multibody with its environment are simulated, using collision detection and collision response algorithms.

As shown in Figure 1.3 an endoscope can be modeled as a linear collection of cylindrical, rigid links interconnected by joints [35]. The endoscope shown consists of two kinds of joints. Passive joints and active joints ("actuators"), which, as opposed to passive joints, exert a force or torque. The passive joints are ball and socket joints which have two degrees of freedom (dof), modelled as two 1-dof-joints with a zero distance. The active joints are 1-dof-joins that exert the forces externally applied to the instrument. For example, a prismatic (translation) and revolute (rotation) joint at the beginning of the instrument model the physician's insertion / withdrawal and twist maneuvers. Also, the deflection of the endoscope's tip is modelled by active tip joints.

To achieve a physically-based dynamic simulation of the manipulator the "forward dynamics problem" has to be solved:

Forward Dynamics Problem:

> Given: The positions and velocities of the n joints of a multibody, the external (contact) forces acting on the body, and the forces and torques being applied by the joint actuators.
>
> Find: The resulting accelerations of the joints.

By integrating the resulting acceleration forward in time, the velocity and position of the links can be found. Note that for the forward dynamics problem the contact forces are assumed to be given.

There are basically two different approaches to solve the forward dynamics problem for multibodies. The first is the $O(n^3)$ Newton-Euler [37], [40] algorithm that explicitly builds the mass matrix for the system and must invert it to solve for joint accelerations. The second is Featherstone's $O(n)$ algorithm [41] [39].

The endoscope model described by Ikuta *et al.* [35] [36] is based on multibody systems. The authors solve the forward dynamics problem by using a simplified Newton-Euler equation of motion. To achieve a real-time simulation, the equation is simplified by omitting inertia, centrifugal and coriolis forces. Since the computational complexity is dependent on the number of links, the authors propose to increase the link length from the tip to the control head of the endoscope. The colon is represented by viscoelastic cylinders of varying diameter. The authors present a contact force model, which is applied to the center point of each joint. Collision between this point and the colon is determined by comparing the distance between this point and the center line of the colon with the cylinder radius. The contact force is calculated using the joint's velocity vector considering static and kinematic friction. The presented experimental results show an endoscope approximated by five links, inserted into a synthetic colon model.

The advantage of this model lies in its physically-based approach. However, the endoscope model shown does not appear realistic due to its small number of links with increasing length. Furthermore, the model does not take the special mechanical constraints of the endoscope's bendable tip into account.

1.3 A Deformable Model for Flexible Instruments (Overview)

In this section, an overview of the endoscope / catheter model is given (see Section 1.4 for a detailed description). At first, the general concept behind the model is described and its basic components and their interactions are outlined. This description should be regarded independently from a possible implementation. Then, in Section 1.3.1 a new algorithm (Filtered Spatial Tree, FST) is introduced as a possible implementation of this concept. The empirical complexity of the FST-algorithm is analyzed. Based on this analysis, two approximative variants of the algorithm are presented in Section 1.3.2, which reduce the FST's empirical complexity from polynomial to linear. Together, both approximations represent the ∼FST-algorithm.

The system for modelling flexible instruments described in this section is based on the following three basic components:

1. A discrete representation of the instrument.
2. A generator that enumerates, based on (1), all possible shapes the instrument can take.
3. Filters that select only those shapes that respect the instrument's mechanical (internal) and physical (external) constraints. Some filter functions are invoked during the enumeration process (within the generator), whereas others are applied to the output of the generator.

(1) A natural way to discretely represent a flexible tube-like structure is as a chain of rigid links, interconnected by "discrete ball and socket joints". A link is represented by a cylinder of certain length and diameter. A joint connects two adjacent links. If the motion of a link with respect to its predecessor is restricted to two degrees of freedom, it is called a "ball and socket joint". If furthermore, a ball and socket joint

facilitates only a finite number of positions (joint positions), it is called a "discrete ball and socket joint". This representation can be regarded as a discrete variant of the multibody model, introduced in Section 1.2.

(2) Based on this representation, the generator enumerates all shapes the instrument can take. Mechanical constraints, such as varying shaft diameter, rigid sleeves and maximum flexibility are directly enforced by filter functions. A "link filter" determines for each link a suitable diameter and length. A "joint filter" determines for each joint a suitable maximum joint range. The maximum flexibility is defined by the diameter of the smallest circle (loop) that the instrument can form (see also Table 1.1 on Page 15). This constraint determines the link length and joint range of the instrument's flexible shaft.

(3) All shapes generated in (2) are filtered according to the instrument's mechanical (internal) and physical (external) constraints. Internal constraints comprise stiffness, torsional rigidity, shaft diameter or maximum flexibility. External constraints comprise the organ wall or a region of interest, defined by the user. While some filters are applied at an early stage during the enumeration process ("link filter" and "joint filter"), others can only be applied to a complete candidate shape. For example, material properties like stiffness and torsional rigidity are enforced by an "energy filter". This filter calculates the deformation energy for each complete shape and only shapes that are below a given threshold pass the filter. The threshold represents the filter's selectivity in the sense that a selectivity of "1" outputs only the shape of "lowest" deformation energy, a selectivity of "2" outputs only the two shapes of lowest deformation energy, and so on.

Model Components:

Let L denote the set of all links, L^n the set of all link sequences of length n and $\mathcal{P}(L^n)$ the power set of L^n. The above introduced concept can formally be described as the concatenation of two functions in the order from right to left:

$$f_{\mathsf{filter}} \circ f_{\mathsf{gen}} \,, \tag{1.1}$$

with $f_{\mathsf{gen}} : L \rightarrow \mathcal{P}(L^n)$ the generator that takes repeatedly single links form L and assembles them to a set of link sequences and $f_{\mathsf{filter}} : \mathcal{P}(L^n) \rightarrow \mathcal{P}(L^n)$ the filter. Algebraically, the generator can be described as the concatenation of two filter functions operating on L:

$$f_{\mathsf{gen}} = f_{\mathsf{joint}}^{u,v,\theta} \circ f_{\mathsf{link}}^{n} \,, \tag{1.2}$$

with n the overall number of links (desired instrument length). Filter $f_{\mathsf{link}}^{n} : L \rightarrow \mathcal{P}(L^n)$ is initialized with a fixed start link $\mathbf{s} \in L$ and assembles the link sequences, while controlling the length and size constraints. Filter $f_{\mathsf{joint}}^{u,v,\theta} : \mathcal{P}(L^n) \rightarrow \mathcal{P}(L^n)$ controls the instrument's maximum flexibility by constraining the joint's maneuverability. Parameters u, v, θ describe the flexibility of a discrete ball and socket joint, by specifying the allowed movement of a link with respect to its predecessor.

The filter is given as a concatenation of a geometry, tube and energy filter:

$$f_{\mathsf{filter}} = f_{\mathsf{energy}}^{\alpha,\beta,p} \circ f_{\mathsf{boundingTube}} \circ f_{\mathsf{geometry}} \,, \tag{1.3}$$

with α, β two material constants for bending and torsion and p the filter selectivity. Filter $f_{\mathsf{geometry}} : \mathcal{P}(L^n) \rightarrow \mathcal{P}(L^n)$ filters those links that collide with the organ wall and $f_{\mathsf{boundingTube}} : \mathcal{P}(L^n) \rightarrow \mathcal{P}(L^n)$ represents a simple bounding tube filter, which based on an insertion protocol defines a region of interest (ROI) in case of the existence of bifurcations. The bounding tube filter "cuts-off" side branches to guide the instrument model directly towards the specified target branch.

Finally, the energy filter $f_{\mathsf{energy}}^{\alpha,\beta,p} : \mathcal{P}(L^n) \rightarrow \mathcal{P}(L^n)$ sorts all remaining link sequences, respectively shapes by their deformation energy and outputs the p shapes of minimal deformation energy. Note that for $p = 1$ the energy filter finds the global minimum of the instrument's deformation energy:

$$f_{\mathsf{energy}}^{\alpha,\beta,p}(I) = \{A \in I \mid (E_{\kappa}(A) + E_{\tau}(A)) = \min_{p}\} \,, \tag{1.4}$$

with $\min_p$ reading "among the p smallest values", $E_\kappa()$ the internal bending energy and $E_\tau()$ the internal torsion energy of an instrument. The two discrete energy terms are given by:

$$E_\kappa(A) = \alpha \sum_{i=1}^{n-1} \kappa(A,i)^2 \qquad \text{and} \qquad E_\tau(A) = \beta \sum_{i=1}^{n-1} \tau(A,i)^2 \,, \tag{1.5}$$

with α the amount of resistance to bending, $\kappa(A,i)$ the angle between link i and $i+1$ of an instrument A, β the amount of resistance to twisting and $\tau(A,i)$ the torsion between link i and $i+1$ of an instrument A.

Justification for the Energy Filter:

The physical basis for the energy filter is the theory of elasticity, see Chapter 5 "Statics of Elastic Bodies" in [42]. As noted in Section 1.1, an endoscope is to some extent an elastic structure. Elastic structural materials have the ability to regain their original shape after a load is removed. According to Hooke's law, stress and strain are proportional in elastic structures. The quotient (stress / strain) is a constant which is denoted as the "Modulus of Elasticity" or "stiffness". Given the modulus of elasticity, possible deformations can be calculated for any material and loading.

However, for the design of the energy filter, a different problem has to be considered. Given a set of shapes as enumerated by the generator, which one is likely to be the shape of an elastic structure, like an endoscope?

The potential energies of elastic bodies restore a deformed shape to its original shape. Elastic bodies want to reach an energy equilibrium by minimizing the deviation of their actual shape from their original shape. Thus, the potential energy should be zero when the body is in its natural state, and the energy should grow larger as the model gets increasingly deformed away from its natural state. Therefore, the energy filter was designed to let pass shapes of minimal deformation energy only.

However, regarding a flexible instrument as a perfect elastic structure is an idealization, especially for the instrument's tip. The internal forces that straighten the instrument are less at the tip than in the middle of the shaft. Therefore, friction has a considerable influence on the shape of the instrument's tip. Since frictional forces inside the human body can vary over time and space they are hard to model. Instead, some degree of uncertainty regarding the exact tip location has to be accepted but should be minimized. One approach is to compute a set of possible tip shapes that "cover" the uncertainty. The "hope" is that the real tip position is close to one element of the set. A natural way to expand the solution set of the model, is to relax the selectivity of filter f_{energy} so that it determines the shapes of the $p > 1$ smallest energies, rather than just one shape of minimal energy.

1.3.1 The Filtered Spatial Tree (FST)

This section gives an overview of a possible implementation of the concept introduced in the previous section. A detailed description of the FST algorithm is given in Section 1.4.3.

The previous section introduced the concept of modelling a flexible instrument by enumerating all possible shapes and filtering the result according to given constraints. A natural way to implement this concept is to recursively create a spatial tree, whose growth is constrained by a set of filter functions. A spatial tree is an ordinary tree data structure, where each node represents a joint in 3D space. Each edge that connects a node with its child represents a link in 3D space. Each path from the root to a leaf represents a chain of links interconnected by joints and therewith is a discrete multibody representation of a flexible instrument. The entire spatial tree (so all paths from the root to the leaves), represents the instrument's full workspace under the given constraints. A spatial tree can be created by a "depth-first spanning tree" algorithm, which recursively attaches links until it can proceed no more and then unwinds (backtracking) to a previous state where it picks a different link and again starts attaching links to it, and so on. Figure 1.4 shows a selection of paths of a filtered spatial tree.

A link is represented by a cylinder, which is attached to a coordinate system or

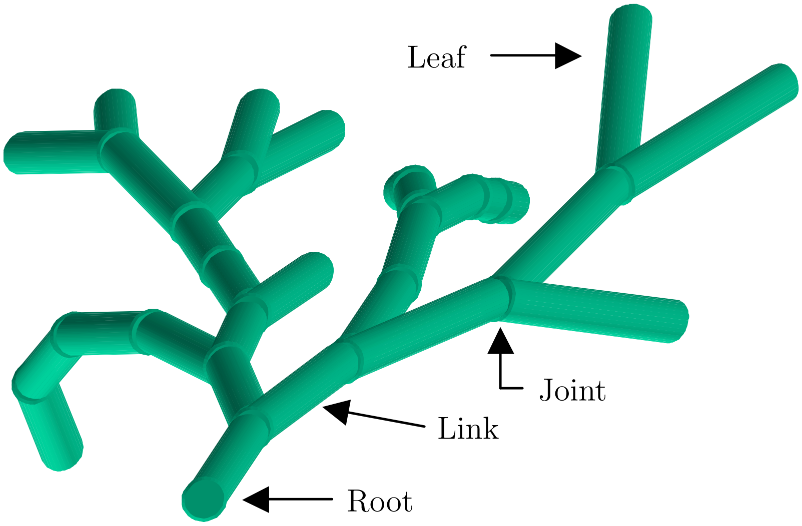

Figure 1.4: Selected paths from the root to the leafs of a filtered spatial tree (FST).

reference frame. The reference frame is represented by a 4 × 4 matrix (homogeneous transform). In other words, the position and orientation of a link in 3D space is described by a matrix, denoted by $\mathbf{F}$.

A joint is defined by the movements, a link can make with respect to its predecessor. Starting from two aligned links (zero degree angle), parameter u describes the number of principal directions in which the second link can be moved. An example for $u = 4$ could be "up", "down", "left" and "right". Parameter v denotes the number of steps a link can be moved in each direction, and θ describes the rotation angel for each step. Figure 1.5 (left) shows an example, where $u = 8$ and $v = 1$. The number of possible positions between two adjacent links is $uv + 1$, where the '+1' contributes for the "straight" (not moved) link, which is the default direction. The joint range is $v\theta$.

An algorithm $f_{\mathsf{gen}}()$ that creates a spatial tree, takes a link $\mathbf{F}$ as input, attaches $uv + 1$ links to $\mathbf{F}$ and recursively calls $f_{\mathsf{gen}}()$ for each attached link:

$$f_{\mathsf{gen}}(\mathbf{F}) = \{\mathbf{F} \circ \mathbf{A} \mid \mathbf{A} \in f_{\mathsf{gen}}(\mathbf{F}\,\mathbf{T}\,\mathbf{R}_{i,j}) \cup \{\mathbf{F}\,\mathbf{T}\} \;\text{ for } i = 1, \ldots, u\,, j = 1, \ldots, v\}\,, \quad (1.6)$$

with $\mathbf{A}$ a link described by a matrix, $\mathbf{T}$ a translation matrix that moves $\mathbf{F}$ along its

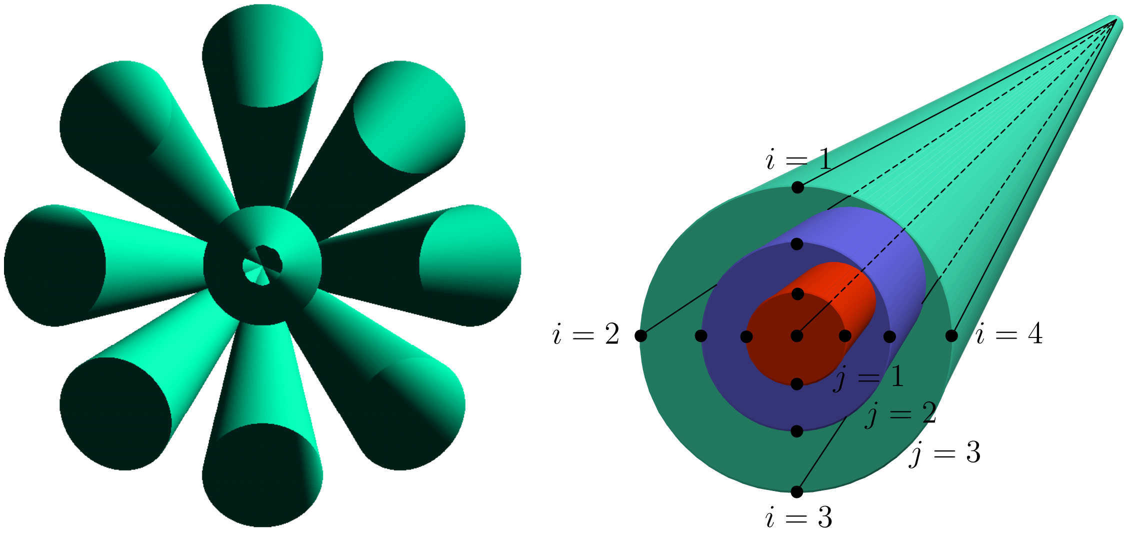

Figure 1.5: Left: A joint of the FST (frontal view), with $u = 8, v = 1$. Each cylinder represents a link in one of 9 $(uv + 1)$ joint positions. Note that the center link corresponds to a straight joint position, with a zero degree angle between two adjacent links. Right: A joint of the FST, with $u = 4, v = 3$. Only the center lines of the links, lying on the surface of concentric cones are depicted.

main axis towards its end and $\mathbf{R}_{i,j}$ a rotation matrix that rotates $\mathbf{F}$ into the new position. '$\circ$' denotes the concatenation of two links (matrices). For details regarding the transformation $\mathbf{F}\,\mathbf{T}\,\mathbf{R}_{i,j}$, see Definition 1.4.4 on Page 45. Set $\{\mathbf{F}\,\mathbf{T}\}$ contributes for the special case of two straight links, where a translation only is performed. The rotation matrix is given as an entry in a pre-computed look-up table of u columns and v rows:

$$\mathbf{R}_{i,j} = \mathbf{R}(\mathbf{u}_i, j\theta) \text{ for } i = 1, \ldots, u\,, j = 1, \ldots, v\,, \tag{1.7}$$

with $\mathbf{R}()$ a rotation matrix, which causes a rotation of $j\theta$ degrees about the i-th rotation axis $\mathbf{u}_i$.

For example a set of 8 rotation axes can be found ($u = 8$ and $v = 1$) that produces a joint as shown in Figure 1.5 (left). A perspective rendering of a frontal view of 9 links $(uv + 1)$ attached to a single link (occluded) is shown. The link in the center corresponds to an attached link, which was translated but not rotated. Figure 1.5 (right) shows an example for $u = 4$ and $v = 3$. The center lines of all links $(i = 1 \ldots u)$,

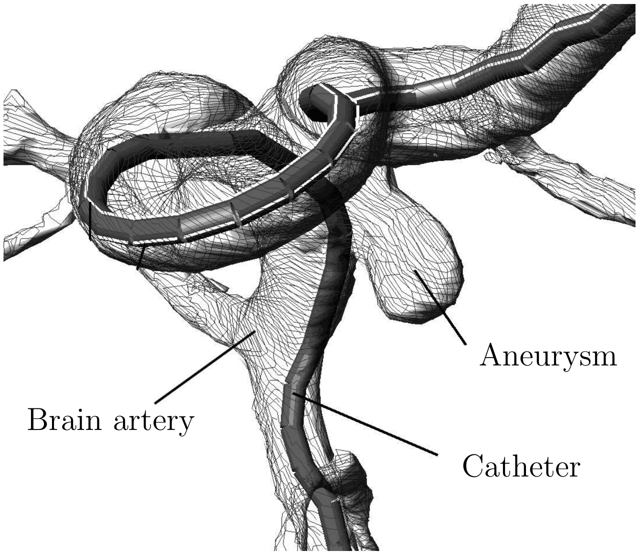

Figure 1.6: Simulated insertion of a catheter into a brain artery. The catheter was approximated by 31 rigid links.

which correspond to the same rotation angle $j\theta$ lie on the surface of a cone. Different rotation angles $j = 1 \ldots v$ correspond to different cones with concentric bases.

In order to prune the spatial tree during creation as much as possible, the filter function $f_{\mathsf{filter}}()$ is placed inside the recursion, so that e.g. colliding links get pruned immediately. For example, if filter $f_{\mathsf{geometry}}()$ determines a collision with the organ wall, this branch is closed by not calling the recursion for the link. Similarly, a branch of the spatial tree is closed for filter $f_{\mathsf{energy}}()$, once the current energy is greater than the current energy minimum.

The look-up table for the rotation matrices is pre-computed, so that a rotation can be done with one 3×3 (only the rotation sub-matrix) matrix multiplication. This multiplication and the collision detection are the most expensive operations within the recursion.

Complexity:

The time and space complexity of the FST-algorithm is $O((u\,v + 1)^n)$. However, depending on how much the growth of the spatial tree is constrained by the filter

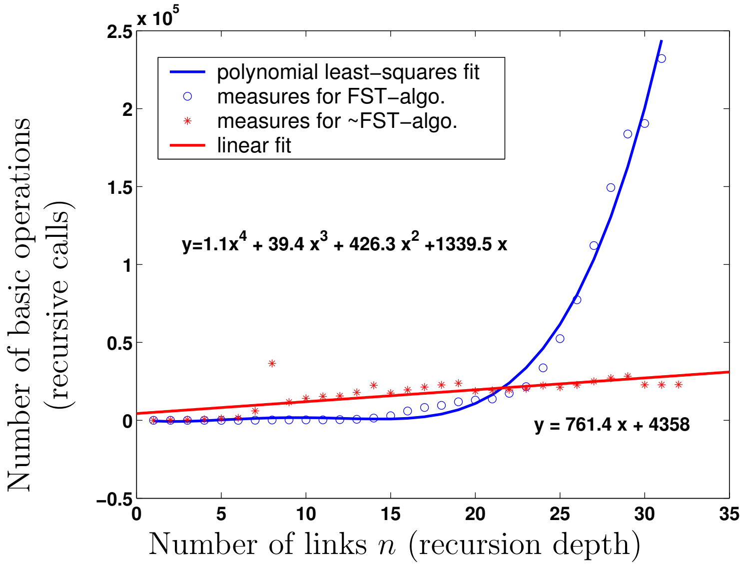

Figure 1.7: Empirical complexity of a catheter inserted into a brain artery. A fourth-order polynomial was fitted to the observed data from the FST-algorithm. A linear fit was found for the data from the ∼FST-algorithm.

functions, the practical complexity, given a real anatomy, is more feasible. Especially tubular structures, such as the tracheobronchial tree and the vasculature greatly limit the growth of the tree. The following experiment confirms this hypothesis:

Experiment 1.1: Complexity of the FST-Algorithm

Objective: *Measuring the empirical time and space complexity of the FST-algorithm.*

Method: *Simulating the insertion of a catheter into a model of a brain artery using the FST-algorithm. For the time complexity, the number of basic operations subject to the insertion depth is measured. Each recursive call is regarded as a basic operation. The space complexity is of the same order of magnitude than the time complexity, since each node of the spatial tree stores a constant amount of information.*

Platform: *PC, Pentium 4 dual, 1.3 GHz, 1 GB; Graphics: nVIDIA GeForce3, 64 MB.*

Material: *Model of a brain artery, reconstructed from planar parallel cross sections*

([43], [44], NUAGES), obtained from a patient specific 3D rotational angiography (C-arm) scan. Number of triangles: ca. 25000.

Design: *For a fixed starting position the flexible instrument model ($u = 8, v = 1$) was used to calculate a catheter of length* n*, inserted into the brain artery (Figure 1.6). This procedure was repeated 31 times, for* $n = 1, \ldots, 31$*. For each calculation the number of recursive calls needed to enumerate all possibilities, given the geometry and energy filter, was recorded.*

Results: *The result is shown in Figure 1.7. A fourth-order polynomial (*$y = 1.1x^4 - 39.4x^3 + 426.3x^2 - 1339.5x + 538.7$*) can be fit (least squares) to the resulting curve, indicating that the empirical complexity for this anatomy is* $O(n^4)$*. The second pair of curves shows the complexity after activation of two approximation methods (see experiment 1.2) described in the following section.*

Observation: *The first 17 links were computed in less than one second.*

1.3.2 The Approximated Filtered Spatial Tree (∼FST)

In the previous section a typical example was given, demonstrating that the empirical complexity of the FST-algorithm is polynomial ($O(n^4)$) for tubular structures. In this section two approximative variants of the FST-algorithm are presented, which together further reduce the empirical complexity to a linear bound. The new algorithm is called “The approximated Filtered Spatial Tree (∼FST)”.

(n', k)-Approximation:

Experiment 1.1 lead to the observation that for a number n' of links, considerably smaller than n,

$$n' << n\,, \tag{1.8}$$

computation can be done in real-time, given off-the-shelf PC hardware. A straight forward idea for accelerating the computation, is to compose an instrument of length

n from several "sub-instruments" of length n'. A sub-instrument is the result of the FST-algorithm for $p = 1$, which means that only one instrument of minimal deformation energy is considered. At first glance, it appears reasonable to calculate a sub-instrument of length n', and to use its last link as the start link for the calculation of a second sub-instrument, and to use its last link as the start link for the calculation of a third sub-instrument and so on. This procedure would result in an instrument of length n, pieced together from several "sub-instruments" of length $n' - 1$ (the length of the last may vary). However, this approach fails in situations, where a sub-instrument ends just before an obstacle, for example a strong bend. In this case it is not possible to attach an other sub-instrument to its last link, without violating the instrument's maximum flexibility.

To avoid running into a "dead-end", it would be necessary to explore the environment lying ahead in order to initiate an early change of direction. This can be achieved by simply following the above described procedure with the modification of taking the $k + 1$-st link, for $k = 1, 2, \ldots$ of a sub-instrument as the start link for the next sub-instrument. In this case, the resulting instrument is pieced together from several units of k links (the length of the last unit may vary). Each unit of k links is called a "segment". Each segment represents the beginning of a sub-instrument of length n'. The "sub-instruments" can be regarded as "tentacles" that reach out to explore the environment ahead.

Figure 1.8 shows an example of an instrument inserted from bottom to top into the beginning of a 2D calibration path. The "M"-shaped path is used in an experiment in Section 1.5.2 (Page 73) for model validation. The example here demonstrates the above described (n', k)-approximation for an overall length of $n = 13$, a sub-instrument ("tentacle") length of $n' = 6$ and a segment length of $k = 1$. At first, a sub-instrument of length six is computed (a). The first link of (a) is the first link of the final instrument (i). The second link of (a) serves as the start link for the second sub-instrument (b). The first link of (b) is the second link of the final instrument (i).

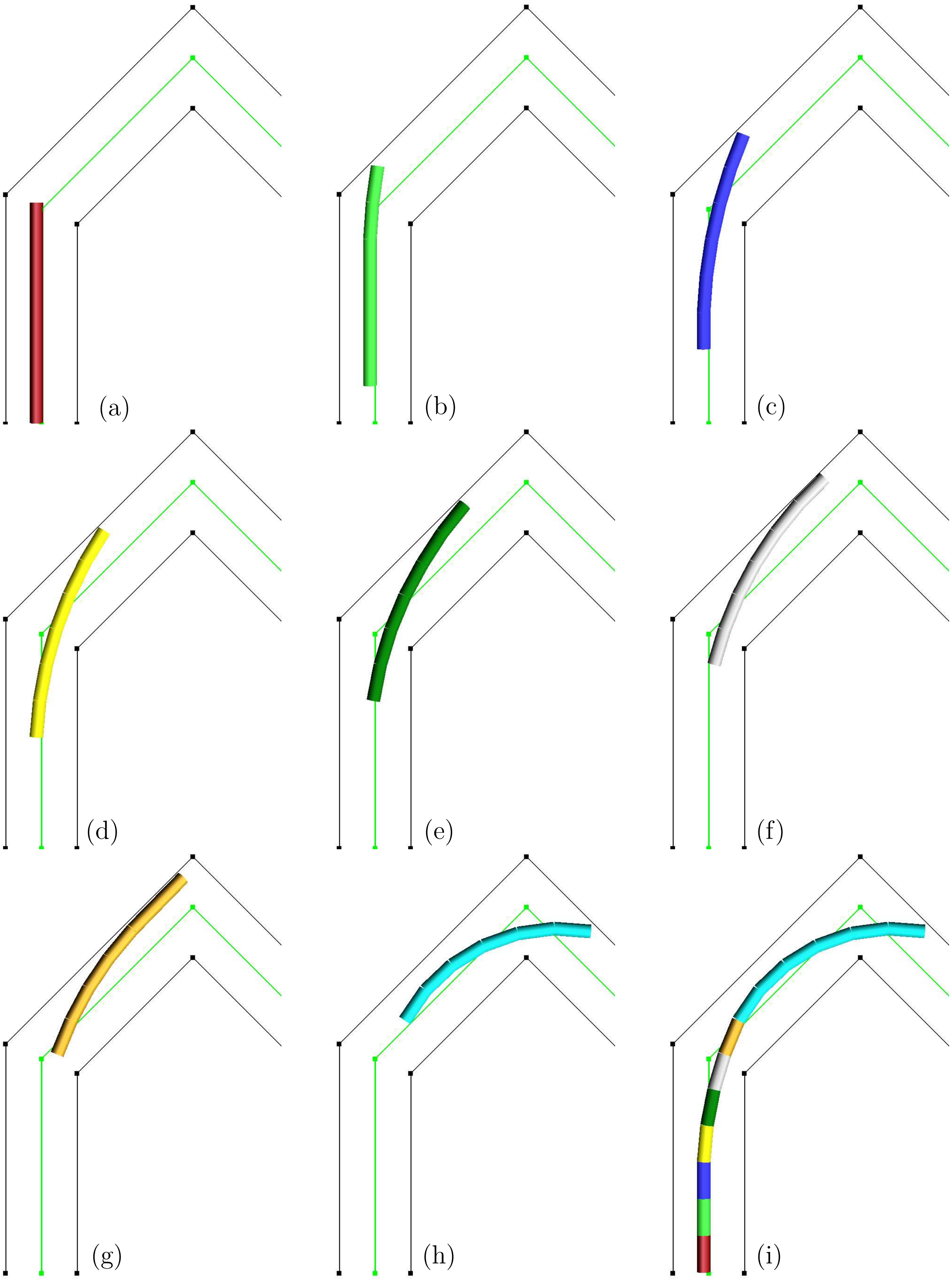

Figure 1.8: Eight “sub-instruments” (a)-(h) and the final instrument (i).

The second link of (b) is the start link for the third sub-instrument (c). The first link of (c) is the third link of the final instrument (i). The second link of (c) is the start link for the fourth sub-instrument (d), and so on.

As can be seen in (d)-(h), the collision of the sub-instrument with the wall causes even the first links to rotate into the future direction. This initiates an early change of direction, which avoids the "dead-end" situation shown in (g).

The final instrument (i) is obtained by concatenating the respective first links of all "sub-instruments". The color of each link corresponds to the color of its sub-instrument. To avoid an overshooting of the tentacle with respect to the desired overall insertion depth n, the procedure was aborted after $n - n'$ (final) links were computed. To these links, one last sub-instrument (h) was concatenated, forming a final instrument of length n. This last sub-instrument is called the "closure sub-instrument" and may be shorter than n'.

In this example, parameters $n' = 6$ and $k = 1$ were chosen to be small, to demonstrate the principle of the approach on a small area. Figure 1.24 (bottom) on Page 78 shows the result of the instrument inserted into the entire "M"-shaped path. Here, the parameters were $n = 44, n' = 20$ and $k = 4$. The top figure shows all "sub-instruments" simultaneously, demonstrating the adequacy of the term "tentacles".

Figure 1.27 on Page 84 represents an example showing the model's "tentacles" and the resulting instrument representing and endoscope inserted into a 3D lung phantom.

It should be noted that for a given "tentacle" length n', one can always construct an (artificial) environment where the above described approach will run into a dead-end. However, in case of the human anatomy it can be expected that one "tentacle" length can be found that works with all possible variations of a given organ. In the case of branching structures, such as lungs or vasculature, a region of interest (ROI) has to be specified, by using a bounding tube filter (see Equation 1.3). The bounding tube filter "cuts-off" side branches to guide the instrument model directly towards the specified target branch.

Formally, the above described idea can be expressed as the iterative concatenation of the first k links of several "sub-instruments". The exact number of "sub-instruments" is given by

$$m := \left\lceil \frac{n - n'}{k} \right\rceil + 1 \,, \tag{1.9}$$

where the '+1' stems from the "closure sub-instrument". Using the ceiling function has the consequence that the length of the "closure sub-instrument" is less or equal n' (see Equation 1.12). The final instrument can now be described by

$$I(\mathbf{s}, n) = \mathop{\bigcirc}_{i=1}^{m} I_i(\mathbf{s}_i, n'_i)[1, \ldots, k] \,, \tag{1.10}$$

where $I(\mathbf{s}, n)$ is the resulting instrument with start link $\mathbf{s}$ and length n, $I_i()$ is the i-th sub-instrument and $I_i()[1, \ldots, k]$ its first k links (segment). Symbol $\circ$ denotes the iterative concatenation of the "sub-instruments". The respective start links are given by:

$$\mathbf{s}_1 = \mathbf{s} \;\;,\;\; \mathbf{s}_i = I_{i-1}(\mathbf{s}_{i-1}, n'_i)[k+1] \,. \tag{1.11}$$

The notation using square brackets $I()[]$ denotes a subset of the instrument's links. For example $I()[k+1]$ and $I()[1, \ldots, k]$ denote the $k+1$-st link and the first to k-th link (in order from head to tip) of instrument I respectively.

The length of the "sub-instruments" is n'. However, as mentioned above, the length of the "closure sub-instrument" is less or equal n':

$$n'_i = \begin{cases} n' & : \;\; i < m \\ n - k(m-1) & : \;\; i = m \end{cases} \,, \tag{1.12}$$

where $k(m-1)$ is the number of final links before the "closure sub-instrument".

The new complexity is given by:

$$O\left(m\,(u\,v+1)^{n'}\right) \;\leq\; O\left((u\,v+1)^{n}\right) \,. \tag{1.13}$$

(u, v)-Approximation:

This section describes a second technique to further speed up the computation. It is based on the observation that the energy filter favors configurations with many small joint angles as opposed to a few large angles. This is because the energy filter minimizes the bending energy, which is calculated as the sum of the squares of joint angles. In other words it tends to distribute bending on many joints, using a small angle for each joint. As a consequence the dispersion of all angles tends to be small. This is particularly true for short tentacles of length $n' << n$ used in the (n', k)-approximation technique.

To better understand the (u, v)-approximation, it is helpful to recall the FST-algorithm, as described in Section 1.3.1. The algorithm can be outlined by:

```
createFST($\mathbf{s}, n, u, v, \theta$)
    ⋮
    (1.7)   $\mathbf{R}_{i,j} = \mathbf{R}(\mathbf{u}_i, j\theta)$ for $i = 1, \ldots, u$ , $j = 1, \ldots, v$ ,
    (1.6)   $f_{\mathsf{gen}}(\mathbf{F}) = \{\mathbf{F} \circ \mathbf{A} \mid \mathbf{A} \in f_{\mathsf{gen}}(\mathbf{F}\,\mathbf{T}\,\mathbf{R}_{i,j}) \cup \{\mathbf{F}\,\mathbf{T}\}$ for $i = 1, \ldots, u$ ,
                                                                  $j = 1, \ldots, v\}$
    ⋮
```

In Equation 1.7 a look-up table of rotation matrices is initialized and Equation 1.6 describes the recursion that enumerates all possible instruments. As can be seen in Figure 1.5 (right), there are $uv + 1$ possible joint positions, for each joint in the instrument. Or, in other words, there are $uv + 1$ possible positions for each link with respect to its predecessor.

The idea behind the (u, v)-approximation is, to constrain for each execution of the FST-algorithm the maneuverability of each joint to only one possible angle. This would correspond in Figure 1.5 (right) to the usage of only one cone, for example the red one in the center ($j = 1$), for all joints of the instrument.

Since this would limit the maximum flexibility of the instrument, a second execution of the FST-algorithm is needed, where only the blue cones (middle, $j = 2$) are

Algorithm 1.1 createFST$^{\text{uv}}$: (u, v)-approximation of createFST()

```
createFST^uv (s, n, u, v, θ)
for $j \leftarrow 1 \dots v$ do
    $\gamma \leftarrow j\,\theta$
    $I_j \leftarrow$ createFST($\mathbf{s}, n, u$, 1, $\gamma$)
end for
$I \leftarrow \min_{j=1}^{v} E(I_j)$        // find the instrument of minimal deformation energy
return $I$
```

used for all joints of the instrument. Finally, a third execution of the FST-algorithm uses only the green cones (outer, $j = 3$). Algorithm 1.1 describes this procedure.

In other words, instead of creating one spatial tree where each node has $uv + 1$ sons, v spatial trees are created, where for each tree, each node has only $u + 1$ sons. For each tree, the angle between two adjacent links is either zero or γ degrees. The '+1' stems from the fact that a zero degree angle (straight joint) is not included in the u principal directions (see Figure 1.5 (right)). The result of each tree is an instrument of minimal deformation energy. Among these v instruments, the one with minimal deformation energy is selected as the resulting instrument.

The ∼FST-Algorithm:

The ∼FST-algorithm combines the (n', k)- and (u, v)-approximation into one common framework. The idea is to apply the (u, v)-approximation to each sub-instrument of the (n', k)-approximation. Algorithm 1.2 shows the final ∼FST-algorithm:

As can be seen, the (u, v)-approximation (line 4) is nested inside the (n', k)-approximation. The (n', k)-approximation is realized by lines 1,2,3,5 and 6. Line 1 and 6 corresponds to Equation 1.11. Line 2 corresponds to Equation 1.9. Finally, line 3 and 5 corresponds directly to Equations 1.10.

Combining both approximation techniques means that the final instrument is pieced together from several shorter segments. Within each segment, the angle between two adjacent links is either zero or $j\,\gamma$ degrees, where j and γ are constants.

Algorithm 1.2 create∼FST($\mathbf{s}, n, n', k, u, v, \theta$): ($n', k$)- and ($u$, v)-approximation of createFST()

```
create∼FST($\mathbf{s}, n, n', k, u, v, \theta$)
$\mathbf{s}' \leftarrow \mathbf{s}$
$m \leftarrow \left\lceil \frac{n-n'}{k} \right\rceil + 1$
for $h \leftarrow 1 \ldots m$ do
    $I_h \leftarrow$ createFST$^{\mathrm{u,v}}(\mathbf{s}', n', u, v, \theta)$
    $I \leftarrow I \circ I_h[1, \ldots, k]$
    $\mathbf{s}' \leftarrow I_h[k+1]$
end for
return $I$
```

However, parameter j may vary between segments, allowing for smaller or greater angles among different segments. This gives the final instrument enough flexibility to adapt to anatomies with a variety of curvatures. Figure 1.9 shows a brain artery (same model as in Figure 1.6) as an example of such an anatomy. The parameters are $n = 31, n' = 10, k = 5, u = 8, v = 20, \theta = 2°$. The final instrument (catheter) is composed of six segments, depicted in different colors. As can be seen, the flexibility of each segment adapts to the artery's local curvature.

The complexity of the ∼FST-algorithm is given by:

$$O\left(m\, v\, (u+1)^{n'}\right) \;\leq\; O\left(m\, (u\, v+1)^{n'}\right) \;\leq\; O\left((u\, v+1)^{n}\right) . \tag{1.14}$$

The following experiment determines the empirical complexity of the ∼FST-algorithm:

Experiment 1.2: Comparison of the ∼FST- with the FST-Algorithm.

Objective: *Comparing the empirical time and space complexity of the FST-algorithm with its ∼FST variant.*

Method: *(1) Simulating the insertion of a catheter into a model of a brain artery using the ∼FST-algorithm. (2) Results from experiment 1.1.*

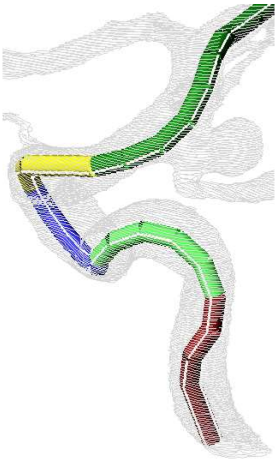

Figure 1.9: Simulated insertion (from bottom to top) of a catheter into a brain artery, using the (n', k) and (u, v)-approximation. The first $k = 5$ links (segment) of overall six "sub-instruments" are shown in different colors. The flexibility of each segment adapts to the artery's local curvature.

Platform: *PC, Pentium 4 dual, 1.3 GHz, 1 GB; Graphics board: nVIDIA GeForce3, 64 MB.*

Material: *Model of a brain artery, reconstructed from planar parallel cross sections ([43], [44], NUAGES), obtained from a patient specific 3D rotational angiography (C-arm) scan. Number of triangles: ca. 25000.*

Design: *For a fixed starting position the ∼FST-algorithm* ($n' = 10$, $k = 5$, $u = 8$, $v = 20$, $\theta = 2°$) *was used to calculate a catheter of length* n*, inserted into the brain artery (Figure 1.9). This procedure was repeated 31 times, for* $n = 1, \ldots, 31$*. For each calculation the number of recursive calls needed to enumerate all possibilities, given the geometry and energy filter, was recorded.*

Results: *The result is shown in Figure 1.7. A linear function* ($y = 761.4\,x + 4358$) *can be fit (least squares) to the resulting curve, indicating that the empirical complexity for this anatomy is* $O(n)$*.*

Observation: *The time for* $n = 31$ *(see Figure 1.9) was 4 seconds.*

1.4 Model Description

This chapter gives a detailed description of the flexible instrument model. Section 1.4.1 describes the model algebraically by defining the building blocks it is made of: An endoscope consists of a “bending section” attached to an “elastic shaft”, which in turn is made of “links” interconnected by “joints”. The problem of creating an elastic shaft is formulated as a successive filtering process, where a series of constraints are imposed to an initially unstructured sequence of links.

Section 1.4.2 describes a model for aligning the instrument’s tip with a target. This is important for endoscopic procedure, where a needle needs to be placed inside a target. The model includes among others a description of the endoscope’s actively bendable tip.

Section 1.4.3 describes the model algorithmically. A pseudo-code, based on the notation introduced in Section 1.4.1, was developed to succinctly describe the algorithms. First, an algorithm “createFST()” is given, which creates a filtered spatial tree. Function “createFST()” calls several filter functions, which are described separately. Then, an algorithm “create∼FST()” is given, which incorporates the $(n', k)-$ and $(u, v)-$approximation techniques, described in Section 1.3.2 to speed-up the execution of algorithm “createFST()”.

1.4.1 Algebraic Description

In this section, an algebraic description of the endoscope model is developed. The endoscope model is described constructively, starting with the definition of two elemental building blocks: “link”, denoted by $\bar{\mathbf{L}}$ and “joint”, denoted by $\bar{\mathbf{J}}$. Based on $\bar{\mathbf{L}}$ and $\bar{\mathbf{J}}$, an “elastic shaft” $\bar{\mathbf{S}}^{\mathrm{E}}$ is described as a sequence of links, interconnected by joints. Its description is given in four consecutive steps, where each step builds on the previous step. The mechanical properties of an elastic shaft are formulated in terms of given internal and external constraints applied to the shaft’s links and

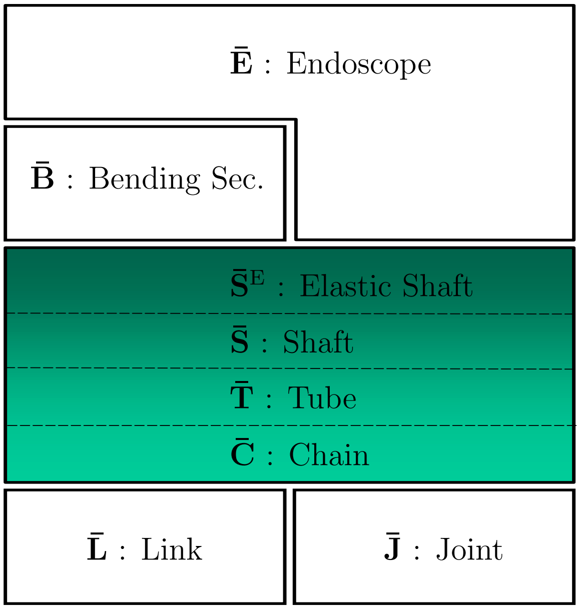

Figure 1.10: Building blocks of the endoscope model. The elemental blocks "Link" and "Joint" represent the foundation of the model.

joints. The definition of $\bar{\mathbf{S}}^{\mathrm{E}}$ is then used to define a "bending section" $\bar{\mathbf{B}}$ as two links, interconnected by an elastic shaft. Based on $\bar{\mathbf{B}}$, an "endoscope" $\bar{\mathbf{E}}$ is finally described as an elastic shaft, connected to a bending section. Figure 1.10 gives an overview of the model's building blocks. The central element in the figure is the elastic shaft $\bar{\mathbf{S}}^{\mathrm{E}}$. Its description is given constructively: the first step is the definition of a structure "chain" $\bar{\mathbf{C}}$, followed by the definition of a structure "tube" $\bar{\mathbf{T}}$, "shaft" $\bar{\mathbf{S}}$ and finally "elastic shaft" $\bar{\mathbf{S}}^{\mathrm{E}}$. Each structure is the result of a filtering process applied to the previous structure, starting with an entirely unconstrained sequence of links. In other words, an elastic shaft is described as the result of a concatenation of four filter functions applied to an initial, unstructured sequence of links.

For each of the four structures $\bar{\mathbf{C}}, \bar{\mathbf{T}}, \bar{\mathbf{S}}$ and $\bar{\mathbf{S}}^{\mathrm{E}}$ there exists an corresponding set of link sequences, denoted by the calligraphic letters $\mathcal{C}$, $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{S}^{E}$. Starting with the unconstrained set of all sequences of "links" $\mathcal{L}$, a "link filter" $f_{\mathsf{link}}()$ is applied to $\mathcal{L}$. The result is a chain $\mathcal{C}$. Then a "joint filter" $f_{\mathsf{joint}}()$ is applied to $\mathcal{C}$. The result is a tube $\mathcal{T}$. To $\mathcal{T}$ a "geometry filter" $f_{\mathsf{geometry}}()$ is applied. The result is a shaft

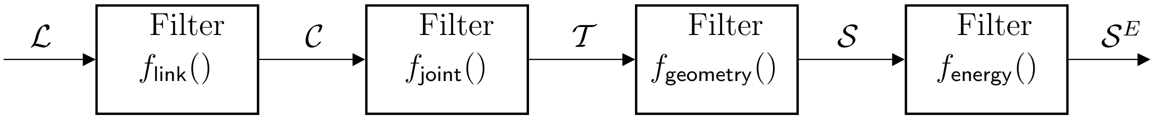

Figure 1.11: A pipeline of filter functions and (from left to right) the intermediate sets "links", "chain", "tubes", "shafts", "elastic shafts".

$\mathcal{S}$. Finally, an "energy filter" $f_{\mathsf{energy}}()$ is applied to $\mathcal{S}$ to form the elastic shaft $\mathcal{S}^E$. Accordingly, the following relation holds:

$$\mathcal{L} \supset \mathcal{C} \supset \mathcal{T} \supset \mathcal{S} \supset \mathcal{S}^E \,. \tag{1.15}$$

Figure 1.11 shows the concatenated filter functions and the notation for the intermediate sets. Filter $f_{\mathsf{link}}()$ has the effect that it determines the physical measures for each link and connects them together to a chain of links. Filter $f_{\mathsf{joint}}()$ has the effect that it restricts the movement of a link with respect to its predecessor. Filter $f_{\mathsf{geometry}}()$ restricts the movement of the instrument to its physical confines. Finally, filter $f_{\mathsf{energy}}()$ imposes mechanical constraints on the instrument. All five stages of this filtering process are visualized in Figure 1.12, 1.13 and 1.16.

1.4.1.1 Link

It follows the description of the elemental building block "link". Position and orientation of a link in 3D space is described by attaching a "reference frame" to it. A reference frame is a "coordinate system" with a specified origin. The orientation of a reference frame with respect to a given world coordinate system can be described by a 3 × 3 rotation matrix. The position of the origin can be described by a 3 × 1 position vector. Let

$$\mathcal{M} = \{(a_{ij}) \mid i, j = 0, \ldots, 3,\ a_{ij} \in \mathbb{R}\} \tag{1.16}$$

be the set of all 4×4 matrices over $\mathbb{R}$. A matrix $\mathbf{F} \in \mathcal{M}$ is regarded as a representation (homogeneous transform) of a reference frame in 3D space:

$$\mathbf{F} = \begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p} \in \mathbb{R}^3 , \tag{1.17}$$

where $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ describe the unit vectors along the frame's positive x-, y- and z-axis with respect to a world coordinate system. Vector $\mathbf{p}$ describes the position of the reference frame's origin with respect to the world coordinate system. The arrangement of $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{p}$ as column vectors of the matrix was chosen to be compliant with OpenGL's representation of a homogeneous transform. Let $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ denote the homogeneous unit vectors $(1001)^{\mathrm{T}}$, $(0101)^{\mathrm{T}}$ and $(0011)^{\mathrm{T}}$. Also, let $\mathbf{0}$ denote the homogenous zero vector $(0001)^{\mathrm{T}}$. Thus, $\mathbf{F}\,\mathbf{0}$ represents the fourth column of matrix $\mathbf{F}$ and therewith the origin of the coordinate system in homogeneous coordinates. In the following text, two special matrices are considered: a translation matrix $\mathbf{T}(\mathbf{v}) \in \mathcal{M}$ and a rotation matrix $\mathbf{R}(\mathbf{v}, \alpha) \in \mathcal{M}$. The first matrix causes a translation to the given vector $\mathbf{v}$ and the second causes a rotation of α degrees about vector $\mathbf{v}$.

Definition 1.4.1 (Link $\bar{\mathbf{L}}$ and set of all links L). A "link" in 3D space is represented by a cylinder of length l and diameter d. The position and orientation of the cylinder is described by an attached reference frame $\mathbf{F}$:

$$\bar{\mathbf{L}} = (\mathbf{F}, l, d), \quad \mathbf{F} \in \mathcal{M},\ l,\ d \in \mathbb{R} .$$

As shown in Figure 1.12 (left), the reference frame is attached in a way that the frame's z-axis corresponds to the centerline of the cylinder. The cylinder's bottom and top bases lie in the $z = 0$ and $z = l$-plane.
Let L denote the set of all links $\bar{\mathbf{L}}$. □

This yields to definition of the unconstrained set of all link sequences $\mathcal{L}$:

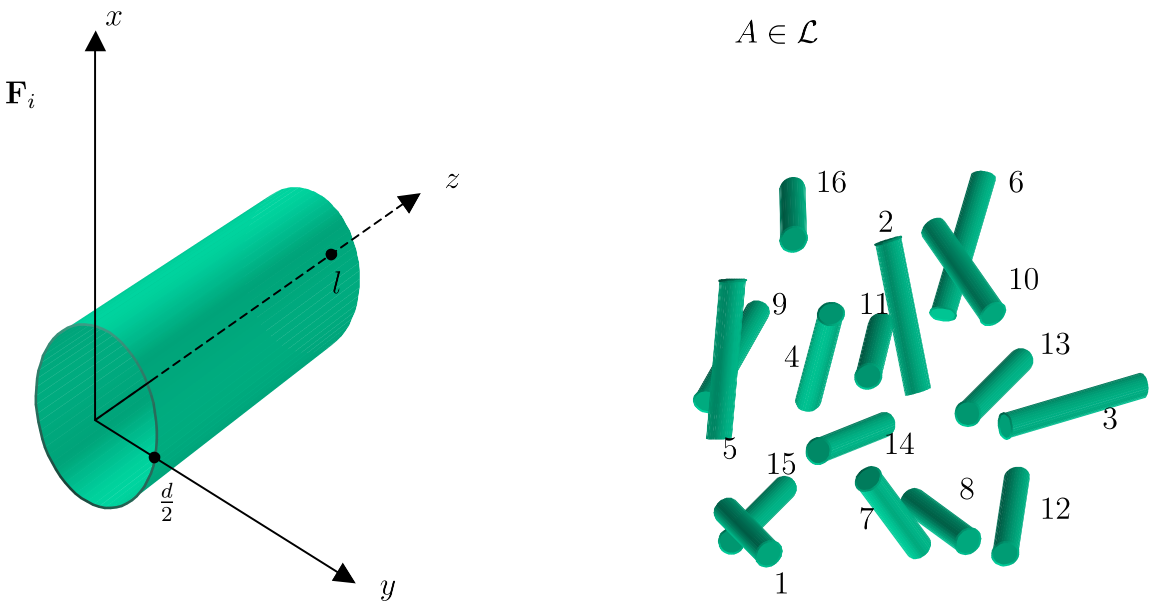

Figure 1.12: Left: A reference frame attached to a link. Right: A sequence of 16 links.

Definition 1.4.2 (Set of all link sequences $\mathcal{L}$). A linearly ordered set of n concatenated links

$$\bar{\mathbf{L}}_n \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1$$

is denoted as a sequence of n links (see Figure 1.12, right). The order is from right to left. The detailed notation for a sequence of n links with length l and diameter d is

$$\bar{\mathbf{L}}_n = (\mathbf{F}_n,\, l_n,\, d_n) \;\circ\; \ldots \;\circ\; \bar{\mathbf{L}}_2 = (\mathbf{F}_2,\, l_2,\, d_2) \;\circ\; \bar{\mathbf{L}}_1 = (\mathbf{F}_1,\, l_1,\, d_1)\ .$$

If the emphasize is more on the link frames, the following notation is used:

$$\mathbf{F}_n \circ \ldots \circ \mathbf{F}_2 \circ \mathbf{F}_1\,, \quad \text{with } \mathbf{F}_i \in \mathcal{M}\ ,$$

where the concatenation is usually realized by a matrix multiplication in the order from right to left.

Given this notation, the set of all link sequences of length n is given by

$$\mathcal{L} = L^n\ ,$$

where L is the set of all links. □

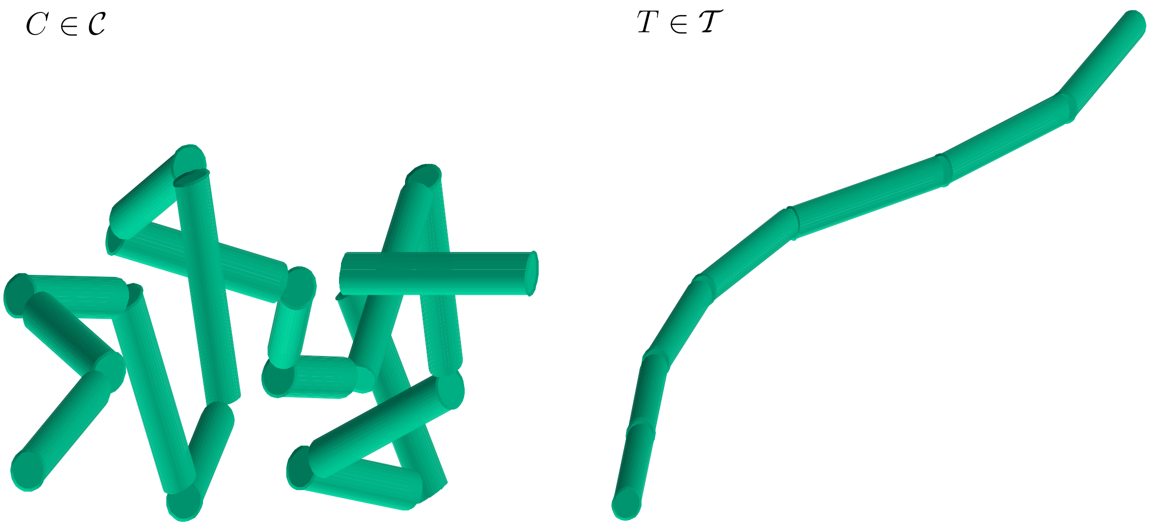

Figure 1.13: Left: A “chain-like” structure. Right: A “tube-like” structure.

1.4.1.2 Chain

Now, the set of all “chains” $\mathcal{C} \subset \mathcal{L}$ is described by applying a “link filter” to $\mathcal{L}$:

Definition 1.4.3 (Chain $\bar{\mathbf{C}}$). A sequence of links $C = (\bar{\mathbf{L}}_n = (\mathbf{F}_n,\, l_n,\, d_n)) \circ \ldots \circ (\bar{\mathbf{L}}_2 = (\mathbf{F}_2,\, l_2,\, d_2)) \circ (\bar{\mathbf{L}}_1 = (\mathbf{F}_1,\, l_1,\, d_1))$ is called a “chain” of length n, if the links in C satisfy the following conditions:

$$\begin{aligned} & |\,C| = n \\ \wedge \quad & \mathbf{F}_{i+1} = \mathbf{F}_i\, \mathbf{T}(\hat{\mathbf{z}} l_i), \quad \text{for all } i = 1, \ldots, n-1 \\ \wedge \quad & (l_i = l_j) \;\wedge\; (d_i = d_j), \;\; \text{for all members of sequence } C\,. \end{aligned}$$

The set of all chains is denoted by $\mathcal{C} \subset \mathcal{L}$. To define a chain, all parameters are grouped in a 4-tuple

$$\bar{\mathbf{C}} = (C,\, n,\, l,\, d)\,. \qquad \square$$

Figure 1.13 (left) shows a perspective rendering of a chain $\bar{\mathbf{C}}$. Note that $\mathbf{T}()$ denotes a translation matrix. The second condition says that the bottom base of a cylinder should be connected to its successor’s top base. Figure 1.14 (left) shows the reference frames of two connected links. The third condition simply says that all links in a chain should have the same length and diameter.

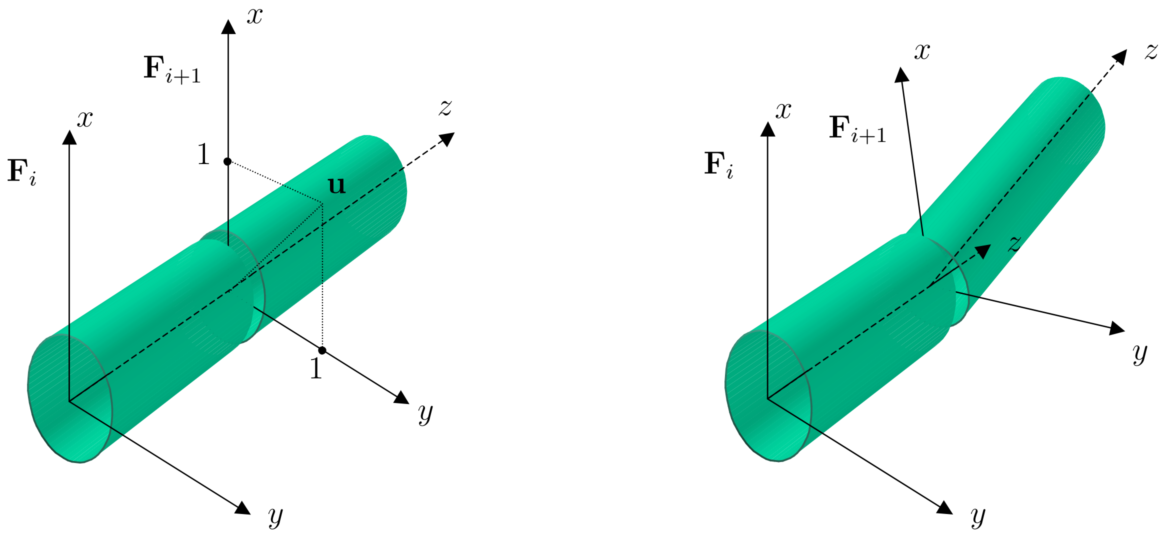

Figure 1.14: A joint with a rotation axis $\mathbf{u}$. Left: Before rotation. Right: After rotation about $\mathbf{u}$.

1.4.1.3 Joint

The second elemental building block is a discrete ball and socket joint:

Definition 1.4.4 (Discrete ball and socket joint $\bar{\mathbf{J}}$). A discrete ball and socket joint is described by the two links it connects and by its degree of maneuverability. If a joint allows a link only to be in a finite number of positions relative to its predecessor, it is called a discrete joint.

The joint's degree of maneuverability is controlled by two parameters: the number of rotation axes u and the number of rotation increments v for each axis. Parameter u describes the number of directions in which the second link can be rotated with respect to the first link. Parameter v gives the maximum number of rotation increments for each direction and a third parameter θ gives the increment in degrees. Thus, the joint "range" is $v\theta$ degrees. Formally, a discrete joint is described by

$$\bar{\mathbf{J}} = (\bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1,\ u,\ \overbrace{\quad v,\ \theta \quad}^{\text{joint range}}\ ),$$

with a defined relation between $\bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_2$, controlled by u, v, θ. Let set $J_{u,v,\theta}$ denote the set of all joints $\bar{\mathbf{J}}$ with a range of motion described by u, v, θ:

$$\begin{aligned}J_{u,v,\theta} = \{(\bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1,\, u,\, v,\, \theta) \;\mid\; & \bar{\mathbf{L}}_1 = (\mathbf{F}_1,\, l_1,\, d_1) \in L, \\ & \bar{\mathbf{L}}_2 = (\mathbf{F}_2,\, l_2,\, d_2) \in L \;\text{ with} \\ & \mathbf{F}_2 = \mathbf{F}_1\; \mathbf{T}(\hat{\mathbf{z}} l_1)\; \mathbf{R}(\mathbf{u}, i\theta), \;\text{ for } i = 0,\, \ldots, v,\; \mathbf{u} \in D_u \\ & D_u \subset D \text{ with } |D_u| = u \\ & D = \left\{ (x\; y\; 0\; 1)^{\mathrm{T}} \mid x,\, y \in \mathbb{R} \right\}\,.\end{aligned}$$

$\square$

In the third line the degree of maneuverability of the second link $\mathbf{F}_2$ with respect to the first link $\mathbf{F}_1$ is defined. A 4×4 translation matrix $\mathbf{T}()$ is used to specify a translation of link $\mathbf{F}_1$ along its positive z-axis by a distance of l_1. After the translation, a 4×4 rotation matrix $\mathbf{R}()$ is used to specify the range of motion. The evaluation order of the transformation $\mathbf{F}_1\, \mathbf{T}()\, \mathbf{R}()$ is from "right to left" (column vectors). For example, the translation $\mathbf{T}(\hat{\mathbf{z}} l_1)$ has to postmultiply $\mathbf{F}_1$, because the translation should be along $\mathbf{F}_1$'s z-axis and not along the z-axis of $\mathbf{F}_1$'s world coordinate system. Similarly, $\mathbf{R}()$ has to postmultiply $\mathbf{F}_1\, \mathbf{T}()$, because the rotation should be about a rotation axis local to $\mathbf{F}_1\, \mathbf{T}()$ and not about an axis of the world coordinate system.

The rotation axes or directions are given by set D_u and the rotation increments are given by $i\theta$. Note that the above notation for $\mathbf{F}_2$ implicitly includes the straight, not rotated link. This is because index i starts from zero, which causes a zero rotation increment. This represents a special case. Consequently, there are $|D_u|v + 1 = uv + 1$ different positions for link $\mathbf{F}_2$, where the '+1' corresponds to the straight link.

Example:

$D_u = \left\{ (x\; y\; 0\; 1)^{\mathrm{T}} \mid x,\, y \in \{-1,\, 0,\, 1\} \right\} \setminus \{(0\;\;0\;\;0\;\;0)^{\mathrm{T}}\}$ contains 8 vectors, which are rotation axes in the $z = 0$-plane, equally spaced by 45 degrees. The zero-vector, which effectively causes no rotation was excluded. The left drawing in Figure 1.14 shows a rotation axis $\mathbf{u} = (1\,1\,0\,1)^{\mathrm{T}}$ for the second link and the right drawing shows this link after the rotation about $\mathbf{u}$. Drawing the resulting link positions for all 8 axes plus the straight link, results in the drawing shown in Figure 1.5 (left) on Page 27. The figure shows the 9 possible positions ($u = 8$, $v = 1$) a link $\bar{\mathbf{L}}_2$ can take relative to its predecessor $\bar{\mathbf{L}}_1$ (occluded).

1.4.1.4 Tube

The above introduced discrete joint is now used to constrain a chain-like structure as shown on the left in Figure 1.13 to a more tube-like structure as shown on the right. A joint is inserted in between two adjacent links of the chain to restrict its flexibility. The tube's flexibility is determined by joint range $v\theta$ and link length l. In the following, an intuitive way to calculate the maximum allowable joint range for a given real endoscope is given.

The idea is to form the smallest possible circle with the given endoscope and measure its radius r. This radius, together with a given link length l determines the endoscope's flexibility and therewith the maximum joint rage. A flexibility function needs to be defined, which calculates $v\theta$ from a given radius r and link length l.

As shown in Figure 1.15, a circle of radius r can be linearly approximated by segments or links. If r is the radius of the smallest circle that can be formed with an endoscope, then the angle between two adjacent links corresponds to the maximum allowable joint range $v\theta$. This joint range models the endoscope's maximum flexibility.

To determine the maximum joint range, it can be seen from the figure that the angle between to adjacent links of length l corresponds to the angle at the center of the circle. To understand this it is helpful to see that the sum of the "outer" angles and the sum of the "inner" angles equals 360°. Thus, the flexibility function, which calculates this angle from a given radius r and link length l is given by:

Definition 1.4.5 (Flexibility function $f_{\sim}()$).

$$f_{\sim} : (\mathbb{R}, \mathbb{R}) \rightarrow \mathbb{R},$$

$$v\theta = f_{\sim}(l, r) = 2 \arcsin \left(\frac{l}{2r} \right) .$$

In the following, radius r is simply called the endoscope's flexibility. □

With this definition a flexible tube can be defined as a chain-like structure, whose links are interconnected by discrete joints (see Figure 1.13, right).

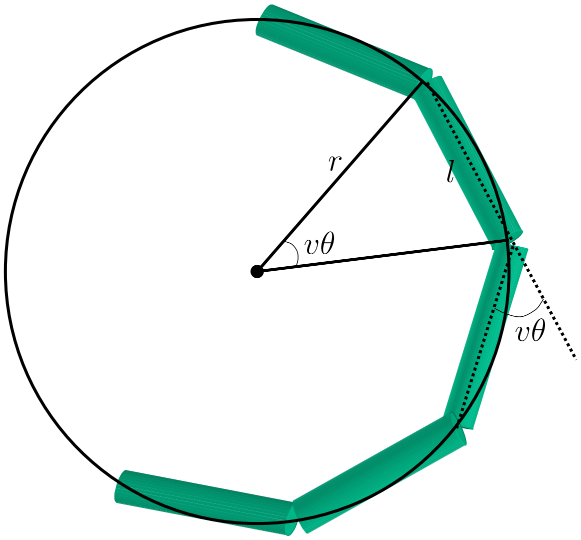

Figure 1.15: A flexible tube at maximum flexibility forms a circle of minimal radius.

Definition 1.4.6 (Tube $\bar{\mathbf{T}}$). A chain $T \in \mathcal{C}$ is called a "tube", if the links in $T = \bar{\mathbf{L}}_n \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1$ satisfy the following condition:

$$(\bar{\mathbf{L}}_{i+1} \circ \bar{\mathbf{L}}_i) \in J_{u,v,\theta}, \text{ for } i = 1, \ldots, n-1\,. \tag{1.18}$$

The set of all tubes is denoted by $\mathcal{T} \subset \mathcal{C}$. A tube with

$$\theta = \frac{f_{\sim}(l,\,r)}{v}\,,$$

where $r > 0$, is called a tube with flexibility r. Parameter r is called flexibility value. All parameters describing a flexible tube are grouped in a 7-tuple

$$\bar{\mathbf{T}} = (\,T,\ \overbrace{n,\,l,}^{\text{tube length}}\ \ d,\ \overbrace{u,\,v,}^{\text{deg. of motion}}\ \ \overbrace{r}^{\text{tube flexibility}}\ )\,.$$

$\square$

In this definition, n and l can be used to reflect the desired tube length and u, v to reflect the desired degree of maneuverability. The condition 1.18 requires that two adjacent links are connected by a discrete ball and socket joint with $uv + 1$ joint positions and a $v\theta$ degree range of motion.

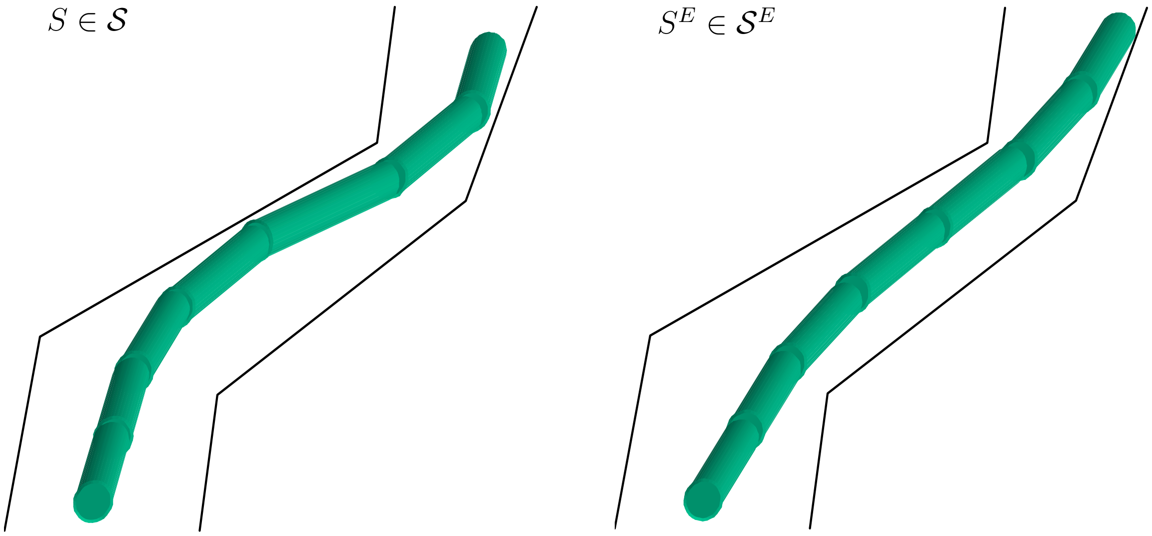

Figure 1.16: Left: A "shaft-like" structure. Right: An "elastic shaft-like" structure of minimal deformation energy.

1.4.1.5 Shaft

In the following step, external constraints are imposed on a flexible tube which leads to a "shaft-like" structure. As shown in Figure 1.16 (left), the movement of a shaft is constrained by its physical confines. To determine a collision between a link and its environment, the geometry of a link is approximated by a set of sample points on its surface:

Definition 1.4.7 (Link sample points $P_{l,d}$). A set $P_{l,d} \subset \mathbb{R}^3$ is called a set of "link sample points", if it is a subset of all surface points of a cylinder with length l and diameter d

$$P_{l,d} \subset \left\{ (x\,y\,z)^{\mathrm{T}} \mid \left(\left(\sqrt{x^2 + y^2} \leq \frac{d}{2} \right) \wedge (0 \leq z \leq l) \right) \right\} ,$$

where x, y, z denote real numbers. □

Definition 1.4.8 (Shaft $\bar{\mathbf{S}}$). A tube $S \in \mathcal{T}$ with link length l and diameter d is called a "shaft", if the following condition holds:

$$\mathbf{F}_i\,\mathbf{p_j} \text{ ``is inside the organ'', for all } \mathbf{p_j} \in P_{l,d},\ \mathbf{F}_i \in S\,. \tag{1.19}$$

Each $\mathbf{F}_i$ is assumed to be given with respect to the world coordinate system. In case an implementation is used, where each link (beside the start link) is given with respect to its predecessor (see also the example on Page 57), the condition becomes

$$\mathbf{F}_i\,\mathbf{F}_{i-1}\,\ldots\,\mathbf{F}_2\,\mathbf{F}_1\,\mathbf{p}_j \text{ ``is inside the organ'', for all } \mathbf{p_j} \in P_{l,\,d},\ \mathbf{F}_i \in S\,. \tag{1.20}$$

The set of all shafts is denoted by $\mathcal{S} \subset \mathcal{T}$. A shaft is described by a 8-tuple

$$\bar{\mathbf{S}} = (S,\, n,\, l,\, d, u,\, v,\, r,\, P_{l,\,d})\,. \qquad \square$$

Condition 1.19 and 1.20 require all sample points on all links to be inside the organ.

1.4.1.6 Elastic Shaft

The last filter in the pipeline filters all shaft-like structures regarding their deformation energy to create an "elastic shaft". An elastic shaft is a shaft with minimal deformation energy based on the theory of elasticity (see Section 1.3).

Definition 1.4.9 (Elastic shaft $\bar{\mathbf{S}}^{\mathrm{E}}$). A shaft $S^E \in \mathcal{S}$ is called an "elastic shaft", if the following condition holds:

$$E(S^E,\, \alpha,\, \beta) < E(A_i,\, \alpha,\, \beta), \text{ for all } A_i \in \mathcal{S},\, A_i \neq S^E\,, \tag{1.21}$$

where $E : (\mathcal{L},\, \mathbb{R}, \mathbb{R}) \to \mathbb{R}$ is an energy function, which calculates the deformation energy of the shaft with material properties α and β. The set of all elastic shafts is denoted by $\mathcal{S}^E$.

The set of all elastic shafts is denoted by $\mathcal{S}^E \subset \mathcal{S}$. An elastic shaft is described by an 10-tuple

$$\bar{\mathbf{S}}^{\mathrm{E}} = (S^E,\, n,\, l,\, d,\, u,\, v,\, r,\, P_{l,\,d},\, \alpha,\, \beta)\,.$$

$\square$

Condition 1.21 requires S^E to have minimal deformation energy among all shafts in $\mathcal{S}$. The energy function is defined as:

Definition 1.4.10 (Energy function $E()$). Given a shaft $\bar{\mathbf{S}} = (S, n, l, d, u, v, r, P_{l,d})$, with $S = \bar{\mathbf{L}}_n \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_i = (\mathbf{F}_i, l, d)$. The deformation energy is the sum of the bending energy and the torsion energy:

$$E(S, \alpha, \beta) = E_\kappa(S, \alpha) + E_\tau(S, \beta),$$

where the bending energy is given by

$$E_\kappa(S, \alpha) = \sum_{i=1}^{n-1} \alpha \, \kappa(S, i)^2,$$

and the torsion energy is given by

$$E_\tau(S, \beta) = \sum_{i=1}^{n-1} \beta \, \tau(S, i)^2 \, ,$$

with α the amount of resistance to bending, $\kappa(S, i)$ the curvature of S at i, β the amount of resistance to twisting and $\tau(S, i)$ the torsion of S at i. $\square$

"Curvature" and "torsion" are fundamental notions in the field of differential geometry of curves. As a reminder, the definitions of curvature and torsion for smooth space curves are given. Both definitions are based on the "Frenet frame", which is introduced first. Based on this, the corresponding notions for piecewise linear "curves" are developed.

Definition 1.4.11 (Frenet or tnb frame and osculating plane). The Frenet frame of a point travelling along a smooth space curve is a triple of mutually orthogonal vectors $(\mathbf{t}, \mathbf{n}, \mathbf{b})$. Vector $\mathbf{t}$ is called the unit tangent vector, $\mathbf{n}$ the principal unit normal vector and $\mathbf{b}$ the unit binormal vector.

Given a parameterized smooth space curve $\mathbf{x}(t) : [a, b] \rightarrow \mathbb{R}^3$, then vector $\mathbf{p} = \mathbf{x}(t)$ is the position vector of a point moving along the curve at time t. The point's velocity vector is $\mathbf{v}(t) = \frac{d\mathbf{x}}{dt}$ and its acceleration vector is $\mathbf{a} = \frac{d\mathbf{v}}{dt}$. Let the distance along the curve from a base point $\mathbf{x}(t_0)$ to a point $\mathbf{x}(t)$ be given by a function $s(t) =$

$\int_{t_0}^{t} ||\mathbf{v}(\tau)||d\tau$. Note that $||\mathbf{v}||$ denotes the norm or length of vector $\mathbf{v}$. Value s is called an arc length parameter for the curve. The Frenet frame is given by

$$\begin{aligned}\mathbf{t} &= \frac{d\mathbf{x}}{ds} = \frac{d\mathbf{x}}{dt}\frac{dt}{ds} = \mathbf{v}\frac{1}{||\mathbf{v}||} = \frac{\mathbf{v}}{||\mathbf{v}||}\,, \\ \mathbf{n} &= \frac{d\mathbf{t}/ds}{||d\mathbf{t}/ds||} = \frac{\check{\mathbf{t}}}{||\check{\mathbf{t}}||}\,, \\ \mathbf{b} &= \mathbf{t}\times\mathbf{n}\,.\end{aligned}$$

Vector $\mathbf{t}$ is simply the unit tangent vector at point $\mathbf{p}$, $\mathbf{n}$ is the unit vector in the direction of the derivative of the unit tangent, and $\mathbf{b}$ is the cross-product of the unit tangent vector with the principal unit normal vector.

The plane orthogonal to $\mathbf{b}$ is called the "osculating plane". The osculating plane is spanned by the three points $\mathbf{x}(t - h_1)$, $\mathbf{x}(t)$, and $\mathbf{x}(t + h_2)$ on the curve as h_1 and h_2 approach zero.

Definition 1.4.12 (Curvature κ and torsion τ of a smooth space curve). The curvature of a space curve in a point $\mathbf{p}$ is the rate of change of the tangent vector in the neighborhood of $\mathbf{p}$:

$$\kappa = \left|\left|\frac{d\mathbf{t}}{ds}\right|\right|\,,$$

with $\mathbf{t}$ the unit tangent vector. The curvature is a measure of how rapidly the curve pulls away from the tangent line in the neighborhood of $\mathbf{p}$.

The torsion of a space curve in a point $\mathbf{p}$ is the rate of change at which the osculating plane turns about $\mathbf{t}$ as $\mathbf{p}$ moves along the curve

$$\tau = -\frac{d\mathbf{b}}{ds}\cdot\mathbf{n}\,,$$

where $\mathbf{b}$ denotes the binormal vector and vector $\mathbf{n}$ the principal normal vector. Note that $d\mathbf{b}/ds$ is parallel to $\mathbf{n}$. The torsion is a measure of how rapidly the curve pulls away from the plane. □

The **tnb** frame was used to define the notions curvature and torsion for smooth (differentiable) space curves. A flexible tube however, is in general not differentiable

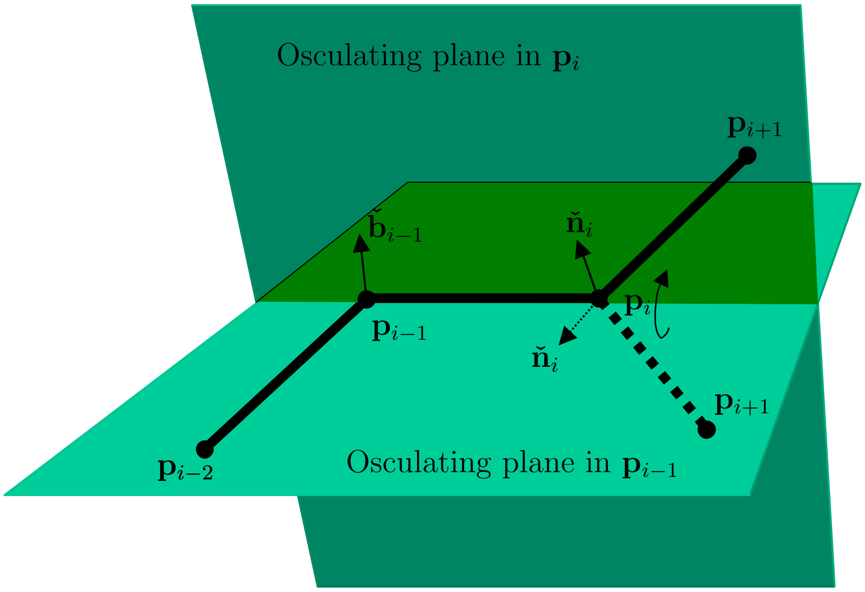

Figure 1.17: Osculating planes in $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ and the respective principal normal vector $\check{\mathbf{n}}_i$ and binormal vector $\check{\mathbf{b}}_{i-1}$.

at its joints, since a joint with a non-zero angle represents a corner where the one-sided derivatives differ. Therefore, in general, the unit tangent vector $\mathbf{t}$ and therewith $\mathbf{n}$ and $\mathbf{b}$ do not exist for piecewise linear curves. However, it is possible to define a "geometrically equivalent" $\check{\mathbf{t}}\check{\mathbf{n}}\check{\mathbf{b}}$ frame:

Definition 1.4.13 (Osculating plane and $\check{\mathbf{t}}\check{\mathbf{n}}\check{\mathbf{b}}$ frame for piecewise linear curves). Given a shaft $\bar{\mathbf{S}} = (S,\, n,\, l,\, d, u,\, v,\, r,\, P_{l,d})$ with $S = \bar{\mathbf{L}}_n \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_i = (\mathbf{F}_i,\, l,\, d)$. Let $(\bar{\mathbf{L}}_{i+1} \circ \bar{\mathbf{L}}_i) \in J_{u,v,\theta}$, for all $i = 1, \ldots, n-1$. Let $\mathbf{p}_i$ be the origin of frame $\mathbf{F}_i$, for $i = 1 \ldots n$.

Consider the above given definition of an osculating plane. The idea can be translated to piecewise linear curves as shown in Figure 1.17. Here, the osculating plane in point $\mathbf{p}_i$ is given by the limit of $\mathbf{p}_{i-1}$ and $\mathbf{p}_{i+1}$ approaching $\mathbf{p}_i$ or simply by the plane passing through $\mathbf{p}_{i-1}, \mathbf{p}_i$ and $\mathbf{p}_{i+1}$.

A point $\mathbf{q}$ on the osculating plane can be described by the plane equation in three-point form:

$$\mathbf{q} = \mathbf{p}_i + r(\mathbf{p}_{i+1} - \mathbf{p}_i) + s(\mathbf{p}_{i-1} - \mathbf{p}_i), \quad r,\, s \in \mathbb{R}\,.$$

With

$$\mathbf{u} = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{||\mathbf{p}_{i+1} - \mathbf{p}_i||} \quad \text{and} \quad \mathbf{v} = \frac{\mathbf{p}_{i-1} - \mathbf{p}_i}{||\mathbf{p}_{i-1} - \mathbf{p}_i||}\,,$$

the equation becomes

$$\mathbf{q} = \mathbf{p}_i + r\,\mathbf{u} + s\,\mathbf{v}, \quad r,\; s \in \mathbb{R}\,.$$

The unit tangent vector can be defined as the vector in the direction of the right-hand derivative or simply as the unit vector in the direction of the joint's second link

$$\check{\mathbf{t}} = \mathbf{u}\,.$$

The principal unit normal vector $\check{\mathbf{n}}$ is defined as the unit vector in direction of the interior bisector of the smaller angle between the two connected links:

$$\check{\mathbf{n}} = \frac{0.5\mathbf{u} + 0.5\mathbf{v}}{||0.5\mathbf{u} + 0.5\mathbf{v}||}\,.$$

The binormal vector in point $\mathbf{p}_i$ is the unit vector orthogonal to the osculating plane in $\mathbf{p}_i$:

$$\check{\mathbf{b}} = \frac{\mathbf{u}\,\times\,\mathbf{v}}{||\mathbf{u}\,\times\,\mathbf{v}||}\,.$$

□

Figure 1.17 shows a piecewise linear curve consisting of three links, where the third link is given at two different stages (dotted, solid). For both stages, the principal normal vector $\check{\mathbf{n}}_i$ in point $\mathbf{p}_i$ is shown. For the first stage (dotted) $\check{\mathbf{n}}_i$ lies in the osculating plane of $\mathbf{p}_{i-1}$. In the second stage, it lies in $\mathbf{p}_i$'s osculating plane. Furthermore, the binormal vector $\check{\mathbf{b}}_{i-1}$ in point $\mathbf{p}_{i-1}$ is shown.

Based on Definition 1.4.12, the notions curvature and torsion are analogously defined for piecewise linear curves.

Definition 1.4.14 (Curvature κ and torsion τ of piecewise linear curves). Given is a shaft $\bar{\mathbf{S}} = (S,\, n,\, l,\, d, u,\, v\; r,\, P_{l,d})$, with $S = \bar{\mathbf{L}}_n \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_i = (\mathbf{F}_i,\, l,\, d)$. Let $(\bar{\mathbf{L}}_{i+1} \circ \bar{\mathbf{L}}_i) \in J_{u,\,v,\,\theta}$, for all $i = 1, \ldots, n-1$.

The curvature of $\bar{\mathbf{S}}$ in joint i is given by

$$\kappa(S,\, i) = ||\check{\mathbf{t}}_i - \check{\mathbf{t}}_{i+1}|| = 2\tan\left(\frac{j\theta}{2}\right),\ \text{with } j \in [1, v]\,,$$

where j denotes the number of increments, by which link $\bar{\mathbf{L}}_{i+1}$ was rotated.

A measure for the torsion in point $\mathbf{p}_i$ is given by the dot product of the binormal vector in $\mathbf{p}_{i-1}$ and the principal normal vector in $\mathbf{p}_i$:

$$\tau(S,\, i) = |\check{\mathbf{b}}_{i-1} \cdot \check{\mathbf{n}}_i|\,,$$

where $|\cdot|$ denotes the absolute value of its argument. Since both factors are unit vectors, the dot product is in $[-1, 1]$ and the torsion is in [0,1]. □

Figure 1.17 makes it clear, how the torsion changes, as the third link moves out of $\mathbf{p}_{i-1}$'s osculating plane (dotted) into the new position. For the initial position, $\check{\mathbf{n}}_i$ lies in $\mathbf{p}_{i-1}$'s osculating plane and the dot product with the binormal vector is zero. As point $\mathbf{p}_{i+1}$ moves out of the plane towards the new position, the angle between $\check{\mathbf{n}}_i$ and $\check{\mathbf{b}}_{i-1}$ gets smaller and the dot product becomes larger (towards one). Would $\mathbf{p}_{i+1}$ move in the other direction, the angle between $\check{\mathbf{n}}_i$ and $\check{\mathbf{b}}_{i-1}$ would get bigger and the dot product becomes smaller (toward -1). In both cases, the torsion increases.

1.4.1.7 Bending Section and Endoscope

Given the previous definitions and the identifier declared in Figure 1.18 a bending section can now formally be described as:

Definition 1.4.15 (Bending section $\bar{\mathbf{B}}$). A bending section $\bar{\mathbf{B}}$ consists of two links (sleeves) interconnected by a elastic shaft:

$$\bar{\mathbf{B}} = \bar{\mathbf{L}}_{s2} \circ \bar{\mathbf{S}}^{\mathrm{E}} \circ \bar{\mathbf{L}}_{s1},$$

with

$$\bar{\mathbf{L}}_{s1} = (\mathbf{F}_{s1},\, l_1,\, d_b)$$

Configuring the Endoscope Features ...			
e	Endoscope length	n	Number of links
s	Shaft length	d_s	Diameter of the shaft
b	Bending section length	d_b	Diameter of the bending section
... by Specifying:			
l	Length of shaft link	l_b	Length of bending section link
l_1	Length of first sleeve	n_s	Number of shaft links
l_2	Length of second sleeve	n_b	Number of bending section links

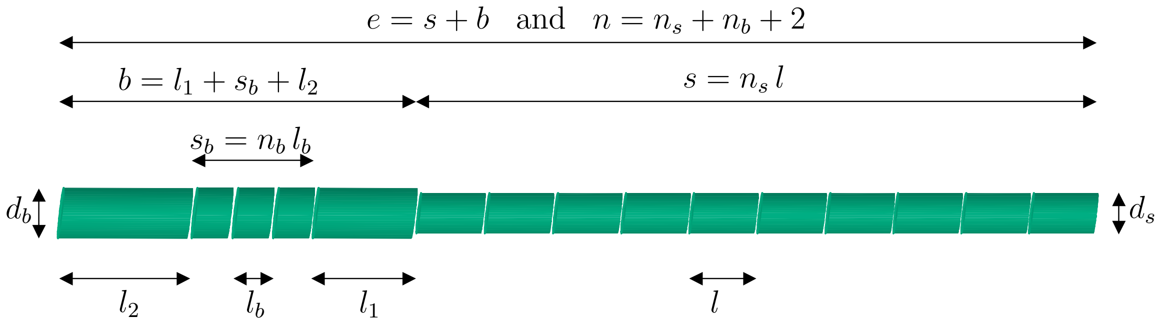

Figure 1.18: Identifiers for the endoscope specification parameters.

is the first link,

$$\bar{\mathbf{L}}_{s2} = (\mathbf{F}_{s2},\, l_2,\, d_b)$$

is the second link and

$$\bar{\mathbf{S}}^{\mathrm{E}} = (S^E,\, n_b,\, l_b,\, d_b,\, u,\, v,\, r,\, P_{l,d},\, \alpha,\, \beta)$$

is an elastic shaft and

$$((\bar{\mathbf{L}}_{s2} \circ \bar{\mathbf{L}}_{n_b}) \in J_{u,v,\theta}) \quad \wedge \quad ((\bar{\mathbf{L}}_1 \circ \bar{\mathbf{L}}_{s1}) \in J_{u,v,\theta})\,, \tag{1.22}$$

where

$$S^E = \bar{\mathbf{L}}_{n_b} \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1\,. \qquad \square$$

The first (second) condition in Equation 1.22 says that the first (last) link of the bending section is connected to the first (last) link of the elastic shaft via a discrete joint.

Given the definition of a bending section, an endoscope can be described as:

Definition 1.4.16 (Endoscope $\bar{\mathbf{E}}$). An endoscope is an elastic shaft, connected to a bending section, via a discrete joint:

$$\bar{\mathbf{E}} = \bar{\mathbf{B}} \circ \bar{\mathbf{S}}^{\mathrm{E}},$$

with

$$\bar{\mathbf{S}}^{\mathrm{E}} = (S^E,\, n_s,\, l,\, d_s,\, u,\, v,\, r,\, P_{l,d},\, \alpha,\, \beta)$$

an elastic shaft and

$$\bar{\mathbf{B}} = \bar{\mathbf{L}}_{s2} \circ \bar{\mathbf{S}}^{\mathrm{E}}_b \circ \bar{\mathbf{L}}_{s1}$$

a bending section and

$$(\bar{\mathbf{L}}_{s1} \circ \bar{\mathbf{L}}_{n_t}) \in J_{u,v,\theta}\ , \tag{1.23}$$

where

$$S^E = \bar{\mathbf{L}}_{n_t} \circ \ldots \circ \bar{\mathbf{L}}_2 \circ \bar{\mathbf{L}}_1\ . \qquad \square$$

Condition 1.23 requires that the last link of the elastic shaft is connected to the first sleeve of the bending section via a discrete joint.

Example: Let the endoscope shown in Figure 1.18 be given by a concatenation of 15 links, where the start link is given with respect to the world coordinate system. For this example an implementation is assumed, where each link, beside the start link is given with respect to its predecessor. Let the position of the endoscope's light source (or the outlet of the working channel, containing a needle) be given by a vector $\mathbf{p}$, where $\mathbf{p}$ is known with respect to the second sleeve's coordinate system $\mathbf{F}_{s2}$. Then the position $\mathbf{q}$ of the light source in world coordinates is given by

$$\mathbf{q} = \mathbf{F}_{s2}\,\mathbf{F}_{b3}\,\mathbf{F}_{b2}\,\mathbf{F}_{b1}\,\mathbf{F}_{s1}\,\mathbf{F}_{10}\,\mathbf{F}_{9}\,\mathbf{F}_{8}\,\mathbf{F}_{7}\,\mathbf{F}_{6}\,\mathbf{F}_{5}\,\mathbf{F}_{4}\,\mathbf{F}_{3}\,\mathbf{F}_{2}\,\mathbf{F}_{1}\,\mathbf{p}\,,$$

where the concatenation of the links is realized by matrix multiplications.

1.4.2 A Model for Aligning the Instrument with a Target

To insert an endoscope needle into a target, the endoscopist first aligns the endoscope's tip with the target, before pushing out the needle. There are three parameters that describe this alignment: Shaft rotation α, angle of tip deflection β and needle length d. This section presents a formula for each alignment parameter, based on the instrument model described in the previous sections.

While the calculation of shaft rotation α and needle length d is relatively straightforward, calculation of the angle of tip deflection β requires a model of the endoscope's active tip deflection. Generally, bending of the endoscope's tip can be caused by two forces: Passive deformation, caused by collisions with the organ wall and active deformation, caused by the physician rotating the bending wheel in the endoscope's control head (see Figure 1.2, left). Experiments showed that the endoscopes's tip deforms passively in the same way as the shaft, if the bending wheel is left untouched.

Passive deformation is naturally taken into account, by modelling the bending section in the same way, the endoscope's shaft is modelled: as a sequence of links, interconnected by joints. By assuming the same maneuverability for the bending section joints and for the shaft joints, the endoscope's tip deforms passively in the same way, the entire endoscope does.

In the following, a model for active tip deformation is presented. The OLYMPUS GIF-100, as described in Section 1.1 served as a reference. Experiments with the GIF-100 resulted in the following four observations:

1. The endoscope's first sleeve remains stationary during active bending.
2. The endoscope's tip moves in a plane during active bending.
3. The movement of the second sleeve's center line during active bending can be approximated by the radii of a semi-circle.
4. A straight tip should always be the zero-position for measuring the angle of tip defection, even though the tip was actually passively bend.

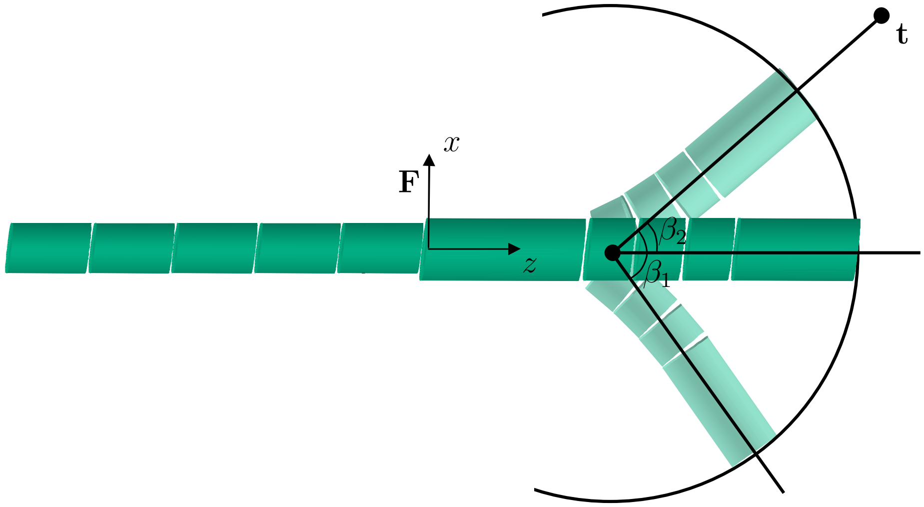

Figure 1.19: A model for active tip deflection.

Figure 1.19 visualizes the above observations. A reference frame $\mathbf{F} \in \mathcal{M}$ was attached to the endoscope's first sleeve. During active bending, this link can be assumed to remain stationary and the tip can be assumed to move in $\mathbf{F}$'s $y = 0$-plane. The center line of the link representing the second sleeve coincides with the radii of the shown semi-circle. For the GIF-100, the center of this circle is 49 mm away from the tip. To understand the last point in the list above, consider an endoscope that was passively bend into the lower position shown in Figure 1.19. The wheel rotation required to bend the tip into the upper position is equal to the required wheel rotation starting from a straight tip. In other words, regarding the required wheel rotation, β_1 equals β_2. This effect is probably caused by the pull-wire mechanism, which connects the tip to the bending wheel. Note that the first sleeve's z-axis corresponds to a straight tip.

To calculate α, β and d required to maneuver the needle tip into a target in 3D space, the following steps have to be taken:

Let l denote the distance between the origin of $\mathbf{F}$ and the center of the semi-circle. Move the first sleeve's reference frame $\mathbf{F}$ l mm along its positive z-axis (straight tip):

$$\mathbf{F} = \mathbf{F}\,\mathbf{T}(\hat{\mathbf{z}}\,l)\,.$$

Let $\mathbf{F}$ and target point $\mathbf{t} \in \mathbb{R}^3$ be given with respect to the same world coordinate system. Apply a coordinate transformation, such that $\mathbf{t}$ is known with respect to $\mathbf{F}$:

$$\mathbf{t}' = \mathbf{F}^{-1}\mathbf{t} \,.$$

Project $\mathbf{t}'$ into $\mathbf{F}$'s $z = 0$ plane and denote the resulting point by $\mathbf{t}''$:

$$\mathbf{t}'' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{t}' \,.$$

Calculate the angle between vector $\mathbf{t}''$ and vector $\hat{\mathbf{x}}$, which denotes $\mathbf{F}$'s x-axis. This is the shaft rotation (about the z-axis), which rotates the bending plane into the target:

$$\alpha = \arccos\left(\frac{\mathbf{t}'' \cdot \hat{\mathbf{x}}}{||\mathbf{t}''||\,||\hat{\mathbf{x}}||}\right) \,. \tag{1.24}$$

After rotating the shaft by α degrees, $\mathbf{t}'$ lies in $\mathbf{F}$'s xz-plane. Calculate the angle between vector $\mathbf{t}'$ and vector $\hat{\mathbf{z}}$. This corresponds to β_2 in Figure 1.19, the angle between a straight tip and the target:

$$\beta = \arccos\left(\frac{\mathbf{t}' \cdot \hat{\mathbf{z}}}{||\mathbf{t}'||\,||\hat{\mathbf{z}}||}\right) \,. \tag{1.25}$$

Calculate the difference between the length of vector $\mathbf{t}'$ and the radius of the semicircle r. This is the required needle length, measured with respect to the working channel's outlet:

$$d = ||\mathbf{t}'|| - r \,. \tag{1.26}$$

After β is determined, the corresponding angle of wheel rotation has to be calculated. At first a calibration table has to be created from experiments with the endoscope. The table should list for a set of equidistant wheel angles the corresponding tip angles, measured with respect to a straight tip. Then, the two closest values to the tip angle have to be looked up in the calibration table and interpolated with the corresponding wheel angles. The result is an approximation for the wheel angle required to bend the endoscope's tip for alignment with the target.

1.4.3 Algorithmic Description

This section gives an algorithmic description of the flexible instrument model. Two main algorithms are described: Algorithm 1.3 "createFST()" creates a filtered spatial tree. And Algorithm 1.9 "create∼FST()" represents a function that incorporates the $(n', k)-$ and $(u, v)-$approximation techniques described in Section 1.3.2 to speed-up the execution of Algorithm "createFST()". Both algorithms are formulated in a generic pseudo-code.

Algorithm "createFST() recursively builds a doubly linked tree data structure, where a path from the root to a leaf represents a flexible instrument in 3D space. The instrument with minimal deformation energy is returned in form of a pointer to its last link (leaf). By following this leaf to the root, the whole instrument can be assembled.

The algorithm consists of 3 main components. Firstly, Lines 4 and 5 (35, 36) represent the joint of the current link. The two loops iterate for each rotation axis over each rotation increment. Secondly, line 7 represents the filtering process as a concatenation of filter functions. Thirdly, in lines 10 - 33 the nodes are created and pointers are assigned. But more importantly, a decision is made whether or not to include the current link into the tree. To understand the significance of this, consider the following:

In order for all instruments to have exactly the same length, all paths from the root to a leaf have to consists of the same number n of nodes. In other words, a node (link) at recursion depth i can not be linked to the tree, until at least one valid path of length $n - i$ (starting at i) was found. Since the decision whether or not to include a node into the tree depends on the "future", the following preorder / postorder mechanism is used: At first, a node is included by default into the tree (lines 11 - 14) before the recursion is called (preorder). Then the recursion is called (line 17) and the flag it returns is evaluated ("future").

Type	Identifier	Description
`Input:`	$\bar{\mathbf{L}} \in L$	Start link
	$node \leftarrow root$	Root of the spatial tree
	$energy \leftarrow 0$	Deformation energy
	$r \leftarrow 0$	Recursion depth
`Output:`	$root$	Spatial tree which represents the instrument's workspace
	$instrument$	Leaf, which if followed to the root, represents a flexible instrument of minimal deformation energy
`Variables:`	$isLinkAttached$	**true**, if at least one link was attached to $\bar{\mathbf{L}}$, else **false**
	$\bar{\mathbf{L}}' \in L$	Link attached to $\bar{\mathbf{L}}$
	$node'$	Son of *node*
	$leaf$	Node without a son
	$minEnergy \leftarrow \infty$	Minimal deformation energy
`Subroutines:`	$f_{\mathsf{link}}()$	Link filter, see 1.4
	$f_{\mathsf{joint}}()$	Joint filter, see 1.5
	$f_{\mathsf{boundingTube}}()$	Bounding tube filter, see 1.7
	$f_{\mathsf{geometry}}()$	Geometry filter, see 1.8
	$f_{\mathsf{energy}}()$	Energy filter, see 1.6
`Parameters:`	n	Instrument length (number of links)
	u	Number of rotation axes
	v	Number of rotation increments
	θ	Angle increment
	α	Material property: resistance to bending
	β	Material property: resistance to twisting
	p	Filter selectivity for deformation energy

Table 1.2: Specification for Algorithm 1.3

Algorithm 1.3 createFST(): Creates a **F**iltered **S**patial **T**ree

`Specification:` see Table 1.2

```
createFST$_{n,u,v,\theta,\alpha,\beta,p}$ ($\bar{\mathbf{L}}$, node, energy, r)
$r \leftarrow r + 1$
$isLinkAttached \leftarrow$ false

for $i \leftarrow 1$ to $u$ do                          // loop over all rotation axes
   for $j \leftarrow 1$ to $v$ do                       // loop over all rotation angle increments

      $\bar{\mathbf{L}}' \leftarrow f_{\mathsf{geometry}}(f_{\mathsf{boundingTube}}(f_{\mathsf{energy}}(energy, minEnergy, f_{\mathsf{joint}}(i, j, f_{\mathsf{link}}(r, \bar{\mathbf{L}})))))$

      if $\bar{\mathbf{L}}' \neq$ NULL then               // if link passed all filters
         if $r < n$ then                                 // if instrument length not reached yet
            $node' \leftarrow$ create node
            insert $\bar{\mathbf{L}}'$ into $node'$
            make $node'$ son of $node$
            make $node$ son of $node'$                            // backward pointer
            $energy \leftarrow energy + E(\bar{\mathbf{L}} \circ \bar{\mathbf{L}}', \alpha, \beta, p)$      // see Def. 1.4.10

            if createFST$_{n,u,v,\theta,\alpha,\beta,p}$ ($\bar{\mathbf{L}}'$, $node'$, $energy$, $r$) = true then
               $isLinkAttached \leftarrow$ true
            else
               remove son $node'$ from $node$
            end if
         else                                              // desired instrument length reached
            $leaf \leftarrow$ create leaf
            insert $\bar{\mathbf{L}}'$ into $leaf$
            make $leaf$ son of $node$
            make $node$ son of $leaf$                             // backward pointer
            $isLinkAttached \leftarrow$ true

            if $energy < minEnergy$ then
               $minEnergy \leftarrow energy$
               $instrument \leftarrow leaf$
            end if
         end if
      end if
   end for
end for
return ($isLinkAttached$)
```

The flag *isLinkAttached* is true, if at least one link could have been successfully attached to the current link, otherwise the flag is false. If the flag is false, the link included by default is removed from the tree (postorder, line 20). The flag *isLinkAttached* is initialized with false (line 2). If the current link was successfully attached and if it is the last link (leaf, $r = n$), then the flag is set to true (line 27). To make sure that all predecessors of that link are not removed, the flag is returned to the calling node (line 37) where it is again set to true (line 18), and so on.

Algorithm 1.4 represents the link filter. It creates a link $\bar{\mathbf{L}}'$, makes assignments to its components $\mathbf{F}', l'$ and d' and returns $\bar{\mathbf{L}}'$ to the calling procedure. The assignments for link length l' and diameter d' depend on where the link is located within the endoscope (see Figure 1.18) and therewith on the current recursion depth r.

Algorithm 1.4 $f_{\mathsf{link}}()$: link filter

`Input:`	r	recursion depth
	$\bar{\mathbf{L}}$	link
`Output:`	$\bar{\mathbf{L}}'$	link
`Measures:`		see Figure 1.18

```
$\mathbf{f}_{\mathsf{link}}(r, \bar{\mathbf{L}} = (\mathbf{F}, l, d))$
  create link $\bar{\mathbf{L}}' = (\mathbf{F}', l', d')$
  $\mathbf{F}' = \mathbf{F}\,\mathbf{T}(\hat{\mathbf{z}}s)$
  $l' = \begin{cases} l & : \quad 1 \le r \le n_s & \text{shaft} \\ l_1 & : \quad r = n_s + 1 & \text{first sleeve} \\ l_b & : \quad n_s + 2 \le r \le n_s + n_b + 1 & \text{flex. bending} \\ l_2 & : \quad r = n_s + n_b + 2 & \text{last sleeve} \end{cases}$
  $d' = \begin{cases} d_s & : \quad 0 \le r \le n_s & \text{shaft} \\ d_b & : \quad n_s + 1 \le r \le n & \text{bending sec.} \end{cases}$
  return $\bar{\mathbf{L}}'$
```

Algorithm 1.5 $f_{\mathsf{joint}}()$: joint filter

```
Input:   $i$                  rotation axis
         $j$                  rotation angle (step)
         $\bar{\mathbf{L}}$   link
Output:  $\bar{\mathbf{L}}'$  link

$\mathbf{f}_{\mathsf{joint}}(i,\ j,\ \bar{\mathbf{L}} = (\mathbf{F},\ l,\ d))$
1: create link $\bar{\mathbf{L}}' = (\mathbf{F}',\ l',\ d')$
2: $\mathbf{F}' = \mathbf{F}\,\mathbf{R}[i, j]$
3: $l' = l$
4: $d' = d$
5: return $\bar{\mathbf{L}}'$
```

Algorithm 1.5 represents the joint filter. It creates a link $\bar{\mathbf{L}}'$, makes assignments to its component $\mathbf{F}'$ and returns $\bar{\mathbf{L}}'$ to the calling procedure. In line 2 a pre-computed look-up table of rotation matrices is used.

Algorithm 1.6 represents the energy filter. It compares the current energy with the minimum energy and returns either `NULL` or just passes through the input link.

Algorithm 1.6 $f_{\mathsf{energy}}()$: energy filter

```
Input:   $e$                         energy
         $e'$                        current energy minimum
         $\bar{\mathbf{L}}$          link
Output:  $\bar{\mathbf{L}}$, NULL    link or NULL

$\mathbf{f}_{\mathsf{energy}}(e,\ e',\ \bar{\mathbf{L}})$
if $e > e'$ then
    return NULL
else
    return $\bar{\mathbf{L}}$
end if
```

Algorithm 1.7 $f_{\mathsf{boundingTube}}()$: bounding tube filter

```
Input:   $\bar{\mathbf{L}}$              link

Output:  $\bar{\mathbf{L}}$, NULL        link or NULL

$\mathbf{f}_{\mathsf{boundingTube}}(\bar{\mathbf{L}} = (\mathbf{F}, l, d))$
 1: if $\bar{\mathbf{L}}$ = NULL then
 2:    return NULL
 3: else
 4:    $\mathbf{p} \leftarrow \mathbf{F}\,\mathbf{0}$
 5:    if $\mathbf{p}$ lies inside the bounding tube then
 6:       return $\bar{\mathbf{L}}$
 7:    else
 8:       return NULL
 9:    end if
10: end if
```

Algorithm 1.7 represents the bounding tube filter. The bounding tube filter (see Equation 1.3) filters all links to define a coarse region of interest (ROI). This is important for example for branching anatomies such as the tracheobronchial tree or the vasculature. In the existence of bifurcations, the bounding tube filter "cuts-off" side branches to guide the instrument model directly towards the specified target branch. The ROI can be defined preoperatively, based on an insertion protocol.

In line 4 the origin of the reference frame attached to the link is determined. Coordinate system **F** is assumed to be given with respect to the world coordinate system. For an implementation, where each link (beside the start link) is given with respect to its predecessor, see Definition 1.4.8 on Page 49. Dependent of whether the origin lies inside the bounding tube or not, either the input link is passed through to the calling procedure, or `NULL` is returned.

Algorithm 1.8 $f_{\mathsf{geometry}}()$: geometry filter

```
Input:   $\bar{\mathbf{L}}$           link

Output:  $\bar{\mathbf{L}}$, NULL     link or NULL

$\mathbf{f}_{\mathsf{geometry}}(\bar{\mathbf{L}} = (\mathbf{F}, l, d))$
 1: if $\bar{\mathbf{L}}$ = NULL then
 2:    return NULL
 3: else
 4:    for all $\mathbf{p}$ in $P_{l,d}$ do
 5:       $\mathbf{p}' \leftarrow \mathbf{F}\,\mathbf{p}$
 6:       if $\mathbf{p}'$ outside geometry then
 7:          return NULL
 8:       end if
 9:    end for
10:    return $\bar{\mathbf{L}}$
11: end if
```

Algorithm 1.8 represents the geometry filter. It checks for all cylinder surface points in $P_{l,d}$, if the corresponding point on the current link (line 5) lies within the organ geometry (line 6).

Function 1.9 "create∼FST()" represents an algorithm that incorporates the $(n', k)-$ and $(u, v)-$approximation techniques described in Section 1.3.2 to speed-up the execution of algorithm "createFST()". In line 6 algorithm "createFST()" is called, however with modified parameters: Instead of an instrument length of n, a significantly shorter length n' is used. And, instead of v rotation increments, just one fixed angle is used.

Algorithm 1.9 create∼FST(): (n', k), (u, v)-approximation of createFST()

`Input:`	n'	length of sub-instrument
	$k < n'$	length of segment
	u	number of rotation axes
	v	number of rotation increments
`Output:`	*instrument*	List of links, which represents a flexible instrument of minimal deformation energy
`Variables:`	$\bar{\mathbf{L}}$	start link of sub-instrument
	minEnergy	minimal deformation energy
	γ	current rotation angle
	spatialTree	root of the spatial tree
	leaf	Node, which if followed to the root, represents a flexible instrument
	I, I_{min}, I^k_{min}	list of links, which represents an instrument
	energy	deformation energy
`Parameters:`	$\bar{\mathbf{L}}_0$	start link
	n	overall instrument length (number of links)
	θ	angle increment
	α, β, p	energy filter parameters (see Table 1.2)

```
create∼FST$_{\bar{\mathbf{L}}_0, n, \theta, \alpha, \beta, p}$ $(n', k, u, v)$
$\bar{\mathbf{L}} \leftarrow \bar{\mathbf{L}}_0$
for $i \leftarrow 1$ to $\lceil \frac{n-n'}{k} \rceil + 1$ do                // loop over all sub-instruments
    $minEnergy \leftarrow \infty$
    for $j \leftarrow 1$ to $v$ do                      // loop over all rotation angle increments
        $\gamma \leftarrow j\,\theta$
        $leaf \leftarrow$ createFST$_{n', u, 1, \gamma, \alpha, \beta, p}(\bar{\mathbf{L}}, spatialTree, 0, 0)$
        $I \leftarrow$traverse $spatialTree$ from leaf $leaf$ to its root
        $energy \leftarrow$ calculate deformation energy of $I$
        if $energy < minEnergy$ then
            $minEnergy \leftarrow energy$
            $I_{min} \leftarrow I$
        end if
    end for
    $I^k_{min} \leftarrow$ copy first $k$ links of instrument $I_{min}$
    $instrument \leftarrow$ attach $I^k_{min}$ to the end of $instrument$
    $\bar{\mathbf{L}} \leftarrow$ copy $k$-th link from $I^k_{min}$
end for
```

1.5 Model Validation

This section describes two experiments to validate the flexible instrument model, described in the previous sections. A general approach to validate a mathematical model of a real world object, is to measure the object with respect to some world coordinate system, to transform the measures into the model coordinate system and to subsequently compare both representations. If the real world object interacts with its environment, a mathematical model of this environment is needed as well.

The biggest problem in this scenario is to measure the real world object without using procedures that influence or alter the object's shape. The object should be acquired in its natural behavior, uninfluenced by external measuring devices. Section 1.5.1 presents an approach to measure a flexible endoscope without changing its shape or behavior. It further describes an experiment that validates the accuracy of these measurements.

Section 1.5.2 uses the technique developed in the previous section to validate the flexible instrument model. The idea is to insert the real endoscope into an artificial calibration phantom and to measure its shape. Also, the digital endoscope model is inserted into a digital model of the calibration phantom. Then, by using a rigid body transformation, the measures of the real endoscope are transformed into the coordinate system of the endoscope model and the two shapes are compared.

In both experiments described in this section, the endoscope's shape is measured, lying on a planar surface. This was done to facilitate the comparison between the endoscope and its model and to allow for an unambiguously display of the result in a 2D image. However, all involved components do not make any explicit assumptions about a planar configuration of the endoscope. The endoscope model, the model of the environment and the measuring device are capable of working with arbitrary non-planar configurations in 3D space. Experiment 3.7 in Section 3.3 represents a fully 3D experimental setup, where the real endoscope's shape is measured, while inserted into a 3D lung phantom.

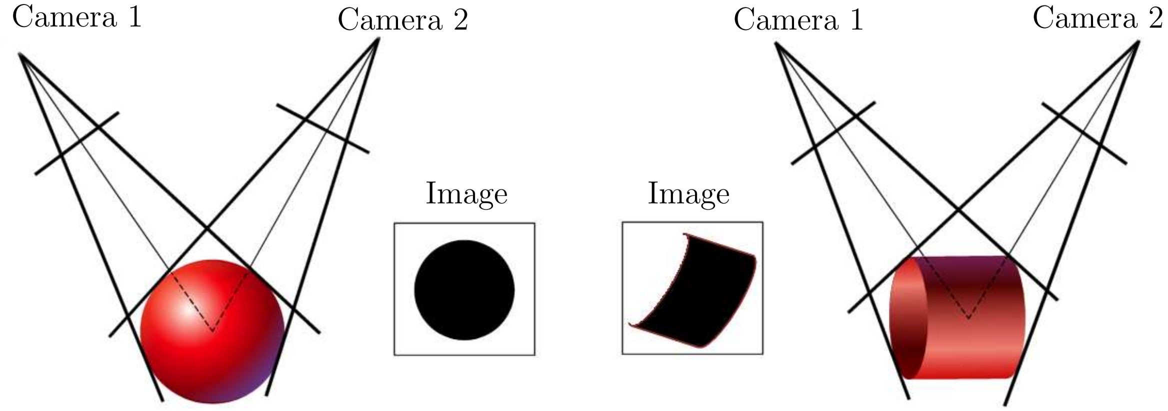

Figure 1.20: An optical tracking system using passive retro-reflective markers. Left: ball-shaped marker. Right: cylindrical marker.

1.5.1 Endoscope Tracking

An obvious approach to measure the shape of a real endoscope is to define a set of characteristic points that describe its shape and to use a tracking system, to track these points in 3D space. Tracking a point in 3D space is usually done by attaching a sensor or marker to it and to track the sensor or marker using a magnetic or optical tracking system. An important requirement regarding the sensor or marker is that they should not significantly increase the size of the endoscope. Hence, magnetic tracking systems are not feasible, since their sensors are quite large and require a cable for each sensor. For the same reason, optical systems, using active markers like IR (infrared) light emitting diodes (LEDs) are not feasible as well. However, optical trackers using passive retro-reflective markers do not have the cable problem. Two types of passive markers are used that could in principle be used for tracking an endoscope: flat disk-shaped markers and ball-shaped markers. For the first type the tracking system tracks the center of the disk, for the latter, the center of the ball.

Assuming a circular cross section of the flexible tube, its shape is best described by its center line or middle axis together with the shaft radius. However, tracking the center line of a flexible tube with disk- or ball-shaped markers is difficult for two reasons: Firstly, the center line itself is inaccessible and secondly, at least three markers have to be attached per point and have to be visible by the IR cameras, in order to non-ambiguously determine the cross sectional circle and therewith the center. Especially the second reason makes the use of ball markers unfeasible, since three of them would largely alter size and shape of the endoscope. The disk marker would not significantly change the endoscope's shape, but it creates the problem of attaching them in a way that at least three are always visible by the IR cameras. What is needed, is a new type of marker, which is passive (does not require cables), is thin and always visible by the IR cameras.

A New Type of Marker:

To design a new type of marker, which works with standard commercially available tracking systems, a deeper understanding of how the position of a standard marker in 3D space is determined, is needed. Figure 1.20 (left) illustrates the principal functionality of an optical infrared stereo tracking system for ball-shaped markers. Each of the two calibrated cameras sees a projection of the ball which is always a perfect disk, assuming a perfect correction for lens distortion. For each image, the center of the disk in the image plane is determined. Assuming that the intrinsic and extrinsic camera parameters are known, a triangulation based on the two center points is used to determine the center of the ball.

The advantage of the ball-shaped marker is that all projections are circles. For cylinders, at least all projections with the optical axis perpendicular to the cylinder axis are rectangles. If furthermore the height of the cylinder is equal to its diameter, all projections are squares. Figure 1.20 (right) shows the scenario for non perpendicular axes.

By using cylindrical markers, there is a possibility that the tracking system's built-in software is able to accurately determine the center of the cylinder's rectangular projections and to correctly triangulate the cylinder's center. The idea is, to wrap a strip of retro-reflective material around the shaft of the endoscope, such that its projection is a square. Figure 1.21 shows two photographs (with and without flash

Figure 1.21: Left: A photograph of an endoscope with 12 cylindrical markers. Right: The light of the camera's flash is reflected by the markers.

illumination) of an endoscope with 12 cylindrical markers.

Since the accuracy of regular off-the-shelf tracking systems is specified only for ball- or disk-shaped markers, their accuracy regarding cylindrical markers was determined with the following experiment:

Experiment 1.3: Tracker Accuracy Using Cylindrical Markers

Objective: *Determining the accuracy of an optical tracking system, which is designed for tracking disk- and ball-shaped markers, to track the center of cylindrical markers in 3D-space.*

Method: *The idea is to place the endoscope on a planar surface and to measure the distance between the center line of the endoscope (given by the center of each cylindrical marker) and the surface. For a perfectly accurate measure the expected value is the radius of the endoscope shaft, plus the marker thickness.*

Platform: *SGI 540 PC, Pentium 3 single, 550 MHz, 500 MB; Graphics board: Cobalt Graphics, 48 MB.*

Material: *Optical tracking system ARTtrack*[2] *1, comprising of two IR cameras and software;*

Video endoscope OLYMPUS GIF-100, 4.75 mm radius;

---

[2] *A.R.T. (Advanced Realtime Tracking) GmbH, Gewerbestr. 17, D-82211 Herrsching, URL: www.ar-tracking.de*

30 passive retro-reflective strip markers (10 mm width, 0.5 mm thickness) wrapped around the endoscope's shaft like a cylinder in a distance of ca. 25 mm (see Figure 1.21);

Planar board with three disk markers;

"Matlab[3]", mathematical computing and visualization software.

Design: *The surface plane of the board was determined using the three disk markers. The endoscope was placed flat on the board and the marker positions were recorded. Since both measures were given with respect to the same coordinate system, the shortest distances between the markers and the plane were directly computed and visualized using Matlab. For the Matlab code, see Appendix A.1.*

Results: *The results show that using cylindrical markers allows to accurately determine the center line of the endoscope in 3D space. The tracking system could determine the position of 26 out of 30 markers. The average distance between the markers and the board is 5.17 mm (Standard deviation: 0.8 mm), given a 5.25 mm overall radius (endoscope radius + marker thickness). These results are expected to be also valid for arbitrary non-planar marker configurations, given that a big enough fraction of each marker is visible in both tracking cameras.*

1.5.2 Calibration Phantom

The previous section describe a method to accurately determine the center line of an endoscope in 3D space. In this section, this method is used together with a calibration phantom for model validation. The idea is to insert the real endoscope into the calibration phantom and to measure its shape. Then, the digital endoscope model is inserted into a digital model of the calibration phantom. By transforming the measured endoscope shape from the coordinate system of the tracker camera into

---

[3] *The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA*

the coordinate system of the digital models, both shapes can be compared with each other.

For this purpose, the calibration phantom should be open, to allow a direct line of sight to the tracking cameras and simple enough so that a digital model can be easily obtained.

Experiment 1.4: Model Validation Using an Optical Tracking System and a Calibration Phantom

Objective: *Model validation and determination of the intrinsic model parameters.*

Method: *Comparing a real endoscope, inserted into a calibration phantom with the digital endoscope model, inserted into a digital model of the calibration phantom.*

Platform: *SGI 540 PC, Pentium 3 single, 550 MHz, 500 MB; Graphics board: Cobalt Graphics, 48 MB.*

Material: *Endoscope with retro-reflective markers and a tracking system as described in experiment 1.3;*

Calibration phantom, which consists of an open, "M"-shaped 2D path with a box-shaped cross section. The path is 1050 mm long, 50 mm wide and is mounted on a planar surface.

A digital model of the calibration path, which was obtained by measuring the length of each section and the angle between two adjacent sections of the "M"-shaped path. Figure 1.23 shows a perspective and frontal view of the model.

Pointing device (see Figure 1.22) with a calibrated tip, which can be used to determine the coordinates of a point in 3D space, with respect to the tracking system described above.

"Matlab[4]", mathematical computing and visualization software.

---

[4] *The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA*

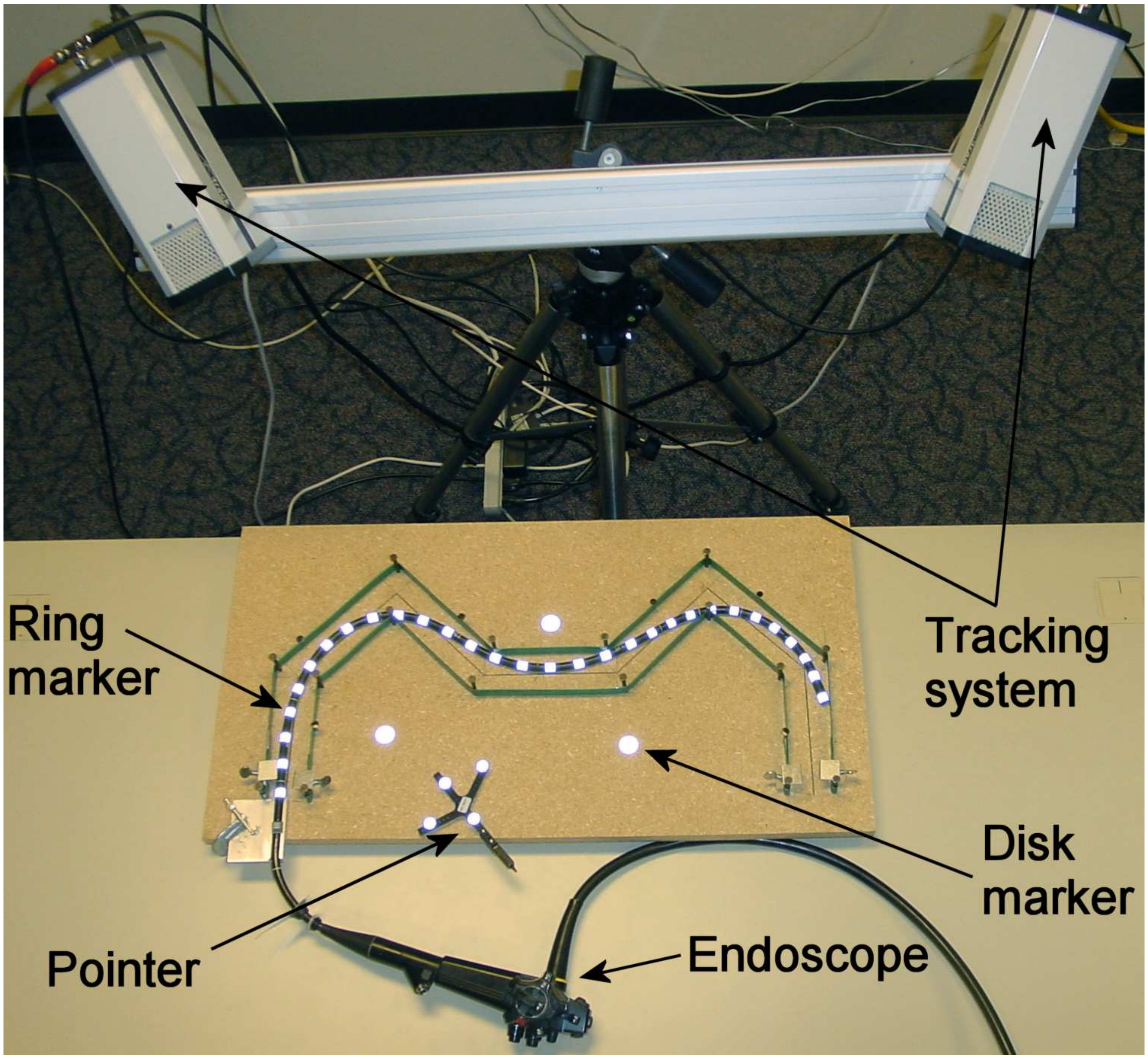

Figure 1.22: Experimental setup. OLYMPUS GIF-100 with retro-reflective markers, inserted into a "M"-shaped calibration path. The endoscope's shape (center line) was measured in 3D space, using an optical stereo tracking system.

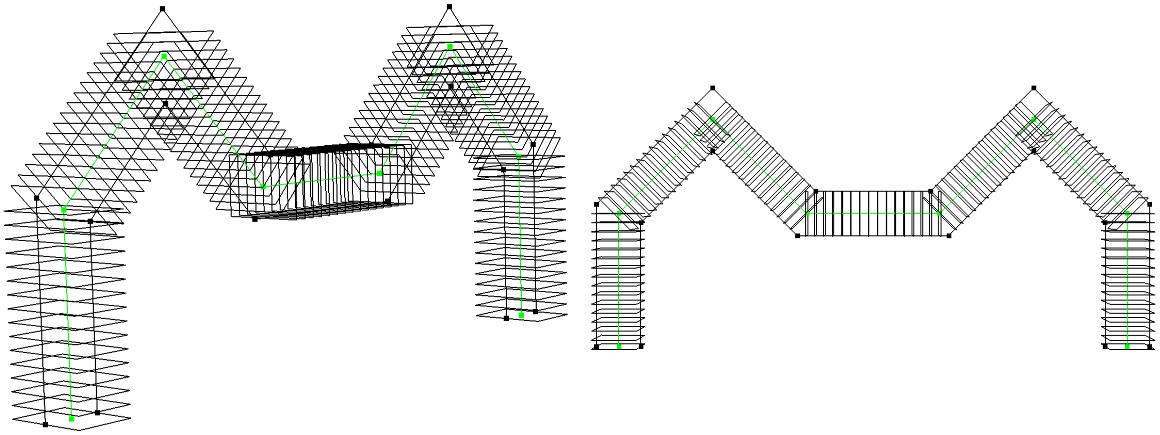

Figure 1.23: Digital model of the "M"-shaped calibration path. Left: Perspective view. Right: Frontal view. The eight corner points of the center line (green) serve as reference points for finding the rigid body transformation that maps a point from tracker camera space into model space.

Design: *The GIF-100 endoscope is inserted a distance of 900 mm into the "M"- path. After the insertion, a reading of all endoscope markers was recorded. To draw the measured markers and the digital model of the "M" in one common reference frame, a rigid body transformation has to be found that maps one coordinate frame into the other. To solve this registration problem, a method according to the ICP (Iterative Closest Point) algorithm [45] is used. At first, the best fit in the least-squares sense between a set of reference points obtained with the pointer and the corresponding virtual points is computed. Then, the resulting non-linear minimization problem is solved, using the "Levenberg-Marquardt" method [46] [47] [48]. See Appendix A.2 for Matlab code that computes the rigid body transformation. To determine the model parameters, the parameters were varied and the best match between the real and virtual instrument was chosen. The Hausdorff distance was taken as a measure for the match between the set of makers and the set of joint positions, because of its sensitivity to even a single "outlying" point. However, a perfect match, where a chain of links covers exactly all markers would in general not give a zero Hausdorff distance, unless all markers and joints are pairwise coincident. Hence, the Hausdorff* *distance was modified in a way that a polygon was created that connects all adjacent markers (joints) and the distance between a joint (marker) and the closest point on the polygon was calculated.*

Results: *Figure 1.24 top shows an example with* $n = 16$ *and* $k = 2$. *It shows as an intermediate result the first 13 segments, which means that 26 links out of* $n = 44$ *have been computed. For the bottom figure* $n' = 20$ *and* $k = 4$. *To take account of the frictional forces acting on the endoscope's tip,* $p = 7$ *(Equation 1.4) was set for the last segment and* $p = 1$ *was set for all others. The figure shows the best match between the markers (black balls) and the model. The model consists of 9 segments (color coded), the first 8 consist of 4, the last consists of 12 links. Each link is 20.45 mm long. By calculating the 7 smallest energies for the last segment, one segment that matched the last 6 markers by a (modified) Hausdorff distance of 0.8 mm was obtained.*

These results are expected to be also valid for arbitrary non-planar endoscope configurations. In fact, Experiment 3.7 in Section 3.3 shows similar results for an endoscope inserted into a 3D lung phantom. The reason is, that no explicit assumptions were made about the endoscope lying on a planar surface. Only the start link was placed to lie on the bottom of the 3D "M"-path model. The torsion term $E_\tau()$ *of the deformation energy function (see Equation 1.4) prevents the model from moving out of this plane.*

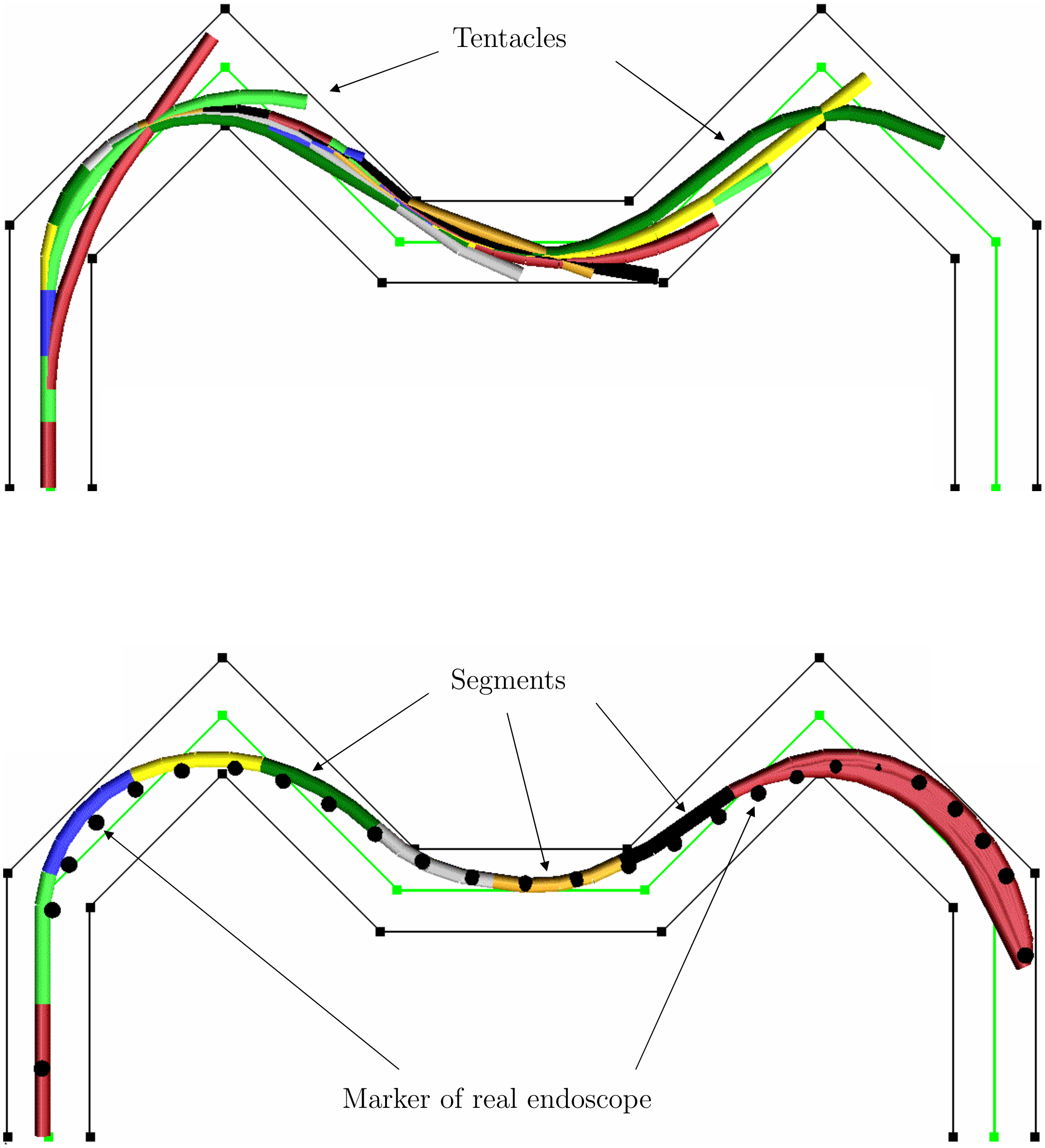

Figure 1.24: Endoscope model inserted into the calibration path model (without the cross sectional contours). Top: Intermediate result ($n = 44$, $n' = 16$, $k = 2$) after the first 13 segments were calculated. All "tentacles" are shown in full length. Bottom: Final result ($n = 44$, $n' = 20$, $k = 4$, $p = 7$), showing only the first $k = 4$ links of each tentacle. For the last segment the energy constraint has been relaxed by computing the $p = 7$ smallest energies. The endoscope model matches largely the measured markers of the real endoscope.

1.6 Results

This section presents results, demonstrating the model's realistic behavior, versatility, robustness and run-time efficiency. A total of 49 screen-shots are given, showing the instrument from different perspectives, inserted into various anatomies, at various depths and in various modes.

For a lung model, reconstructed form a scan of a patient, the instrument model was configured to represent a thin (5 mm diameter) and highly flexible bronchoscope. For a lung phantom, the instrument model was configured to represent the OLYMPUS GIF-100 Gastroscope, described in Section 1.1. And finally, for a brain artery model, reconstructed from a patient scan, the model was configured to represent a catheter. For each of these three scenarios Figure 1.25 to 1.34 present examples, demonstrating different aspects of the model, like its bending section geometry, tentacles, workspace, bounding tube and deformation energy.

Figures 1.32, 1.33 and 1.34 show results of three insertion simulations. The goal was to demonstrate the model's realistic behavior and robustness by producing animations, which show the instrument sliding down different anatomies. Starting with an initial insertion depth, a model of the respective instrument was calculated and the result was displayed. Then, the insertion depth was increased by a fixed step size and another model was calculated and displayed, and so on. The first group of four columns in Table 1.3 shows the overall insertion distance, the initial and final depth and the step size for the respective simulations. The second group of four columns shows the number of links that were needed to model the respective distances.

For example, as shown in the second row of Table 1.3, one simulation computes $n = 240$ instrument models of 140 to 380 mm length, with a 1 mm step size. Consecutively displaying the result of each model shows a continuous and smooth movement of the endoscope into the lung phantom (see Figure 1.33). This demonstrates the model's realistic behavior and robustness.

Another objective of calculating the animations was to demonstrate the model's

Insertion [mm]				Number of Links				Run-Time [s]				
				Shaft				Single Instr.			Cont. Ins.	
Overall dist.	From	To	Step size	From	To	BS[1]	Total max.	#[2]	Total	per ∼FST	last only	Frame rate
Lung phantom (active bending), ca. 14000 triangles, see Figure 1.32:												
120	140	260	2	1	6	4	10	60	60	1	4	ca.15
Lung phantom, ca. 14000 triangles, see Figure 1.33:												
240	140	380	1	1	10	4	14	240	27	0.11	3	ca.80
Brain artery, ca. 25000 triangles, see Figure 1.34:												
72	0	72	1	1	30	0	30	72	7	0.09	< 1	ca.72

[1] Bending Section. [2] Number of ∼FST calls (Overall dist. divided by step size)

Table 1.3: Run-time analysis of three insertion simulations. Hardware: PC, Pentium 4 dual processor, 1.3 GHz, 1 GB; Graphics board: nVIDIA GeForce3, 64 MB.

run-time efficiency. As described in Section 1.1, the goal of this Chapter was to develop a model that calculates a flexible instrument for a given insertion depth that resembles as closely as possible the shape of a real endoscope inserted by the same insertion depth. Consequently, the average time needed to calculate a single instrument model for a given anatomy and insertion depth was determined. A new and self-contained approximated FST was built for each step of the insertion simulation. For each step the ∼FST-algorithm was executed and the resulting shape was visualized. In other words, no information of the model at insertion depth x mm was used to compute a new model at insertion depth $x + 1$ mm, given a 1 mm step size. The first three columns of the third group in Table 1.3 show the timing results for single instruments. The first and second column show the number of ∼FST executions and the total time. Since all models were computed separately, column three (first column divided by second column) shows the average time needed to calculate a single instrument model, independent of a given length. This number could be useful for a physician to asses the algorithm's average run-time for a typical anatomy.

Although appropriate for determining the algorithm's average run-time for single instruments, the above described implementation is inefficient for calculating continuous insertion simulations. For such a task, only one spatial tree should be build in a single call to algorithm $\sim$FST, since all information accumulated for insertion step i can be used to calculate step $i+1$. In other words, if all information at step i were used in the computation at step $i+1$, the overall run-time for n steps would have been similar to the run-time needed for the last step n only. A tree of depth n includes all leaves of a tree of depth $n-1$, which includes all leaves of a tree of depth $n-2$ and so on. The last two columns of Table 1.3 give run-time information for a continuous insertion simulation. The first column gives the time needed for the last (longest) instrument of the first implementation. Based on this number, the second column ("last only" divided by "#") predicts a frame rate for a continuous insertion simulation, had the second implementation been used.

The first row of the Table demonstrates the model's ability to simulate an active bending of the endoscope's tip. To allow a higher flexibility for the bending section, the ranges of the corresponding joints are much higher than for shaft joints. Consequently, v, a joint's number of rotation steps is higher, which significantly increases the run-time for this model. The second example, which uses no active bending shows a much faster average run-time, regardless the higher number of links. The third example shows an even faster average run-time for about double the number of links. The reason for this lies in the better ratio between instrument diameter and organ diameter, which significantly reduces the number of collision detections. Also, the endoscope's starting length is five links (one shaft plus four bending section links), while the catheter's starting length is one link.

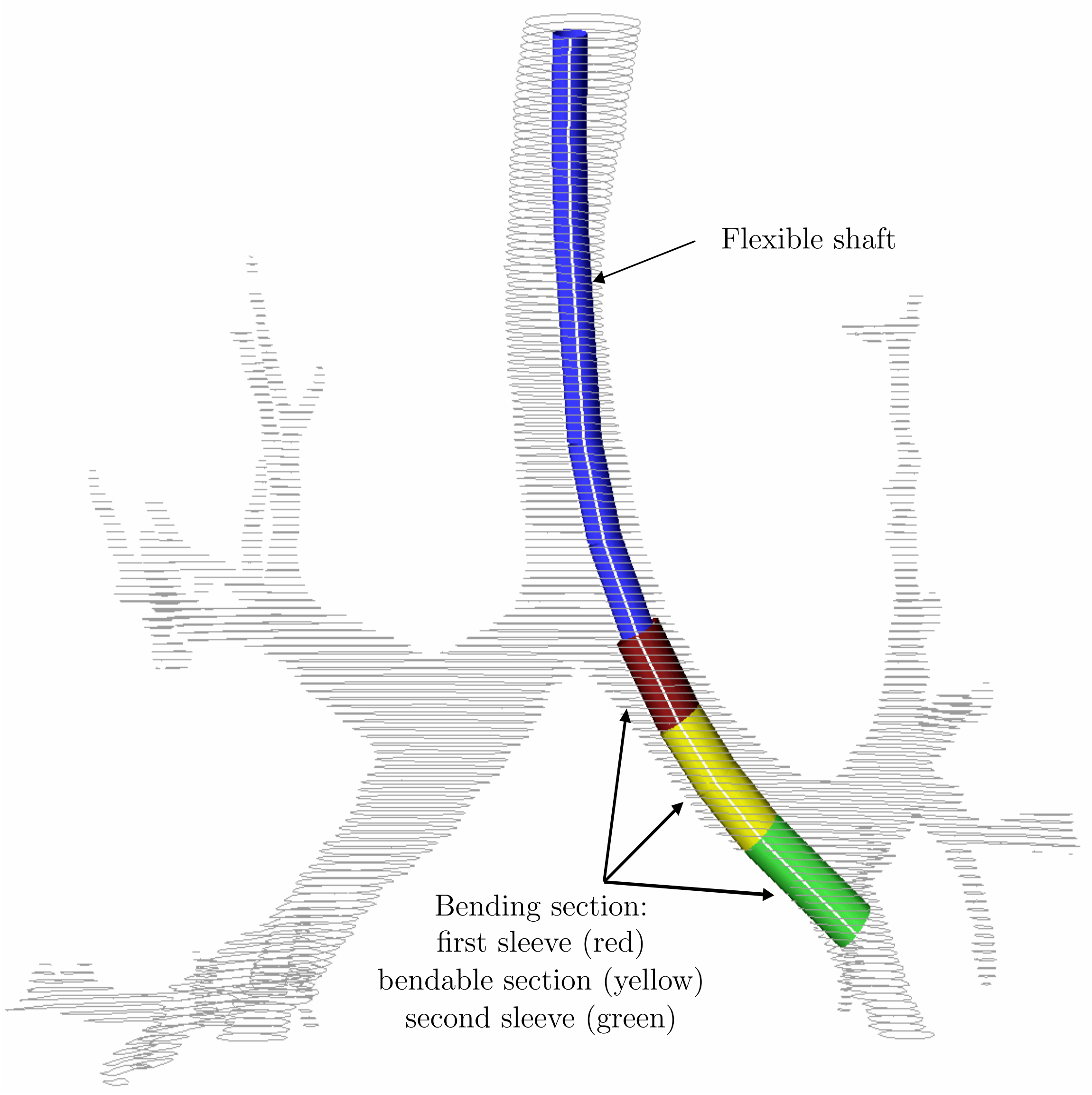

Figure 1.25: Endoscope model inserted into a tracheobronchial tree model. The 3D model of the anatomy (ca. 14000 triangles) was reconstructed from 132 planar parallel cross sections (slices), obtained from a CT scan of the patient. The model is depicted by drawing the contour of each slice. The endoscope model consists of four parts: Flexible shaft (blue), rigid sleeve (red), bendable section (yellow) and an other rigid sleeve (green). Note the difference between shaft diameter and bending section diameter. This represents one of the model requirements, described in Section 1.1. The white line along the entire instrument was created by drawing in each local cylinder frame the same line onto the cylinder's surface.

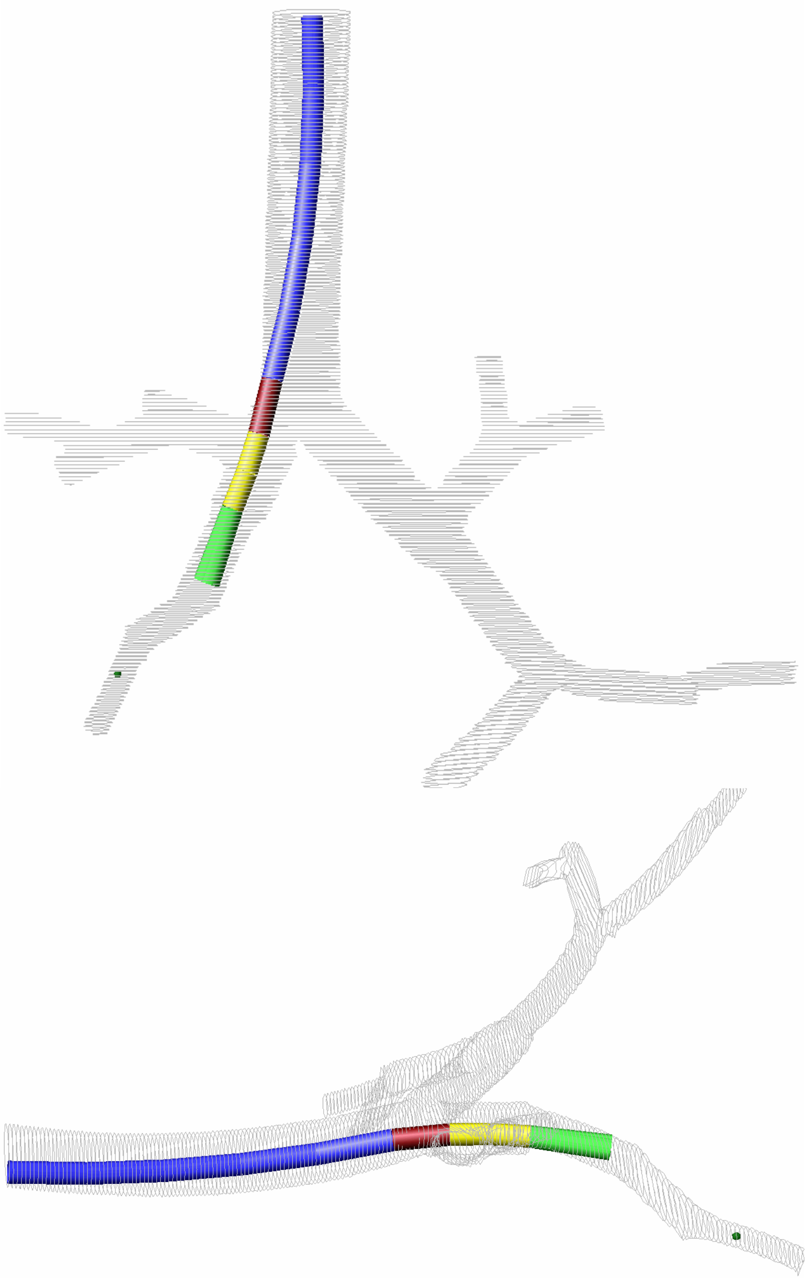

Figure 1.26: Endoscope model inserted into a 3D model of a lung phantom. The phantom model was reconstructed (ca. 14000 triangles) from 207 planar parallel cross sections (slices), obtained from a CT scan of the phantom. The model is depicted by drawing the contour of each slice. The endoscope model was configured to reflect the specifications of the OLYMPUS GIF-100 videoscope (see Table 1.2 on page 62). The overall length shown is 275 mm. Top: Frontal view. Bottom: Side view.

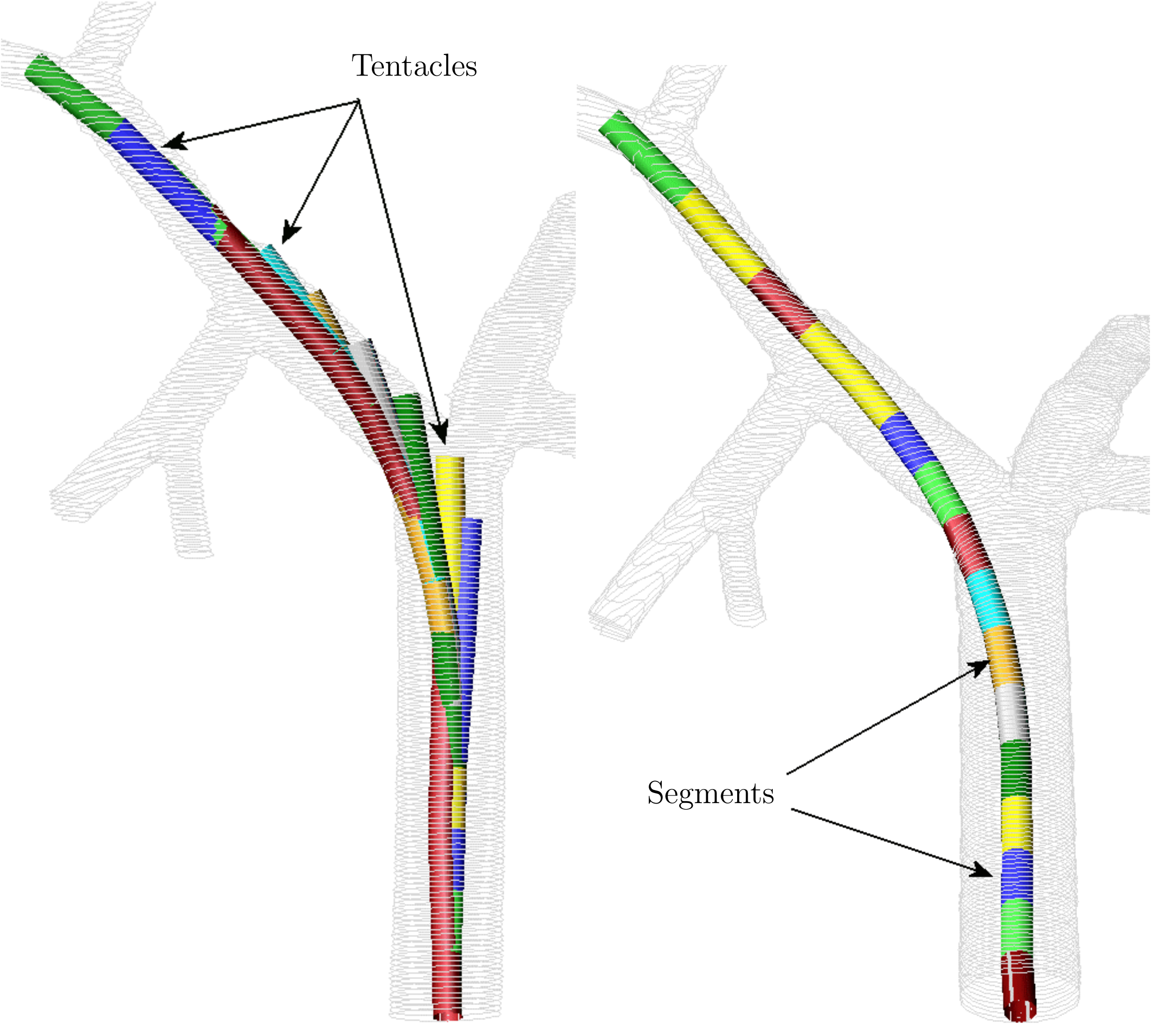

Figure 1.27: Endoscope model inserted into a 3D model of a lung phantom. Left: Showing all “tentacles” in full length. Right: Showing only the first link of each “tentacle” ($k = 1$).

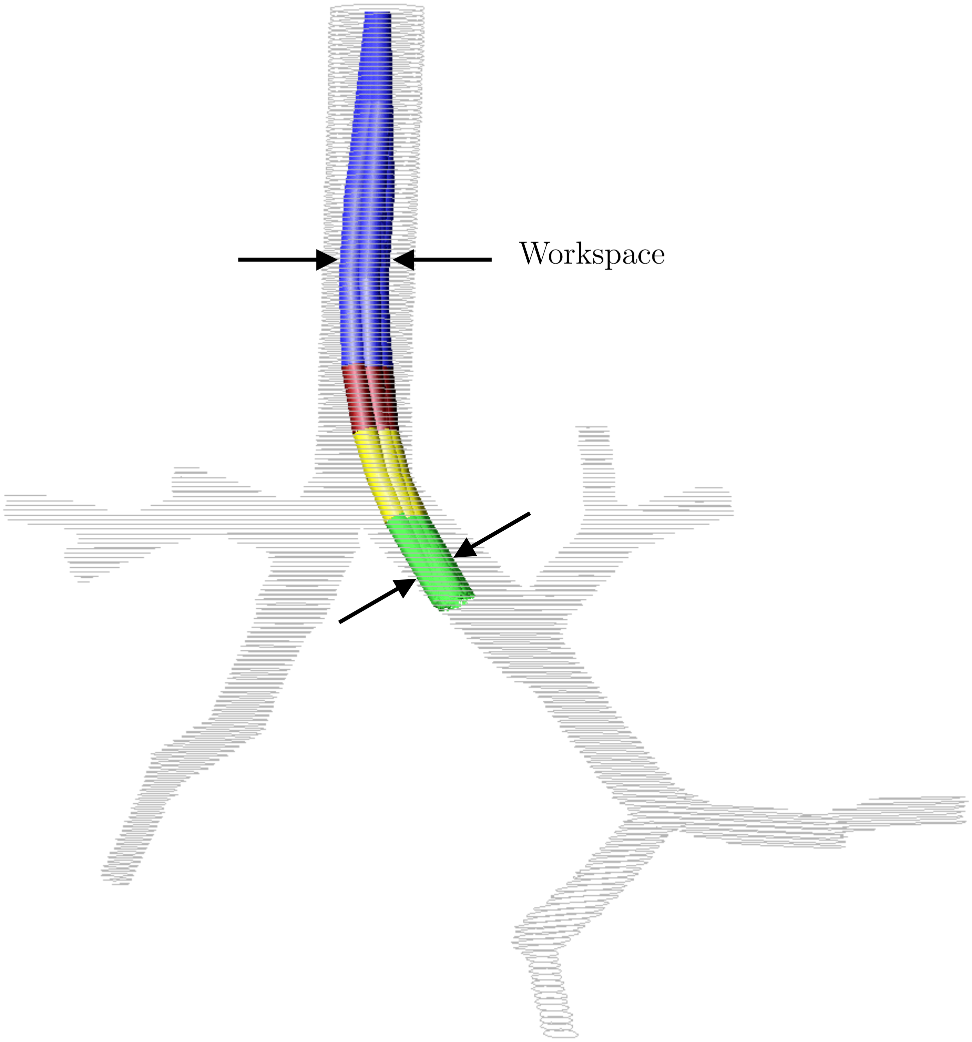

Figure 1.28: The endoscope model was configured to calculate the workspace of the OLYMPUS GIF-100 by relaxing the selectivity of the energy filter. The 150 endoscopes of lowest deformation energy ($p = 150$) are depicted.

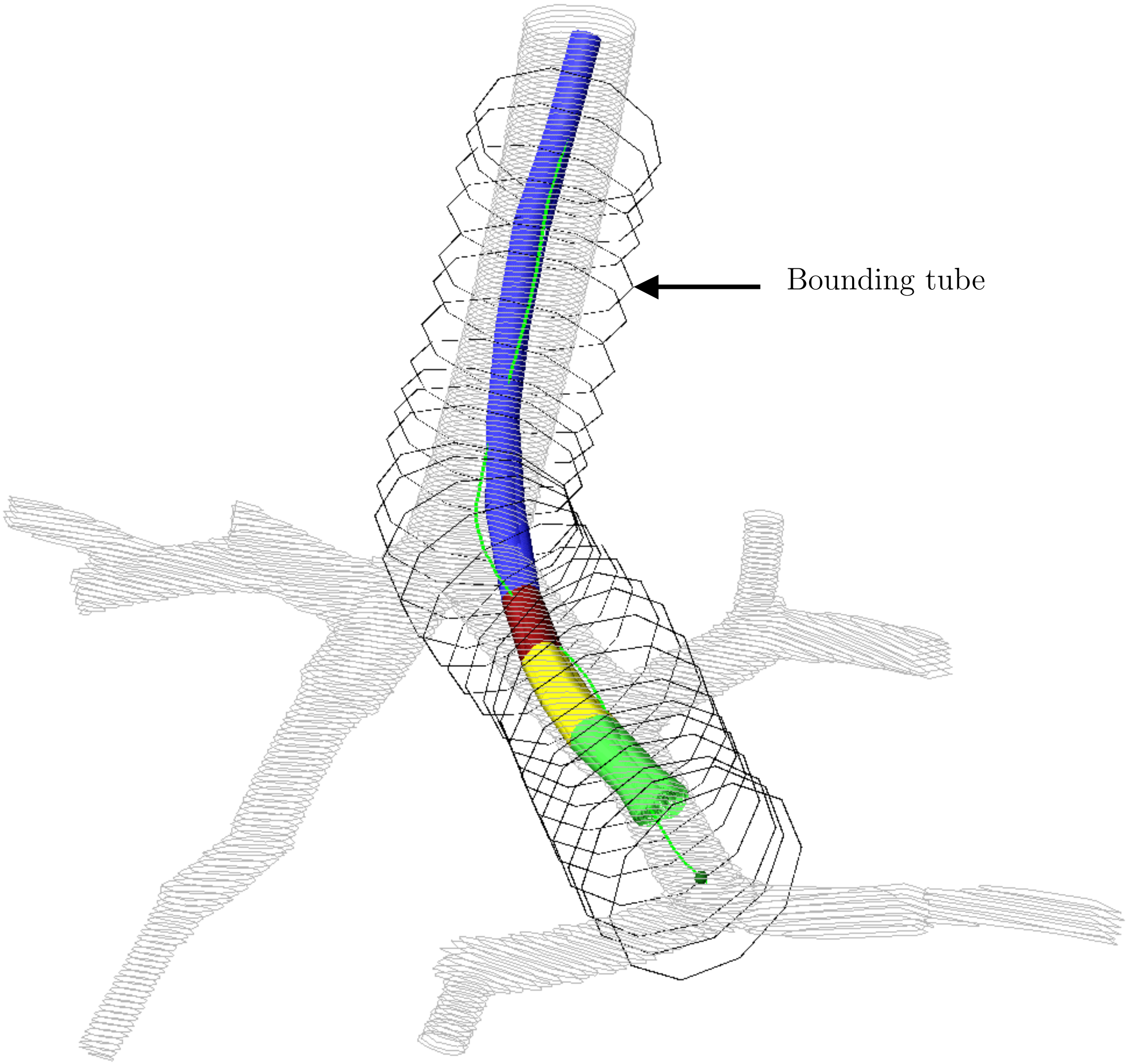

Figure 1.29: In this figure, as well as in the previous three figures, a bounding tube filter was used to restrict the endoscope model to a target branch. Here, the rings of the bounding tube are depicted.

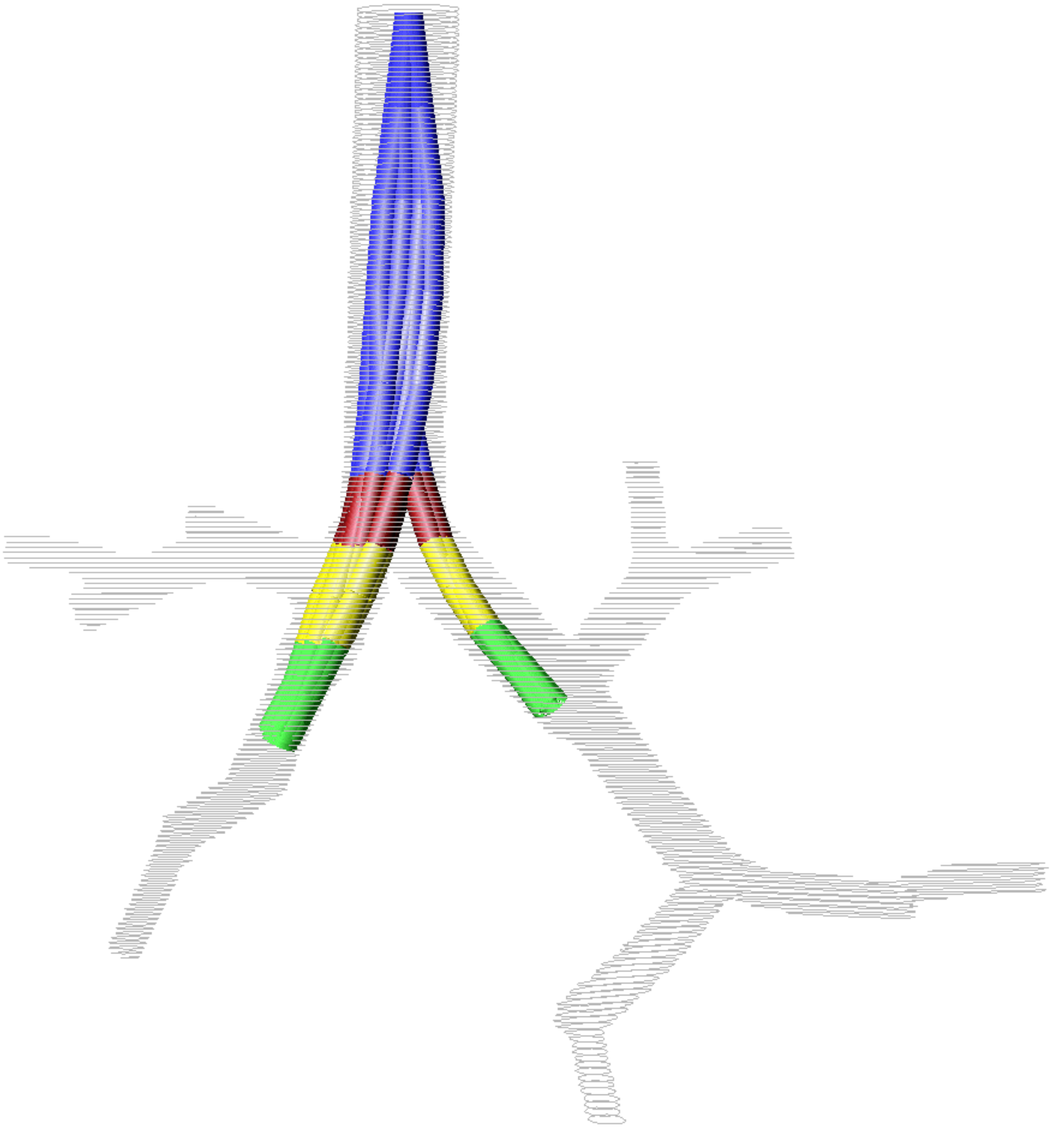

Figure 1.30: The bounding tube filter was switched off, to calculate where an endoscope of a given insertion depth can reach. Note that the endoscope can not reach into the upper right lobe (shown on the right), without an active bending of its tip. See Figure 1.32 for an "active bending" example.

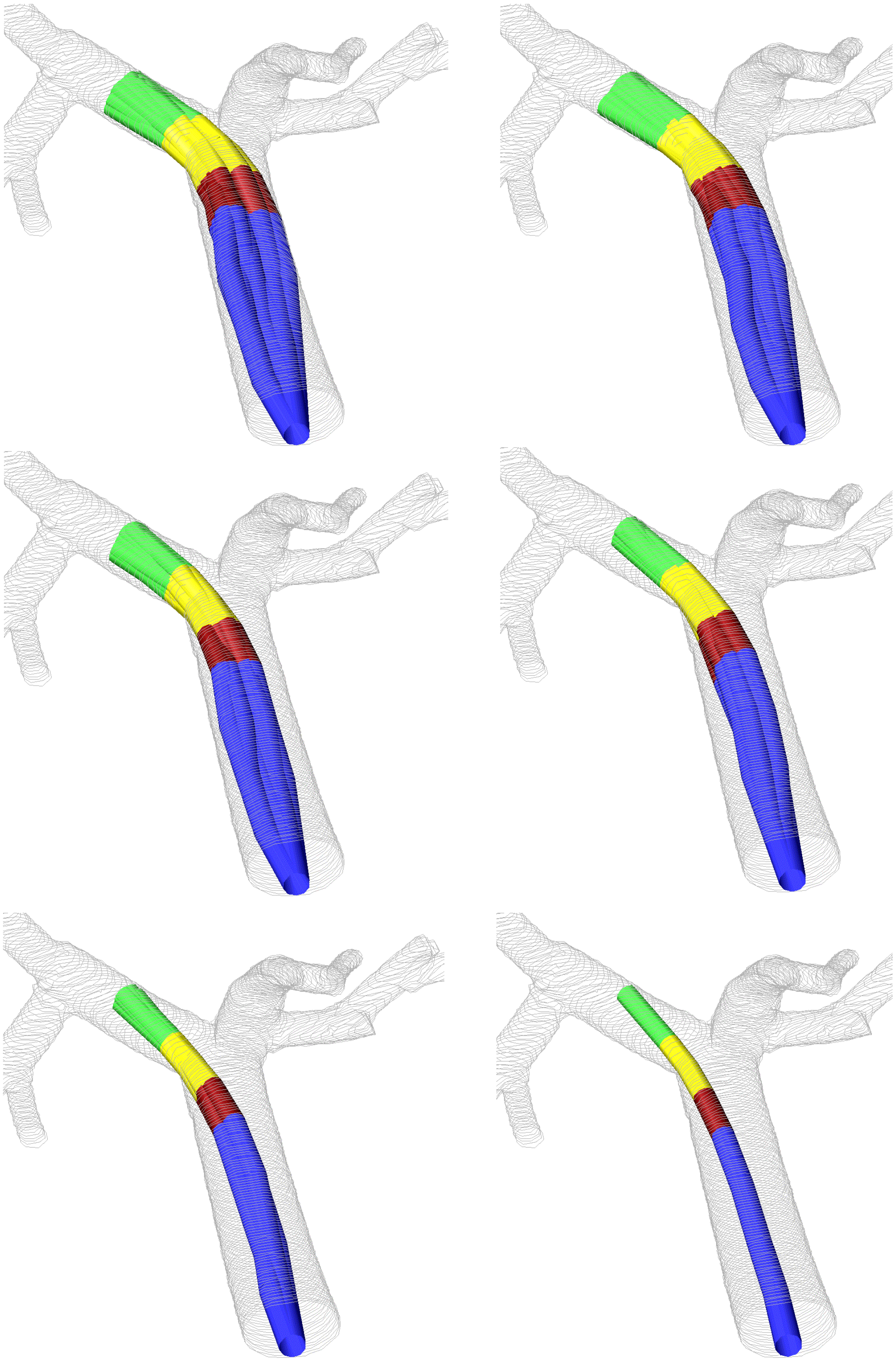

Figure 1.31: The deformation energy decreases from top, left to bottom, right.

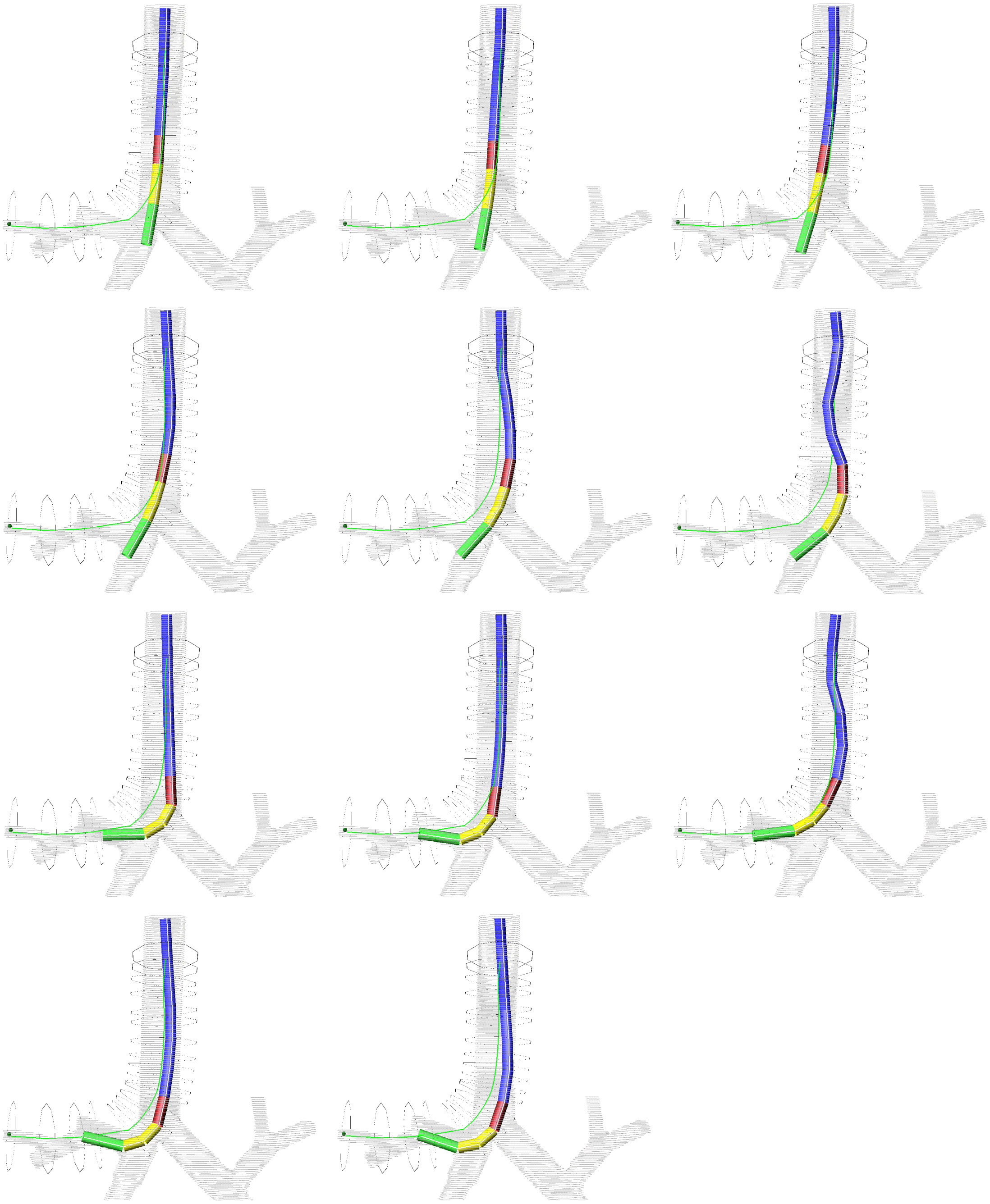

Figure 1.32: How far can the OLYMPUS GIF-100 reach into the upper right lobe of the lung phantom? This simulation shows the last 11 insertion steps (top, left to bottom, center, Step size: 5 mm) before the endoscope model got stuck. The flexible part of the bending section (yellow) was configured to have a maximum range of 90°. The result shown in the last image corresponds precisely to the maximum achievable insertion depth of the real GIF-100 inserted into the same branch of the lung phantom.

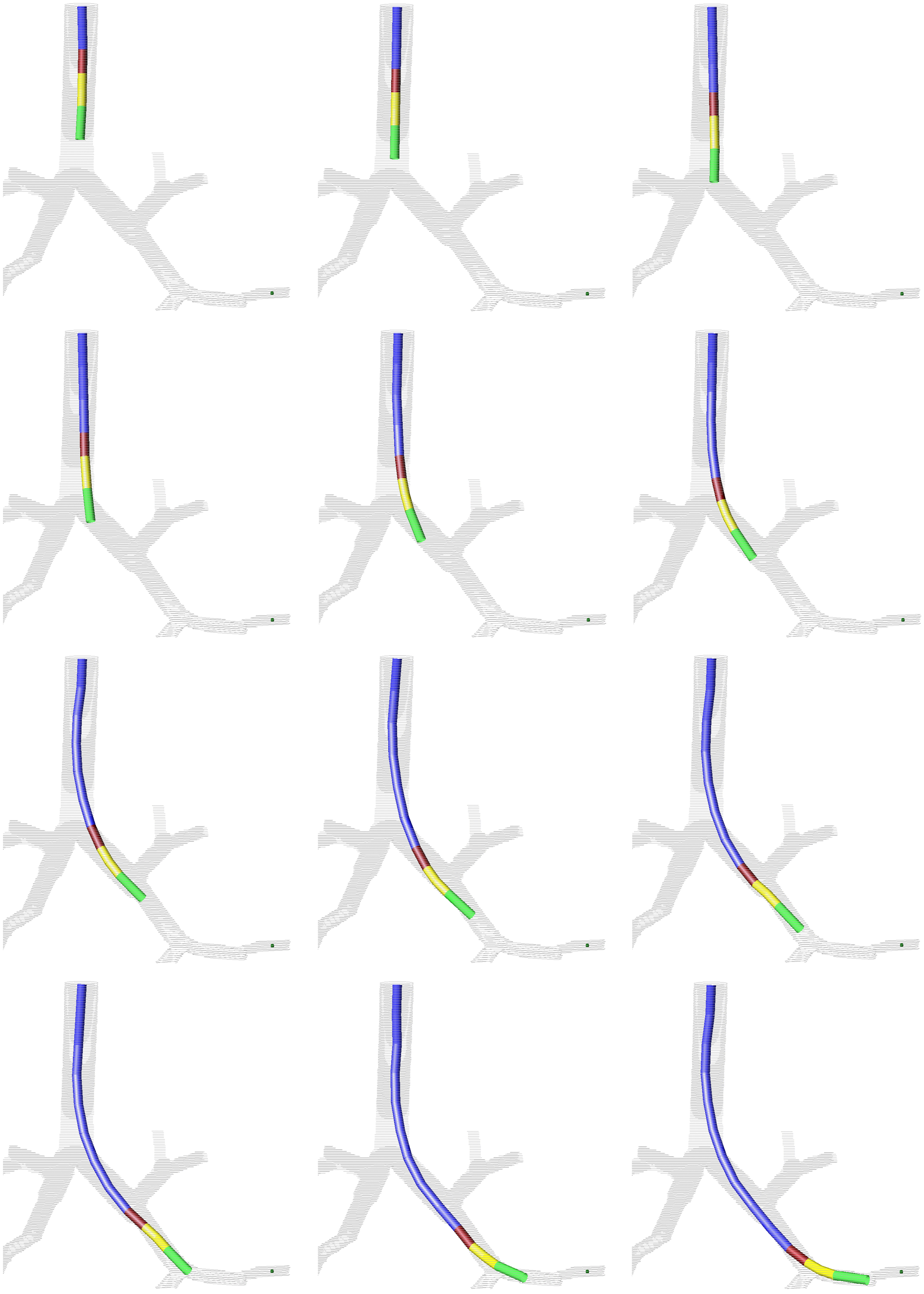

Figure 1.33: Insertion simulation at 12 different stages (from top, left to bottom, right). After an insertion depth of 420 mm, the endoscope got stuck, due to an insufficient branch diameter.

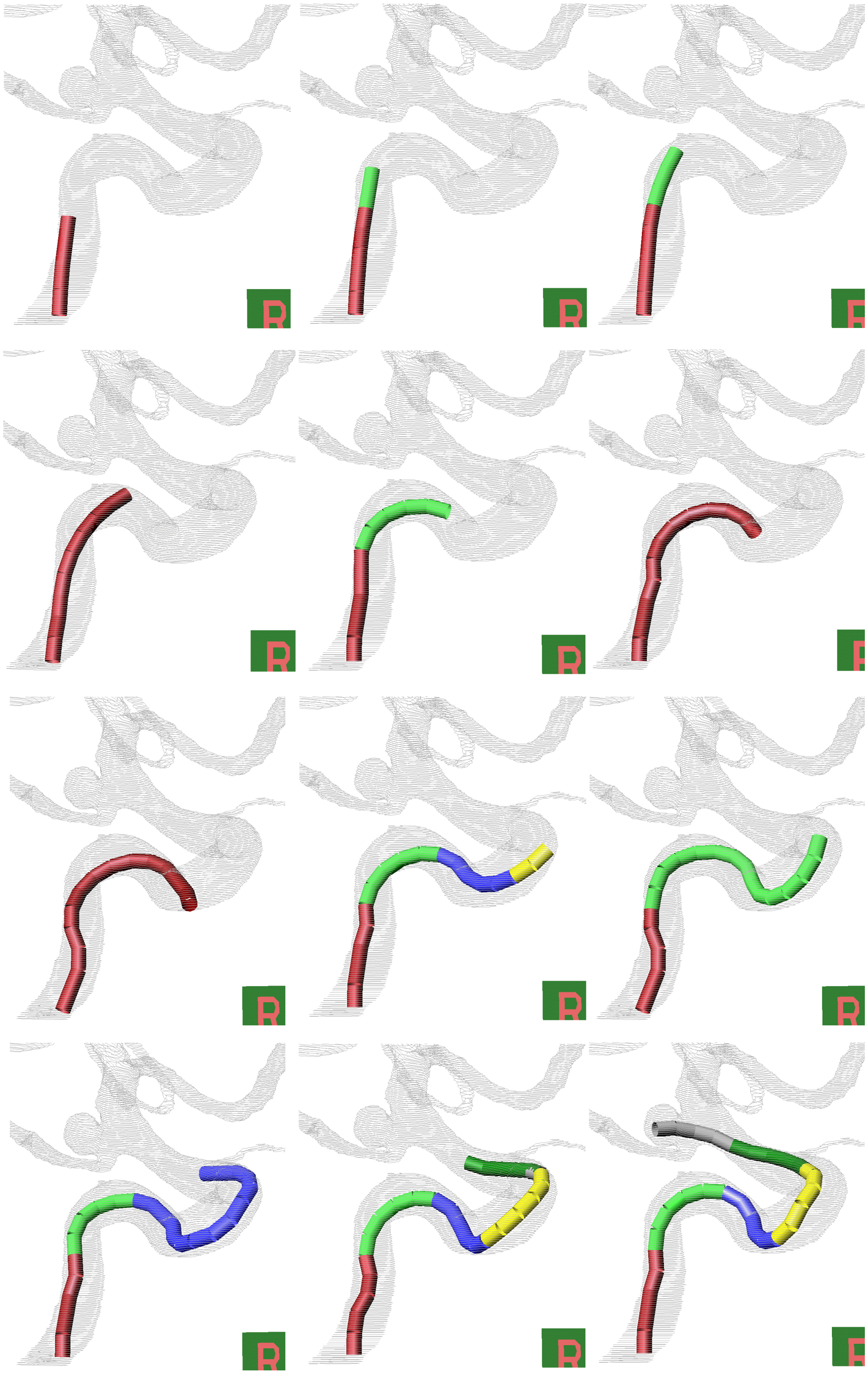

Figure 1.34: Simulation of a catheter (diameter 1.5 mm, max. length 72 mm) inserted into a brain artery, depicted at 12 different stages (from top, left to bottom, right).

Chapter 2

An “Optimal” Needle Placement Strategy Given an Approximate Initial Needle Position

A biopsy is a minimally invasive surgical procedure, often used in the diagnosis and staging of cancer patients. In general, the goal is to take a sample of the suspicious tissue (target) by placing a biopsy needle inside the target. Since the target is often not directly visible for the physician, numerous methods for guiding biopsies have been developed. Procedures that have attracted special attention in recent years include the biopsy of the prostate, breast, liver and lung.

In many cases it is common practise to take more than one tissue sample, in order to increase the probability of hitting the target. Instead of using a simple trial-and-error approach, biopsy strategies have been developed, among others for prostate cancer biopsies [49],[50]. The “k-Needle Placement Strategy” is a biopsy protocol that specifies how to place k (biopsy) needles, such that the probability of success is maximized. The placement of a needle is specified by a suitable parameterization of its degrees of freedom, e.g. by two angles and an insertion depth.

This chapter presents an "optimal" k-needle placement strategy for a special class of biopsy problems, where the initial needle position is known approximately. A typical example for such a procedure is a "transbronchial needle aspiration biopsy" (TBNA)[31]. Traditionally this biopsy is performed by maneuvering a bronchoscope to a suitable site within the tracheobronchial tree. Then the bronchoscopist inserts a needle through the bronchoscope and punctures the bronchial wall in order to hit the target lying behind. Generally, methods to guide TBNAs are based on determining the position and orientation of the bronchoscope's tip: In the "imaging–based" approach, standard imaging technology, such as fluoroscopy is used to visualize the instrument. In the "sensor–based" approach, sensors like a six degree of freedom position and orientation sensor are attached to the bronchoscope. In the "vision–based" approach, the video images coming from the CCD camera inside the bronchoscope's tip are analyzed to achieve a continuous tracking of the tip. The "model–based" approach proposed in this thesis, estimates the pose of the tip from a given insertion depth and region (branch) of interest. All four approaches determine the tip position approximately, due to visualization, sensor, video tracking or model inaccuracies. In other words, the initial position from where the biopsy is taken, is given with some error. Or, in the case of the model-based approach, where a mathematical model of the endoscope facilitates the calculation of it's workspace (see Chapter 1), a number of possible tip positions and orientations is given.

This chapter addresses the problem of finding an "optimal" strategy for placing k biopsy needles, given a large number of possible initial endoscope positions. Two variations of the problem are considered: (1) Calculate the smallest set of needles[1], needed to guarantee a successful biopsy. (2) Given a number k, calculate k needles such that the probability of a successful biopsy is maximized.

Both problems are formulated in terms of two general, NP-hard optimization

---

[1] We use "needle" as short hand for the parameter vector that specifies the needle placement

problems. The solution to both problems is "optimal" with respect to the best approximative algorithm known for the respective NP-hard problem. For the latter problem there exists an approximative algorithm which requires virtually no implementation effort and is guaranteed to be within a factor of $1 - \frac{1}{e} \approx 0.63$ of the exact solution. For both variations of the problem success probabilities for each needle are provided to the physician.

The outline of this chapter is as follows: Section 2.1 gives a formal description of the problem and presents a "naive" solution. Based on the shortcomings of this approach, Section 2.2 formulates the problem as an optimization problem. A solution to the two variations (see above) of this optimization problem is found by considering a dual problem in the needle parameter domain. The first variation is formulated as the "Set Covering Problem", the second as the "Maximum k-Coverage Problem", both classic NP-hard optimization problems. For the first variation, the result is an algorithm that finds for a given set of possible initial needle positions, the smallest set of needles needed to guarantee a successful biopsy. For the second variation the result is an algorithm that maximizes the coverage of the possible initial positions for a given maximum number of k needles.

Section 2.3 describes an approximative algorithm for the second variation, which at each stage greedily constructs a solution by selecting the sub-solution, which gives maximum improvement. Finally, Section 2.4 describes an experiment to validate the approach by simulating a TBNA, using the endoscope model described in Chapter 1. The algorithm described in Section 2.3 was implemented to calculate a list of needle parameters and probabilities for a given number of needles. The resulting list shows that the approach can provide valuable decision support for the physician in choosing how many needles to place and how to place them.

2.1 Problem Definition

This section introduces the main components of the problem of finding an "optimal" k-needle placement strategy, given an approximate initial needle position. At first, a list of general assumptions is given. Then, a straightforward solution to the problem is presented and its shortcomings are analyzed. They represent the motivation for the following section to formulate the problem as an optimization problem.

2.1.1 Assumptions

The problem is based on the following three assumptions.

1. There exists an initial position domain $P \subset \mathcal{M}$, which is a set of possible initial locations for the endoscope, before needle placement. The endoscope is assumed to be given by the model described in Chapter 1, namely by a sequence of links, interconnected by joints. As described in Section 1.4.2, the endoscope's first bending section link (sleeve, see Figure 1.18 on Page 56) can be assumed to remain stationary during the alignment of the tip with a target (active tip deflection). Therefore, this link alone is sufficient to describe the endoscope's initial position before needle placement. In other words, each initial position $\mathbf{p} \in P$ can be represented by a single 4×4 matrix.

   Let $\tilde{\mathbf{p}} \in P$ denote the real, but unknown position of the endoscope after insertion. It is assumed that $\tilde{\mathbf{p}}$ does not change during needle placement.

2. There exists a target domain $T \subset \mathbb{R}^3$.

3. There exists a function $f : P \times T \to N$, which computes for a given $\mathbf{p} \in P$ and $\mathbf{t} \in T$ the necessary needle parameter $\mathbf{n} \in N$ to hit $\mathbf{t}$ from position $\mathbf{p}$. $N \subset \mathbb{R}^3$ is denoted as the "needle parameter domain". Function $f()$ represents the model described in Section 1.4.2 of the endoscope's active tip deflection. In the notation of that section, where a link was described by a matrix $\mathbf{F}$, function

$f()$ can be written as:

$$f(\mathbf{F},\,\mathbf{t}) = \begin{pmatrix} \alpha \\ \beta \\ d \end{pmatrix}. \tag{2.1}$$

It computes the necessary shaft rotation α, needle of tip deflection β and needle length d (Equation 1.24, 1.25 and 1.26), to hit a target $\mathbf{t} \in T$ from an initial endoscope position $\mathbf{p}$, described by a single link $\mathbf{F}$.

There also exists a dual function $\bar{f} : P \times N \to \mathbb{R}^3$, which computes for a given position $\mathbf{p}$ and a needle parameter $\mathbf{n}$ the resulting position of the needle tip. Given the same model of active tip deflection as mentioned above, the realization of $\bar{f}$ is straightforward.

Note that the codomain of $\bar{f}$ is $\mathbb{R}^3$ (and not T), because for $\mathbf{p} \neq \mathbf{q}$:

$$(\,\mathbf{n} = f(\mathbf{p},\,\mathbf{t})\,) \;\not\Rightarrow\; (\,\bar{f}(\mathbf{q},\,\mathbf{n}) \text{ is an element of } T\,), \quad \mathbf{p}, \mathbf{q} \in P, \;\; \mathbf{t} \in T \tag{2.2}$$

Given these assumptions, the k-needle placement problem is to determine a set $N_* \subset N$ of k needle parameters, such that P is covered as well as possible. In the remainder of this Chapter, three such sets N_* and their corresponding sets P_* in the position domain P are considered:

Definition 2.1.1 ($N_{\{\mathsf{naive},\ \mathsf{better},\ \mathsf{opt}\}}$ and $P_{\{\mathsf{naive},\ \mathsf{better},\ \mathsf{opt}\}}$). Let

$$N_{\mathsf{naive}},\; N_{\mathsf{better}},\; N_{\mathsf{opt}}$$

be subsets of the needle parameter domain N. Then the corresponding sets in the position domain P are denoted by

$$P_{\mathsf{naive}},\; P_{\mathsf{better}},\; P_{\mathsf{opt}}$$

and defined as:

$$P_i = \{\mathbf{p} \in P \mid \bar{f}(\mathbf{p},\,\mathbf{n}) \text{ is an element of } T,\; \mathbf{n} \in N_i\}\,,$$

for $i \in \{\mathsf{naive},\ \mathsf{better},\ \mathsf{opt}\}$. □

Set P_i is the set of all $\mathbf{p} \in P$ that are mapped into the target by a needle of set N_i.

2.1.2 Naive Method

This Section presents a naive solution to the problem of finding a k-needle placement strategy, given an approximate initial needle position. A naive method to find k needle parameters that cover P, is to firstly select a set P^{Δ} of k samples from P. For each sample $\mathbf{p}_i \in P^{\Delta}$ a needle parameter is calculated that would bring the needle tip into the center of the target:

$$N_{\mathsf{naive}} = f(P^{\Delta},\ \mathbf{t}_{center}), \quad |P^{\Delta}| = k\,. \tag{2.3}$$

Note that this abbreviated notation is used as an equivalent for:

$$N_{\mathsf{naive}} = \{\mathbf{n} = f(\mathbf{p}_i, \mathbf{t}_{center}) \mid \mathbf{p}_i \in P^{\Delta}\}\,. \tag{2.4}$$

In other words, N_{naive} is a set of needle parameters that hit the target from at least all positions $\mathbf{p} \in P^{\Delta}$. It is “hoped” that N_{naive} maps as many $\mathbf{p} \in P$ into the target as possible.

This “strategy” has at least two shortcomings. The first shortcoming is that P is not necessarily well covered:

$$P^{\Delta} \ \subseteq\ P_{\mathsf{naive}} \ \subseteq\ P\,. \tag{2.5}$$

It is not guaranteed that for all endoscope positions $\mathbf{p} \in P$ there exists a needle in N_{naive}, which hits the target. Secondly, N_{naive} is not necessarily minimal. It may exist a set $N_{\mathsf{better}} \subset N$ such that

$$\begin{aligned} |N_{\mathsf{better}}| \ &<\ |N_{\mathsf{naive}}| \ \text{ and} \\ P_{\mathsf{better}} \ &\supseteq\ P_{\mathsf{naive}}\,. \end{aligned} \tag{2.6}$$

P_{better} covers at least as much of P as P_{naive}, while needing fewer needles.

These observations suggest the formulation of the k-needle placement problem as an optimization problem.

2.2 An “Optimal” Strategy

In this section, a solution to the problem of finding an “optimal” k-needle placement strategy is developed. The basic idea is to find needles that “cover” as many of the initial endoscope positions as possible. A needle “covers” an area, if for any endoscope within this area the needle in question hits the target. One goal is to solve the problem of minimizing the number of needles needed for a full coverage. The problem of finding the smallest set of needles that cover all initial positions is formulated as the problem of finding the “minimum set cover” in the needle parameter domain. This problem in turn, can be directly formulated as the “Set Covering Problem”, a well known NP-hard optimization problem.

Another goal is to maximize the number of initial positions covered by a given number of k needles. This problem is formulated as the “Maximum k-Coverage Problem”, likewise a NP-hard, general optimization problem.

2.2.1 Formulation as an Optimization Problem

An “optimal” k-needle placement strategy is a set $N_{\mathsf{opt}} \subset N$ of needle parameters, such that

$$\begin{array}{ll} 1. & P_{\mathsf{opt}} = P \quad \text{and} \\ 2. & |N_{\mathsf{opt}}| = \text{ minimal}\,. \end{array} \tag{2.7}$$

In other words, for all endoscope positions $\mathbf{p} \in P$ there exists a needle in N_{opt} which hits the target and no set smaller than N_{opt} guarantees the same.

Similar to Definition 2.1.1, let P_i denote the set of all $\mathbf{p} \in P$ that are mapped into the target by a needle parameter $\mathbf{n}_i \in N_{\mathsf{opt}}$. By definition, N_{opt} induces a coverage of P by k subsets P_i. Figure 2.1 shows an example for $k = 5$. It shows five subsets $P_1, \ldots, P_5$, with each P_i induced by a needle parameter $\mathbf{n}_i \in N_{\mathsf{opt}}$. This example makes it clear that any given real initial endoscope position $\tilde{\mathbf{p}}$ will fall inside a subset P_i and a corresponding needle parameter $\mathbf{n}_i$ will map $\tilde{\mathbf{p}}$ inside the target. Since $\tilde{\mathbf{p}}$ is

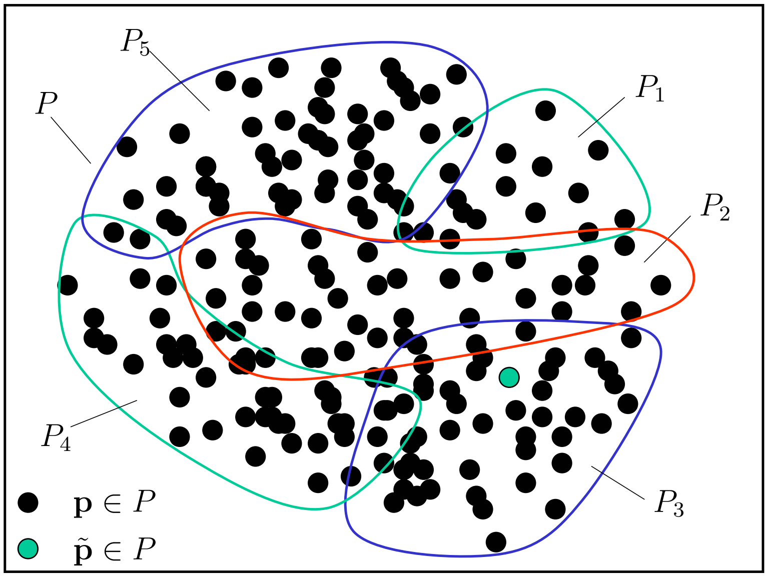

Figure 2.1: Set P divided into subsets P_i, induced by N_{opt}.

unknown, all five needle parameters have to be tested, one at a time. In this example, the first, second, fourth and fifth needle will fail and the third will hit the target.

The basic idea behind finding the smallest set of subsets in P, is to consider a "dual problem" in the needle parameter domain N. The problem is transformed into N by sampling P and calculating a "scan" of target T from the "perspective" of each sample. The dual problem is then to find a minimum number of points in N such that each scan covers at least one point. This set of points is equivalent to N_{opt}.

2.2.2 Transformation Into the Needle Parameter Domain

To transform the problem into the needle parameter domain, the following definition is used:

Definition 2.2.1 ($S^T(\mathbf{p})$). $S^T(\mathbf{p})$ denotes a "scan" of T from a given position $\mathbf{p} \in P$:

$$S^T(\mathbf{p}) = f(\mathbf{p}, T) .$$

$S^T(\mathbf{p}) \subset N$ is the set of all needle parameters needed to hit all $\mathbf{t} \in T$ from a fixed $\mathbf{p}$. Position $\mathbf{p}$ is called the "viewpoint" of the scan. □

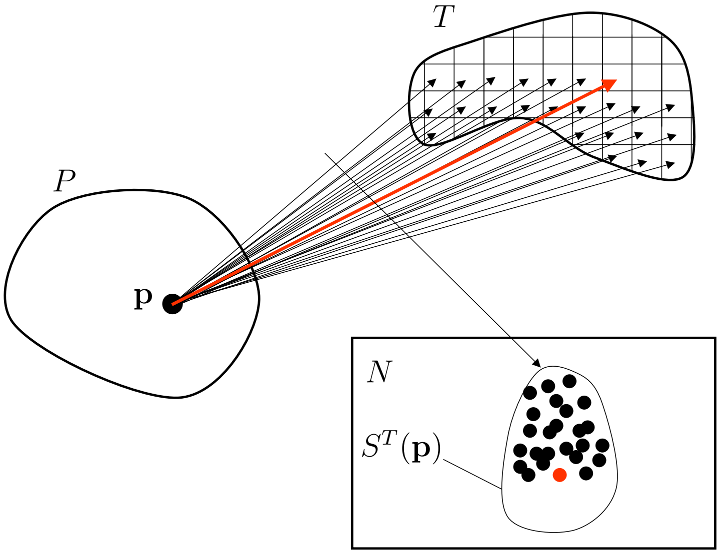

Figure 2.2: A “scan” of T from viewpoint $\mathbf{p}$: $S^T(\mathbf{p})$

Figure 2.2 makes it clear that $S^T(\mathbf{p})$ is the result of T scanned from viewpoint $\mathbf{p}$ and that it forms a point cloud in the needle parameter domain.

Let target T be discretized into T^Δ, which consists of voxels or cells of side length Δ_T. A discretization of T also requires a discretization of N in the sense that two needle parameters which map a position $\mathbf{p} \in P$ into the same voxel of T^Δ, can be regarded as one needle parameter.

Definition 2.2.2 (N^Δ). The needle parameter domain N is discretized into cells. The centers of all cells represent the discretized needle parameter domain N^Δ. Cell size Δ_N is derived from the cell size Δ_T in T^Δ. Let $d()$ be the euclidian distance:

$$\Delta_N = d(\mathbf{n}_1, \mathbf{n}_2) \rightarrow \max \ \text{ such that for a } \mathbf{p} \in P : d(\bar{f}(\mathbf{p}, \mathbf{n}_1), \bar{f}(\mathbf{p}, \mathbf{n}_2)) \leq \Delta_T$$

□

In the following the transition is made from $S^T(\mathbf{p}) \subset N$ to $S^{T^\Delta}(\mathbf{p}) \subset N^\Delta$, where $S^{T^\Delta}(\mathbf{p})$ is the scan of T^Δ from viewpoint $\mathbf{p} \in P$. The idea is to “round” each $\mathbf{n} \in S^{T^\Delta}(\mathbf{p})$ to the center of the cell it falls in. If one or more $\mathbf{n}$ fall into the same cell,

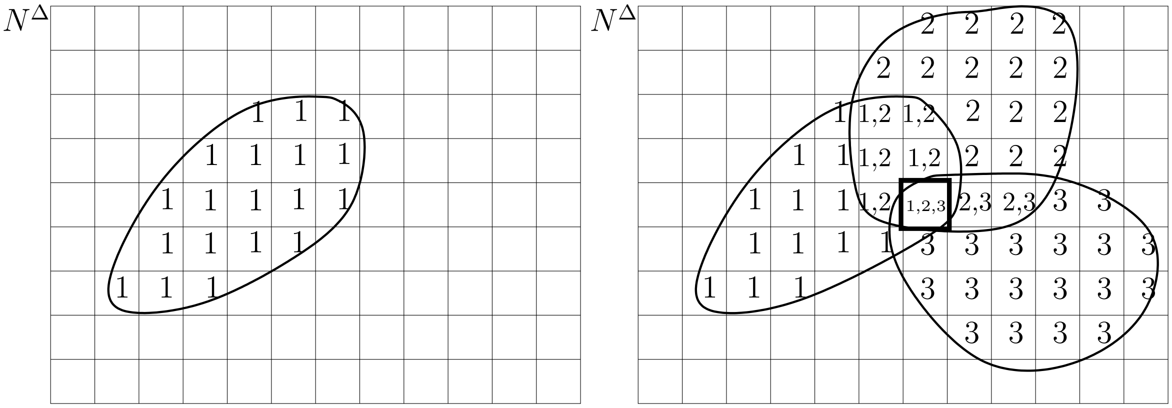

Figure 2.3: Left: A scan $S^{T^\Delta}(\mathbf{p}_1)$ in N^Δ. Each cell shows its set of viewpoints V_i (subscript of $\mathbf{p}_1$ only). Right: Three scans from viewpoint $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$. Set V_i of cell i gives the indices of the viewpoints, whose scan cover cell i. Only one cell (boxed) is covered by all three scans.

we say the cell is "covered" by the scan. Consequently, it is sufficient to store for each cell the viewpoint $\mathbf{p}$ of the scan, which "covers" the cell. This yields the following:

Definition 2.2.3 ($\mathbf{n}_i^c$ and V_i). Each cell of N^Δ stores two pieces of information:

1. $\mathbf{n}_i^c \in N^\Delta$ the center of cell i,
2. $V_i \subseteq P$ the set of viewpoints of cell i

The "center of cell i" is the needle parameter in the center of cell i. Set V_i is the set of viewpoints of all scans that cover cell i. □

Figure 2.3 (left) shows N^Δ divided into cells and a scan $S^{T^\Delta}(\mathbf{p}_1)$. For each cell the set of viewpoints V_i is given. The set is either $\{1\}$ (subscript of viewpoint $\mathbf{p}_1$) if the cell is covered by the scan or the empty set $\{\}$, if the cell is not covered.

To transform the problem from P to N^Δ, P is sampled and a scan $S^{T^\Delta}(\mathbf{p}_i)$ is calculated for each $\mathbf{p}_i \in P$. Figure 2.3 (right) shows an example for three samples $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$. Each cell's set of viewpoints V_i is given. Note that one cell (boxed) is covered by all three scans. With $\mathbf{n}_i^c$ the center of this cell, this can be interpreted as:

$$\bar{f}(\mathbf{p}_1, \mathbf{n}_i^c) \in T^\Delta \ \wedge \ \bar{f}(\mathbf{p}_2, \mathbf{n}_i^c) \in T^\Delta \ \wedge \ \bar{f}(\mathbf{p}_3, \mathbf{n}_i^c) \in T^\Delta \tag{2.8}$$

In other words, only needle parameter $\mathbf{n}_i^c$ is needed to map all three positions into the target. Positions $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ are members of the same subset, induced by $\mathbf{n}_i^c$. The goal of dividing P into a minimum number of subsets can now be formulated as the problem of selecting a minimum number of cells in N^{Δ}, such that each scan covers at least one selected cell. This problem is reduced to the following "classic" optimization problem.

2.2.3 "Set Covering Problem" and "Maximum k-Coverage Problem"

The "Set Covering Problem" or short SCP is a well known NP-hard combinatorial optimization problem, which can be formulated as:

Set Covering Problem (SCP): A finite set U of elements and a class $\mathcal{S}$ of subsets of U is given. Let S_i denote the i-th subset in $\mathcal{S}$.

The task is to select subsets S_i, such that every element in U belongs to at least one S_i. A selection $\mathcal{W} \subseteq \mathcal{S}$ with this property is called a set cover of U with respect to $\mathcal{S}$.

The optimization problem is to find a set cover $\mathcal{W}$ of minimum cardinality:

$$\mathsf{SCP}(U, \mathcal{S}) = \{\mathcal{W} \mid \mathcal{W} \text{ is a set cover of } U \text{ of minimum cardinality}\}. \tag{2.9}$$

□

The SCP is a subject of numerous publications in the operations research and mathematical literature. Many applications of the set covering problem to real-world problems, such as resource allocation and scheduling have been described. Exact solutions for modestly sized problems using a dual heuristic, have been reported by Fisher and Kedia [51]. For large problems, approximative schemes have been suggested by Beasley [52].

An interesting variation of the SCP is the "Maximum k-Coverage Problem" (kCP).

Maximum k-Coverage Problem (kCP): A set U and a class of subsets $\mathcal{S}$ is given, as in the SCP, as well as an integer k. Each element $u \in U$ has an associated weight $w(u)$.

The optimization problem is to select k subsets S_i from $\mathcal{S}$, such that the weight of the elements in $\bigcup_{i=1}^{k} S_i$ is maximized. □

Hochbaum and Pathria [53] have shown that the greedy approach to this NP-hard problem, which selects at each stage the subset that gives maximum improvement, is guaranteed to be within a factor of $1-(1-\frac{1}{k})^k > 1-\frac{1}{e}$ of the optimal solution.

2.2.4 Formulation of the Problem as an SCP and kCP

The connection between our problem and the SCP can be established as follows:

Let P^Δ be a set of samples of P, $V_i \subseteq P^\Delta$ the set of viewpoints of cell i and $\mathcal{W}$ an arbitrary minimal set cover:

$$\mathcal{W} \in \mathsf{SCP}(U, \mathcal{S}), \text{ where} \tag{2.10}$$

$$U = P^\Delta, \;\; \mathcal{S} = \{V_1, V_2, \ldots, V_{|N^\Delta|}\}$$

Let $\mathbf{n}_i^c \in N^\Delta$ be the needle parameter in the center of cell i. Then an "optimal" k-Needle placement strategy is given by:

$$N_{\mathsf{opt}} = \{\mathbf{n}_i^c \,|\, V_i \in \mathcal{W}\}\,. \tag{2.11}$$

The $P_{\mathsf{opt}} = P$ condition of Equation 2.7 follows from the SCP condition that every element in U belongs to at least one selected subset S_i. The '$|N_{\mathsf{opt}}|$ = minimal' condition follows from the minimization of the set covers' cardinality.

For example, given the situation shown in Figure 2.3 (right), $U = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$, $\mathcal{S} = \{\{\}, \{\mathbf{p}_1\}, \{\mathbf{p}_2\}, \{\mathbf{p}_3\}, \{\mathbf{p}_1, \mathbf{p}_2\}, \{\mathbf{p}_2, \mathbf{p}_3\}, \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}\}$, $\mathcal{W} = \{\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}\}$ and $N_{\mathsf{opt}} = \{\mathbf{n}_i^c\}$, where i is the boxed cell.

With this formulation, a subset of P, induced by a $\mathbf{n}_i^c \in N_{\mathsf{opt}}$ is given by V_i. It is important to note that the quality of solution N_{opt} depends on the sample density of P^{Δ}.

The connection between our problem and the kCP follows directly from the above theorem, with the weight function given by: $w(u) = 1$, for all $u \in U$. This weight function favors cells that are covered by many scans, since the kCP maximizes the sum of the weights of all elements of all selected subsets.

The kCP is an interesting variation for two reasons: Firstly, the greedy approach is easy to implement, by simply selecting at each stage the cell with the highest cardinality of V_i and subsequently updating all V_i. Secondly, as shown by Hochbaum *et al.* [53] for small k, a greedily constructed solution is within an acceptable factor from the exact solution. For example for $k < 3$ the factor is > 0.7.

The needle parameters given by N_{opt} should be executed in the order of decreasing probability of success. Regarding a chosen sample density, the probability of hitting target T with a needle parameter $\mathbf{n}_i^c \in N_{\mathsf{opt}}$ is given by $\frac{|V_i|}{|P^{\Delta}|}$.

2.3 Algorithm kCP_Greedy()

This section describes an algorithm "kCP_Greedy()" which represents a greedily constructed solution to the "Maximum k-Coverage Problem". The algorithm is described on the basis of a concrete application, namely a transbronchial needle aspiration (TBNA).

The idea is to use the endoscope model described in Chapter 1 to simulate a TBNA by calculating a set of possible endoscope shapes (workspace), given a target branch and an insertion depth. This set of candidate shapes represents the initial position domain P^{Δ}. For each shape in P^{Δ} an artificial target $T^{\Delta} \subset \mathbb{R}^3$, modelled as an elliptic point cloud is scanned ($S^{T^{\Delta}}(\mathbf{p})$). An example of such a scenario is shown in Figure 2.4 (left).

For each point $\mathbf{t}$ of the target point cloud three endoscope parameters are determined, which move the tip of the biopsy needle into position $\mathbf{t}$: shaft rotation α, angle of tip deflection β and needle length d (see Equation 2.1). These parameters represent a point in the needle parameter domain $N^{\Delta} \subset \mathbb{R}^3$. A complete scan of T^{Δ} from one viewpoint results in a point cloud in N^{Δ}. A complete scan from all viewpoints results in a set of point clouds in N^{Δ}. This set of point clouds (see Figure 2.3 (right)) represents the input to the kCP_Greedy() algorithm.

The set of point clouds is given by:

$$\mathbf{I} = (\mathbf{x},\, \mathbf{y},\, \mathbf{z},\, \mathbf{p})\,, \tag{2.12}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^l$ and $\mathbf{p} \in \mathbb{N}^l$ are l-dimensional column vectors:

$$l := |P^{\Delta}|\,|T^{\Delta}|\,.$$

Let x_i denote the i-th element of vector $\mathbf{x}$ so that $(x_i,\, y_i,\, z_i,\, p_i)$ represents the i-th "row" of $\mathbf{I}$, for $i = 1, \ldots, l$. The first three parameters of a row represent the $x-$, $y-$ and $z-$coordinates of a point in the needle parameter domain N. The fourth parameter p_i represents the number of the corresponding viewpoint. As in Figure 2.3, the viewpoints are numbered 1 to $|P^{\Delta}|$, so $p_i \in \{1, 2, \ldots, |P^{\Delta}|\}$.

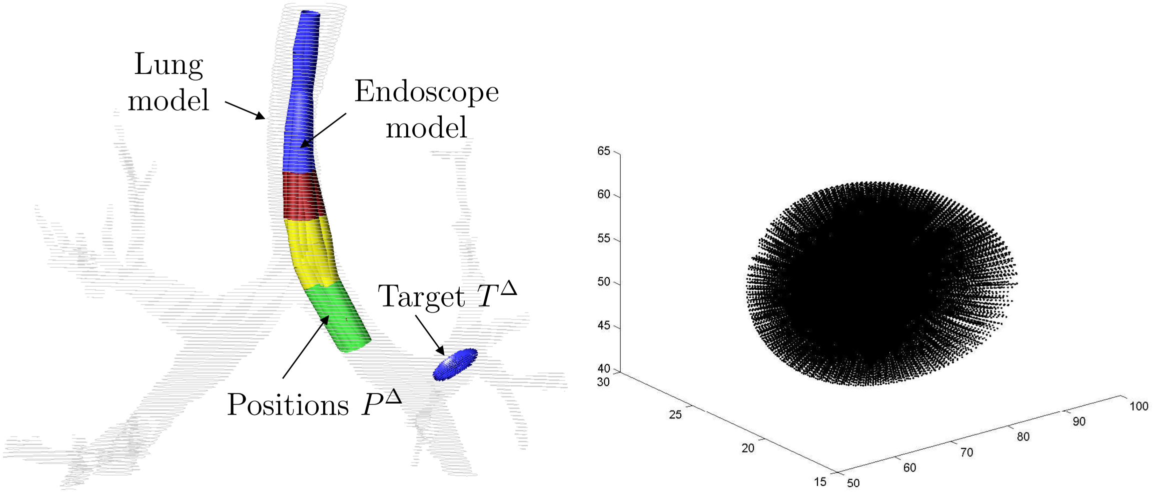

Figure 2.4: Left: Endoscope model (workspace) inserted into a lung model. Target T represented by an elliptical point cloud. Right: One full scan of T, plotted as a point cloud in the needle parameter domain N.

Algorithm 2.1 describes the kCP_Greedy() approach in pseudo-code. The description follows the basic structure of an implementation, realized in "Matlab[2]". The full Matlab code of this implementation can be found in Appendix A.3.

Array $A()$ (line 2) can be regarded as a set of 3-dimensional arrays, where each 3D array corresponds to a scan from one of the viewpoints in P^{Δ}. In other words, the size of $A()$ along the fourth dimension is $|P^{\Delta}|$. The size along the first three dimensions ($x-$, $y-$ and $z-$coordinates of a needle parameter) is given by the difference between the maximum and minimum parameter value, divided by the desired cell size Δ_N, respectively.

The instruction in line 11 corresponds to a projection of $A()$ from 4D to 3D. All 3D (needle parameter) arrays are merged (summed) together to one 3D array.

S_{idx} in line 23 is a set of subscripts. Each subscript denotes a 3D array (scan) that has a "1" at position (i, j, k).

---

[2] The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA

Algorithm 2.1 kCP_Greedy()

```
Input:   k'                  Number of needles .
         $\mathbf{I} = (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p})$   Set of point clouds or scans in $N^{\Delta}$.
         $l$                 Number of rows in $\mathbf{I}$ .
         $|P^{\Delta}|$      The number of candidate shapes, resp. viewpoints

Output:  $N_{\mathsf{opt}}$  Set of $k'$ needle parameters
         on screen           The $k'$ corresponding probabilities.

kCP_Greedy($k'$, $\mathbf{I}$)
$N_{\mathsf{opt}} \leftarrow \emptyset$
Create a 4-dimensional array $A()$
$A(:, :, :, :) \leftarrow 0$                    // Initialize all cells of $A$ with zero

for row $m \leftarrow 1$ to $l$ of $\mathbf{I}$ do
    $[i, j, k] \leftarrow$ indices of the cell, point $(x_m, y_m, z_m)$ falls in.
    $A(i, j, k, p_m) \leftarrow 1$
end for

for needle $n \leftarrow 1$ to $k'$ do
    $A_{\mathsf{proj}} \leftarrow \mathsf{sum}(A, 4)$// Sum $A$ along the 4th dimension. $A_{\mathsf{proj}}$ is 3-dimensional.

    $[i, j, k] \leftarrow \mathsf{max}(A_{\mathsf{proj}})$      // $(i, j, k)$ are the indices of the cell with max. value.
    $a \leftarrow A_{\mathsf{proj}}(i, j, k)$                // The number of scans that cover cell $(i, j, k)$.
    if $a = 0$ then
        exit
    end if

    $\mathbf{n}_n \leftarrow$ Needle parameter corresponding to the center of cell $(i, j, k)$
    $N_{\mathsf{opt}} \leftarrow N_{\mathsf{opt}} \cup \mathbf{n}_n$
    output: Probability $p_n \leftarrow \frac{a}{|P^{\Delta}|}$.

    $S_{idx} \leftarrow \mathsf{find}(A(i, j, k, :) == 1)$       // Find all scans that cover cell $(i, j, k)$ ...
    $A(:, :, :, S_{idx}) \leftarrow 0$                     // ... and remove them
end for
```

2.4 Validation

This section describes an experiment for testing the kCP-approach to the problem of finding an optimal biopsy strategy for a transbronchial needle aspiration (TBNA). The scenario is shown in Figure 2.4 (left). The model described in Chapter 1 was used to calculate a set of candidate shapes for the endoscope. An optimal set of needle parameters that covers these candidate shapes and the corresponding probabilities are computed using Algorithm 2.1 "kCP_Greedy()". The result is a table that could give valuable decision support to the bronchoscopist for choosing how many biopsies to perform and how to perform them.

Experiment 2.5: A TBNA Biopsy Strategy

Objective: *(1) Visualization: The first objective is to validate the approach by visualizing a scan of target* T^{Δ} *in the needle parameter domain* N^{Δ}*. Based on this visualization, several scans from different viewpoints are rendered to assess the amount of overlap.*

(2) Biopsy strategy: The second objective is to calculate needle parameters $\mathbf{n}_i \in N^{\Delta}$ *for* $i = 1, 2, \ldots, k$ *needles and the corresponding probabilities.*

Method: *The set of initial positions* P^{Δ}*, is given by the first rigid sleeve (red link in Figure 2.4) of the bending section of all endoscope shapes in the workspace (*$|P^{\Delta}| = 283$*).*

Function $f_{\mathsf{free}} : P^{\Delta} \times T^{\Delta} \to N^{\Delta}$ *represents a model for the endoscope's bending section (see Section 1.4.2), without any restrictions for its movement. In other words, all* $\mathbf{t} \in T^{\Delta}$ *can be reached by the needle. As described by Equation 2.1, for a given link and target* $\mathbf{t}$*,* f_{free} *calculates the following three parameter, needed to bring the biopsy needle into position* $\mathbf{t}$*: shaft rotation* α*, bending angle* β *and needle length* d*. These three parameters form the needle parameter domain* $N^{\Delta} \subset \mathbb{R}^3$*.*

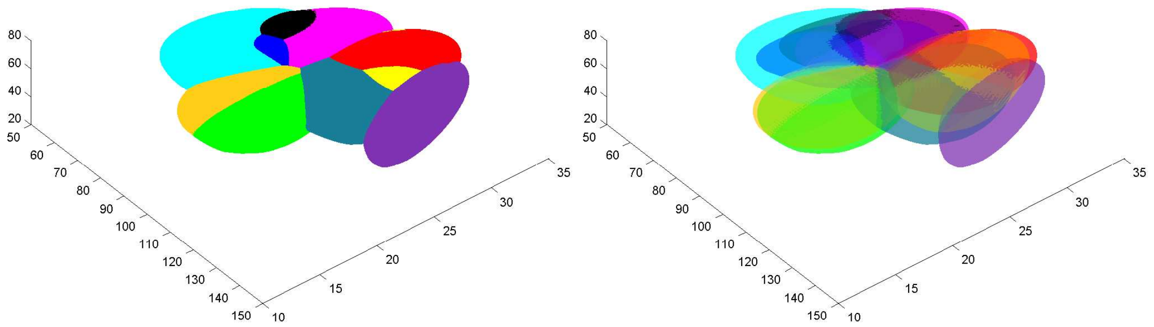

Figure 2.5: Color rendering. Left: Ten scans of T^{Δ}*, rendered as "flat shaded", convex hulls of the respective point clouds. Right: Same as left, but each scan rendered with a 0.5 transparency value (alpha blending). Bending model:* f_{free}*.*

The unrealistic assumption of an unrestricted tip movement was omitted with function f_{coll}*. This model uses collision detection to determine whether or not the tip of the endoscope can be bent to reach a target* $\mathbf{t}$*.*

Platform: *SGI 540 PC, Pentium 3 single, 550 MHz, 500 MB; Graphics board: Cobalt Graphics, 48 MB.*

Material:

1. *Patient scan (CT) of the tracheobronchial tree, Figure 2.4 (left).*
2. *Deformable model of a flexible endoscope as described in Chapter 1. The model calculates a set of possible endoscope shapes for a given insertion depth and target branch. The model can be configured to calculate several candidate shapes, which represent the endoscope's workspace under the given constraints.*
3. *A function* $f_{\mathsf{free}} : P^{\Delta} \times T^{\Delta} \rightarrow N^{\Delta}$ *and* $f_{\mathsf{coll}} : P^{\Delta} \times T^{\Delta} \rightarrow N^{\Delta}$*, which represent a model for the endoscope's actively bendable tip as described in Section 1.4.2.*
4. *The target of a TBNA is an enlarged lymph node, which is of circular or elongated shape and typically about 10 mm long. Such a lymph node was*

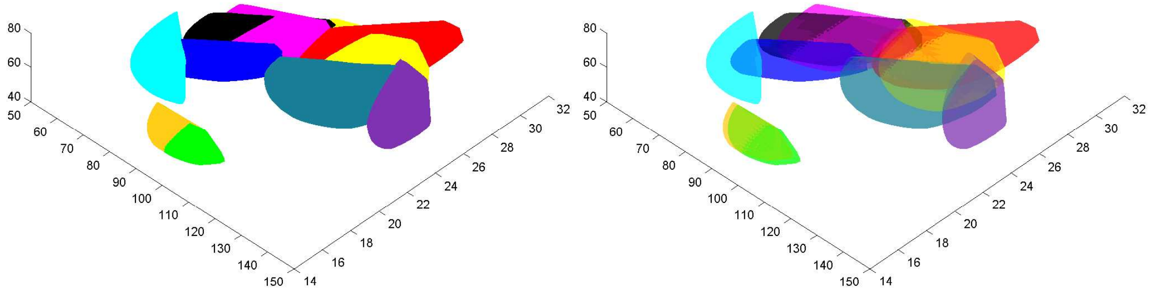

Figure 2.6: Color rendering. Same as Figure 2.5, however using bending model f_{coll}.

modeled as an ellipsoid of 10 mm length and 4 mm width. The ellipsoid is represented and rendered as a point cloud T^{Δ} *consisting of 5000 points.*

5. *Visualization of the scans was done using "Matlab*[3]*").*

6. *A Matlab implementation of Algorithm 2.1 "kCP_Greedy()". For the full code see Appendix A.3.*

Results: *(1) Visualization: Figure 2.4 (right) shows the visualization of one single scan of target* T^{Δ}*, using bending model* f_{free}*. Each point in* $\mathbb{R}^3$ *represents a value for shaft rotation, bending angle and needle length.*

Figure 2.5 (left) shows ten scans, each represented as a "flat shaded" convex hull of the original point cloud. Figure 2.5 (right) shows the same scenario, with each scan rendered with a 0.5 transparency value (alpha blending). Note the considerable overlap of these randomly chosen scans in 3D.

Figure 2.6 corresponds to Figure 2.5, with the difference that now bending model f_{coll} *was used. The movement restriction of the tip, caused by the physical confines, does not allow for a "full" scan of the target from a given initial position. Consequently, only portions of the scans in the previous figure remain.*

---

[3] *The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA*

However, the overlap between the scans is still considerable.

(2) Biopsy strategy: Figure 2.7 shows a series of screen shots from the Matlab implementation of algorithm "kCP_Greedy()", using bending model f_{coll}*. Figures (a)-(l) depict the needle parameter domain* N^{Δ}*. The grid indicates the cells in* N^{Δ}*. The cell size is given by* $5° \times 5° \times 2\,mm$*. Each scan of target* T^{Δ} *is depicted as a transparent (alpha blending) convex hull. Figure (a) shows the scans from all viewpoints in* P*. In Figure (b) the cell of maximum coverage was determined, all scans covering this cell were removed and a "1" was drawn in the center of that cell. Given the remaining scans, the cell of maximum coverage was determined, all scans covering this cell were removed and a "2" was drawn in the center of that cell. The result is shown in Figure (c). This procedure was repeated until 99.9% of the scans were removed (Figure (l)). Each number i drawn in* N^{Δ} *represents an alignment parameters* α_i, β_i, d_i*.*

The percentage of scans removed in each step corresponds to the probability of success for the respective biopsy needle placed. This probability and the cumulative probability was calculated for needle $i = 1 \ldots 6$ *as* $\frac{|V_i|}{|P^{\Delta}|=283}$*. The result is shown in Figure (b)-(g) and in the following table:*

Needle	*1*	*2*	*3*	*4*	*5*	*6*
Probability [%]	*42*	*28*	*16*	*4*	*4*	*3*
Sum	*42*	*71*	*86*	*90*	*94*	*97*

The table shows that two needles cover 71% and three needles 86% of all initial positions. The table represents a valuable decision support tool for the bronchoscopist. Depending on the concrete condition of the patient, he/she can decide whether or not a third or even a fourth needle is advisable. Based on this table, a third needle gives a considerable improvement of 16%, whereas the improvement of a fourth, fifth or sixth needle is negligible ($\leq 4\%$*).*

Computation time for Algorithm 2.1 was about two minutes.

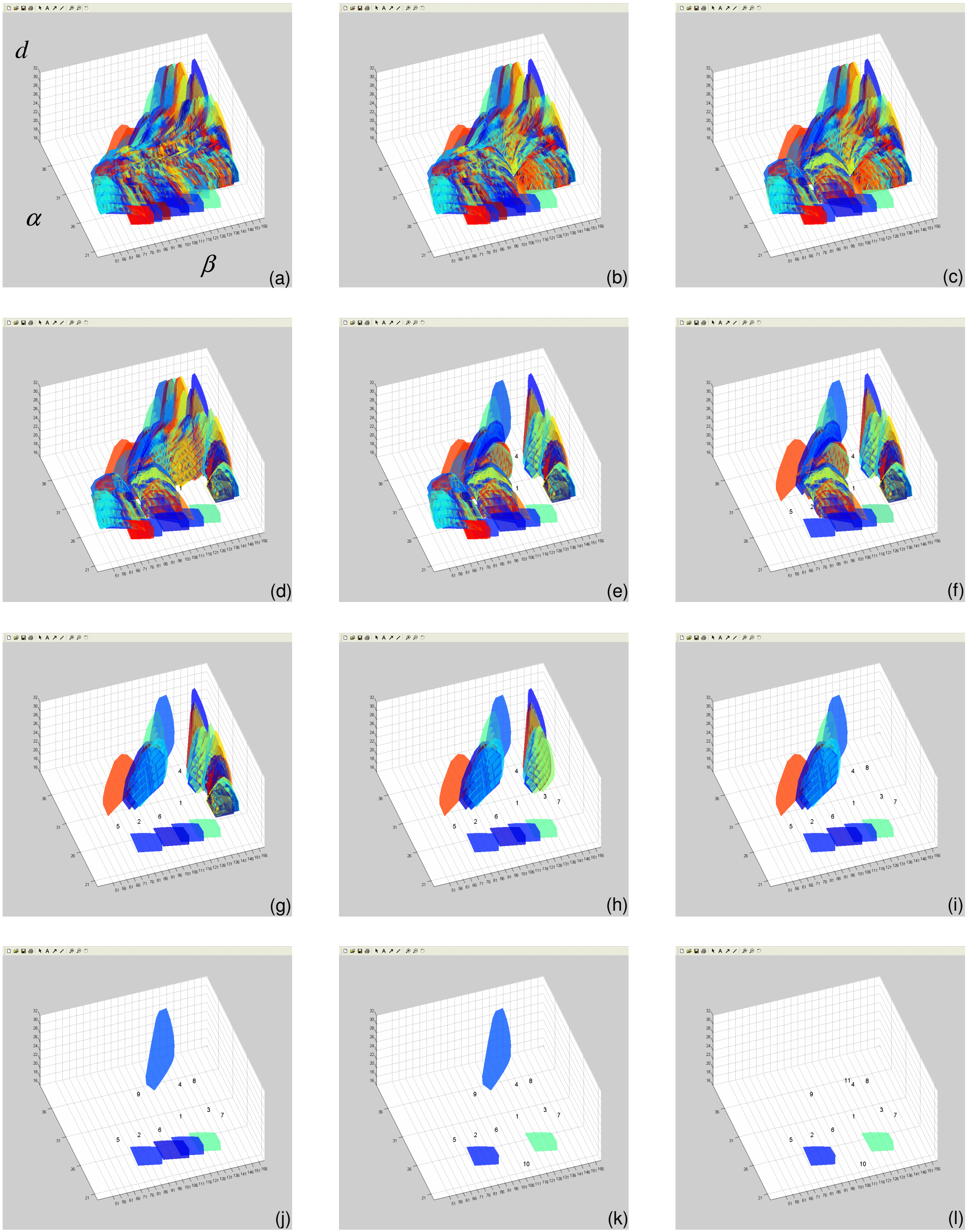

Figure 2.7: Color rendering. Screen shots from the "kCP_Greedy()" Matlab simulation. For the full code see Appendix A.3. Figures (b)-(g) correspond to needles 1-6 of the table.

Chapter 3

Model-based Guidance of Transbronchial Needle Aspirations (TBNA) Without a Computer in the Operating Room

Lung cancer is the most common cause of cancer deaths in men and women. Prognosis for the patient depends on the extent of the disease at the time of diagnosis. For early detection of lung cancer, a low dose chest CT (computed-tomography) scan has been used as a screening test. A screening examination is a study performed in order to detect a disease process at a time before signs or symptoms are present. Using sophisticated screening tests, it is hoped that if a disease is present it will be detected at an early stage. Preliminary results suggest that chest CT screening may lead to early detection of lung cancer in certain populations and may therefore significantly improve patients' survival rate.

One of the most common radiological findings in lung cancer screening examinations is a solitary pulmonary nodule. It is characterized as a single well-defined, round or oval lesion within the lung up to 6 cm in diameter. After detection, the nodule has to be classified into benign or malign. To assess the extent of the disease (staging [54]) or to establish ground-truth, a tissue sample has to be taken for a histological classification of the cell type. One way to get tissue, is to insert a needle from the outside directly into the suspicious lesion. However, a needle biopsy may cause a pneumothorax, a serious condition where air enters the intrapleural space. Another, more elegant way that avoids the risk of pneumothorax is a transbronchial needle aspiration.

Transbronchial needle aspiration (TBNA) [55] is a valuable minimal invasive procedure in the bronchoscopic diagnosis and staging of patients with lung cancer. The procedure allows nonsurgical access to mediastinal and hilar lymph nodes from the inside of the tracheobronchial tree. Traditionally this biopsy is performed by maneuvering a bronchoscope to a suitable site within the tracheobronchial tree (see Figure 3.1). Then the surgeon inserts a needle through the bronchoscope and punctures the bronchial wall in order to hit the target behind. This is literally a “blind” puncture since the target object is at no time visible by the bronchoscope. The relatively “blind” nature of the procedure and the physician’s lack of confidence about where to position the needle are obstacles to the widespread use and positive diagnostic yield of TBNA. In a 1991 survey [56] only 11.8% of experienced bronchoscopists routinely performed TBNA. The lack of direct sight and tactile feedback, together with the complicated hand-eye coordination, requires excellent education and training for the endoscopist [57].

In order to increase the chance of hitting the tumor, the surgeon takes more than one tissue sample from each target lesion. Studies have shown that up to five needle aspirations in the same site can be safely performed, although the optimal number of aspirations is yet to be clarified [58]. But despite the fact that the surgeon performs more than one needle aspiration in a single TBNA, this procedure has a failure rate of 60 to 80%, if the bronchial wall is not yet affected [59] [31]. If the subsequent histological examination of the tissue sample shows that the sample was useless, the

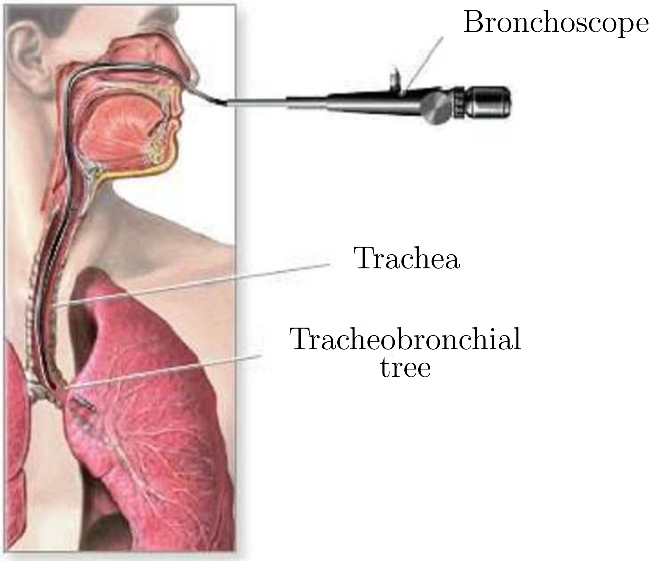

Figure 3.1: A bronchoscope is inserted through either mouth or nose into the tracheobronchial tree.

patient has to come back to the hospital and undergo another biopsy. This results in discomfort for the patient and higher costs for the health care system. For that reason, numerous approaches to guide TBNA biopsies have been developed.

Most of the approaches found in the literature aim for a continuous tracking of the bronchoscope's tip. Their goal is to determine at any point in time the position of the endoscopic camera and its viewing direction. It appears that guiding TBNA biopsies is regarded as an application of this more general problem. However, for the special case of TBNA biopsies, a continuous tracking seems over-engineered. From discussions with practicing bronchoscopists it turned out that they do not need to know at any point in time during the operation, where they are and in which direction they are looking. Bronchoscopists know the anatomy by heart and are very well capable of reaching a target branch without any guidance. However, once they reach the target branch, they need guidance with the fine-tuning of aligning the bronchoscope's tip with the target. In this chapter, a dedicated solution to guiding TBNA biopsies is presented that abandons the idea of a continuous tracking of the bronchoscope.

By performing a TBNA the surgeon faces two major problems: Firstly, hitting the target requires 3D imagination (coordination of the learned three dimensional anatomy with a fish-eye distorted 2D video image) together with the handling of the endoscope (hand-eye-coordination). Secondly, assessing how many tissue samples to take is a trade-off decision between patient safety and success probability.

This chapter presents a new guidance method for TBNA biopsies that helps the bronchoscopist with both problems. The basic idea is to calculate a TBNA-protocol. Regarding the first problem, the protocol describes in detail how to perform a number of tissue samples, by prescribing for each sample, how to handle the bronchoscope, in order to move the biopsy needle into the target lesion. Regarding the second problem, the protocol gives probabilities of success for each tissue sample. This allows the bronchoscopist to decide whether or not the gain of an additional biopsy justifies the associated discomfort respectively risk for the patient. During the operation, the bronchoscopist executes the protocol by setting the bronchoscope to the prescribed configuration. To gain control over the current configuration of the bronchoscope, a set of passive controls is used to monitor its degrees of freedom.

The endoscope model described in Chapter 1 is used to preoperatively simulate the insertion of an bronchoscope into a target branch of the tracheobronchial tree. The result is a set of candidate shapes for the real shape of the bronchoscope during the intervention. This set represents the initial position domain for the "optimal" $k-$needle placement strategy, introduced in Chapter 2.

The foundation of this TBNA guidance technique is an accurate model of a flexible endoscope, including its actively bendable tip. This suggests to denote the approach described here as a "model-based" approach.

3.1 Related Work

Techniques found in the literature for guiding TBNA can be classified into three different groups: imaging-based, vision-based and sensor-based approaches.

Imaging-based approaches use standard imaging techniques like conventional fluoroscopy, computed-tomography (CT), CT-fluoroscopy and ultrasonography, to visualize the endoscope, the advancing needle and the target lesion. Conventional fluoroscopy (C-arm) produces in real-time a two-dimensional projection image (x-ray) with limited inferior contrast resolution. The target lesion is usually not visible [60]. In order to keep the needle in the image plane, the position of the C-arm has to be updated frequently.

Conventional CT produces images of adequate resolution but the procedure is cumbersome and time-consuming since real-time imaging is not possible and each sequence must be prescribed in advance.

CT-fluoroscopy is a term for continuous-imaging CT that allows the visualization (one slice) of dynamic processes in real-time, like the insertion of a needle into the target lesion. White *et al.* [60] report a single case, where a TBNA was successfully performed under CT-fluoroscopy guidance. However, this technique is limited to axial images and requires significant CT-scanner time and causes additional radiation.

Shannon *et al.* [58] use ultrasonography to visualize the target lesion by inserting a catheter-enclosed ultrasound transducer through the working channel of the bronchoscope. A motor in the ultrasound unit rotates the transducer within the catheter, producing a cross-sectional ultrasound image oriented perpendicular to the long axis of the catheter. However, this study shows no significant difference in sensitivity compared to unguided TBNA. Images obtained from the sonography probe are of variable quality, user dependant (probe pressure) and may not be diagnostic. Furthermore, the transducer and the needle share the same port and cannot be inserted simultaneously, thus real time imaging during needle insertion is not possible.

Vision-based approaches use solely the image information from the bronchoscope's optical system (fiberoptics or CCD chip) to track the instrument's tip. This is an elegant approach since it generally requires no additional devices. The idea was pioneered by Bricault *et al.* [61] [62] [63]. The authors are able to compute the position and orientation of the tip in a special situation, where a bifurcation is shown in the image. The tracking fails, if no bifurcation is shown.

Sherbondy *et al.* [64] propose a two-step approach, which requires the cooperation of the bronchoscopist. In step one (preoperative), the user interacts with the 3D CT data by means of an virtual endoscopy environment and selects a "key-site" from which he/she wants to perform the biopsy. A virtual view from this site is recorded. During stage two (intraoperative) the bronchoscopist moves the real endoscope as close as possible to the "key-site". Then, the virtual view is registered to the current real view, using an iterative mutual-information based matching. All presented results are based on key-sites, showing a bifurcation.

Mori *et al.* [65] report to have achieved a continuous tracking of the bronchoscope's tip, even in the absence of strong features like a branching structure. The authors use epipolar geometry analysis and an image-based registration technique. The main idea is to match real endoscopic views with virtual endoscopic views. The computation time is six seconds per frame, which is not feasible for real-time procedures.

Sensor-based approaches use external sensors attached to the bronchoscope to determine the tip's position and orientation. A new technology called ShapeTape (Measurand Inc., Canada, www.measurand.com) might be used to visualize the entire endoscope within the tracheobronchial tree. ShapeTape is a lightweight, flexible ribbon with an array of fiber optic sensors along its length that measures its bend and twist. Attaching the ribbon to a flexible endoscope could allow measuring the endoscope's shape in 3D. This technology has not yet been applied to guiding flexible endoscopic procedures.

The Biosense intrabody navigation system (Biosense, Setauket, NY) uses electromagnetic fields to track a 1.5 mm sensor that can be attached to the bronchoscope's tip. Solomon *et al.* use this system in an animal study (swine) [56] to determine the feasibility of using real-time tracking technology coupled with preoperative CT data to enhance TBNA. Before the CT scanning, 10 to 20 metallic nipple markers are secured on the animals' anterior chest wall and later their coordinates are identified in the CT dataset. To register the sensor's position with the CT images, the sensor, while attached to the tip of the bronchoscope is touched to the nipples. Another registration problem is caused by the respiratory motion during the intervention, which causes deformations, not included in the static preoperative CT dataset. Drawing continuously the sensor's position in the static CT dataset will sometimes show the tip of the bronchoscope at physically impossible positions, for example outside the airways. According to the study, this situation was assessed by the bronchoscopist as "confusing".

To compensate for that, a second position sensor was attached to the animals' chest wall to gate respiratory motion. The position of the bronchoscope's tip was updated on the CT image monitor when the animal was in a stage of the breathing cycle that corresponds to the breathing state during image acquisition. The study showed an in-vivo accuracy of 4.2 mm ±2.6 mm (standard deviation).

Using the same system, Solomon *et al.* report in a later publication [66] an accuracy of 5.6 mm ±2.7 mm in a study with 15 adult patients. The authors also investigated the feasibility of a new registration method that involves touching internal structures of the tracheobronchial tree, instead of external skin markers (metallic nipple). This new method was subjectively judged to be superior for registering the position of the bronchoscope. However, the exact accuracy of the new method was not determined.

3.2 A Model-based Approach to TBNA Guidance

3.2.1 Motivation

The motivation for developing a new approach to TBNA guidance comes from the shortcomings of the existing approaches:

Imaging-based approaches like fluoroscopy or CT imaging cause additional radiation for the patient. Furthermore, guidance by fluoroscopy is cumbersome and time consuming, since the position of the C-arm has to be frequently updated. In addition, lung nodules are soft tissue, a tissue type for which this modality has a poor contrast resolution. Guidance by CT imaging is also cumbersome and time consuming, since the patient has to be moved frequently in and out of the scanner.

Vision-based approaches can not be considered as a mature technology since tracking is either not reliable or not capable of real-time processing. The approaches by Bricault and Sherbondy rely on the existence of strong features in the images, which is not always given. The approach by Mori *et al.* shows satisfactory tracking results, however, with a processing time of six seconds per frame, the approach is by far not ready for a real-time clinical application.

Considering a peak acceleration of the bronchoscope during the intervention and an additional simultaneous acceleration, caused by an active bending of the tip, it can be assumed that a processing time of about 30 frames per second is necessary to assure reliable tracking.

Sensor-based approaches seem to be the most obvious way to guide TBNA biopsies. One general problem however is the increase in size of the bronchoscope by affixing sensors to the outside of its tip or shaft. Furthermore, cables have to be passed all the way along the outside of the shaft to the sensors. This represents a big problem for the bronchoscope's sterilization. A potential solution to the latter problem is to design a new type of bronchoscope, which contains the sensor and cables inside. This however requires the hospitals to buy new and probably expensive equipment. An other concern is that this approach would counteract the current promising trend of developing ultra-thin bronchoscopes that are capable of reaching down several levels of bifurcations.

Registration between the sensor coordinate frame and the CT dataset represents an other difficulty. Solomon *et al.* use a large number of metallic nipples that need to be affixed to the patients chest prior to the scan. This leads to two problems: Firstly, the time between the scan and the TBNA should be kept to an minimum, since with increasing temporal distance, the chances increase that the patient changes or the nipples move or even fall off. Secondly, the patient has to be scanned twice. The first scan leads to a diagnosis and to the need for the patient to undergo a TBNA. Then the patient has to be scanned again, this time with the metallic markers on.

3.2.2 Calculating a TBNA-protocol

The approach described here to guide TBNA biopsies is based on the idea of preoperatively calculating a TBNA-protocol. The term "protocol" is used in the sense of a detailed plan (step-by-step instructions) for a medical procedure. The protocol prescribes how to handle the bronchoscope to achieve a successful biopsy. Describing the handling of a bronchoscope requires a parameterization of all its degrees of freedom. Figure 3.2 shows an endoscope and its degrees of freedom regarding an insertion into a tubular structure like the tracheobronchial tree. Given that the bronchoscope resides "somewhere" inside a predetermined target branch, the following four parameters well-define a TBNA-biopsy:

1. Bronchoscope insertion depth l.

2. Shaft rotation about the principal axis α.

3. Angle of tip deflection β controlled by the bronchoscope's angling wheel.

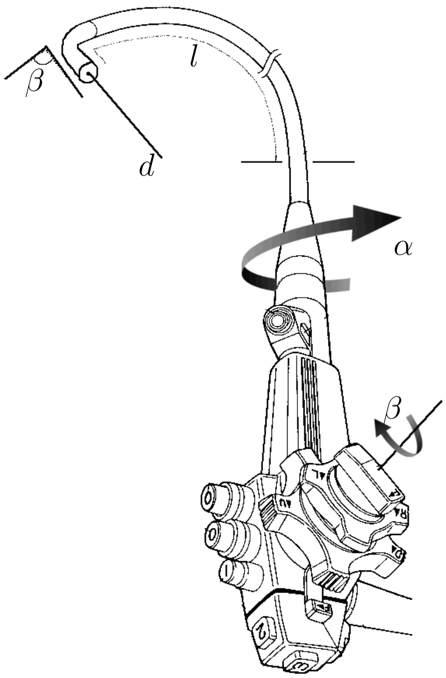

Figure 3.2: The handling of an endoscope can be parameterized using four parameters: l, α, β, d.

4. Needle length d.

These parameters are denoted as an “endoscope configuration” $\mathbf{c} = (l,\ \alpha,\ \beta,\ d)$. Parameters α, β, d are also called “alignment parameters”. An endoscope configuration $\mathbf{c}$ is called a “successful configuration”, if the needle tip of an endoscope set to $\mathbf{c}$, is located inside the target lesion.

During the operation, the bronchoscopist executes the protocol by setting the bronchoscope to the prescribed configurations. To ensure control over the current configuration of the bronchoscope, a set of passive controls is used to monitor parameters l, α, β, d (see Section 3.2.4).

The goal now is to calculate for a given biopsy scenario, a minimal set of endoscope configurations, of which one is a successful configuration. To allow the bronchoscopist to assess how many tissue samples to take, the probability of success for each sample is needed as well.

Algorithm 3.1 createTBNAprotocol()

```
Input:        Latest chest CT scan of the patient.
          $k$ Maximum possible number of tissue samples.
Output:       Set of $k$ endoscope configurations $\mathbf{c}_i$
              The $k$ corresponding probabilities of success.

createTBNAprotocol()
1: Reconstruct from the planar parallel cross sections a 3D model of the tracheobronchial tree and the target mass.
2: The radiologist / bronchoscopist preoperatively plans biopsy site $\mathbf{b}$.
3: Calculate the bronchoscope length $l$, needed to reach $\mathbf{b}$.
4: Calculate the bronchoscope's workspace $W$ (see Chapter 1), given its length $l$ and the branch containing $\mathbf{b}$ as the branch of interest.
5: Calculate a needle placement strategy (see Chapter 2), given workspace $W$ (candidate shapes) as the initial position domain $P$.
```

Algorithm 3.1 lists the necessary steps to calculate a TBNA-protocol, which includes both information. The algorithm describes a function "createTBNAprotocol()", which calculates for a given maximum number of k tissue samples a list of k endoscope configurations and the corresponding probabilities of success. The algorithm uses the results of Chapter 1 and 2 to calculate the protocol. The basic idea behind the algorithm is to regard the procedure of performing a TBNA biopsy as a two-step process: Firstly, the bronchoscope is inserted into the target branch. After the insertion, the bronchoscope's insertion depth is kept fixed. Secondly, the bronchoscope's tip is aligned (fine tuning) with the target lesion and the needle is advanced into the tumor. This allows to calculate the endoscope parameters α, β, d, based on a fixed insertion depth l.

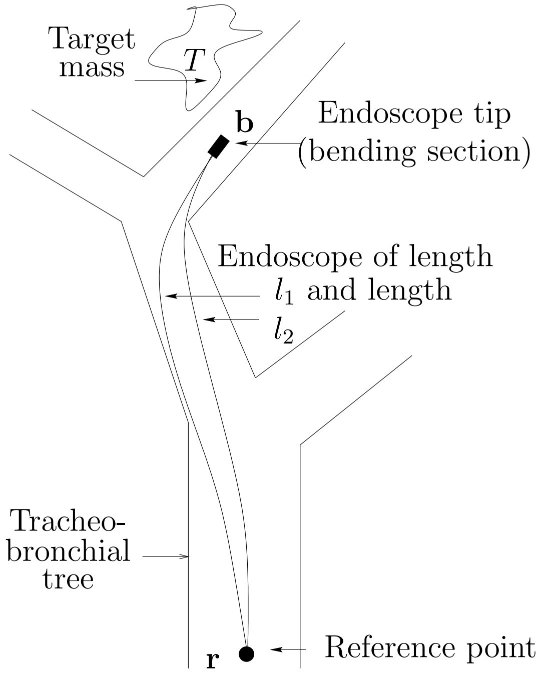

Figure 3.3: The TBNA scenario.

In the first step of the algorithm, a patient specific 3D model of the tracheobronchial tree and the target mass is reconstructed from a preoperative CT chest scan. In step two, the physician preoperatively plans the ideal biopsy site. The scenario is shown in Figure 3.3. Biopsy site $\mathbf{b}$ represents roughly the position and orientation of the bronchoscope's tip within the tracheobronchial tree from where to perform the biopsy. In step three the endoscope model described in Chapter 1 is used to calculate the insertion depth l, needed to reach biopsy site $\mathbf{b}$. The length is calculated with respect to reference point $\mathbf{r}$. In a later referencing step (Section 3.2.3), $\mathbf{r}$ is related to the actual position of the passive length control outside the body. Based on insertion depth l, Algorithm 1.3 "createFST()" calculates the bronchoscope's workspace in step four. The workspace is a set of bronchoscope shapes, representing possible candidate shapes for the real bronchoscope, inserted into the same branch. Since all shapes of the workspace are of length l, the alignment parameters α, β, d can be calculated in step five, using Algorithm 2.1 "kCP_Greedy()".

3.2.3 Registering the Virtual with the Real Bronchoscope

This section deals with the issue of registering the bronchoscope parameters l, α, β, d, derived from a preoperative 3D dataset, with the current configuration of the bronchoscope during the intervention. For each parameter a mutual zero-point regarding preoperative planning and the configuration during the intervention has to be found. For parameter β and d, such a mutual zero-point is obvious. For the first, it is the straight tip (zero degrees bending). For the latter, it is the needle length just before leaving the working channel (zero mm infeed).

For parameter l and β, two mutual zero-points outside the bronchoscope have to be found, which can be easily identified in the CT dataset as well as during the intervention. Such zero-points could then be used for registration in the following way: Firstly, in the preoperative planning phase, the needed insertion depth l and shaft rotation β is calculated as an offset to the respective zero-points. Secondly, during the intervention, the bronchoscopist resets the bronchoscope by setting its insertion depth and shaft rotation to the respective zero-position. This state is recorded by marks on the external (outside the patient's body) controls (see Section 3.2.4). For performing the biopsy, the bronchoscope is then set to l mm insertion depth and β degrees shaft rotation relative to these zero-marks on the controls.

An obvious candidate for a zero-point would be the entry point of the bronchoscope into the body, so either the mouth or nose. This however is not feasible, since a regular chest CT usually images the body starting from below the vocal chords. In other words, a natural landmark within the tracheobronchial tree has to be selected. Studying the anatomy of the lungs revealed that the "main carina", a keel-shaped part of the tracheobronchial tree, which marks the bifurcation of the trachea into the left and right lung, can serve as a mutual zero-point for both parameters. Figure 3.4 (right) shows an endoscopic image of the main carina.

Regarding the insertion depth, the idea is to touch the carina during the intervention with the bronchoscope. This state can be recorded as the zero-position for

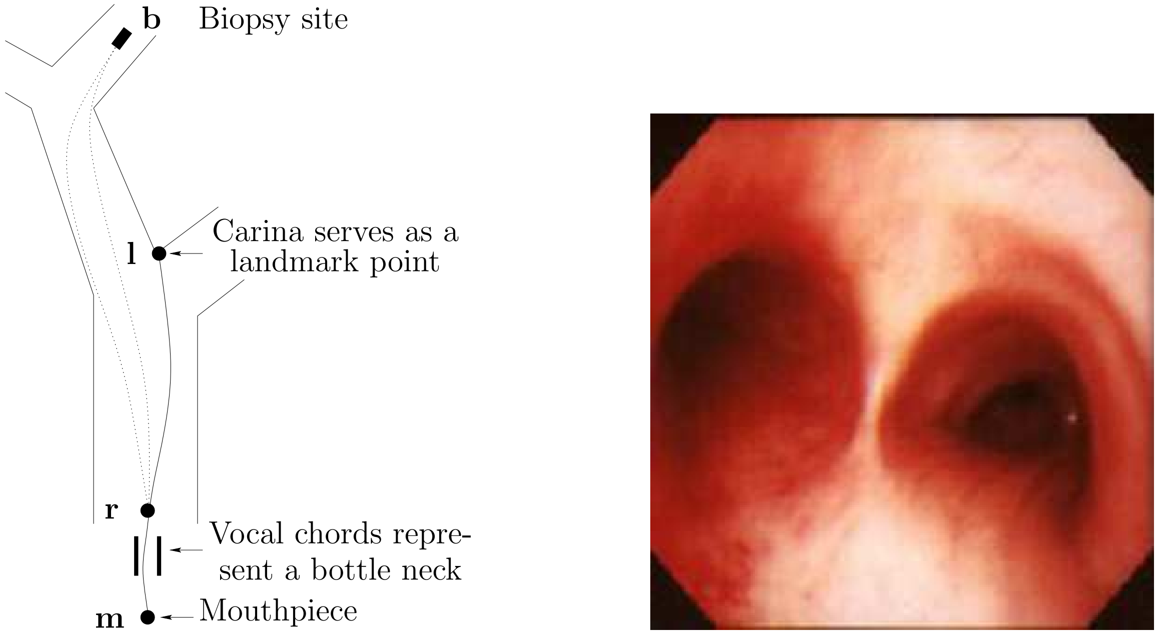

Figure 3.4: Landmark based registration. Left: The insertion depth to the biopsy site can be given as an offset to the insertion depth to the main carina. Right: Endoscopic view of the main carina.

the insertion depth by marking off the endoscope length, for example at the patient's mouthpiece. The corresponding protocol parameter l has to be calculated relative to this length. The situation is shown in Figure 3.4 (left). Let $\widetilde{\mathbf{ml}}$ denote the length of the bronchoscope from the mouthpiece to the landmark (carina). The insertion depth l, needed for the biopsy, can then be given as an offset to $\widetilde{\mathbf{ml}}$. This offset can be calculated as the difference between $\widetilde{\mathbf{rb}}$ and $\widetilde{\mathbf{rl}}$.

$$l = |\widetilde{\mathbf{rb}} - \widetilde{\mathbf{rl}}| \tag{3.1}$$

Reference point $\mathbf{r}$ was chosen to lie right behind the vocal chords, because the vocal chords form a bottleneck regarding all possible path a bronchoscope can take. This justifies the use of algorithm “createFST()” for length calculations, since this algorithm requires a fixed start link.

Regarding the shaft rotation, the rotation angle, which shows the carina to appear, for example vertically, can serve as a zero-point. During the intervention, after

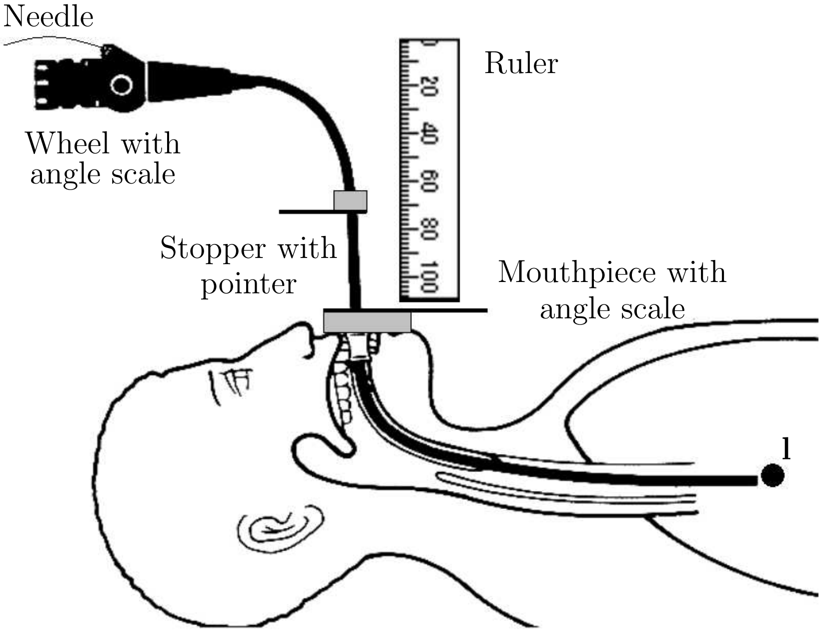

Figure 3.5: Passive controls for monitoring the endoscope configuration.

touching the carina, the bronchoscopist withdraws the bronchoscope until the carina is fully visible. Then he / she rotates the shaft until the carina appears vertically. This angle can be recorded as the zero-position by a mark on the control outside the patient. During the planning phase the same angle can be determined by rendering a view from a virtual endoscopic camera. The radiologist/bronchoscopist rotates this virtual view until it shows the main carina to appear vertically. The shaft rotation β, needed for the biopsy, is then given as an offset to this angle.

3.2.4 Controlling the Protocol Execution

In order to set the bronchoscope to the configuration given by the protocol parameters, control over the inserted length l, the shaft rotation α, the tip deflection β and needle length d is needed during the intervention. Figure 3.5 shows a set of passive (non electronic) controls, for controlling all parameters.

Since the biopsy is performed based on a fixed bronchoscope length, a stopper is used to prevent the bronchoscope from penetrating the body deeper than desired

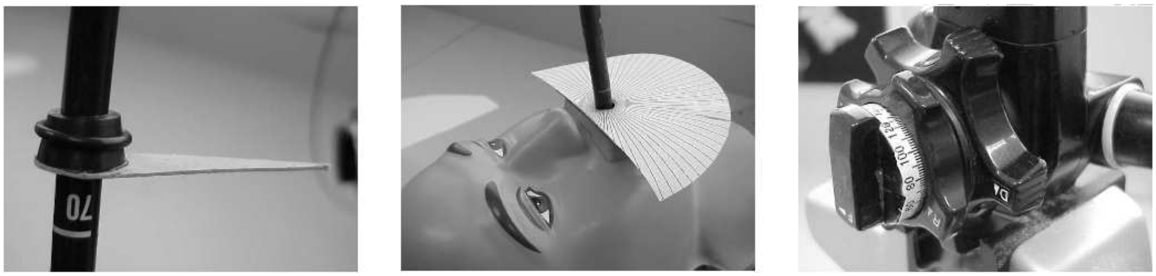

Figure 3.6: Passive Controls. Left: Stopper with pointer. Center: Mouthpiece with angle-scale. Right: Angle-scale attached to the bending wheel.

(Figure 3.6 (left)). To set the insertion depth of the bronchoscope, the stopper is moved l mm away from the base-plate of the mouthpiece (using a ruler), while the bronchoscope is in zero-position, which means while touching landmark **l**. The stopper is locked at this position, using a rubber ring (see Figure 3.6 (left)). Note that the use of a mouthpiece (bite guard) is highly recommended [67] to avoid damage to the bronchoscope by the patient's teeth.

To control the shaft rotation, a pointer has been attached to the stopper and a goniometer (angle-scale) was attached to the base plate of the mouthpiece (center figure). When the bronchoscope is in zero-position regarding the shaft rotation, which means when the bronchoscopic image shows the main carina to appear vertically, the pointer is moved to a zero-mark on the goniometer. When the base plate stops the insertion of the bronchoscope (due to the locked stopper), any shaft rotation moves the pointer along the angle-scale, indicating the current offset to the (vertical) main carina. To set the shaft rotation, the bronchoscope is rotated β degrees relative to the zero-mark.

Tip bending is controlled by an angle-scale attached to the angling wheel located at the bronchoscope's control head (right). During the planning phase, a look-up table is computed that maps the angle of tip deflection to the wheel angle. The needle length is controlled by marks on its proximal end and by using the opening of the biopsy port as zero-point.

3.2.5 The TBNA-protocol

Figure 3.7 shows the universal TBNA-protocol. It can be divided in three parts:

- Lines 1-5 contain the registration instructions.
- Lines 6-7 describe how to reach the biopsy site.
- Lines 8-10 describe the alignment of the tip with the target.

Protocol parameters $l, \alpha_i, \beta_i, d_i$ represent the patient specific part of the protocol. The instructions themselves remain unchanged for different patients.

Instructions 1-5 realize the registration between the reference frame of the endoscope model and the reference frame of the real bronchoscope. Executing these instructions defines a zero-point for the endoscope length and shaft rotation. Line 1-2 defines the carina as the zero-point for the endoscope length and line 4-5 defines the vertical carina as the zero-point for the shaft rotation. The withdrawal in line 3 moves the bronchoscope away from the carina to the rotation site, in order to see the carina in the bronchoscope's view.

For lines 6-7 there are two alternatives: The left column represents the instructions for a biopsy site that is distal to the rotation site. The right column is for biopsy sites proximal to the rotation site. The protocol parameter l in line 6 gives the bronchoscope length to the biopsy site as an offset to the bronchoscope length to the carina.

After execution of line 7, the bronchoscope reached the biopsy site and the insertion depth is fixed by the stopper. The following instructions 8-10 use the alignment parameters α, β, d to align the bronchoscope's tip with the target lesion. Line 8 sets the shaft rotation, controlled by the stopper's pointer to α degrees. This angle is given as an offset to the rotation angle, which shows the main carina to appear vertically.

Once the shaft rotation is calibrated and the insertion depth is fixed (lines 1-7), it is possible to perform a series of tissue samples, according to the k-needle placement

1. Touch carina with the bronchoscope tip.
2. Move stopper to mouthpiece.
3. Withdraw bronchoscope 30 mm.
4. Rotate shaft until carina appears vertically in the bronchoscope's view.
5. Move stopper to mouthpiece and set its rotation-pointer to 0 degrees.

6. Move stopper $l + 30$ mm away from the mouthpiece and lock stopper.	6. Withdraw bronchoscope $l - 30$ mm.
7. Insert bronchoscope into the branch of biopsy-site $\mathbf{b}$ until the stopper hits the mouthpiece.	7. Move stopper back to mouthpiece and lock stopper.

8. Rotate endoscope shaft until the shaft rotation-pointer points to α_i degrees.
9. Set rotation of the bending-wheel to β_i degrees.
10. Insert needle d_i mm beyond the working channel outlet.

Figure 3.7: The TBNA-protocol, including patient unspecific instructions and patient specific parameters $(l, \alpha_i, \beta_i, d_i)$.

strategy. The arrow indicates that lines 8-10 can be executed repeatedly. Each pass through the loop corresponds to one tissue sample, determined by a triple (α_i, β_i, d_i) for $i = 1, \ldots, k$.

3.3 Experiments and Results

This section presents three experiments to investigate the feasibility of guiding TBNA biopsies by executing a TBNA-protocol (model-based approach). Experiment 3.6 investigates the feasibility of the proposed registration method and the feasibility of using the passive controls for monitoring the current bronchoscope configuration. The experimental setup consists of a lung phantom connected to a head phantom. Experiment 3.7 investigates the accuracy of the endoscope model (Chapter 1) in predicting the shape of a real bronchoscope inserted into the lung phantom. The shape of the real bronchoscope is measured using the optical tracking system introduced in Section 1.5.1, together with cylindrical markers. Experiment 3.8 tests the entire approach of a protocol based TBNA guidance. For various targets, a TBNA-protocol is calculated and executed and the overall accuracy is determined.

Experiment 3.6: Repeatability

Objective: *This experiment represents the first step in investigating the feasibility of guiding TBNA biopsies by executing a TBNA-protocol. The first questions that need to be answered are: (1) Can the insertion depth, shaft rotation, bending angle and needle length of the endoscope be reliably controlled by the passive controls described in Section 3.2.4? (2) How accurate is the proposed landmark based registration method, described in Section 3.2.3? Or in other words, how reliably can the endoscope be set to the respective zero-position for insertion depth and shaft rotation?*

Material:

1. *Video endoscope OLYMPUS GIF-100 (Gastroscope)*
2. *A lung phantom, made from transparent PVC tubes and a head phantom, made from styrofoam wearing a plastic face mask (see Figure 3.8). The lung phantom was enlarged by factor 1.5 (trachea diameter: 30 mm) in*

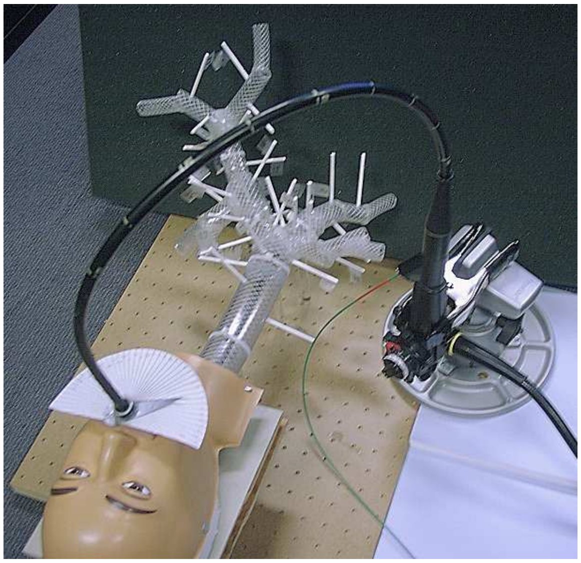

Figure 3.8: Experimental setup, showing lung and head phantom and the endoscope with the biopsy needle.

comparison to an adult lung, to compensate for the fact that no bronchoscope (average shaft diameter: 6.0 mm) but a "Gastrointestinal Videoscope" (Olympus GIF 100, shaft diameter: 9.5 mm) was available for this non-sterile experiment. Only the lung phantom was scanned (CT, $512 \times 512 \times 382$*, 1 mm slice distance, 1.2 mm slice thickness) and a 3D model of the tracheobronchial tree was reconstructed from the 382 planar cross sections.*

3. *A set of passive controls, as shown in Figure 3.6.*

Design: *The idea is to measure the repeatability of hitting a target by executing the registration procedure of the TBNA-protocol. The task is to measure the variation of parameter* α, β *and* d*, given a fixed insertion depth* l*, to hit a target point visible on the inner wall of the lung phantom. The first step is to perform the registration (line 1-5 of the TBNA-protocol). In the second step, the endoscope is moved into a target branch of the lung phantom. The third step is to use the CCD camera in the endoscope's tip to touch the target point with the biopsy* *needle. In step four, the current shaft rotation, bending angle and needle length are read from the controls and recorded. Steps 1-4 are repeated 20 times for the same target.*

This test is performed five times, each time for a different target point. Consequently, five datasets are recorded, each containing 20 measures of the parameters α, β, d*. To calculate the overall variation, the smallest and largest outlier of each parameter in each dataset were rejected. This is to compensate for the fact that several outliers were obviously the result of measuring errors. Each parameter's range was calculated as the difference between the smallest and largest value.*

Results: *The following table shows the worst and average range among the five datasets for each parameter:*

	Range	
Parameter	worst	average
α	7°	6°
β	6°	4°
d	7 mm	4 mm

For TBNAs, a needle length of about 3 cm is an upper limit for the length the needle is pushed out into the target. Given this maximum needle length and an average range of 6° *(0.1 radians) for the shaft rotation, the average distance (error) at the needle's tip is about 3 mm. The same reasoning regarding the bending angle, results in an average error of about 2 mm at the needle tip. An average error of 4 mm at the needle tip is directly caused by the average range of the needle length.*

These variations of 3, 2 and 4 mm at the needle tip have to be assessed regarding a target diameter of at least 10 mm and up to 60 mm.

Experiment 3.7: Endoscope Model Accuracy

Objective: *This experiment investigates the following question: How accurately can the shape of the OLYMPUS GIF-100, inserted into the lung phantom, be predicted in a pre-operative simulation, using the endoscope model described in Chapter 1?*

An accurate prediction of the real endoscope's shape can be used to measure the length needed to touch the main carina ($\widetilde{\mathbf{rl}}$*) and to reach the biopsy site (*$\widetilde{\mathbf{rb}}$*). Both measures are essential in the calculation of a TBNA-protocol (see Section 3.2.3).*

Material: *Hardware:*

1. *Optical tracking system ARTtrack*[1] *1, comprising of two IR cameras and software.*
2. *Pointing device (see Figure 1.22 on Page 1.22) with a calibrated tip, which can be used to determine the coordinates of a point in 3D space, with respect to the tracking system described above.*
3. *Calibration wand, which represents a metal rod with several ball-shaped markers attached to it. The distance between the markers is known with a high precision. The wand comes with the tracking system and is used to calibrate the intrinsic and extrinsic camera parameters.*
4. *Video endoscope OLYMPUS GIF-100, 4.75 mm radius.*
5. *30 passive retro-reflective strip markers (10 mm width, 0.5 mm thickness) wrapped around the endoscope's shaft like a cylinder in a distance of ca. 25 mm (see Figure 1.21 on Page 72).*

---

[1] *A.R.T. (Advanced Realtime Tracking) GmbH, Gewerbestr. 17, D-82211 Herrsching, URL: www.ar-tracking.de*

6. *The transparent lung and head phantom from the previous experiment. Additionally, 37 holes (4 mm diameter) were randomly drilled into the lung model.*
7. *"Matlab*[2]*", mathematical computing and visualization software.*

Software: The lung model was reconstructed (ca. 14000 triangles) from 207 planar parallel cross sections (slices), obtained from a CT scan of the lung phantom. Before the scan, a marker stick, made from cardboard (cylindrical, 65 mm length, 4 mm diameter) was placed in each hole, so that its base was aligned with the inner surface of the tube (see Figure 3.8). The cardboard markers were clearly recognizable in the CT dataset, since their X-ray attenuation (density) lies between the density of air and PVC plastic. This allows to manually determine the center of the hole in the CT dataset (in the plane of the inner tube surface), by determining the center of the cylinder's (stick's) base.

Design: *The idea is to use the optical tracking system to measure the shape of the OLYMPUS GIF-100 inside the lung phantom. Figure 3.9 shows the experimental setup, consisting of the tracking system, the lung phantom and the endoscope with the cylindrical markers. The measured shape is then compared to the shapes generated by the endoscope model.*

The endoscope is manoeuvered to five different sites within the lung phantom. At each site, the position of the markers detected by the tracking system are recorded. Also, the insertion depth is recorded for each site.

Then the virtual endoscope is "manoeuvered" to the same sites, with the same insertion depth, using Algorithm "create∼FST()". For each site, 10 endoscope shapes are calculated by setting the selectivity of the energy filter to $p = 10$.

For each site, the measured shape, given by the coordinates of the cylindrical markers, is compared to the 10 virtual endoscopes.

---

[2] *The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA*

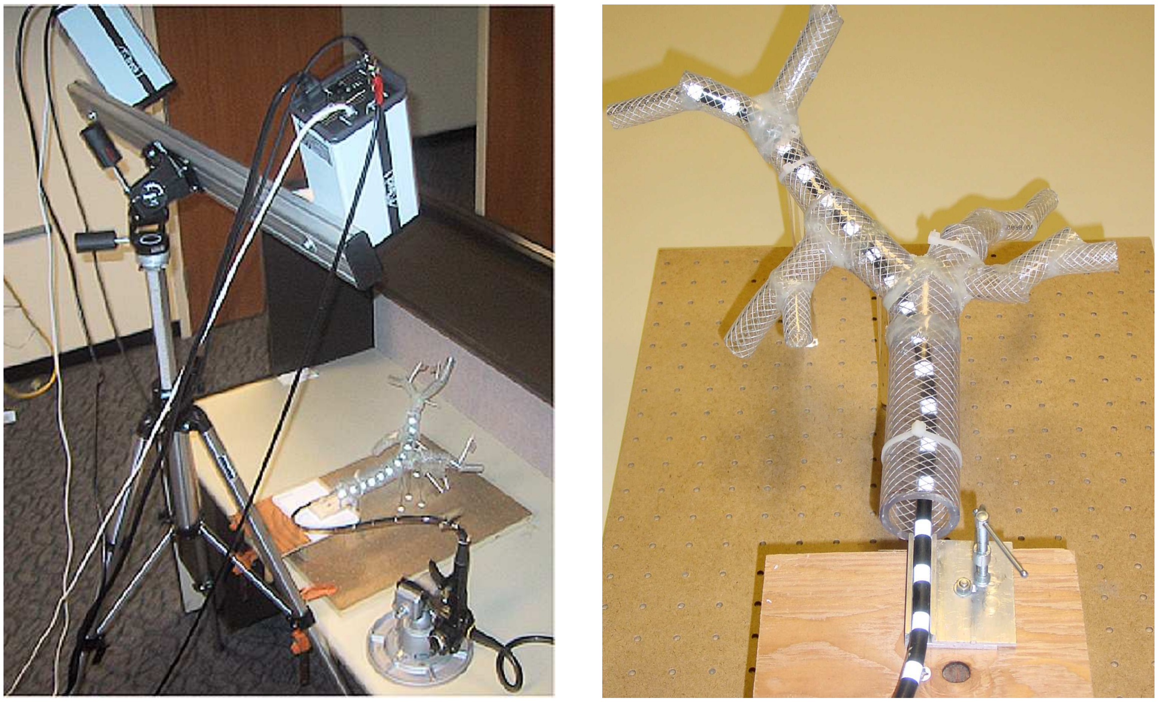

Figure 3.9: Experimental setup. Left: Tracking cameras and lung phantom. Right: The OLYMPUS GIF-100 with reflective markers inserted into the lung phantom.

Methods: *It has already been demonstrated in Experiment 3.3 that the endoscope can be tracked accurately, using cylindrical markers. In this experiment another difficulty arises. The tracking cameras "see" the markers through the transparent PVC tubes. Does the refraction, caused by the tubes diminish the accuracy of the measurements?*

To answer this question, the calibration wand was inserted into the tubes and the position of each marker was recorded. Then the distances between the markers were calculated and compared to the ground-truth.

In order to compare the measured endoscope with the virtual endoscope, a coordinate transformation has to be found that maps the coordinate system of the tracking system into the coordinates system of the endoscope model. To solve this registration problem, a method according to the ICP (Iterative Closest Point) algorithm [45] is used. At first a set of corresponding reference points

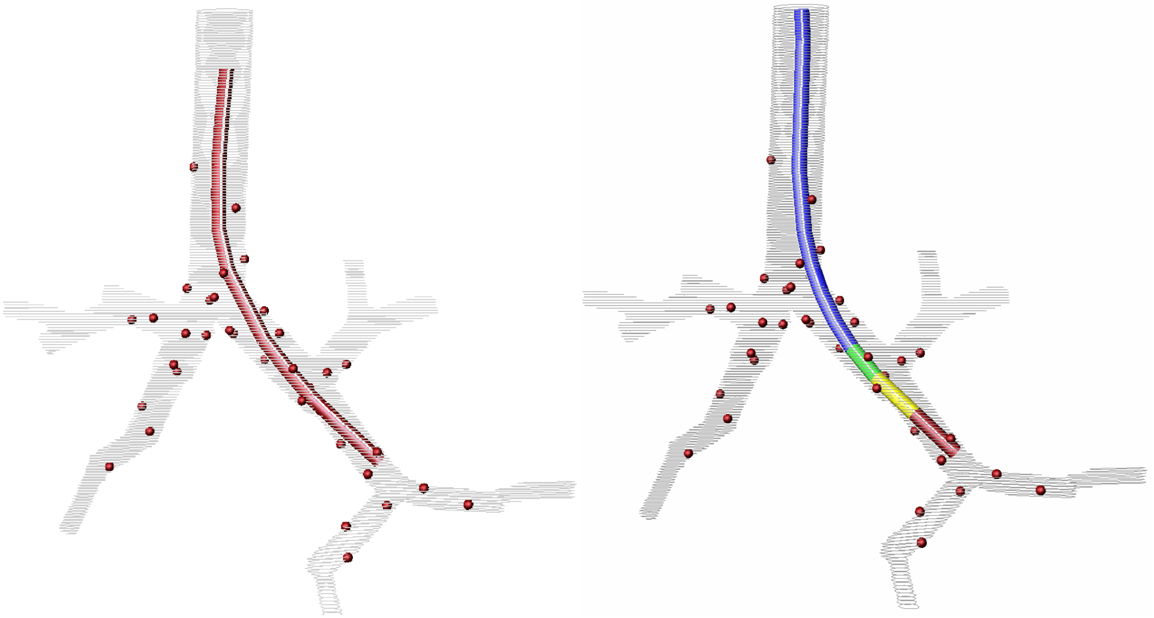

Figure 3.10: Model accuracy. Left: Measured shape. Right: Predicted shape.

is specified in both datasets. In the CT dataset, the centers of the holes (see Material, Software) can serve as a set of reference points. The corresponding points in the lung phantom can be found by inserting the pointing device into each hole until the tip is aligned with the inner surface of the tube. Then the position of the tip is recorded.

Subsequently, a rigid body transformation is calculated that represents the best fit in the least-squares sense between the two sets of reference points. To solve the resulting non-linear minimization problem, the "Levenberg-Marquardt" method [46] [47] [48] is used. See Appendix A.2 for Matlab code that computes the rigid body transformation.

Results: *The influence of refraction is negligible. The divergence between the measured marker distance and the ground-truth is within the accuracy of the tracking system (0.1 mm).*

Figure 3.10 shows one of the five measured shapes of the OLYMPUS GIF-100 *(left) and the corresponding virtual endoscope (right), generated by Algorithm "create∼FST()". In the left figure, two adjacent marker positions were connected by a cylinder. To calculate the matching error between the two endoscopes, it is sufficient to compare the tips (second sleeves, see Figure 1.2 on Page 14) only, in consideration of the fact that the application is a needle biopsy. Consequently, the measured position and orientation of the OLYMPUS GIF-100's tip is compared to the position and orientation of the virtual endoscope's tip. Note that two cylindrical markers were affixed to the second sleeve of the real endoscope to measure its position and orientation.*

The virtual endoscope shown in Figure 3.10 (right) represents one out of the 10 generated shapes. Both figures also show the set of reference points that was manually obtained from the CT dataset.

For each of the five sites, the best matching result between the real (measured) endoscope tip and the 10 virtual endoscope tips is considered. The following table shows the best and worst result out of these five matches:

	Accuracy	
Sleeve	worst	best
Position	2.5 mm	0.7 mm
Orientation	7°	4°

As mentioned earlier, a needle length of about 3 cm is an upper limit for the length a TBNA biopsy needle is pushed out into a target. Given this maximum needle length and an orientational error of maximum 7° (0.12 radians), the average distance (error) at the needle's tip is about 4 mm.

This error of 4 mm at the needle tip and an maximum positional error of 2.5 mm at the needle tip have to be assessed regarding a target diameter of at least 10 mm and up to 60 mm.

Experiment 3.8: TBNA-protocol Accuracy and Timing

Objective: *This experiment investigates the overall accuracy of the TBNA-protocol approach. The objective is to measure the error of predicting a set of endoscope configurations to hit a target with the biopsy needle. This error is a combination of the following three sources of error: Registration accuracy (Experiment 3.6), accuracy of predicting the endoscope's shape after reaching the target site (Experiment 3.7) and the accuracy of calculating the alignment parameters* α, β, d.

A second objective is to measure the time needed to calculate the TBNA-protocol and the time needed to execute it.

Material:

1. *Video endoscope OLYMPUS GIF-100, 4.75 mm radius.*
2. *Lung and head phantom as shown in Figure 3.8.*
3. *Figure 3.11 (left) shows five out of the 37 reference points of the previous experiment that have been selected as target points for this experiment. To make the holes better visible by the endoscopic camera, a white marker stick has been inserted into each hole and aligned with the inner surface of the tubes. As shown in Figure 3.11 (right), the holes appear in the endoscopic view as white disks of 4 mm diameter . The center of each disk has been manually identified in the CT dataset.*

Platform: *SGI 540 PC, Pentium 3 single, 550MHz, 500MB; Graphics board: Cobalt Graphics, 48MB.*

Design: *The biopsy target is assumed to be a point (center of disk) on the inside of the lung phantom. To measure the error of hitting the center, parameters* α, β, d *of the predicted configuration(s) are compared to the "ground-truth" for this target. The ground-truth is the average of 10 configurations, each representing the resulting configuration of the endoscope after its needle was manually*

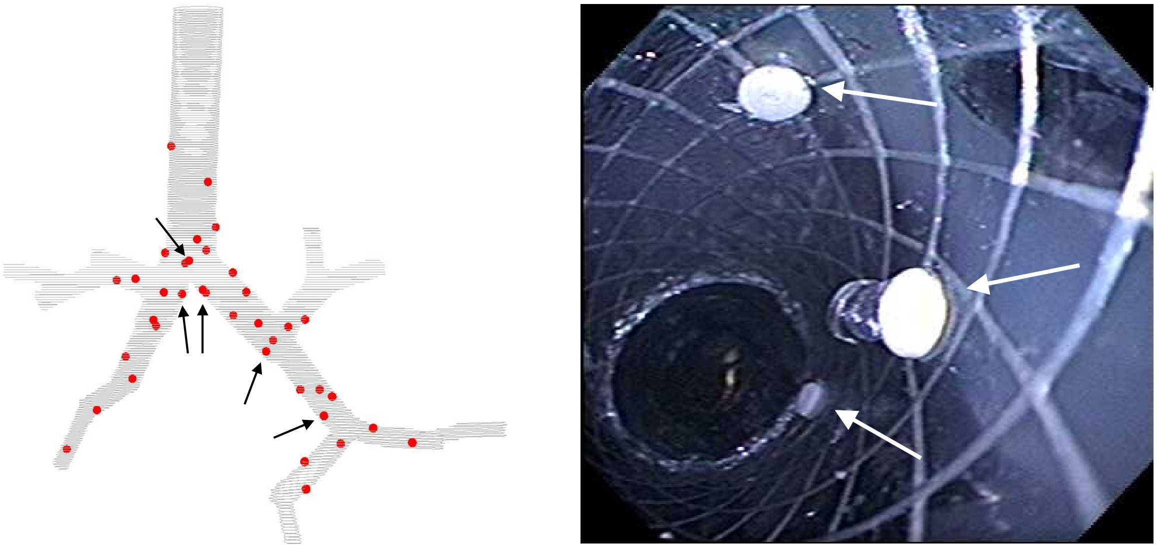

Figure 3.11: Target points. Left: The five target points subject of this experiment. Right: Endoscopic image (OLYMPUS GIF-100) of the target points from inside the lung phantom.

maneuvered into the target. The predicted insertion depth l *was taken as the insertion depth for determining the ground-truth.*

Two TBNA-protocols have been calculated for each of the five target points shown in Figure 3.11 (left). The first calculates one configuration to take one tissue sample ($k = 1$)*, the second calculates three configurations for taking three tissue samples* ($k = 3$)*. In the latter case, only the best prediction result among the three is regarded. By using a simple error model, the difference between the predicted alignment parameters and the ground-truth is expressed as a single distance between the needle tip according to the predicted configuration and the target point.*

Results: *The results show that shaft rotation* α *is by far the most sensitive parameter with an average error of* 12°*. The angle of tip deflection* β *could be predicted with an average error of* 3° *and needle length* d *with an average error of* $1\,mm$*.*

The following table shows the worst and average error out of the five target sites after performing one and three biopsies:

	Accuracy	
TBNA-protocol	worst	average
1 Biopsy	24 mm	13 mm
3 Biopsies	12 mm	5 mm

The table demonstrates the great yield of performing several biopsies systematically. By performing one biopsy, spherical targets of 48 mm diameter can be hit reliably. In contrast, by performing three biopsies, targets of 24 mm diameter can be hit reliably and targets of 10 mm diameter can be hit in the average.

It took about 3 minutes for each biopsy site, to calculate the corresponding TBNA-protocol. This period includes the time needed for user interaction for specifying the biopsy site and rotating the virtual camera until the carina appears vertically.

To execute a protocol, the additional time needed (with respect to an unguided biopsy) is about nine minutes. It took about five minutes to setup the controls, about three minutes for the registration procedure and about one minute to set the endoscope to the prescribed configuration.

3.4 Towards Clinical Evaluation

The experiments described in the previous section are based on a phantom of the tracheobronchial tree. The two main differences between this phantom study and a patient study are the existence of mucus within the tracheobronchial tree and the motion and deformation of the tracheobronchial tree during the respiratory cycle.

Mucus inside the tracheobronchial tree influences frictional forces and may therefore change the movement of the bronchoscope's tip. In a first step towards clinical evaluation, it has to be investigated if the proposed instrument model can cover these variations by simply producing several candidate shapes. If the real shape is similar to one of the candidate shapes, the subsequent "optimal" needle placement strategy guarantees a successful biopsy.

The question regarding the influence of respiratory motion on the overall accuracy can be divided in two sub-questions: The influence of respiratory motion on the registration and the influence on the prediction of the endoscope's tip.

The landmark based registration technique used in this work is less sensitive to respiratory motion than for example a technique that is based on a set of external markers. External markers that are attached to the patient's chest, move considerably during the respiratory cycle. This represents a problem in relating these moving markers to the corresponding markers in the static CT data set of the patient. In contrast, the anatomical landmark (carina) is in close proximity to the center of the body and therefore is expected to move less than a marker located on the patient's chest. But more importantly, it can be assumed that the landmark and target move in sync during respiration. In other words there is only little movement of the landmark relative to the target, which may also result in good accuracy in a breathing patient.

Regarding the influence of respiratory motion on the prediction of the endoscope's tip, a closer look on the characteristic and extent of this motion has to be taken. The analysis of organ motion and deformation during the respiratory cycle is an active field of research. Wood [68] investigated the changes of the three-dimensional lung structure itself during inflation, using computed tomography. In a study involving four dog lungs, she determined the effect of lung inflation on branch segment length, cross sectional area and angle. The results show that, as the lung expands in three spatial dimensions with inflation, the individual airway segments elongate along their axes. Also, the airways were elliptical at lower pressure and changed to a more circular shape during inflation. However, the results show very little change in branching angle with lung inflation: The change in the average angle for each lung is $3°, 4°, 3°$ and $0°$. "As the lung inflates, the parenchyma pulling on the airway to change its shape in one direction seems balanced by a pull in the opposite direction. This complies with the isotropic theory of lung inflation.".

If an animal study would show a significant decrease in the overall accuracy caused by respiratory motion, the static lung model used in this work could be replaced by a dynamic model. Generally, the findings described by Wood [68] are encouraging in the sense that only the length and cross-sectional shape of each branch are significantly affected by respiratory motion, while the angle between branches vary very little. It appears possible to incorporate these changes into the patient specific but static model of the tracheobronchial tree used in this work.

One approach would be to scan the patient once, for example at the end of expiration. The lung model reconstructed from this scan can serve as the starting point for generating a set of synthetic lung models that represent various stages of the lung during inspiration. To create the first synthetic model form the initial reconstructed model, the medial axis of the tracheobronchial tree [69][70] has to be calculated. Then, the shape of each branch can be approximated by an elliptic contour, which is locally perpendicular to the respective medial axis. Once the first synthetic model is obtained, the above described changes induced by lung inflation can be incrementally incorporated, producing a new model at each stage. As the lung inflates, each branch is slightly elongated and its cross-sectional contour is changed from elliptic to more circular. Then the instrument model is inserted into each synthetic lung model. The sum of candidate shapes of all simulations represents the uncertainty of the tip position and orientation under respiratory motion. A preoperative insertion simulation of a flexible bronchoscope into a dynamic lung model could be the final step needed to bring the idea of a “device-less” guidance of TBNA to clinical reality.

Conclusions

In this thesis, a new approach to the problem of intraoperative guidance of flexible endoscopy has been proposed. The main contribution of this work is given in the three chapters 1, 2 and 3:

The contribution of the first chapter is a new model for flexible instruments. The model can be configured to represent either catheters or endoscopes. Special attention has been paid to model certain mechanical constraints often found with flexible endoscopes: A tip that can be bent to a much higher degree than the general shaft flexibility, rigid sections (sleeves) within the bendable tip, a considerable shaft diameter and a slightly bigger diameter of the tip.

An interesting property of the model is the option to generate several similar shapes for the instrument's tip. The model requires no initialization in form of an initial "good guess" (spline control points) for the final shape and no preprocessing.

Furthermore, a method to measure the center line of a real endoscope has been developed. The method uses a new type of marker that does not significantly alter the shape of the endoscope and therefore does not influence the outcome of the measures. It has been used to validate the accuracy of the instrument model during insertion into a lung phantom. The results show that for a given insertion depth, the model can accurately "predict" the position and orientation of the instrument's tip. The instrument model was applied to a synthetic data set, a data set obtained from a lung phantom CT scan and two data sets obtained from a patient CT scan (airways and brain artery). Assuming a moderate insertion velocity and acceleration, a real-time simulation of an endoscope inserted into a tracheobronchial tree has been achieved.

The contribution of the second chapter is an “optimal” strategy for placing k biopsy needles given a large number of possible initial needle positions. This problem arises for example in guided, endoscopic needle biopsies, where the position of the endoscope’s tip is known with some error. Beside the actual needle parameters, a table is provided to the physician, which contains a probability of success for each needle. By placing the needles in order of decreasing probability, the physician can decide after each needle, whether the gain in the overall probability of success by employing the next needle outweighs the risk to the patient. Overall, the approach can provide valuable decision support to the physician regarding how many needles to place and how to place them.

The contribution of the third chapter is a new approach to the problem of intraoperative guidance of flexible endoscopy. The advantage of this approach over existing approaches (imaging-, vision- and sensor-based, see Chapter 1.2) is its simplicity. No computer, or other electrical devices are needed in the operation room. As a result, the time needed for a guided endoscopic procedure is only marginally longer than for a conventional, unguided procedure.

Also, in contrast to vision-based approaches the approach proposed is inherently “real-time” during the intervention. The bronchoscopist operates at his / her own speed, without having to wait for, or worry about being lost by the tracking software. This also contributes to an overall faster execution of the procedure.

Furthermore, the approach doesn’t require a specialized endoscope. Standard fiberoptic or video endoscopes can be used without any structural modifications. This makes the approach very cost efficient, since hospitals and physicians can use their existing equipment. The passive controls attached to the endoscope remain outside the patient’s body and can be removed after the procedure. This allows to sterilize the endoscope as usual.

Finally, no additional scan of the patient is needed for registration. For the sensor-based approach, a second scan might be necessary with the patient wearing a set of external markers, used for registration. Since the external markers move considerably during respiration, this marker-based registration technique also requires an additional monitoring of the breathing motion. In contrast, the approach proposed in this thesis uses a landmark-based registration technique, which doesn't require any external markers.

The approach presented in this thesis opens up a new paradigm for intraoperative guidance of endoscopic procedures. It represents a shift away from the extensive usage of additional hardware in the operating room, towards a more minimalist approach. Existing approaches seem to focus on solving the more general problem of a continuous tracking of the endoscope's tip. They regard the problem of guiding TBNA as an application to this more complex problem. In contrast, the approach described in this thesis represents the first dedicated system for guiding TBNA. The solution appears to be minimal regarding the required additional hardware, effort, time and costs in the operating room, with respect to a conventional unguided procedure. All three parameters are critical factors in a clinical environment that is increasingly forced to operate as effectively and cost efficiently as possible.

With this work, a proof of concept has been established for the feasibility of a "device-less" guidance of flexible endoscopy. The results are encouraging and justify further research. The biggest open question that needs to be answered, is the accuracy of the approach in a clinical study. As discussed in Section 3.4, the two main differences between the phantom study conducted in this thesis and a patient study are the existence of mucus within the airways and the respiratory motion. In a first step towards clinical evaluation, an animal study has to be conducted to clarify the influence of these two factors on the number of aspirations (biopsy needles) needed, to guarantee a successful TBNA. If the number of aspirations is less or equal than the maximum number possible for the individual patient, the simplicity and cost efficiency of this approach justifies its use over more expensive and time consuming approaches.

Appendix A

Matlab Functions

Matlab[1] is a programming language from “The Math Works, Inc.” for scientific computing and visualization. This chapter gives listings of Matlab code (Version 6) for various functions used in this project. The character “%” denotes a comment line.

A.1 Distance Between Markers and a Planar Surface

This function determines the average distance and standard deviation between the cylinder shaped markers and the board plane, given by three disk markers. For more details, see Experiment 1.3 in Section 1.5.1.

```
function = CalcAverageDistance();


% read the input file
p = dlmread('..\..\Marker\series2\series3\3boardMarker.txt');


% p1, p2, p3 are points on the plane
```

---

[1] The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760 USA

```
p1 = [p(1, 5:7),1]';

p2 = [p(2, 5:7),1]';

p3 = [p(3, 5:7),1]';


% calc normal


r1 = p2 - p1;

r2 = p3 - p1;


n = cross(r1, r2);


l = norm(n);


n = n * (1/l);


% A plane is defined by its normal n and a point on the plane p1


% P = px-p1     for any point px on the plane


% n dot P = 0


% Let n=(a, b, c)',  p1 = (x1, y1, z1)',  px = (xx, yx, zx)'


% P = (xx - x1, yx -y1, zx - z1)


% a(xx - x1) + b(yx - y1) + c(zx - z1) =0
```

```
% a xx + b yx + c zx - a x1 -b y1 -c z1 =0

% D = - (a x1  + b y1 + c z1)

% minimum distance of point q = (xq, yq, zq) to the plane:
% (a xq + b yq + c zq + D) / sqrt(a^2 + b^2 + c^2)

D = - dot(n, p1);

q = p3 + (0.4711 * n);  % test vector with distance to plane = 0.4711

dist = (dot(n, q) + D ) / (sqrt(sum(n.^2)));

disp(dist);  % should be 0.4711


% load marker and calculate average distance

Q = dlmread('.\markerInModelFrame.txt');

Q = [Q(6:21,:)]; %discard the markers that were not in contact
% with the board during the experiment (markers close
% to the instrument's control head)

sumDist = 0;

for i = 1:16
    q = [Q(i,:)]'
```

```
    dist = (dot(n, q) + D ) / (sqrt(sum(n.^2)));
    dA(i) = dist;
    sumDist = sumDist + dist;
end

disp('mean distance: ');
disp(sumDist / 16);

disp('standard deviation: ');
disp(std(dA));
```

A.2 Rigid Body Transformation

The following function calculates a rigid body transformation T, which maps a point with respect to the tracking camera coordinate system into a point with respect to the "M"- model coordinate system. For more information, see Experiment 4 in Section 1.5.1.

```
function T = findRigidBodyTransform();

% read input files

% Pt: reference points with respect to the tracker’s reference frame
Pt = dlmread ('.\MICCAI02\TrackerData\refPoints_all8.txt', ' ');

% reference points with respect to the "M" model’s reference frame;
% arbitrary order
Pm = dlmread ('.\MICCAI02\TrackerData\refPointsInModelFrame.txt', ' ');

% visualization

hold on;
plot3(Pt(:,1), Pt(:,2), Pt(:,3), 'o');

figure;
hold on;
plot3(Pm(:,1), Pm(:,2), Pm(:,3), 'o');

% find transformation H
% H maps a point from tracker space to model space
```

```
% m = [tx, ty, tz, wx, wy, wz] is an initial value for the transform
% three translational coordinates, three rotation angles
% here, an initial value (guess) is not known
m = [0 0 0 0 0 0];
H=rigidbody_find(Pm, Pt, m);

disp('mapping transform:');
disp(H);

% calculate average mapping error:

l =0;
for i = 1:8
    p =[Pt(i,1), Pt(i,2), Pt(i,3), 1];
    p=p';
    q = H*p;
    q = q';
    r =[Pm(i,1), Pm(i,2), Pm(i,3), 1];
    s = q-r;
    l = l + norm (s);
    disp(norm(s));
end
disp('average mapping error:');
disp(l/8);

T=H;
return
```

```
% function H=rigidbody_find(Pp, P, m);
% find the rigid body transformation between two sets of 3D points.
% m=[tx ty tz wx wy wz] is an initial guess for the transformation.

function H=rigidbody_find(Pp, P, m);

SP=size(P); if (SP(1)~=3 & SP(2)==3)
   P=P';
elseif (SP(1)==3 & SP(2)~=3) else
   disp('Error: the dimension is wrong');
end

SPp=size(Pp); if (SPp(1)~=3 & SPp(2)==3)
   Pp=Pp';
elseif (SPp(1)==3 & SPp(2)~=3) else
   disp('Error: the dimension is wrong');
end

LB=[-10000 -10000 -10000 -2*pi -2*pi -2*pi];
UB=[ 10000  10000  10000  2*pi  2*pi  2*pi];

options = optimset('Display','iter','Jacobian','off',
'LevenbergMarquardt','on','LargeScale','off', ...
'MaxIter',6000,'MaxFunEval',6000,'TolFun',eps,'TolX',eps);

[m, a, b, c, d, e, f] = lsqcurvefit('rigidbody_map', m, P, Pp, LB, UB,
options);
```

```
% convert m from [tx, ty, tz, wx, wy, wz] to a homogeneous transform H

%  translation vector: T
tx = m(1);
ty = m(2);
tz = m(3);

T=[ tx; ty ; tz];

% rotation sub-matrix: R
wx = m(4);
wy = m(5);
wz = m(6);

sx = sin(wx);
cx = cos(wx);
sy = sin(wy);
cy = cos(wy);
sz = sin(wz);
cz = cos(wz);

R(1,1) = cz * cy;
R(1,2) = cz * sy * sx - sz * cx;
R(1,3) = sz * sx + cz * sy* cx;
R(2,1) = sz * cy;
R(2,2) = sz * sy * sx + cz * cx;
R(2,3) = sz * sy * cx - cz * sx;
```

```
R(3,1) =-sy;
R(3,2) = cy * sx;
R(3,3) = cy * cx;

% transform H:
H=[ R     T;
    0 0 0 1];

return
```

A.3 Algorithm kCP_Greedy()

```
function kNeedle()

clear all;

% read input data

% without restrictions to the tip movement
%I = dlmread ('.\k-NeedleBiopsy\kNeedleData_NoCollision.txt', ',');

% remove % in next line for reading the data set based on
% restricted tip movement (collision detection)
I = dlmread ('.\k-NeedleBiopsy\kNeedleData.txt', ',');

% data I is given as (X, Y, Z, P), four column vectors
% each row (x, y, z, p) represents a point (x,y,z) in the needle
% parameter domain from viewpoint p


% Visualization of the scans
h = kNeedleVisualization(I);


%################################################################
%
%    Subdivision of needle parameter domain N into cells Nd
%
```

```
% cell dimension:
xc = 5;
yc = 5;
zc = 2;

% "round" X component towards the nearest grid point
Xmin = min(I(:, 1));
Xmax = max(I(:, 1));

Xx = [floor(Xmin) : xc : ceil(Xmax)+xc];       % generate grid (cells)
Xy = [1 : length(Xx)];                          % generate cell indices

Xd = interp1(Xx, Xy, I(:, 1), 'nearest');      % "round" to grid

% "round" Y component towards the nearest grid point
Ymin = min(I(:, 2));
Ymax = max(I(:, 2));

Yx = [floor(Ymin) : yc : ceil(Ymax)+yc];       % generate grid (cells)
Yy = [1 : length(Yx)];                          % generate cell indices

Yd = interp1(Yx, Yy, I(:, 2), 'nearest');      % "round" to grid

% "round" Z component towards the nearest grid point
Zmin = min(I(:, 3));
Zmax = max(I(:, 3));
```

```
Zx = [floor(Zmin) : zc : ceil(Zmax)+zc];       % generate grid (cells)
Zy = [1 : length(Zx)];                         % generate cell indices

Zd = interp1(Zx, Zy, I(:, 3), 'nearest');     % "round" to grid



% Visualization of the grid
axisH = gca;
% set ticks to size of cell array
set(axisH, 'XTick', Xx);
set(axisH, 'YTick', Yx);
set(axisH, 'ZTick', Zx);

grid on                     % show cell grid

% ####################################################################
%
%     Create a discretized needle parameter domain Nd
%     Nd is represented by a four dimensional array of cells
%     The first three dimensions represent the three needle parameters
%     (alpha, beta, d)
%     The fourth dimension (pages) represents the "scans" of T from
%     viewpoint p \in P
%
%     '0' : cell is not covered
%     '1' : cell is covered
%
```

```
Pd = I(:,4);
Pmax = max(Pd);                  % number of 'scans'

% create and initialize cell array with zeros
Nd = zeros(length(Xy), length(Yy), length(Zy), Pmax);

% mark the covered cells with '1'.
Ind = sub2ind(size(Nd), Xd, Yd, Zd, Pd);
Nd(Ind) = 1;

% sub2ind is used to create a linear index from multiple subscripts.

% ######################################################################
%
%         Greedy approach to the set covering problem (SCP):
% Select at each step the subset that gives the highest improvement
% (coverage)

p = 0;      % probability
p_sum = 0;  % cumulative probability
n = 0;      % needle number
a = 1;      % number of scans that cover a cell

disp 'n,    p,    p_sum';  % display on screen

while a > 0     % loop over all needles until all 'scans' are selected
    n = n +1;   % needle number
```

```
% to calculate by how many subsets each cell is covered, we sum
% up Nd along the fourth dimension (scans).

Proj = sum(Nd, 4);

[dummy, ind] = max(Proj(:));
[i, j, k] = ind2sub(size(Proj), ind);

% (i, j, k) is the index of the cell with the highest coverage

a = Proj(i, j, k);  % number of scans that cover cell (i, j ,k)

if a == 0   % if no scans left
    break
end

p = (100 / Pmax ) * a;                      % ratio

p_sum = p_sum + p;

disp(strcat(int2str(n), ': ', num2str(p,2), ', ',
                                            num2str(p_sum,2)));

% after selecting a subset, we have to remove it from the set
% of all subsets
% first step: find all "scans" that cover cell (i, j, k)
Idx = find(Nd(i, j, k, :) == 1);
```

```
    % second step: remove the ’scans’ by setting all cells of
    % the respective page to ’0’
    Nd(:, :, :, Idx) = 0;

    % Visualization:

    %  transform a point (i, j, k) \in Nd into a point (x , y, z) in N
    x = interp1(Xy, Xx, i, ’linear’);
    y = interp1(Yy, Yx, j, ’linear’);
    z = interp1(Zy, Zx, k, ’linear’);

    % print needle number in cell center
    t = text(x, y, z , int2str(n));
    set(t, ’FontSize’, 15);

    axis manual;                          % turn off automatic scaling

    set(h(Idx), ’Visible’, ’off’);        % removes the covered ’blobs’
end


%######################################################################

function h = kNeedleVisualization(I)

% data is given as (X, Y, Z, P), four colun vectors
% each row (x, y, z, p) represents a point (x,y,z) in the needle
% parameter domain from viewpoint p
```

```
Pmax = max(I(:, 4));

color = colormap;

figH = figure;      % handle to figure

set(figH, ’Renderer’ , ’OpenGL’);
set(figH, ’DoubleBuffer’, ’on’);

grid on

hold on;

axis auto;

P = [1 : Pmax];

for i = P

    [idx] = find(I(:, 4) == i);

    %plot3(I(idx, 1), I(idx, 2), I(idx, 3),’c.’);

    C = convhulln(I(idx, 1:3));

    h(i) = patch;       % handle to patch
```

```
    set(h(i), 'Vertices', I(idx, 1:3));
    set(h(i), 'Faces', C);

    r = ceil(rand * 64);    % random color

    set(h(i), 'FaceColor', color(r, :));

    set(h(i), 'FaceAlpha', 0.5);

    set(h(i), 'FaceLighting', 'Phong');

    set(h(i), 'EdgeColor', 'none');
end
```

Bibliography

[1] D. J. Vining et al. Virtual Bronchoscopy: A New Perspective for Viewing the Tracheobronchial Tree. Radiology 189(P):438, 1993.

[2] D. J. Vining and D. W. Gelfand. Noninvasive Colonoscopy Using Helical CT Scanning, 3D Reconstruction, and Virtual Reality (Abstr.). SGR Scientific Program, 1994.

[3] D. J. Vining et al. Virtual Colonoscopy (Abstr.). Radiology 193:446, 1994.

[4] B. Geiger and R. Kikinis. Simulation of Endoscopy. In *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science - 905, pages 277–281. Springer-Verlag, April 1995.

[5] W. E. Lorensen et al. The Exploration of Cross-Sectional Data With a Virtual Endoscope. In *Interactive Technology and the New Paradigm for Healthcare*, pages 221–230. IOS Press and Ohmsha, 1995.

[6] G.D. Rubin, C.F. Beaulieu, V. Argiro, H. Ringl, A.M. Norbash, J.F. Feller, M.D. Drake, R.B. Jeffrey, and S. Napel. Perspective Volume Rendering of CT and MR Images: Applications for Endoscopic Viewing. Radiology 199, pages 321–330, 1996.

[7] S. Napel, G. Rubin, C. Beaulieu, R. Jeffrey Jr., and V. Argiro. Perspective Volume Rendering of Cross-Sectional Images for Simulated Endoscopy and Intra-Parenchymal Viewing. In *SPIE's Medical Imaging*, volume 2707-07, page 16, Newport Beach, CA, 1996.

[8] A. P. Wunderlich, T. Fleiter, A. J. Aschoff, and H.-J. Brambs. Virtuelle Endoskopien aus CT- und MR Datensaetzen. In *Bildverarbeitung fuer die Medizin 1998*, Informatik aktuell. Springer-Verlag, 1998.

[9] S.F. Sheikh, D.S. Paik, C.F. Beaulieu, G. D. Rubin, R.B. Jeffrey Jr., and S. Napel. Wide-Angle Virtual Endoscopy using Multiple-View Rendering: The Virtual Cockpit. In *RSNA-EJ*, volume 2, 1998.

[10] D.S. Paik, C.F. Beaulieu, R.B. Jeffrey Jr., G.D. Rubin, and S. Napel. Automated Flight Path Planning for Virtual Endoscopy. In *Medical Physics*, volume 25(5), pages 629–637, 1998.

[11] David J. Vining et al. Free Flight: A Virtual Endoscopy System. In *CVRMed-MRCAS '97*, Lecture Notes in Computer Science - 1205, pages 413–416. Springer-Verlag, March 1997.

[12] L. M. Auer, D. Auer, and J. F. Knoplioch. Virtual Endoscopy for Planning and Simulation of Minimally Invasive Neurosurgery. In *CVRMed-MRCAS '97*, Lecture Notes in Computer Science - 1205, pages 315–318. Springer-Verlag, March 1997.

[13] C. v. Pilcher, K. Radermacher, W. Boeckmann, and G. Rau G. Jakse. 3D-Visualisation for Image Guided Surgery - A Case Study in Video Endoscopy. In *CVRMed-MRCAS '97*, Lecture Notes in Computer Science - 1205, pages 311 – 314. Springer-Verlag, March 1997.

[14] S. Aharon, B. M. Cameron, and R. A. Robb. Computation of Efficient Patient Specific Models From 3-D Medical Images: Use in Virtual Endoscopy and

Surgery Rehearsal. In *Information Processing in Medical Imaging*, Lecture Notes in Computer Science - 1230, pages 429–434. Springer-Verlag, June 1997.

[15] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, and K. Katada. A Fast Rendering Method Using the Tree Structure of Objects in Virtualized Bronchus Endoscope Systems. In *Visualization in Biomedical Computing*, Lecture Notes in Computer Science - 1131, pages 33–42. Springer-Verlag, September 1996.

[16] R. Shahidi, V. Argiro, S. Napel, L. Gray, H.P. McAdams, G. D. Rubin, C. F. Beaulieu, R.B. Jeffrey, and A. Johnson. Assessment of Several Virtual Endoscopy Techniques Using Computed Tomography and Perspective Volume Rendering. In *Visualization in Biomedical Computing*, Lecture Notes in Computer Science - 1131, pages 521–528. Springer-Verlag, September 1996.

[17] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, and K. Katada. Automated Extraction and Visualization of Bronchus from 3D CT Images of Lung. In *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science - 905, pages 542 – 548. Springer-Verlag, April 1995.

[18] K. Darabi, K.D.M. Resch, J. Weinert, U. Jendrysiak, and A. Perneczky. Real and Simulated Endoscopy of Neurosurgical Approaches in an Anatomical Model. In *CVRMed-MRCAS '97*, Lecture Notes in Computer Science - 1205, pages 323 – 326. Springer-Verlag, March 1997.

[19] T. Todd Elvins. A Survey of Algorithms for Volume Visualization. *Computer Graphics*, 26(3):194–201, August 1992.

[20] Maria Magnusson, Reiner Lenz, and Per-Erik Danielsson. Evaluation of Methods for Shaded Surface Display of CT Volumes. *Computerized Medical Imaging and Graphics*, 15(4):247–256, 1991.

[21] Marc Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[22] Blinn J. F. Models of Light Reflection for Computer Synthesized Pictures. *SIGGRAPH*, pages 192–198, 1977.

[23] J.T. Kajiya. The Rendering Equation. *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, 20:143–150, August 1986.

[24] Markus Kukuk and Bernhard Geiger. A Real-Time Deformable Model for Flexible Instruments Inserted Into Tubular Structures. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 2489 of *Lecture Notes in Compter Science – LNCS*, pages 331–338. Springer-Verlag, 2002.

[25] Markus Kukuk and Bernhard Geiger and Heinrich Müller. TBNA-protocols - Guiding TransBronchial Needle Aspirations Without a Computer in the Operating Room. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 2208 of *Lecture Notes in Compter Science – LNCS*, pages 997–1006. Springer-Verlag, 2001.

[26] Markus Kukuk. Modeling the Internal and External Constraints of a Flexible Endoscope for Calculating its Workspace: Application in Transbronchial Needle Aspiration Guidance. In *Medical Imaging 2002: Visualization, Image-Guided Procedures, and Display*, volume 4681 of *Proceedings of SPIE*, pages 539–550, 2002.

[27] Markus Kukuk and Bernhard Geiger. Registration of Real and Virtual Endoscopy - A Model and Image Based Approach. In James D. Westwood et al., editors, *Medicine Meets Virtual Reality – MMVR*, volume 70 of *Studies in Health Technologie and Informatics*, pages 168–174. IOS Press, 2000.

[28] Siemens Corporate Research, Inc. and Markus Kukuk. Modeling a Flexible Tube. Patent Application. Docket No.: 2001E13595US (8706-527).

[29] Siemens Corporate Research, Inc. and Markus Kukuk. System and Method for Guiding Flexible Instrument Procedures. Patent Application. Docket No.: 2001P07456US (8706-558).

[30] Siemens Corporate Research, Inc. and Markus Kukuk. Optimal K-Needle Placement Strategy Considering an Approximate Initial Needle Position. Patent Application. Docket No.: 2002E0404US (8706-581).

[31] Udaya B. S. Prakash, editor. *Bronchoscopy.* Raven Press - New York, 1993. ISBN:0-7817-0095-7.

[32] Theo van Walsum et al. Deformable B-splines for Catheter Simulation. In *CARS'99*, 1999.

[33] J. Anderson, W. Brody, et al. daVinci - A Vascular Catheterization and Interventional Radiology-Based Training and Patient Pretreatment Planning Simulator. In *Proc. of Society of Cardiovascular and Interventional Radiology (SCVIR)*, March 1996.

[34] C. K. Chui, H. T. Nguyen, Y. P. Wang, R. Mullick, R. Raghavan, and J. A. Anderson. Potential Field of Vascular Anatomy for Realtime Computation of Catheter Navigation. In *First Visible Human Conference*, Bethesda, MD, USA, October 1996.

[35] Koji Ikuta et al. Virtual Endoscope System with Force Sensation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 1496 of *Lecture Notes in Compter Science – LNCS*. Springer-Verlag, 1998.

[36] Koji Ikuta et al. Portable Virtual Endoscope System with Force and Visual Display for Insertion Training. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 1935 of *Lecture Notes in Compter Science – LNCS*, pages 907 – 920. Springer-Verlag, 2000.

[37] John J. Craig. *Introduction to Robotics - Mechanics and Control.* Addison-Wesley, 1986. ISBN: 0-201-10326-5.

[38] Jeffrey Michael Wendlandt. *Control and Simulation of Multibody Systems.* PhD thesis, University of California at Berkeley, 1997.

[39] Brian Vincent Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems.* PhD thesis, University of California at Berkeley, 1996.

[40] Ahmed A. Shabana. *Dynamics of Multibody Systems.* Wiley-Interscience, 1989. ISBN: 0-471-61494-7.

[41] R. Featherstone. The Calculation of Robot Dynamics Using Articulated-Body Inertias. *International Journal of Robotics Research*, 2(1):13–30, 1983.

[42] E. U. Condon and Hugh Odishaw, editors. *Handbook of Physics.* McGraw-Hill, second edition, 1967.

[43] J.-D. Boissonnat. Shape Reconstruction From Planar Cross-Sections. *Comput. Vision Graph. Image Process.*, 44:1–29, 1988.

[44] J-D. Boissonnat and B. Geiger. Three Dimensional Reconstruction of Complex Shapes Based on the Delaunay Triangulation. In R. S. Acharya and D. B. Goldgof, editors, *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 964–975. SPIE, 1993.

[45] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)* , 2001.

[46] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer-Verlag, 1999.

[47] K. Levenberg. A Method for the Solution of Certain Nonlinear Problems in Least Squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[48] D. W. Marquardt. An Algorithm for Least-squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.*, 2:431–441, 1963.

[49] D. Shen et al. A Statistical Atlas of Prostate Cancer for Optimal Biopsy. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 2208 of *Lecture Notes in Compter Science – LNCS*, pages 416–424. Springer-Verlag, 2001.

[50] J. Zeng et al. Distribution of Prostate Cancer for Optimized Biopsy Protocols. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 1935 of *Lecture Notes in Compter Science – LNCS*, pages 287–296. Springer-Verlag, 2000.

[51] M. L. Fisher and P. Kedia. Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristics. In *Management Science-36*, pages 674–688, 1990.

[52] J. E. Beasley. A Genetic Algorithm for the Set Covering Problem. In *European Journal of Operational Research*, pages 392–404, 1994.

[53] D. S. Hochbaum and A. Pathria. Analysis of the Greedy Approach in Problems of Maximum k-Coverage. In *Naval Research - 45*, pages 615–627, 1998.

[54] E. M. Harrow et al. The Utility of Transbronchial Needle Aspiration in the Staging of Bronchogenic Carcinoma. In *American Journal of Respiratory and Critical Care Medicine*, volume 161, pages 601–607, 2000.

[55] Ko Pen Wang et al. Flexible Transbronchial Needle Aspiration for Staging of Bronchogenic Carcinoma. In *CHEST*, volume 84,5, pages 571 – 576, 1983.

[56] Stephen B. Solomon et al. Real-time Bronchoscope Tip Localization Enables Three-dimensional CT Image Guidance for Transbronchial Needle Aspiration in Swine. In *CHEST*, volume 114/5, pages 1405–1410, 1998.

[57] E. F. Haponik et al. Education and Experience Improve Transbronchial Needle Aspiration Performance. In *American Journal of Respiratory and Critical Care Medicine*, volume 151, pages 1998–2002, 1995.

[58] John J. Shannon et al. Endobronchial Ultrasound-Guided Needle Aspiration of Mediastinal Adenopathy. In *American Journal of Respiratory and Critical Care Medicine*, volume 153, 1996.

[59] P. C. Gay and W. M. Brutinel. Transbronchial Needle Apiration in the Practice of Bronchoscopy. In *Mayo Clinic Proceedings 64*, pages 158–162, 1989.

[60] C. S. White et al. CT-Assisted Transbronchial Needle Aspiration: Usefulness of CT Fluoroscopy. In *American Journal of Roentgenology 169*, 1997.

[61] I. Bricault, G. Ferretti, and P. Cinquin. Computer-Assisted Bronchoscopy: Aims and Research Perspectives. In *Medical Robotics and Comuter Assisted Surgery - MRCAS '95*, pages 124–131, Nov. 1995.

[62] I. Bricault. A Fast Morphology-Based Registration. In *CVRMed-MRCAS '97*, Lecture Notes in Computer Science - 1205, pages 417–426. Springer-Verlag, March 1997.

[63] I. Bricault, G. Ferretti, and P. Cinquin. Multi-level Strategy for Computer-Assisted Transbronchial Biopsy. In *Medical Image Computing and Computer-Assisted Intervention- MICCAI*, Lecture Notes in Computer Science - 1496, pages 261–268. Springer-Verlag, Oct. 1998.

[64] Anthony J. Sherbondy et al. Virtual Bronchoscopic Approach for Combining 3D CT and Endoscopic Video. In *Proc. of SPIE Medical Imaging*, volume 3978, 2000.

[65] Kensaku Mori et al. A Method for Tracking the Camera Motion of Real Endoscope by Epipolar Geometry Analysis and Virtual Endoscopy System. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 2208 of *Lecture Notes in Compter Science – LNCS*, pages 1–8. Springer-Verlag, 2001.

[66] Stephen B. Solomon et al. Three-dimensional CT-Guided Bronchoscopy With a Real-Time Electromagnetic Position Sensor. In *CHEST*, volume 118/6, pages 1783–1787, 2000.

[67] A. C. Mehta et al. The High-Price of Bronchoscopy. In *Chest 98*, pages 448–454, 1990.

[68] Susan Alyson Wood. *Analysis of Three Dimensional Lung Structure During Inflation Using Computed Tomography.* Umi dissertaion services, Johns Hopkins University, 300 North Zeeb Road, P.O. Box 1346, Ann Arbor, Michigan 48106-1346, 1994. phone: 1-800 5210600.

[69] R.D. Swift, A.P. Kiraly, A.J. Sherbondy, A.L. Austin, E.A. Hoffman, G. McLennan, and W.E. Higgins. Automatic Axes-Generation for Virtual Bronchoscopic Assessment of Major Airway Obstructions. *Computerized Medical Imaging and Graphics*, 26(2):103–118, March-April 2002.

[70] Kensaku Mori et al. Automated Labeling of Bronchial Branches in Virtual Bronchoscopy System. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, volume 1496 of *Lecture Notes in Compter Science – LNCS*, pages 870–878. Springer-Verlag, 1998.