

Ein System zur Überwachung und Steuerung von realen Endoskopiephantomen

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Matthias Vietz
aus Tönisvorst

Düsseldorf, Juli 2014

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. V. Aurich

Korreferent: Prof. Dr. E. Wanke

Tag der mündlichen Prüfung: 09. Juli 2014

Vorwort

In dieser Dissertation wird häufig von den Personen die Rede sein, die ein Training am Phantom betreuen und von denen, die trainieren. Um nicht jedes Mal sprachliche Verrenkungen betreiben zu müssen, werde ich daher bei der Nennung dieser Personen stets das Generische Maskulinum wählen, also z.B. *der Trainierende* oder *der Tutor*, es sei denn, es handelt sich explizit um eine weibliche Person. Dies stellt selbstverständlich keine Diskriminierung des weiblichen Geschlechts dar, sondern dient der Klarheit der Sprache.

Zusammenfassung

Für die Aus- und Weiterbildung von Ärzten existiert eine Vielzahl von sog. *Trainingsphantomen*. Darunter versteht man eine künstliche Nachbildung eines oder mehrerer Organe des Menschen, anhand der Eingriffe trainiert werden können. Besonders realistische Vertreter stellen die Endoskopiephantome unserer Forschungspartner aus der Arbeitsgruppe um Prof. Dr. Grund vom Universitätsklinikum Tübingen dar. Diese Phantome sind modular konzipiert; es existieren bereits Module für den Oberen und den Unteren Gastrointestinaltrakt sowie für das Gallengangsystem. Weitere Arten von Modulen sind geplant bzw. in aktiver Entwicklung, beispielsweise ein Phantom des Bronchiensystems. Anhand dieser Phantome lassen sich verschiedene diagnostische und therapeutische endoskopische Eingriffe unter sehr realistischen Bedingungen trainieren.

In dieser Arbeit stellen wir ein kombiniertes System aus Hard- und Software vor, das wir „Simtux“ nennen. Es bildet eine einheitliche Plattform, mit der eine effektive Überwachung und Steuerung der verschiedenen Arten von Tübinger Phantomen mit Hilfe eines Computers möglich ist. Ein Phantom wird dazu um spezielle Sensorikhardware erweitert, deren Daten an einen PC gesendet werden. Auf diesem wird die Software ausgeführt, die den aktuellen Fortschritt des trainierten Eingriffs erfasst und auswertet. Dadurch können dem Trainierenden während des Trainings wichtige Informationen über den Trainingsverlauf geboten werden.

Die Software wurde im Rahmen dieser Dissertation entwickelt. Während der Konzeption und Entwicklung haben wir besonderen Wert auf eine strenge Modularisierung unseres Systems gelegt. Dies stellt eine gute Erweiterbarkeit für zukünftige Arten von Phantomen sicher, für die einfach zusätzliche Module entwickelt und in das Gesamtsystem integriert werden können. Bisher haben wir zwei Module für *Simtux* entwickelt:

Das erste Modul, „GastroInt-Sim“, für Endoskopien am Oberen und Unteren Gastrointestinaltrakt bietet einem Trainierenden in erster Linie Informationen über den aktuellen Diagnosefortschritt während einer Gastro- oder Koloskopie am Phantom. Dazu werden in einer grafischen Übersicht die Bereiche des Phantoms markiert, die aktuell unter endoskopischer Beobachtung stehen. Diese Informationen sind wichtig, da es für eine erfolgreiche Gastro- oder Koloskopie unerlässlich ist, dass jeder Bereich von Magen oder Kolon gründlich untersucht wurde. Darüber hinaus können dem Trainierenden

gezielte Aufgaben zur Navigation innerhalb des Phantoms gestellt werden, sodass er seine Fähigkeiten im Umgang mit der komplexen Steuermechanik des flexiblen Endoskops verfeinern kann.

Das zweite Modul, „X-Sim“, dient der Überwachung des sog. „Tübinger Biliphanten“, eines Phantoms zum Erlernen und Trainieren der *Endoskopisch Retrograden Cholangiopankreatikographie (ERCP)*. Dabei handelt es sich um einen schwierigen und riskanten Eingriff mittels endoskopischer Instrumente am menschlichen Gallengangsystem. Eine reale ERCP muss unter permanenter Röntgendurchleuchtung des Patienten durchgeführt werden. Auch eine auf herkömmliche Weise, d.h. ohne die Unterstützung durch das *Sim-tux*-System, am Biliphanten durchgeführte ERCP musste bisher unter Röntgenkontrolle stattfinden, weil es sonst keine Möglichkeit gab, den komplexen Eingriff nachzuverfolgen. Dies bedeutete eine erhebliche Strahlenbelastung aller Beteiligten während des Trainings.

Als entscheidende Neuheit des vorgestellten Systems haben wir daher eine *realitätsnahe virtuelle Röntgensimulation* entwickelt, die die Verwendung eines echten Röntgengeräts auch für komplexe Eingriffe am Gallengangsystem unnötig macht und damit einen wichtigen Beitrag zum Strahlenschutz leistet.

Abstract

Education and training of physicians is often performed using so-called *training phantoms*. These are artificial replica of one or more human organs that can be used to train medical procedures. A variety of such systems is available on the market. The endoscopy phantoms built by our research partners, the research group of Prof. Dr. Grund at *Universitätsklinikum Tübingen*, are especially realistic examples. The basic concept of these phantoms is their modular structure. Modules for the upper and lower gastrointestinal tract and the biliary system have already been created. Other types of modules are either planned or in active development, such as a phantom of the bronchial system. Using these phantoms, different diagnostic and therapeutic endoscopic procedures can be trained hands-on, under very realistic conditions.

In this work we present a combined system of hardware and software which we call „Simtux“. It allows an effective monitoring and controlling of the *Tübingen phantoms* using specific sensor hardware connected to a computer. The phantoms are extended by the sensors which send data to the computer running the detection and evaluation software that we created in the context of this dissertation. This provides the trainee with important information about the training course.

Simtux forms a unified platform for monitoring the various types of *Tübingen phantoms*. Special emphasis was put on strict modularization of all parts of our system. This allows good extensibility for future types of phantoms as additional software modules can easily be integrated into the overall system. So far, we developed two modules for *Simtux*:

„GastroInt-Sim“ is a module for monitoring endoscopies in the upper and lower gastrointestinal tract. It provides information about the current diagnostic progress during a gastro- or colonoscopy in a phantom. This information is important because it is essential to thoroughly examine every region of the stomach or the colon in order to perform a successful gastro- or colonoscopy. In addition, specific navigational tasks can be assigned to a trainee in order to refine his skills in dealing with the complex control mechanism of the flexible endoscope.

„X-Sim“ is the second module we developed. It serves monitoring the so-called „Tübingen Biliphant“ which is a phantom for learning and training the *Endoscopic retrograde cholangiopancreatography (ERCP)*. An ERCP is a difficult and risky procedure on the human

biliary system using endoscopic instruments. A real ERCP has to be performed under continuous fluoroscopy of the patient. Even a conventional ERCP on the *Biliphant*, i.e. without the support of our *Simtux* system, had to be performed under fluoroscopy because there was no other way of tracking the complex operation. This meant a significant radiation exposure of all participants during a training session.

Therefore, we developed a *realistic virtual X-ray simulation* as a key novelty of the proposed system. This makes the usage of a real X-ray generator completely unnecessary even for complex operations on the biliary system, thus making an important contribution to radiation protection.

Danksagung

Ohne die Unterstützung vieler lieber Menschen hätte ich diese Arbeit nicht erfolgreich fertigstellen können. Ich möchte daher einige Dankesworte an diese Personen richten. Mein besonderer Dank geht an meinen Doktorvater Prof. Dr. Aurich. Die vielen Gespräche und Diskussionen über den Inhalt meiner Dissertation und mein Forschungsgebiet sowie über viele andere Themen waren sehr anregend und inspirierend. Ich danke ihm auch für die sehr angenehme Zusammenarbeit in den vergangenen Jahren und dafür, dass er mir immer große Freiheiten und Spielräume ermöglicht hat.

Während meines Studiums und zu Beginn meiner Doktorandenzeit war mir Dr. Andreas Beck eine jederzeit ansprechbare Hilfe. Von seinem enormen Wissen und seiner Erfahrung habe ich so viel es ging aufgesogen und zehre bis heute davon.

Als nächstes möchte ich meinen Eltern danken. Sie haben mich jederzeit gefördert und unterstützt, gaben mir Wurzeln und Flügel. Ohne sie wäre ich niemals dorthin gekommen, wo ich heute stehe. Vielen Dank!

Weiterhin geht mein Dank an Matthias, Felix, Nurdagül und Henning, die mir beim Korrekturlesen eine große Hilfe waren und durch kritisches Hinterfragen viele Aspekte dieser Arbeit positiv beeinflusst haben. In diese Reihe gehört selbstverständlich auch mein Vater, der die vollständige Arbeit mehrfach korrekturgelesen hat.

Zu guter Letzt, aber ganz sicher nicht an letzter Stelle, danke ich meiner Verlobten Duygu, die meine komplette Dissertation Satz für Satz mehrfach mit mir durchgegangen ist, obwohl sie selbst gerade in ihrer Abschlussphase steckt. Über so viele Jahre hinweg hat sie mich liebevoll unterstützt und viel Verständnis dafür aufgebracht, wenn mir die Dinge manchmal über den Kopf gewachsen sind. Sie hat mich jedes Mal wieder auf den Boden der Tatsachen zurückgeholt und, wenn ich mal nicht weiter wusste, mich dazu ermutigt, einfach weiterzumachen. Für all das danke ich dir von ganzem Herzen und freue mich darauf, was die Zukunft noch alles für uns bringen wird!

Inhaltsverzeichnis

1. Einleitung	1
1.1. Fortschritt durch diese Arbeit	5
1.2. Aufbau der Arbeit	6
2. Architektur des Gesamtsystems	7
2.1. Anforderungen an das System	7
2.1.1. Modularisierung des Systems	7
2.1.2. Interaktion mit dem System	7
2.2. Überblick über die Hardwareimplementierung	8
2.2.1. Die Sensoren	9
2.2.2. Sensorbus und Busmaster	10
2.3. Grundlagen der Softwareimplementierung	11
2.3.1. Wahl der Programmiersprache	11
2.3.2. Der Aufbau von Simtux	12
2.4. Programmkomponenten von Simtux	13
2.4.1. Sensorserver	14
2.4.2. Simtux-Kern	15
2.4.3. Module	15
3. Oberer und Unterer Gastrointestinaltrakt	17
3.1. Endoskopien am Oberen bzw. Unteren Gastrointestinaltrakt	18
3.1.1. Gastroskopien	18
3.1.2. Koloskopien	19
3.2. Anforderungen an das Softwaremodul GastroInt-Sim	20
3.3. Existierende Navigationssysteme	20
3.4. Überlegungen zur Umsetzung	21
3.4.1. Erkennung der beobachteten Oberflächenbereiche	22
3.4.2. Modifikationen am Tübinger Phantom	23
3.4.3. Umgang mit der spärlichen Sensorverteilung	24
3.4.4. Mögliche Rekonstruktionsfehler	30
3.4.5. Definition des Abstandsbegriffs	31
3.4.6. Theoretische Überlegungen zur exakten Lokalisierung	35

3.4.7.	Diskussion des Ansatzes	37
3.5.	Implementierung des Softwaremoduls	38
3.5.1.	3D-Bereich	38
3.5.2.	Skripte	40
3.6.	Weitere Arten von Sensoren	41
3.6.1.	Lagesensoren	41
3.6.2.	Drucksensoren	42
4.	Gallengangssysteme	43
4.1.	Die ERCP	43
4.1.1.	Tübinger Biliphant	45
4.2.	Anforderungen an das Softwaremodul X-Sim	46
4.3.	Simulation eines Röntgenbildes	49
4.3.1.	Physikalische Betrachtungen	50
4.3.2.	Berechnung eines Röntgenbildes	52
4.3.3.	Umsetzung im Rechner	53
4.4.	Konzeption der Sensorik für die Anfängerphantome	56
4.4.1.	Kalibrierung der Sensoren	58
4.4.2.	Probleme in der Praxis	60
4.4.3.	Theoretische Betrachtung des Fehlers bei der Längenmessung	61
4.4.4.	Messungen zur Kapazitiven Sensorik	64
4.4.5.	Fazit der Messung	68
4.5.	Darstellung des Instruments in den Anfängerphantomen	70
4.5.1.	Definition der Instrumente mittels Splines	70
4.5.2.	Zuordnung der Sensoren zu Kurvenabschnitten	73
4.5.3.	Auswerten und Zeichnen der Splines	76
4.5.4.	Splinebäume	77
4.5.5.	Auswahl des richtigen Pfades	80
4.5.6.	Verschiebung von Knotenpunkten in Abhängigkeit von Sensorwerten	83
4.6.	Optische Erfassung von Instrumenten	85
4.6.1.	Modifikationen am Aufbau	86
4.6.2.	Die Erkennung der eingeführten Instrumente	88
4.6.3.	Korrektur bei elastischen Deformationen	92
4.6.4.	Darstellung des Instruments	95
4.7.	Implementierung des Softwaremoduls	99
4.7.1.	Fenster für das Röntgenbild	99
4.7.2.	Bedienung	100
4.8.	Erkennung der Begradigung	103
4.9.	Herstellung der transparenten Gallengänge	104
4.9.1.	PVC-Isolierschläuche	105

4.9.2.	Glas	105
4.9.3.	Kunstharz	106
5.	Fazit und Ausblick	109
5.1.	Bronchophant	110
5.1.1.	Ideen zur Entwicklung eines Moduls für Simtux	110
5.1.2.	Implementierungsideen für das Softwaremodul Broncho-Sim	111
A.	Anhänge	113
A.1.	Das Programmframework Taraxacum	113
A.1.1.	OpenGL	113
A.1.2.	Grafisches Subsystem	115
A.1.3.	Skripting-Schnittstelle	116
A.1.4.	Szenenbeschreibung	117
A.2.	Dateiformate	121
A.2.1.	Szenenbeschreibungsdateiformat	121
A.2.2.	Volumendateiformat	123
A.3.	Algorithmen und Implementierungen	123
A.3.1.	Berechnung von Geodäten	123

1. Einleitung

Wie in allen Berufen ist auch und besonders für Mediziner eine gute Ausbildung äußerst wichtig. Immerhin wird dem behandelnden Arzt die Verantwortung für Leben und Gesundheit seines Patienten übergeben und es wäre daher geradezu fahrlässig, ihn diese Verantwortung ohne die entsprechende Vorbereitung übernehmen zu lassen. Die Ausbildung lässt sich dabei grob in zwei Bereiche einteilen: Zunächst gibt es die theoretische Ausbildung. Das Wissen wird in der Regel durch Fachbücher, Fachzeitschriften, Tagungen und natürlich im Laufe des Medizinstudiums in Form von Vorlesungen vermittelt. Auf der anderen Seite steht die praktische Ausbildung. Gelernt wurde und wird häufig direkt am Patienten. Dabei steht dem Auszubildenden im Idealfall ein erfahrener Experte zur Seite, der ihm Anleitung gibt, seine Handlungen beaufsichtigt und im Notfall eingreifen kann. Wie jedoch Prof. Dr. K.-E. Grund einmal sinngemäß sagte: „Wenn Sie bei einem Barbier zum Rasieren sind und der Lehrling Ihnen das Rasiermesser an die Kehle hält, beruhigt es Sie da wirklich, wenn der Meister daneben steht?“. Gemeint ist dabei das zum Teil erhebliche Risiko, wenn ein unerfahrener Chirurg einen komplizierten Eingriff vornimmt. Manchmal können selbst kleine Unaufmerksamkeiten und Fehler katastrophale Folgen für den Patienten haben, die auch durch das schnelle Eingreifen eines Experten nicht immer verhindert werden können. Aufgrund der potentiell großen Risiken ist daher das Lernen am Patienten in vielen Fällen ethisch fragwürdig. Um diese Risiken für Patienten zu minimieren, gibt es für viele chirurgische Anwendungsfälle das Training an sogenannten *Phantomen*, welches gewissermaßen einen Mittelweg zwischen theoretischer und praktischer Ausbildung darstellt.

Ganz allgemein bezeichnet der Begriff *Phantom* eine künstliche Nachbildung eines oder mehrerer Organe des Menschen, anhand derer Eingriffe trainiert werden können. Daneben gibt es noch andere Arten von Phantomen, die eher der Validierung von bestimmten Verfahren dienen, wie z.B. CT-Phantome oder ähnliche. Um solche Phantome soll es in dieser Arbeit jedoch nicht gehen; stattdessen konzentrieren wir uns ganz auf Trainingsphantome, an denen Chirurgen diagnostische oder therapeutische Eingriffe trainieren können.

Die Arbeitsgruppe von Prof. Dr. Grund am Institut für experimentelle endoskopische Chirurgie der Universität Tübingen arbeitet an der Herstellung solcher Phantome speziell für das Training *endoskopischer Verfahren* [GID⁺12, VzGA13, GIS⁺12a, GSFa, GSFb].



Abb. 1.1.: Flexibles Endoskop. Quelle: Nutzer „Kalumet“, Wikimedia Commons, entnommen am 13.05.2014

Unter *endoskopischen Verfahren* versteht man diagnostische oder therapeutische Eingriffe, die mit Hilfe eines in der Regel flexiblen *Endoskops* vorgenommen werden. Ein flexibles Endoskop ist ein Instrument, das aus einem Bedienteil, einem langen, flexiblen, schlauchförmigen Schaft und an dessen Ende einer Optik und einer starken Lichtquelle besteht. Dieses Schaftende lässt sich mit Hilfe des Bedienteils in den Richtungen hoch/-runter bzw. links/rechts neigen. Zumeist kann die Neigung in drei der vier Richtungen bis 90° geschehen, während sie in der vierten Richtung sogar bis zu 180° betragen kann. Man spricht dann von einer „Invertierung“ des Endoskops. Der Schaft enthält in der Regel einen oder mehrere sog. *Instrumentierkanäle*, über den Werkzeuge vorgeschoben werden können, sowie Luft- und Wasserleitungen. Abb. 1.1 zeigt ein solches Endoskop.

Mit einem Endoskop ist es möglich, minimalinvasiv durch natürliche Körperöffnungen in das Innere des Körpers vorzudringen und dort mit Hilfe der Optik Diagnosen zu erstellen oder mit Instrumenten, die über den Instrumentierkanal des Endoskops vorgeschoben werden können, therapeutische Eingriffe vorzunehmen. Beispiele für diagnostische Eingriffe sind *Koloskopien*, umgangssprachlich als „Darmspiegelungen“ bekannt, oder *Gastroskopien*, die auch als „Magenspiegelungen“ bezeichnet werden. Ein Beispiel für einen therapeutischen endoskopischen Eingriff stellt die sog. *ERCP* (Endoskopische Retrograde Cholangiopankreatographie) dar, mit der Eingriffe am System der Gallengänge vorgenommen werden können.

Die Tübinger Phantome sind modular aufgebaut, wobei stets dasselbe Chassis zum Einsatz kommt, in das je nach Anforderung unterschiedliche Module eingesetzt werden können. Für alle Module, die Endoskopien nachstellen, welche oral durchgeführt werden,



Abb. 1.2.: Typisches Chassis eines Tübinger Phantoms mit montiertem Kopf.

kann das Chassis mit einem nachgebildeten Kopf ausgestattet werden, damit bereits das Einführen des Endoskops durch Speise- oder Luftröhre so realistisch wie möglich nachgestellt werden kann. Abb. 1.2 zeigt ein solches (geschlossenes) Chassis mit Kopf.

Es existieren bereits Module für den oberen und den unteren Gastrointestinaltrakt (*Gastro-* bzw. *Kolophant*), also das menschliche Verdauungssystem und für das Gallengangsystem (*Biliphant*). Weitere Arten von Phantomen sind geplant, bzw. deren Entwicklung ist zum Zeitpunkt des Drucks dieser Arbeit bereits fortgeschritten, beispielsweise ein Phantom des Bronchiensystems (*Bronchophant*). Diese Phantome sind allesamt in den wesentlichen Eigenschaften wie Größe und Gestalt, der Farbe und vor allem den physikalischen Eigenschaften, z.B. beim chirurgischen Schneiden, der menschlichen Anatomie so weit wie möglich nachempfunden und bedienen sich vollständig künstlicher Materialien.

Insbesondere wird eine vollkommene „Schweinfreiheit“ garantiert. Dieser Ausdruck leitet sich davon ab, dass zu Übungszwecken für Interventionen am Gastrointestinaltrakt häufig auf Schweine zurückgegriffen wird, da ihre Anatomie der eines Menschen relativ ähnlich ist [Noa95]. Es kommen sowohl Interventionen am lebenden Schwein als auch an *ex-vivo*-Phantomen aus frisch geschlachteten Schweinen zum Einsatz. Beispielsweise wird der Erlanger *Endo-Trainer* [NME⁺00] aus Innereien von Schweinen hergestellt, die für die Trainingseingriffe präpariert und zusammengenäht werden. Dies bringt jedoch eine ganze Reihe von Nachteilen mit sich. Die wichtigsten sind:

- Der logistische Aufwand: Die einzige verlässliche Quelle für Schweineinnereien ist ein Metzger, d.h. es muss ein Metzger vor Ort sein, der diese verfügbar hat. Das Phantom muss vor dem Training zusammengenäht werden und ist dann nur für wenige Stunden benutzbar, da das Fleisch nach einer gewissen Zeit zu verwesen beginnt, was sehr unangenehme Gerüche mit sich bringt.
- Trotz der Ähnlichkeiten weicht die Anatomie eines Schweins in manchen Punkten stark von der menschlichen ab. So ist zum Beispiel das Kolon eines Schweins glatt, während ein menschliches Kolon gefurcht ist. Weiterhin münden Pankreas- und Gallengang beim Menschen in einer gemeinsamen Auslassöffnung, der sog. *Papille*, in das Duodenum, während sie beim Schwein in zwei getrennten Auslassöffnungen münden [Noa95].
- Aus hygienischen und ethischen Gründen darf ein Endoskop, das einmal in einem tierischen Phantom benutzt wurde, nie wieder für einen Eingriff am Menschen verwendet werden. Die dadurch entstehenden Kosten sind angesichts des hohen Preises für ein Endoskop im mindestens vierstelligen Bereich nicht zu vernachlässigen.
- Vor allem in den jüdisch oder islamisch geprägten Ländern des Nahen und Mittleren Ostens gibt es in der Regel überhaupt keine Möglichkeit, an frisch geschlachtete Schweine zu gelangen; und selbst wenn welche verfügbar wären, ist die Akzeptanz eines Phantoms aus Schweineinnereien in diesen Ländern naheliegenderweise sehr problematisch.

Alle diese Negativpunkte treten bei den Tübinger Phantomen nicht auf. Damit kann ein breites Spektrum an diagnostischen und therapeutischen Eingriffen sehr realitätsnah und beliebig oft wiederholbar trainiert werden. Besonders hervorzuheben ist, dass keinerlei spezielle Ausrüstung vonnöten ist, sondern normale Endoskope, wie sie im alltäglichen Einsatz sind, verwendet werden können. Nach einer normalen Reinigung stehen diese wieder für den Einsatz am Menschen zur Verfügung.

Fairerweise muss erwähnt werden, dass Phantome aus natürlichem Gewebe auch Vorteile gegenüber künstlichem Gewebe bieten, da z.B. Operationstechniken noch realistischer trainiert werden können. Für die Endoskopie überwiegen jedoch die Vorteile künstlichen Gewebes eindeutig gegenüber den Nachteilen.

Neben solchen sog. „Hands-on-Phantomen“, also Phantomen, die dem Trainierenden unter realistischen Bedingungen das Training mit echten Instrumenten und damit echter Haptik ermöglichen, gibt es noch die Kategorie der reinen *Virtual-Reality-Simulatoren*. Das sind Simulatoren, bei denen die Eingriffe vollständig durch eine Kombination aus spezieller Hard- und Software simuliert werden. Beispielsweise gibt es von der Firma Simbionix den *GI Mentor* [Sim14] zur Simulation des Gastrointestinaltrakts oder von der

Firma CAE Healthcare den *CAE EndoVR* [Hea14]. Dabei hat der Benutzer im Wesentlichen eine *Black Box* vor sich, aus der die Bedienelemente eines Endoskops herausragen. Damit lässt sich das „Endoskop“ ähnlich wie sein reales Pendant bedienen; sämtliche Details des Eingriffs werden jedoch im Inneren der *Black Box* durch verschiedene Mechanismen simuliert, z.B. werden Kräfte auf das „Endoskop“ ausgeübt, wodurch sich die Haptik ähnlich anfühlen soll wie bei einem realen endoskopischen Eingriff am Menschen. Auf einem oder mehreren Monitoren wird dem Benutzer ein computergeneriertes Bild präsentiert, welches dem Blick durch das virtuelle Endoskop entspricht.

Ich hatte bei einem Besuch bei Prof. Dr. Häussinger am Universitätsklinikum Düsseldorf Gelegenheit, eine kurze Vorführung des *GI Mentors* zu erhalten und ihn selbst auszuprobieren. Als Laie schien mir die Simulation zufriedenstellend, jedoch sind solche virtuellen Simulatoren nach Aussage von Prof. Dr. Grund nur bedingt brauchbar, um damit wirkliche therapeutische Eingriffe zu trainieren. [TLD14] gibt eine Übersicht über Studien, die sich mit der Wirksamkeit des Trainings mit virtuellen Simulatoren befassen. Diese kommen zu dem Schluss, dass zumindest die Navigation mit dem Endoskop durchaus durch ein Training mit virtuellen Simulatoren verbessert werden kann. Es wird jedoch vor allem der hohe Preis von je nach Ausbaustufe beim *GI Mentor* 64.500 \$ - 114.000 \$ und beim *CAE EndoVR* mindestens 46.750 \$ als Kriterium dafür genannt, dass virtuelle Simulatoren keine flächendeckende Verbreitung finden.

Trainingskurse mit den Tübinger Phantomen

Die von Prof. Dr. Grund entwickelten Phantome werden bereits seit 1999 erfolgreich in Trainingskursen eingesetzt. Typischerweise wird ein solcher Kurs von einem auf dem entsprechenden Gebiet erfahrenen Tutor geleitet, der die Trainingsteilnehmer einweist, anleitet und betreut. Die Erfahrungsstufe der Teilnehmer kann je nach Zielsetzung des Kurses von absoluten Anfängern bis zu bereits fortgeschrittenen Ärzten reichen. Als Beispiel für einen Fortgeschrittenen-Kurs seien die Kurse mit dem *Biliphanten* genannt, in welchen die Teilnehmer häufig bereits einen soliden endoskopischen Hintergrund aufweisen, jedoch bei dem komplizierten Eingriff der *ERCP* noch Trainingsbedarf haben oder vielleicht auch spezielle oder neue Techniken erlernen wollen.

1.1. Fortschritt durch diese Arbeit

Während ein Simulator wie der *GI Mentor*, der vollständig auf virtueller Realität basiert, nach Expertenmeinung für das Training von therapeutischen Eingriffen unzureichend ist, würden die Tübinger Phantome durchaus stark von einer Computerunterstützung

profitieren. Beispielsweise sind Protokollierfunktionen zur Überwachung eines Trainingseingriffs nützlich, um im Nachhinein oder schon während des Trainings die Tätigkeit des Trainierenden analysieren und gegebenenfalls auswerten zu können. Eine virtuelle Darstellung auf einem Monitor kann den Benutzer unterstützen, während er mit realen Instrumenten *hands-on* den Eingriff am Phantom vornimmt oder sogar, wie im Fall der Röntgensimulation beim *Biliphanten*, welche in Kapitel 4 beschrieben wird, einen ganz entscheidenden Mehrwert bedeuten.

Im Rahmen dieser Dissertation entstand in enger Forschungspartnerschaft mit der Gruppe von Prof. Dr. Grund eine Kombination aus Hard- und Software, die mittels speziell entwickelter Sensoren eine Überwachung und Steuerung sowie eine grafische Darstellung der Phantome am Rechner ermöglicht. Für die Sensorikhardware zeichnet Prof. Dr. Aurich verantwortlich. Die Protokoll-, Steuer- und Darstellungssoftware wurde von mir im Rahmen dieser Dissertation entwickelt. Entstanden ist ein modulares System zur Steuerung, Auswertung und Überwachung eines Phantoms, welches wir „Simtux“ getauft haben. Die Sensorikhardware ist dabei fest in das Tübinger Phantom eingebaut. Die Verbindung zwischen der Hardware des Phantoms und dem Rechner, auf dem die Software läuft, wird ausschließlich über ein USB-Kabel hergestellt. Dadurch wird ein minimaler Aufwand für die Inbetriebnahme ermöglicht.

1.2. Aufbau der Arbeit

Nach diesem einleitenden Kapitel wird in Kapitel 2 zunächst detailliert auf die Anforderungen an das Gesamtsystem und im Speziellen auf die Architektur der Software eingegangen. Dort wird außerdem ein Überblick über die Sensorikhardware, den Sensorbus und das Protokoll, mit dem die Hardware mit dem „Hostrechner“ genannten Steuerungscomputer kommuniziert, gegeben. In den nachfolgenden Kapiteln 3 und 4 werden die konkreten Ideen, Algorithmen und die Implementierungen der beiden bisher entwickelten Softwaremodule *GastroInt-Sim* und *X-Sim* vorgestellt. Zu Beginn dieser Kapitel werden kurz die medizinischen Hintergründe beschrieben. Diese Beschreibungen sind in der Regel entweder dem Fachbuch [Gas03] entlehnt oder stellen mündlich durch Prof. Dr. Grund mitgeteiltes Wissen dar.

Kapitel 5 liefert schließlich ein Fazit und einen Ausblick auf zukünftige Entwicklungen, bzw. aktuelle Entwicklungen, die nicht mehr Einzug in diese Arbeit gefunden haben, wie z.B. ein Modul zur Überwachung des aktuell in Entwicklung befindlichen *Bronchophanten*, welcher in Zukunft für simulierte Bronchoskopien eingesetzt werden soll.

2. Architektur des Gesamtsystems

2.1. Anforderungen an das System

2.1.1. Modularisierung des Systems

Bis dato existieren bereits Tübinger Phantommodule für verschiedene Einsatzbereiche; für die Zukunft sind noch weitere Module geplant. Zielsetzung des Projekts war daher von Beginn an, dass das entstandene Gesamtsystem „Simtux“ eine einheitliche Basis für die Überwachung, Steuerung und Darstellung aller existierenden und zukünftigen Phantommodule darstellen soll. Die einzelnen Phantommodule unterscheiden sich mehr oder weniger stark voneinander in Hinblick auf die eingesetzte Sensorik und die Art und Weise, wie der simulierte Eingriff erfasst, ausgewertet und dem Benutzer präsentiert wird. Trotz der Unterschiede gibt es jedoch zwischen allen Phantomtypen große Gemeinsamkeiten. Deswegen lag es von Anfang an nahe, das Gesamtsystem analog zu den Tübinger Phantomen modular auszulegen, sodass gemeinsam genutzte Komponenten einmal implementiert und dann von verschiedenen Stellen aus verwendet werden können. Besonders in Hinblick auf zukünftige Entwicklungen ist eine strenge Modularisierung wichtig, damit das System für neue Arten von Phantomen leicht angepasst und erweitert werden kann.

2.1.2. Interaktion mit dem System

Die Zielgruppe des *Simtux*-Systems sind Anwender mit einem nicht-technischen Hintergrund. Für diese ist eine komplizierte Bedienung mit vielen einzustellenden Parametern unzumutbar. Im Idealfall soll das System aufgebaut und eingeschaltet werden können und dann sofort ohne großen Aufwand einsatzfähig sein. Alle notwendigen Konfigurationen sollten automatisch anhand vordefinierter Konfigurationsdateien vorgenommen werden. Weiterhin sollten notwendige Interaktionen so natürlich wie möglich in den Trainingsablauf integriert werden, sodass sich der Gesamtprozess nicht wesentlich von einer realen Endoskopie unterscheidet. Eine Bedienung mit Maus und Tastatur während des Trainings würde den Benutzer zu sehr in Anspruch nehmen. Deswegen kommt beispielsweise für

die Interaktion mit dem Biliphanten im Modul *X-Sim* ein Touchscreen, der ein Bedienfeld ähnlich dem eines realen Röntgengeräts zeigt, sowie ein Fußpedal, das dem realen Fußpedal für das Auslösen eines Röntgengeräts nachempfunden wurde, zum Einsatz.

Außerdem sollten die je nach Modul unterschiedlichen Auswertungen und Statusanzeigen derart präsentiert werden, dass die Anwender diese ohne Schwierigkeiten sofort erfassen und einordnen können. Alle menschliche Erfahrung zeigt, dass eine optisch gefällige Präsentation häufig größeren Zuspruch beim Anwender findet als nüchterne, „langweilige“ Statusmeldungen. Natürlich darf die Darstellung auf der anderen Seite auch nicht überfrachtet wirken, damit die relevanten Informationen klar erkannt werden können und der Anwender nicht abgelenkt wird. Wie in späteren Abschnitten genauer beschrieben wird, wurde daher auf eine optisch ansprechende, aber dennoch nicht überfrachtete oder unübersichtliche 3D-Darstellung zurückgegriffen. Diese Darstellung erfordert keine besondere Spezialhardware. Vielmehr ist *Simtux* auf handelsüblichen Notebooks lauffähig.

2.2. Überblick über die Hardwareimplementierung

Die oberste Modularisierungsstufe ist die Trennung des Systems in eine Hardware- und eine Softwareseite. Die Hardwareseite umfasst dabei sowohl die „echte“ Hardware wie die Sensorik und ein Bussystem, mit Hilfe dessen die Werte der einzelnen Sensoren eingesammelt werden, als auch Mikrocontroller, die diese Sensordaten aufbereiten und weiterleiten. Diese Mikrocontroller sind in der Sprache *C* programmierbar; dadurch besteht die Hardwareseite streng genommen zu einem Teil auch aus Software, aber der Klarheit halber sei dies dennoch unter dem Begriff „Hardwareseite“ zusammengefasst. Für die Entwicklung der Hardwareseite zeichnet Prof. Dr. Aurich verantwortlich; im Rahmen dieser Dissertation wurde ausschließlich die Software konzipiert und realisiert. Aus diesem Grund wird im Folgenden nur ein Überblick über die Hardware gegeben, während deren Details aus Sicht der Softwareentwicklung als *Black Box* aufgefasst und nicht näher erläutert werden.

Das Gesamthardwaresystem besteht aus den Sensoren, einem *Bus*, der Daten von und zu diesen Sensoren transportiert und einem *Busmaster*, welcher als Schnittstelle zwischen der Hard- und der Softwareseite fungiert. Das Gegenstück des Busmasters auf der Softwareseite ist der *Hostrechner*, auf dem der sog. „Sensorserver“ ausgeführt wird. Dieser kann mehrere Instanzen von *Simtux*, die per TCP/IP-Verbindung mit diesem verbunden sind, gleichzeitig mit Daten versorgen. Abb. 2.1 zeigt einen skizzenhaften Überblick.

In der momentanen Implementierung können über den Bus nicht mehr als 127 Sensoren adressiert werden, was sich jedoch bisher nie als Problem herausgestellt hat. Sollte es in Zukunft erforderlich werden, mehr als diese 127 Sensoren gleichzeitig einzusetzen, muss das Protokoll modifiziert werden. Weiterhin sind die Daten, die über den Sensorbus

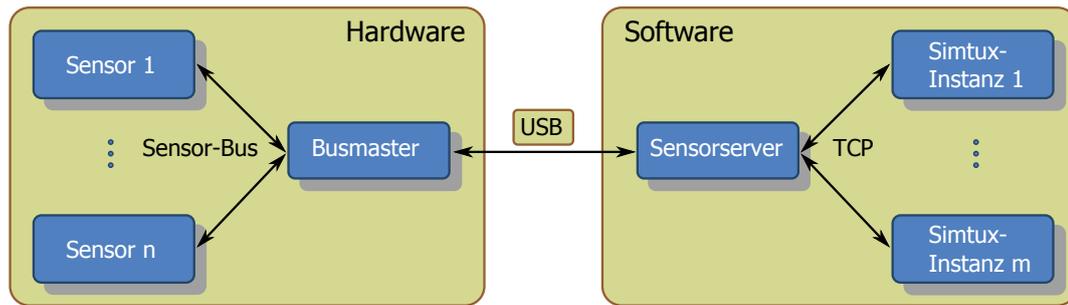


Abb. 2.1.: Sisse des Simtux-Systems.

übertragen werden können, auf 8-Bit-Werte beschränkt. Es gibt aber einige Arten von Sensoren, die 16-Bit-Werte erzeugen. Damit deren Werte über den Bus übertragen werden können, werden solche Sensoren technisch als zwei einzelne Sensoren behandelt, von denen der eine das High-Byte und der andere das Low-Byte des Sensorwerts liefert.

2.2.1. Die Sensoren

Für das Projekt wurden für die unterschiedlichen individuellen Anforderungen der einzelnen Phantommodule verschiedene Arten von Sensoren entwickelt. Ein Sensor besteht dabei stets aus der eigentlichen Sensorelektronik und einem Mikroprozessor, der die Rohdaten, die von der Sensorik geliefert werden, auswertet, aufbereitet und über den Bus an den Busmaster weiterleitet. Unter anderem kommen die folgenden Sensoren zum Einsatz:

- Lichtsensoren, die aus einem Fotowiderstand bestehen, dessen elektrischer Widerstand von der Stärke des einfallenden Lichts abhängt.
- Gravitationssensoren, die die Rotation des Sensors bezüglich zweier Raumachsen bestimmen können. Die Drehung um die senkrechte Achse, das ist die Achse, die zum Gravitationszentrum der Erde zeigt, lässt sich mittels Gravitationssensoren prinzipiell nicht bestimmen.
- Drucksensoren, die einen mechanischen Druck messen können.
- Kapazitive Sensoren, die die äußerst geringe elektrische Kapazität zwischen einem Instrument, wie z.B. einem Endoskop, und dem Sensor bestimmen können.

Auf Einsatz- und Funktionsweise der verschiedenen Sensoren soll an dieser Stelle nicht weiter eingegangen werden; vielmehr wird, wenn es in den entsprechenden Kapitel notwendig wird, noch einmal eine genauere Beschreibung der jeweiligen Sensorik geliefert werden.

2.2.2. Sensorbus und Busmaster

Wie oben bereits erwähnt wurde, leiten die Mikroprozessoren, die Teil eines jeden Sensors sind, die aufbereiteten Sensordaten über einen Bus weiter an den Busmaster. Der Bus verwendet ein einfaches, maßgeschneidertes Protokoll zur Übertragung der Daten. Daten können bidirektional, d.h. sowohl von den Sensoren zum Busmaster als auch für einige Spezialfälle in der anderen Richtung vom Busmaster zu den Sensoren übertragen werden. Der Busmaster ist ein weiterer Mikroprozessor, der die Verbindung zwischen den über den Bus angekoppelten Sensoren und dem Hostrechner herstellt. Dazu fragt er die Daten aller angeschlossenen Sensoren ab und schickt sie per USB-Verbindung an den Hostrechner. In der Gegenrichtung nimmt der Busmaster Daten vom Hostrechner entgegen, die er an die Sensoren weiterleitet. Das Übertragungsprotokoll zwischen Busmaster und Hostrechner sieht dabei vor, dass die Kommunikation in Zyklen nach folgendem Schema stattfindet:

1. Der Busmaster sendet über den USB-Bus den Wert 0 als Synchronisationssignal. Darauf folgt ein Datenpaket, das aus einer vier Byte großen Phantom-ID und den anschließenden Werten der 127 Sensoren besteht. Die ID des Phantoms ist nicht einfach eine zufällige Nummer, sondern besitzt eine wohldefinierte Struktur, welche im Anschluss an diese Beschreibung des Kommunikationsschemas beschrieben wird.

Um bei der Datenübertragung Verwechslungen mit dem Synchronisationsdatum 0 zu vermeiden, kommt in keiner gültigen Phantom-ID die Ziffer 0 vor und der Wert 1 ist der kleinstmöglich übertragbare Sensorwert. Erreicht wird dies, indem vor dem Sendevorgang zu allen Sensorwerten der Wert 1 addiert wird. Dies muss bei der späteren Auswertung der Daten natürlich berücksichtigt werden. Die Nutzung eines ausgezeichneten Synchronisationsdatums dient dazu, dass ein Host auch nach Verbindungsabbrüchen wie z.B. neu gestartetem Programm oder Verlust der USB-Verbindung jederzeit eine erneute Synchronisation durchführen kann, indem er so lange Daten einliest und verwirft, bis erneut das Datum 0 empfangen wurde. Von da an ist er wieder synchron mit dem Busmaster.

2. Nun verarbeitet der Sensorserver auf dem Hostrechner die empfangenen Werte und sendet anschließend eine Folge von 127 Werten $\in \{0, 1\}$ an den Busmaster zurück. Diese werden vom Busmaster an die entsprechenden Sensoren weitergeleitet und können von diesen unterschiedlich interpretiert werden; z.B. können die Lichtsensoren, die in Kapitel 3 Verwendung finden, mit LEDs ausgestattet werden. Diese Sensoren interpretieren das vom Busmaster empfangene Datum als Befehl, die LED ein- bzw. auszuschalten. Falls der Busmaster für eine bestimmte Zeit keine Daten vom Hostrechner empfängt („timeout“), überspringt er diesen Schritt und beginnt den nächsten Zyklus.

Phantomtyp	Erstes Byte
X-Sim	0x11
GastroInt-Sim (Kolon)	0x21
GastroInt-Sim (Magen)	0x22
Broncho-Sim	0x31

Tabelle 2.1.: Alle bisher verfügbaren Phantomtypen und ihre Codierungen.

Struktur der Identifikationsnummer eines Phantoms

Die Phantom-ID enthält codiert den Typ des Phantoms und eine fortlaufende Nummer, mit der alle Phantome eines Typs durchnummeriert werden. Der Typ des Phantoms wird in den ersten 8 Bits der ID codiert, während die letzten 8 Bits die fortlaufende Nummer enthalten. Die übrigen Bits werden zur Zeit nicht benutzt, könnten aber in Zukunft z.B. dazu benutzt werden, nähere Einzelheiten zum Phantom zu beschreiben. Tabelle 2.1 listet alle bisher verfügbaren Phantomtypen mit der zugehörigen Typ-Codierung auf.

2.3. Grundlagen der Softwareimplementierung

2.3.1. Wahl der Programmiersprache

Eine grundsätzliche Entscheidung zu Beginn des Projektes bestand in der Wahl der Programmiersprache. Dabei wurden die Gesichtspunkte Ausführungsgeschwindigkeit, Programmierkomfort und Verbreitung berücksichtigt. Die Verbreitung war deswegen wichtig, weil sich für verbreitete Programmiersprachen in der Regel wesentlich mehr Anleitungen, Beispiele und Tutorien finden lassen, was die Entwicklung des Programms erheblich erleichtert. In die enge Auswahl kamen die Sprachen *C++*, *Java* in der damaligen Version 6 und *C#* in der damaligen Version 3.5. Von diesen drei Sprachen ist C++ mit Abstand die vielseitigste und auch die performanteste. Allerdings ist die Programmierung in C++ nach subjektiven Erfahrungen deutlich weniger komfortabel als in C# und Java, was die Entwicklungszeit unnötig verlängert hätte. C# und Java ähneln sich sowohl vom zugrundeliegenden Konzept als auch von der Klassenbibliothek und der Syntax sehr stark und beide bieten auch eine ähnliche Ausführungsgeschwindigkeit, welche nach subjektiven Erfahrungswerten zwischen 90% und 100% der Geschwindigkeit eines vergleichbaren C++-Programms beträgt.

Java bietet gegenüber C# den Vorteil, dass die Sprache plattformunabhängig ist, sodass dasselbe Programm ohne die Notwendigkeit einer Neukompilierung auf den verschiedensten Plattformen lauffähig ist. Demgegenüber setzt C# auf das .Net-Framework der Firma Microsoft auf, welches ausschließlich für das Microsoft-Windows-Betriebssystem

erhältlich ist. Es existiert zwar eine Open-Source-Alternative namens *Mono*, die für die meisten verbreiteten Plattformen verfügbar ist, aber diese beinhaltet nicht alle Klassen des .Net-Frameworks. Speziell die für die Erstellung grafischer Benutzerschnittstellen (*GUI*) konzipierte Bibliothek *WPF* (Windows Presentation Foundation) wurde aufgrund ihres Umfangs im Mono-Framework nicht umgesetzt.

Auf der anderen Seite jedoch war *C#* in der damaligen Version ausdrucksstärker als Java und bietet bei allen Ähnlichkeiten eine, wiederum nach rein subjektiven Gesichtspunkten, übersichtlichere und durchdachtere Standard-Klassenbibliothek. Die Wahl fiel nach Abwägung aller Vor- und Nachteile auf die Programmiersprache *C#* und das .Net-Framework von Microsoft. Damit ist zwar keine vollständige Plattformunabhängigkeit gewährleistet, aber dies ist kein Problem:

Erstens ist *Simtux* keine Standardsoftware, die von einem möglichst großen Nutzerkreis auf vielfältigsten Zielplattformen genutzt werden soll, sondern ist vielmehr eine Spezialanwendung, die für genau den Zweck konzipiert wurde, im Zusammenhang mit den Tübinger Endoskopiephantomen unter wohldefinierten Bedingungen zu funktionieren. Aus diesem Grund stellt die momentane Beschränkung auf die Microsoft-Windows-Familie in der Praxis keine echte Einschränkung dar.

Zweitens stellt eine Portierung für andere Betriebssysteme wie z.B. Linux oder MacOS, sollte sie in Zukunft erforderlich werden, keine große Herausforderung dar. Mit Ausnahme der GUI setzen sämtliche Programmmodule vollständig auf Bibliotheken auf, die auch im Mono-Framework vorhanden sind, sodass sie auch mit diesem problemlos ausgeführt werden könnten. Die GUI könnte innerhalb weniger Tage angepasst werden, sodass das Gesamtprogramm vollständig plattformunabhängig wäre.

2.3.2. Der Aufbau von *Simtux*

Die Softwareseite des *Simtux*-Systems ist entsprechend der in 2.1.1 formulierten Anforderungen modular aufgebaut. Die Basis des Systems bildet das Programmframework *Taraxacum*. Dabei handelt es sich um eine Bibliothek, die an der Universität Düsseldorf parallel zum *Simtux*-Projekt entstanden ist und sich teilweise aus früheren Projekten entwickelt hat, teilweise durch den vorlesungsbegleitenden Übungsbetrieb zur Veranstaltung „Computergrafik“ weiterentwickelt wurde und zu einem wesentlichen Teil durch die speziellen Anforderungen von *Simtux* vorangetrieben wurde. *Taraxacum* umfasst eine *Grafikengine*, mittels welcher dreidimensionale Grafiken dargestellt werden können, ein *Skriptingsystem* zur Erweiterung der Engine und ein darauf aufbauendes umfangreiches *Entitysystem* zur Reaktion auf Ereignisse und der Definition und Steuerung logischer Abläufe in dreidimensionalen Szenen. In Anhang A.1 wird näher auf den Aufbau von *Taraxacum* eingegangen.

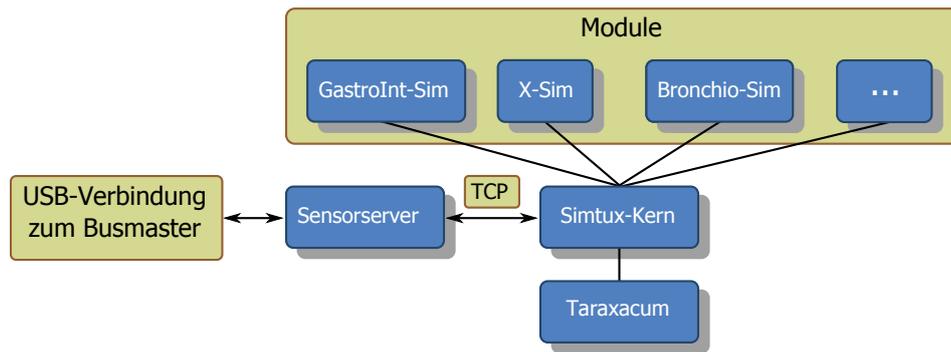


Abb. 2.2.: Aufbau der Softwareseite des *Simtux*-Systems. Diese Skizze ist gewissermaßen eine Ausschnittsvergrößerung der Softwareseite in Abb. 2.1.

Dieses Framework wurde vor allem deswegen für die Entwicklung von *Simtux* eingesetzt, weil hier an den verschiedensten Stellen dreidimensionale Grafiken und Skripte zum Einsatz kommen, für deren Darstellung und Programmierung *Taraxacum* hervorragende Dienste leistet: Von Beginn an war die Präsentation des Trainingsablaufs einer der zentralen Punkte bei der Entwicklung von *Simtux*. Diese sollte einerseits übersichtlich und funktional sein, damit sie den Benutzern im täglichen Einsatz keine unnötigen Hindernisse durch komplizierte Bedienung in den Weg stellt, andererseits war jedoch auch klar, dass sie ansprechend aussehen sollte, da – man mag dies bedauern – eine ansprechende Präsentation für Anwender und Kunden ein wichtiges Bewertungskriterium ist.

Prinzipiell sind zwei Vorgehensweisen vorstellbar. Die eine Möglichkeit der Präsentation besteht darin, auf eher schematische Ansichten zu setzen, die übersichtlich alle benötigten Informationen anzeigen. Der andere Weg, der letztendlich auch eingeschlagen wurde, besteht darin, eine möglichst realistische 3D-Darstellung zu erzeugen. Die Wahl der dreidimensionalen Darstellung hat sich spätestens bei der Entwicklung der Röntgensimulation des Moduls *X-Sim*, welche in Kapitel 4 behandelt wird, als sehr vorteilhaft erwiesen.

2.4. Programmkomponenten von *Simtux*

Die komplette Softwareseite des *Simtux*-Systems, die auf dem Hostrechner ausgeführt wird, besteht aus den drei Teilen *Sensorserver*, *Simtux*-Kern und den verschiedenen Modulen wie z.B. dem Modul für die Steuerung des Gastro-/Kolophanten „GastroInt-Sim“. Diese Module nutzen die durch den Kern bereitgestellte Basisfunktionalität. Für die Module gilt die Entwicklungsrichtlinie, dass für jeden Typ von Tübinger Phantom (Magen/Darm, Gallenwege, Bronchien, etc.) ein eigenes Softwaremodul erstellt wird. Die Modularisierung wird skizzenhaft in Abb. 2.2 verdeutlicht.

2.4.1. Sensorserver

Der *Sensorserver* ist ein separates Programm, das auf dem Hostrechner ausgeführt wird. Er dient als Schnittstelle zwischen dem eigentlichen *Simtux* auf der einen und dem *Busmaster* des angeschlossenen Phantoms auf der anderen Seite. Über den USB-Bus empfängt der Sensorserver Datenpakete vom Busmaster, wie bereits weiter oben in Abschnitt 2.2.2 beschrieben wurde. Auf der anderen Seite sendet der Sensorserver mit derselben Frequenz, mit der er die Daten vom Busmaster empfängt, an alle registrierten *Simtux*-Instanzen Datenpakete, die exakt so aufgebaut sind wie diejenigen, die vom Busmaster empfangen wurden. Der Sensorserver dient als Datenquelle für eine beliebige Anzahl von *Simtux*-Instanzen. Damit diese sich bei dem Sensorserver registrieren können, um Daten zu erhalten, horcht der Sensorserver permanent auf dem TCP-Port 4223 nach eingehenden Verbindungsanfragen eines neuen *Simtux*-Clients. Dabei spielt es keine Rolle, ob die *Simtux*-Instanz auf demselben physikalischen Rechner läuft wie der Sensorserver oder ob die Verbindung über das Netzwerk aufgebaut werden soll. Sobald eine *Simtux*-Instanz eine Verbindung aufbaut, wird sie registriert und dann kontinuierlich mit Sensordaten versorgt. Diese Entkoppelung von *Simtux* und der Verbindung zum Busmaster durch Einfügen einer Abstraktionsschicht in Form des Sensorservers führt nur scheinbar unnötige Redundanz ein. Tatsächlich ist das Sensorserverkonzept aus mehreren Gründen sehr nützlich.

Debugging

Der Hauptnutzen besteht in den stark verbesserten Möglichkeiten der Fehlersuche. Während der Entwicklung von *Simtux*, als die Software noch nicht ausgereift war und infolge dessen viele Fehler auftraten, kam es regelmäßig zu schwer auffindbaren Problemen bei der Kommunikation zwischen Busmaster und *Simtux*. Mit der Einführung des Sensorservers konnten viele Fehler besser gefunden werden. Insbesondere bei einem Absturz von *Simtux* ist es sehr praktisch, wenn die Sensordaten weiterhin empfangen und auch dargestellt werden können, sodass eine Klärung der Fehlersituation deutlich erleichtert wird.

Mehrere verschiedene *Simtux*-Instanzen

Weiterhin könnte eine vorstellbare zukünftige Erweiterung darin bestehen, dass *Simtux* in Prüfungssituationen eingesetzt wird, in denen Prüfling und Prüfer zwei verschiedene Rechner mit unterschiedlich konfigurierten *Simtux*-Instanzen nutzen, sodass dem Prüfer mehr Informationen zur Verfügung stehen als dem Prüfling. Beide *Simtux*-Instanzen müssten aber an dasselbe Phantom angeschlossen sein, was mit dem *Sensorserver*-Konzept leicht zu realisieren wäre.

2.4.2. *Simtux*-Kern

Der *Simtux*-Kern umfasst die von allen Modulen gemeinsam genutzte Basisfunktionalität für die Steuerung der Phantome sowie einen *Modullader*, welcher beim Start der *Simtux*-Anwendung als erstes gestartet wird. Sobald er läuft, meldet er sich automatisch an einem Sensorserver an, sodass von da an kontinuierlich Daten von diesem empfangen werden. IP-Adresse und Port des Servers können über Konfigurationsdateien festgelegt werden. Wie in 2.4.1 beschrieben wurde, enthält jedes Datenpaket im Paketheader die ID des Phantoms. Der *Modullader* untersucht diese und entscheidet anhand des in der ID codierten Phantomtyps, welches Modul für die Steuerung und Simulation des angeschlossenen Phantoms zuständig ist. Daraufhin startet er das passende Modul. Der *Modullader* ist so konzipiert, dass das Phantom im laufenden Betrieb ausgewechselt werden kann. Diesen Fall erkennt der *Modullader* daran, dass plötzlich eine andere Phantom-ID empfangen wird als bisher. Sollte dies der Fall sein, stoppt der *Modullader* das aktuell laufende Modul und startet stattdessen das Modul, welches mit dem Phantomtyp, der in der neuen Phantom-ID codiert ist, assoziiert wird.

Nachdem das Modul gestartet wurde, werden automatisch alle weiteren für dieses Phantom relevanten Daten von der Festplatte geladen. Dies ist mit Hilfe des Mechanismus zum Laden von Szenen, wie er in Anhang A.2.1 beschrieben wird, realisiert. Die meisten der für die Funktionsweise des Moduls notwendigen Objekte wie Sensoren, grafische Repräsentationen z.B. von Magen, Kolon oder Galle, oder interne Strukturen wie Kameras und Lichtquellen, sind mit Hilfe von *Entities*, also logischen Einheiten, denen mittels Skripten ein individuelles Verhalten zugewiesen werden kann, implementiert (vgl. Anhang A.1.3). Im Sinne der Definition in A.1 stellt das Phantom damit eine *Szene* dar, deren Beschreibung in einer XML-Datei abgespeichert ist. Diese Datei trägt als Dateinamen die hexadezimale Darstellung der Phantom-ID, wodurch sie durch den *Modullader* leicht aufgefunden werden kann.

2.4.3. Module

Die einzelnen Module umfassen alle notwendigen Datentypen für die Steuerung, Simulation und Darstellung des Phantomtyps, für den sie zuständig sind. Diese lassen sich grob in zwei Kategorien einteilen: Objekte, die für alle Phantome eines Typs gleich sind und solche, die für jedes Phantom unterschiedlich sind. Beispielsweise sind alle Objekte der grafischen Benutzerschnittstelle für alle Phantome desselben Typs gleich. Aus diesem Grund sind sie als statisch kompilierte *C#*-Klassen implementiert. Objekte, die für jedes Phantom eines Typs unterschiedlich sind, sind vor allem solche, die Daten zu Details des Phantoms beinhalten, wie z.B. Angaben zu Sensorpositionen, -kalibrierungswerten oder den zur grafischen Darstellung verwendeten Volumen- oder Dreiecksmeshdatensatz.

Solche Objekte sind, wie in A.1 beschrieben, als geskriptete *Entities* realisiert, deren Quellcode in Python geschrieben ist und die zur Laufzeit dynamisch kompiliert werden.

Im Kern besteht ein Modul mindestens aus einer Hauptklasse, die das Interface `IModule` implementiert, welches die drei Methoden `Init`, `Start` und `Stop` vorgibt. Der Modullader erzeugt eine Instanz dieser Klasse und ruft die Methoden `Init` und `Start` auf, um das Modul zu starten. Wird das Modul gestartet, erzeugt die Instanz der Hauptklasse in der Regel ein oder mehrere Bildschirmfenster und alle weiteren Objekte, die für die ordnungsgemäße Funktion des Moduls benötigt werden. Anschließend werden die individuellen Daten des Phantoms anhand der Informationen in „seiner“ XML-Datei geladen. Detailliertere Beschreibungen zu den einzelnen Modulen finden sich in den entsprechenden Kapiteln der verschiedenen Phantomtypen. Soll das Modul wieder beendet werden, z.B. weil die Anwendung beendet wird oder weil ein anderes Phantom angeschlossen wurde und daher ein anderes Modul gestartet werden soll, ruft der Modullader die Methode `Stop` auf und beendet damit das Modul.

3. Oberer und Unterer Gastrointestinaltrakt

Dieses Kapitel behandelt die Ideen und Algorithmen des Softwaremoduls, das die Endoskopien am Gastro- und Kolophanten, den Phantomen des Oberen und des Unteren Gastrointestinaltraktes, steuert. Gastro- und Koloskopien am Phantom unterscheiden sich in vielen wesentlichen Merkmalen kaum voneinander. Daher kommt in beiden Fällen zur rechnergestützten Überwachung und Steuerung eine weitgehend identische Sensorik zum Einsatz. Die Abb. 3.1 und 3.2 zeigen einleitend vier Fotos der Tübinger Phantome für Oberen bzw. Unteren Gastrointestinaltrakt.



Abb. 3.1.: Gastrophant mit montiertem Kopf.

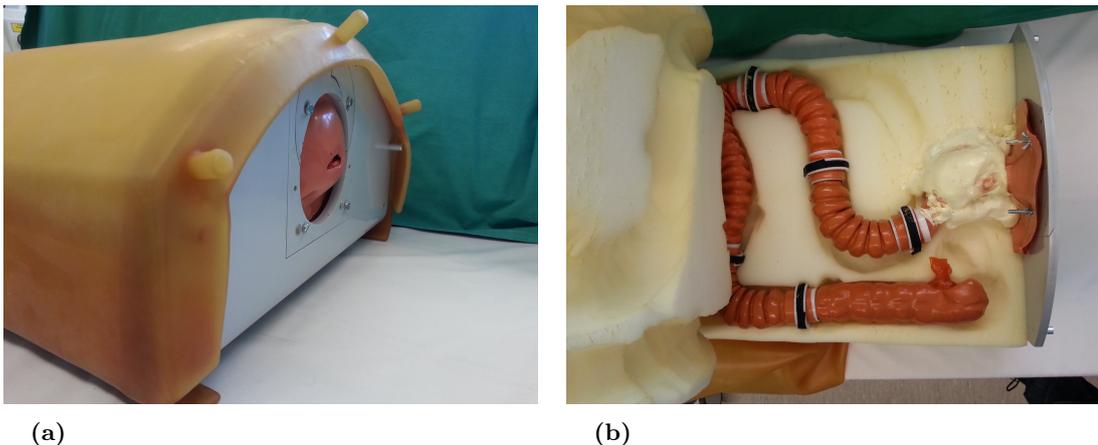


Abb. 3.2.: Kolophant. a) Rektum. b) Geöffnet mit Blick auf das Kolon. Das Rektum befindet sich auf der rechten Seite.

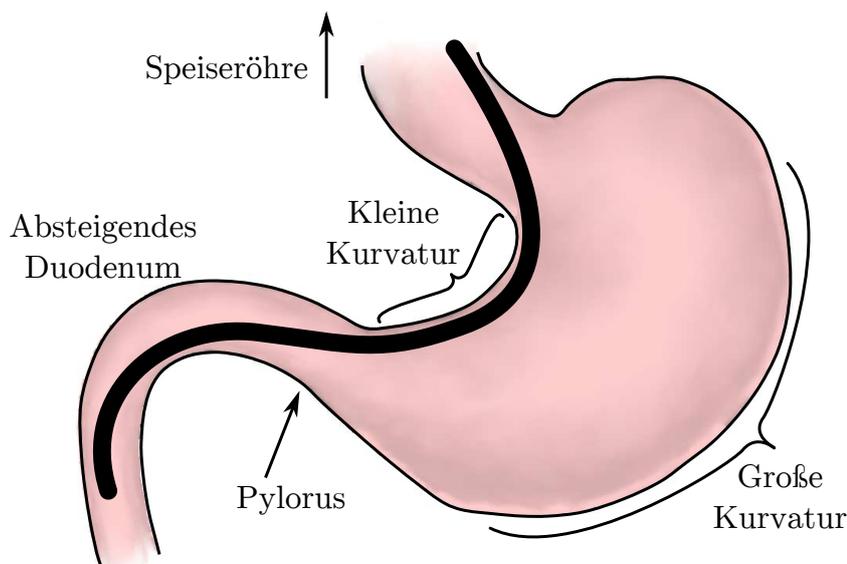


Abb. 3.3.: Gastroskopie: Vorschub des Gastroskops bis ins Duodenum. Abbildung von Duygu Özmen.

3.1. Endoskopien am Oberen bzw. Unteren Gastrointestinaltrakt

3.1.1. Gastrokopien

Es folgt zunächst eine einleitende Beschreibung des Ablaufs einer *Gastroskopie*. Diese Beschreibung lehnt sich eng an [Gas03] an:

Bei einer Gastroskopie wird ein spezielles Endoskop, das sog. *Gastroskop*, dazu benutzt, einen direkten Einblick in den Oberen Gastrointestinaltrakt zu erhalten, um dort Pathologien diagnostizieren zu können. Das Gastroskop wird zunächst in einem als *Intubation* bezeichneten Vorgang in den Rachen eingeführt und anschließend durch die Speiseröhre in den Magen und von dort durch die sog. *Pylorus*¹-Passage in das *absteigende Duodenum*² weitergeführt.

Während des Vorschubs werden alle durchquerten Bereiche systematisch und gründlich untersucht, damit sichergestellt wird, dass keine Stelle übersehen wird. Die Gastroskopspitze lässt sich vollständig invertieren, wodurch die Blickrichtung entgegengesetzt zur Vorschubrichtung gedreht wird, was für die Inspektion einiger Bereiche des Magens unverzichtbar ist. Abschließend wird das Gastroskop wieder zurückgezogen, wobei Duodenum, Magen und Speiseröhre nochmals gründlich untersucht werden.

¹Magenpförtner, also der Ringmuskel am Übergang vom Magen zum Duodenum

²Zwölffingerdarm

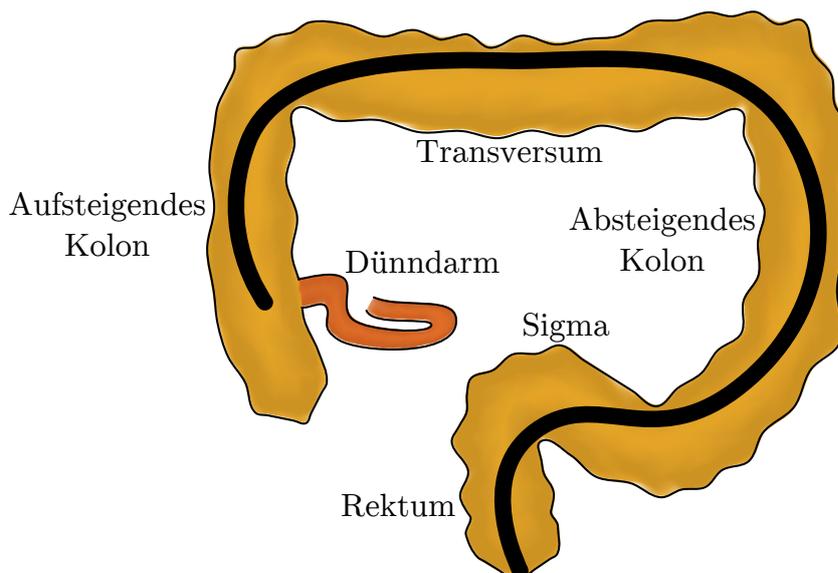


Abb. 3.4.: Koloskopie: Vorschub des Koloskops bis zur Dünndarmmündung. Abbildung von Duygu Özmen.

Herausforderungen bei einer Gastroskopie sind auf der einen Seite, dass keine Bereiche übersehen werden dürfen, damit keine Pathologien unentdeckt bleiben und auf der anderen Seite auch die Vermeidung der direkten Risiken von Komplikationen schon während des Eingriffs selbst, die nach [Gas03] mit einem Risiko von 0,11% auftreten können, wie z.B. eine Magenperforation, bei der die Wand des Magens mit dem Gastroskop verletzt wird.

3.1.2. Koloskopien

„Durch erfahrene Untersucher können eine komplette Koloskopie und Ileoskopie sicher und schnell bei nahezu allen Patienten durchgeführt werden, allerdings ist die Kunst der Koloskopieeinführung schwer erlernbar. Dies beruht darauf, dass eine Koloskopie im Wesentlichen die Passage einer flexiblen Röhre in einer anderen beinhaltet.“[Gas03].

Die Koloskopie ist umgangssprachlich auch unter dem Begriff *Darmspiegelung* bekannt. Hierbei wird ein Endoskop, das sog. *Koloskop*, rektal in das zuvor vorbereitete Kolon eingeführt. Es wird zunächst über die gesamte Länge des Kolons vom Mastdarm über das Sigma und das absteigende Kolon in das Transversum und von dort ins aufsteigende Kolon vorgeschoben, bis die Mündung des *terminalen Ileums*, des letzten Abschnitts des Dünndarms, erreicht wurde (Abb. 3.4). Danach wird das Koloskop langsam zurückgezogen und dabei sorgfältig hinter jede Darmfalte geschaut, damit dort keine Pathologien wie z.B. Darmpolypen oder Tumore übersehen werden.

3.2. Anforderungen an das Softwaremodul GastroInt-Sim

Für eine erfolgreich durchgeführte Gastro- bzw. Koloskopie ist es außerordentlich wichtig, dass der Untersucher keine Stelle der Oberfläche des Magens oder Darms übersieht, weil sich im Prinzip überall Pathologien verbergen können. Damit ein Untersucher jede Stelle inspizieren kann, muss er einerseits die Steuerung des Endoskops beherrschen und andererseits ein gutes räumliches Vorstellungsvermögen entwickeln. Zu Beginn der Entwicklung des Softwaremoduls zur Überwachung und Steuerung der Phantome des Gastrointestinaltraktes wurden daher die folgenden Ziele festgelegt:

- Für Anfänger, die die Navigation von Gastro- oder Koloskop gerade erlernen, soll auf dem Bildschirm eine unmittelbare Rückmeldung erscheinen, die ihnen zeigt, welchen Bereich des Magens oder Kolons sie aktuell betrachten.
- Für die anschließende Auswertung soll ein Protokoll darüber erstellt werden, welche Bereiche der Trainierende beobachtet hat und welche übersehen wurden.
- Damit der Trainierende die geforderten Fähigkeiten erlernen und verbessern kann, sollen ihm gezielt Aufgaben gestellt werden können, z.B. die Aufgabe „Beobachte die folgenden Stellen im Magen in der folgenden Reihenfolge so schnell und gründlich wie möglich“. Diese Aufgaben sollen durch das System automatisch oder durch einen Tutor manuell gestellt werden können. Dieser Vorgang muss für den Tutor mit möglichst wenig Interaktion komfortabel durchgeführt werden können.

Im Laufe der Arbeiten an diesem Projekt hat sich die Gewichtung dieser Anforderungen von den Protokollierfunktionen weg zu einer reinen Darstellung der aktuell betrachteten Bereiche sowie dem Stellen von Navigationsaufgaben verschoben. Daher wurden die Protokollierfunktionen letztendlich nicht implementiert.

3.3. Existierende Navigationssysteme

Für die Navigation eines Endoskops vor allem im Kolon existieren bereits Navigationssysteme auf dem Markt. Das dahinterstehende Prinzip wird als *MEI, Magnetic Endoscope Imaging*, bezeichnet [Gas03]. Bei diesen Systemen sind direkt im Endoskop in regelmäßigen Abständen kleine Spulen angebracht, die der Reihe nach ein Magnetfeld erzeugen. Auf der Körperoberfläche des Patienten liegt eine Detektionsvorrichtung mit Empfängerspulen auf, in denen durch das Magnetfeld der Sendespulen des Endoskops ein Strom induziert wird. Anhand der Stromstärke kann auf den Abstand zwischen Sender- und Empfängerspulen rückgeschlossen werden. Durch Triangulation kann daraus die Position der Senderspulen im Raum relativ zum Empfänger berechnet werden. Da diese

in bekannten Abständen entlang des Endoskops angebracht sind, lässt sich daraus die Gestalt des Endoskops exakt ermitteln.

Ein solches System könnte für *Simtux* im Allgemeinen und das Modul *GastroInt-Sim* im Speziellen prinzipiell hervorragende Dienste leisten; jedoch war eine der zentralen Vorgaben für die Entwicklung von *Simtux*, dass keinerlei spezielle Gerätschaften vonnöten sein sollen. Das System soll mit jedem beliebigen Endoskop funktionieren können, damit es überall dort, wo Trainingskurse angeboten werden sollen, auch garantiert funktioniert. Aus diesem Grund kamen solche navigierten *MEI*-Systeme für den Einsatz in *Simtux* nicht in Frage.

3.4. Überlegungen zur Umsetzung

Die Kernaufgabe des Moduls *GastroInt-Sim* besteht darin, den Bereich der Oberfläche des Gastrointestinaltraktes zu ermitteln, den der Trainierende gerade beobachtet. Prinzipiell sind dazu zwei Vorgehensweisen denkbar:

1. Die Position der Endoskopspitze wird erfasst. Dann kann bei bekannter Geometrie des Gastrointestinaltraktes auf den Bereich von dessen Oberfläche geschlossen werden, der aktuell beobachtet wird.
2. Es wird direkt ermittelt, welche Oberflächenbereiche beobachtet werden, ohne den „Umweg“ über die Erkennung der Lage der Endoskopspitze zu gehen.

Für die konkrete Umsetzung wurde letztendlich die zweite Variante gewählt. Da, wie im vorigen Abschnitt erläutert wurde, ein Spezialendoskop mit integrierter Detektorvorrichtung nicht in Frage kommt, muss diese Aufgabe mit den Mitteln gelöst werden, die bei jedem Endoskop zur Verfügung stehen. Insbesondere besitzt jedes Endoskop eine starke Lichtquelle, die genau in dessen Blickrichtung leuchtet. Gelingt es also zu erfassen, welcher Teil der Oberfläche beleuchtet wird, kann daraus darauf zurückgeschlossen werden, welcher Bereich momentan beobachtet wird. Im Folgenden wird beschrieben, wie dies realisiert wurde. Es wird außerdem in Abschnitt 3.4.6 theoretisch untersucht, ob bei einer erfolgreichen Erkennung der beleuchteten Oberflächenbereiche sogar die genaue Lage der Endoskopspitze im Raum ermittelt werden kann.

Vereinbarung: Da sich die Überlegungen und Umsetzungen für Magen und Kolon stark ähneln, wird folgende sprachliche Vereinfachung vorgenommen: Wenn im Weiteren von der „Oberfläche des Magens“ oder in ähnlichen Zusammenhängen vom Magen die Rede ist, sei stillschweigend immer auch das Kolon gemeint. Sollte wirklich einmal explizit nur der Magen gemeint sein, wird dies entsprechend kenntlich gemacht.

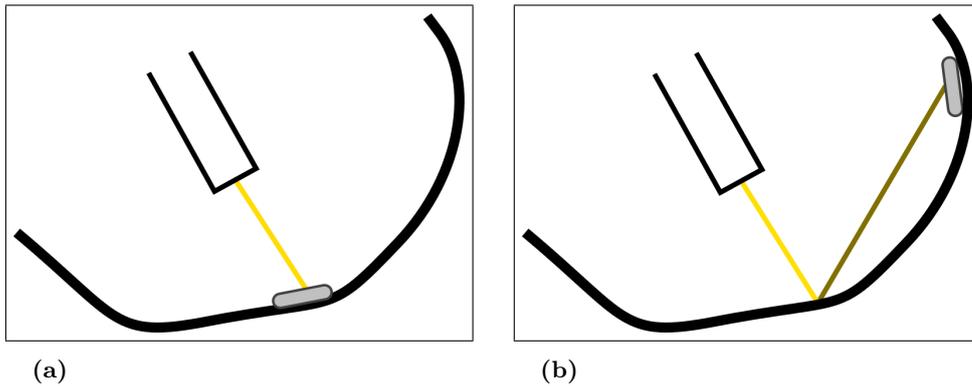


Abb. 3.5.: Einfall des Lichtscheins. a) Lichtschein trifft direkt auf den Fotowiderstand. b) Der Lichtschein trifft zunächst auf eine Oberfläche, wird dort reflektiert und trifft den Fotowiderstand indirekt.

3.4.1. Erkennung der beobachteten Oberflächenbereiche

Definition 3.1. *Unter der Bestrahlungsstärke versteht die Physik die Leistung pro Fläche, die durch auf die Fläche einfallendes Licht transportiert wird.*

Die Bestrahlungsstärke I lässt sich z.B. durch einen Fotowiderstand messen. Dabei handelt es sich um ein elektrisches Bauteil, dessen elektrischer Widerstand proportional zu I ist. Durch Messen dieses Widerstands kann daher direkt auf die Bestrahlungsstärke geschlossen werden, die die Oberfläche in der unmittelbaren Nähe des Fotowiderstandes erfährt. Die Grundidee ist dabei die Folgende: Befindet sich an einer bestimmten Stelle der Oberfläche des Magens ein solcher Lichtsensor, der eine Bestrahlungsstärke > 0 misst, gibt es genau zwei mögliche Interpretationen:

1. Der Sensor wird vom Endoskop beleuchtet und demzufolge blickt das Endoskop diese Stelle direkt an (Abb. 3.5a).
2. Der Sensor wird nur indirekt beleuchtet, d.h. das Endoskop blickt in eine andere Richtung und durch Reflexion an der Magenwand und Streuungseffekte wird ein Teil des Lichts auf den Sensor reflektiert (Abb. 3.5b).

Bei der Reflexion des Lichts an der Magenwand wird die Bestrahlungsstärke aufgrund von Streuungseffekten stark verringert. Daher lassen sich die beiden Fälle anhand des Sensorwerts relativ verlässlich voneinander unterscheiden: Ist der Sensorwert groß genug, lässt sich dieser nicht durch Reflexionen oder Streuungen erklären. Stattdessen muss der Sensor direkt von der Endoskoplichtquelle beleuchtet werden. Ist der Sensorwert zu klein, gibt es im Prinzip zwei Möglichkeiten: Entweder der Sensor wird nur indirekt beleuchtet, oder die Endoskopspitze ist weit vom Sensor entfernt bzw. der Einfall des Lichts auf den

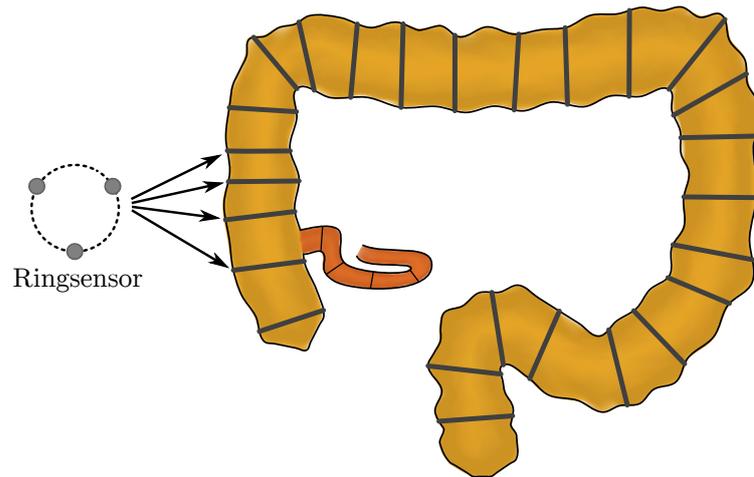


Abb. 3.6.: Skizze der Ringsensoren für das Kolon. Drei Fotowiderstände sind jeweils in Reihe geschaltet und bilden einen Ringsensor. Abbildung von Duygu Özmen.

Sensor geschieht relativ tangential. Diese beiden Situationen lassen sich anhand der Sensorwerte nicht mehr zuverlässig voneinander unterscheiden, was aber kein echtes Problem darstellt: Blickt der Trainierende tangential auf die Magenwand oder ist er weit davon entfernt, ist es ihm im Allgemeinen nicht möglich, eine genaue Inspektion durchzuführen, sodass dieser Bereich in beiden Fällen als nicht inspiziert angesehen werden kann. Was „groß genug“ und „zu klein“ bedeutet, muss durch eine einmalige Kalibrierung der Sensoren festgelegt werden.

3.4.2. Modifikationen am Tübinger Phantom

Zur Messung der Bestrahlungsstärke wurde das Tübinger Phantom dahingehend modifiziert, dass auf der Oberfläche von Magen bzw. Kolon eine Menge von Fotowiderständen verteilt wurde. Im Falle des Magens beträgt der mittlere Abstand zwischen zwei benachbarten Sensoren ca. 4 cm. Für das Kolon wurden jeweils drei Fotowiderstände in Reihe geschaltet und bilden so sehr grob angenähert einen ringförmigen Lichtsensor. Solche Ringe wurden alle zwei bis fünf Darmfalten, das entspricht ungefähr 3 cm bis 6 cm, um das Kolon gelegt, wie Abb. 3.6 zeigt.

Zusätzlich wurden auf dem Magen des *Gastrophanten* direkt neben einigen der Fotowiderstände rote LEDs aufgebracht, deren Licht durch die Magenwand scheint, sodass es aus Sicht eines eingeführten Gastroskops gesehen werden kann. Diese sind unter anderem entlang zweier markanter Strukturen des Magens angeordnet: Der großen und der kleinen Krümmung des Magens.

Diese Krümmungen verlaufen längs der beiden gekrümmten Konturen „rechts“ bzw. „links“ am Magen (siehe Abb. 3.3). Diese LEDs lassen sich gezielt ein- und ausschalten. Damit

kann ein wichtiges, grundlegendes Manöver bei der Gastroskopie trainiert werden: Die gezielte Ansteuerung eines Zielpunktes mit dem Gastroskop. Dabei stellt eine eingeschaltete LED diesen Zielpunkt dar. Mit Hilfe des Fotowiderstandes, der sich direkt neben einer solchen LED befindet, ist es möglich, zu erkennen, ob die Region um LED und Fotowiderstand direkt angeleuchtet wird (siehe Abschnitt 3.5.2) und damit, ob der Zielpunkt erfolgreich angesteuert wurde.

3.4.3. Umgang mit der spärlichen Sensorverteilung

Angenommen die Oberfläche wäre dicht mit Lichtsensoren bedeckt. Dann könnte für jeden Punkt auf der Oberfläche anhand des Messwerts des dort liegenden Sensors entschieden werden, ob er beleuchtet wird oder nicht. Da es jedoch aufgrund des nicht zu vertretenden Aufwands in der Praxis nicht möglich ist, die Oberfläche wirklich dicht-andicht mit Sensoren zu überdecken, muss stattdessen stets mit einer spärlichen Sensorverteilung gearbeitet werden. Es stellt sich also die Frage, wie mit Oberflächenpunkten verfahren wird, an denen sich kein Sensor befindet.

Die naheliegende Lösung besteht darin, die wenigen Punkte, für die Sensorwerte vorhanden sind, als Stützstellen eines Interpolationsproblems aufzufassen. Dann kann mittels Interpolation für alle Punkte, die zwischen diesen Stützstellen liegen, ein Zwischenwert errechnet werden. Auf diese Weise kann die spärliche Verteilung der Sensorwerte auf die gesamte Oberfläche erweitert werden. Das Interpolationsverfahren soll die folgenden beiden Eigenschaften aufweisen:

1. Es soll plausible Resultate erzielen, d.h. die Helligkeitsverteilung, die auf diese Weise berechnet wird, soll der Realität entsprechen.
2. Der Aufwand bei der Berechnung muss gering sein, damit diese echtzeitfähig ist.

Shepard-Interpolation

Ein sehr alte und einfache Interpolationsmethode ist die *Globale Shepard-Interpolation* [She68], die manchmal auch *Inverse Distanzgewichtung* genannt wird. Diese interpoliert einen Wert $u(x)$ anhand vorgegebener Werte u_i in den (paarweise verschiedenen) Stützstellen x_i und ist gegeben durch:

$$u_p(x) = \begin{cases} \frac{1}{w_{sum}} \cdot \sum_{i=1}^N w_i(x) \cdot u_i & \text{falls } \|x - x_i\|_2 > 0 \ \forall i \\ u_i & \text{falls } x = x_i \end{cases}$$

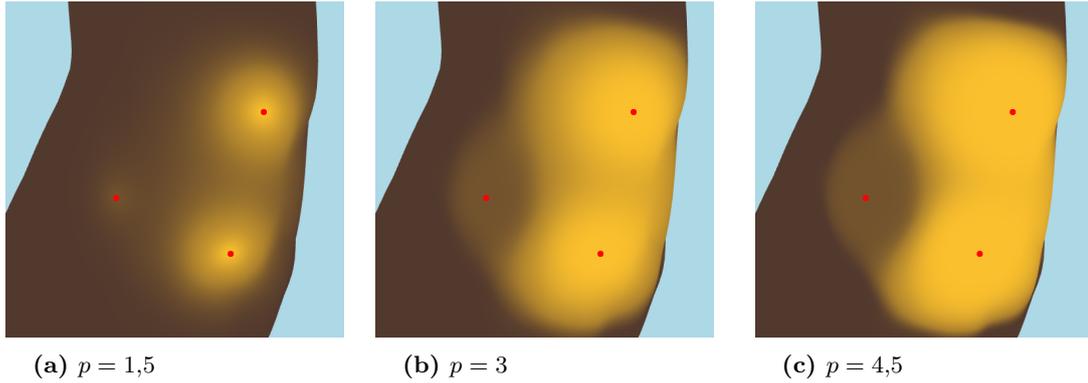


Abb. 3.7.: Mit Shepard-Interpolation berechnete Helligkeiten, dargestellt in gelblichem Farbton auf der ansonsten homogen braunen Magenoberfläche. Die Positionen der Sensoren sind hier rot markiert.

Hierbei sind $w_i(x) = \frac{1}{\|x - x_i\|_2^p}$ und $w_{sum}(x) = \sum_{i=1}^N w_i(x)$.

Der Parameter p bestimmt, wie „schnell“ der Einfluss einer Stützstelle x_i mit wachsendem Abstand zwischen x_i und dem zu interpolierenden Punkt x abnimmt. Im Zusammenspiel mit der Normierung der Gewichte aufgrund der Division durch die Gewichtssumme führt dies dazu, dass die Stützstelle x_n , die dem Punkt x am nächsten liegt, umso dominanter wird, je größer p ist. Dennoch haben stets alle Stützstellen einen Einfluss > 0 auf das Interpolationsresultat, weshalb das Verfahren *global* genannt wird. Nur im Grenzwert $p \rightarrow \infty$ strebt das Interpolationsergebnis gegen eine Voronoi-Zerlegung, bei der das Interpolationsresultat bei x vollständig durch den Wert u_n in der Stützstelle x_n bestimmt wird und damit *lokal* ist.

Abb 3.7 zeigt Resultate der Shepard-Interpolation mit verschiedenen Werten für den Parameter p . In diesem Fall empfangen drei benachbarte Sensoren Licht, zwei davon gleich viel, der dritte ca. $\frac{1}{4}$ davon, während alle übrigen Sensoren kein Licht messen. Der interpolierte Helligkeitswert in jedem Punkt auf der Magenoberfläche wird farbcodiert in gelblichem Farbton dargestellt. Dabei ist die Farbe umso heller, je größer die interpolierte Helligkeit ist.

Analyse

In Abb. 3.7a ist eine deutliche „Hütchenbildung“ zu sehen, bei der die Helligkeit nur in unmittelbarer Nähe zu den Stützstellen hoch ist und dazwischen deutlich abfällt. Dieses Verhalten entspricht nicht dem realen Lichtschein, weil in der Realität davon auszugehen ist, dass der Lichtschein zwischen drei hell beschienenen Punkten ebenfalls hell ist. In Abb. 3.7c mit $p = 4,5$ ist bereits eine deutliche Tendenz zu einer Voronoizerlegung zu erkennen, wodurch sich Plateaus mit relativ konstanten Helligkeitswerten ergeben, die

an deren Rand abrupt wechseln. Auch dieses Verhalten ist nicht realistisch, weil in der Realität eher von einem glatten Helligkeitsverlauf zwischen den drei Sensoren auszugehen ist und weil der Rand des Lichtscheins in der Realität weicher ist. In der Praxis hat sich gezeigt, dass ein Wert $p \approx 3$ die besten Ergebnisse erzielt, wie in Abb. 3.7b zu sehen ist. Hier sind die Ränder des Lichtscheins relativ weich und zwischen den drei Sensoren ist ein glatter Helligkeitsverlauf zu sehen.

Die Wahl $p = 3$ liefert durchaus brauchbare Ergebnisse, ist allerdings noch nicht vollständig zufriedenstellend. Die Helligkeit zwischen den beiden rechten Sensoren ist nämlich, ähnlich wie in Abb. 3.7a, nur deutlich schwächer ausgeprägt, ein wenig geringer als an den Stützstellen selbst. Dies liegt daran, dass bei einem Parameter $p = 3$ die globale Natur des Verfahrens noch erkennbar ist und daher auch die übrigen Sensoren, die gar kein Licht empfangen und relativ weit von den Punkten zwischen den drei beschienenen Sensoren entfernt liegen, noch einen gewissen Einfluss auf das Interpolationsergebnis haben. Dieser Einfluss lässt sich nur mit einem größeren Exponenten beseitigen, was jedoch, wie eben gezeigt, zu einer unschönen Plateaubildung führt.

Zweiter Ansatz zur Modellierung der Lichtverteilung

Um die Nachteile der Shepard-Interpolation zu beheben, wurde ausgehend von folgender empirisch motivierter Überlegung ein zweites Verfahren entwickelt, bei dem der Einfluss einer Stützstelle nicht mehr global ist: Das Sichtfeld des Endoskops und damit auch der Bereich, in den dessen Lichtquelle strahlt, ist ein Kegel. Daher gilt, dass immer dann, wenn ein Sensor S_i ein nennenswertes Signal liefert, S_i innerhalb eines solchen Kegels liegen muss. Die genaue Lage von S_i innerhalb des Kegels kann aber aufgrund von Mehrdeutigkeiten nicht bestimmt werden. Der verfolgte Ansatz besteht nun darin, die genaue Position von S_i innerhalb des Kegels zu ignorieren. Stattdessen wird die vereinfachende Annahme getroffen, dass der Lichtschein stets auf den Sensor S_i zentriert ist und dort senkrecht auf die Oberfläche trifft. Dadurch kann man davon ausgehen, dass das Gebiet, das in der Umgebung von S_i liegt, ebenfalls beleuchtet wird. Diese Umgebung kann durch eine Kugel K_{r_i} mit einem bestimmten Radius r beschrieben werden.

Da in der Praxis die Sensordichte zwar nicht beliebig groß, aber dennoch immerhin groß genug ist, sodass zumeist mehr als nur ein Sensor vom Lichtschein des Endoskops getroffen wird, muss obige Überlegung natürlich für alle beleuchteten Sensoren vorgenommen werden. Bei der Konstruktion des neuen Verfahrens wird für einen Punkt p auf der Magenoberfläche daher zunächst einfach die Summe

$$V_{ges}(p) = \sum_{i=1}^N V_i \cdot \chi_{K_{r_i}}(p)$$

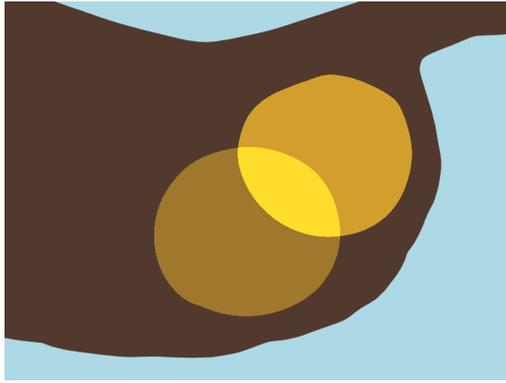
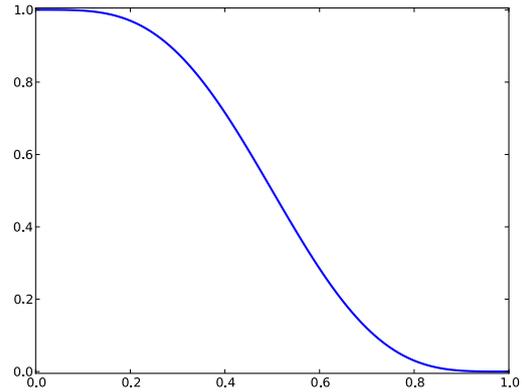


Abb. 3.8.: Überlappende Kreisscheiben.

Abb. 3.9.: Graph der Funktion $h_1(d)$.

mit der charakteristischen Funktion $\chi_{K_{r_i}}(p)$ als Helligkeitswert angenommen. Anschaulich bedeutet dies, dass um jeden Sensor eine Kugel mit Radius r_i gelegt wird, innerhalb der die Helligkeit konstant als V_i angenommen wird, während sie außerhalb 0 ist. Überlappen sich in einem Punkt zwei solche Kugeln, werden dort die Helligkeiten addiert. Diese simple Addition hat zwei unschöne Eigenschaften, die in Abb. 3.8 exemplarisch abgebildet sind:

1. Am Rand einer solchen Kugel tritt ein Sprung in der Helligkeit auf.
2. Überlappen sich zwei Kugeln, ist die Helligkeit in der Überlappungsregion deutlich höher als dort, wo nur eine einzige Kugel vorliegt.

Beide Effekte hängen damit zusammen, dass die Helligkeitsverteilung innerhalb einer Kugel als konstant angenommen wurde. Ein realistischeres und auch optisch ansprechenderes Resultat erhält man, wenn die Helligkeit stattdessen als zum Rand der Kugel hin stetig differenzierbar abfallend modelliert wird. Eine im Umfeld der OpenGL-Shadersprache *GLSL* (siehe Anhang A.1.1) häufig verwendete Funktion für einen solchen sanften Abfall ist die dort sogenannte `smoothstep`-Funktion, die definiert ist als

$$\text{smoothstep}_r(d) := \begin{cases} t^2 \cdot (3 - 2t) \text{ mit } t = \frac{d}{r} & \text{falls } d \leq r \\ 1 & \text{sonst.} \end{cases}$$

Diese Funktion ist für den gewünschten Effekt gut geeignet, wenn anstelle von `smoothstepr` die Funktion $h_r(d) := 1 - \text{smoothstep}_r(d)$ verwendet wird. Abb. 3.9 zeigt den Graphen dieser Funktion. Mit dem Parameter r lässt sich das Intervall $[0, r]$ festlegen, innerhalb dessen $h_r(d)$ sanft von 1 auf 0 abfällt. In der Praxis wird damit der Radius des Einflussbereichs eines Sensors festgelegt.

Ausgehend von dieser Helligkeitsverteilung kann nun die endgültige Interpolationsformel konstruiert werden, die im Folgenden als das „Additionsverfahren“ bezeichnet wird:

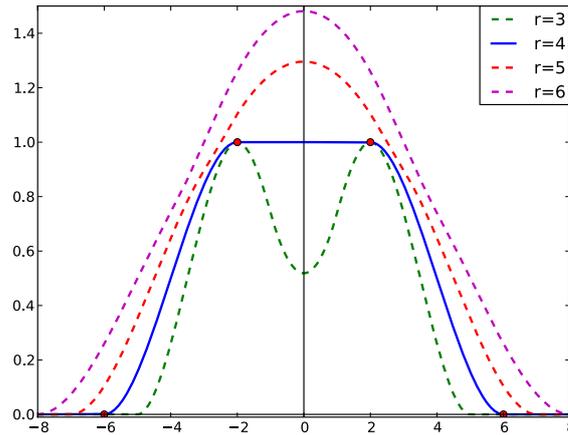


Abb. 3.10.: Funktionsgraphen des Additionsverfahrens im Eindimensionalen für verschiedene Parameter r . Interpoliert werden die Punkte $(-6,0)$, $(-2,1)$, $(2,1)$ und $(6,0)$. Für $r > 4$ berechnet das Verfahren keine Interpolation mehr.

Ähnlich wie bei der *Shepard*-Interpolation wird auch beim neuen Ansatz eine gewichtete Summe über die Werte aller Stützstellen vorgenommen. Diesmal allerdings werden die Gewichte wie oben beschrieben durch die Funktion h_r definiert und ganz bewusst nicht auf 1 normiert, damit der Einflussbereich eines Sensors unabhängig von den übrigen Sensoren ist:

$$V_r(x) = \sum_{i=1}^N w_i(x) \cdot V_i \text{ mit } w_i(x) = h_r(\|x - x_i\|_2) \quad (3.1)$$

Ohne eine Normierung der Interpolationsgewichte kann der interpolierte Wert in einer Stützstelle größer sein als der dort tatsächlich vorgegebene Wert, weshalb das Verfahren streng genommen kein echtes Interpolationsverfahren, sondern eher ein Approximationsverfahren ist. Solange die Sensoren aber zueinander einen Abstand haben, der in der Größenordnung von r liegt, tritt dieser Effekt nicht oder nur vernachlässigbar auf.

Begründung für diese Wahl

Für eine anschauliche Erklärung dieser Wahl von r werden in Abb. 3.10 Funktionsgraphen dargestellt, die mit einer eindimensionalen Variante des Additionsverfahrens interpoliert wurden. Man sieht, dass für $r = 4$, was in dem Beispiel genau dem Abstand zwischen zwei benachbarten Stützstellen entspricht, das Verfahren sowohl alle vier Stützpunkte interpoliert, als auch keine „Durchhänger“ produziert. Für $r < 4$ werden die Punkte zwar ebenfalls interpoliert, aber es ergeben sich deutliche „Durchhänger“ zwischen zwei Stützstellen. Für $r > 4$ sind die berechneten Werte in den Stützstellen größer als die dort vorgegebenen Stützwerte.

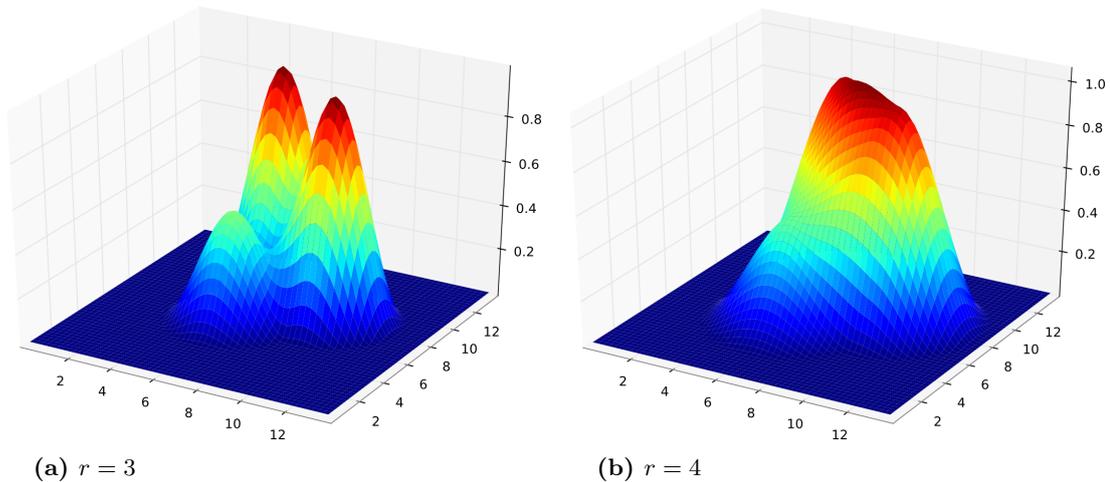


Abb. 3.11.: Funktionsgraphen des Additionsverfahrens im Zweidimensionalen für verschiedene Parameter r . Interpoliert werden die Punkte $(6, 6, 0,5)$, $(10, 6, 1)$ und $(8, 9,46, 1)$.

Abb. 3.11 zeigt Graphen für die auch tatsächlich zum Einsatz kommende zweidimensionale Variante des Verfahrens. Es liegen drei Stützstellen mit Werten > 0 vor, deren Abstand untereinander jeweils 4 ist. Auch in diesem Fall sieht man, dass für $r = 3$ deutliche „Durchhänger“ auftreten. Bei $r = 4$ werden die Stützstellen exakt interpoliert und es gibt keine „Durchhänger“ mehr.

Vergleich des Shepard- und des Additionsverfahrens

Abb. 3.12 zeigt exemplarisch den Vergleich zwischen einer mit dem Shepard- und einer mit dem Additionsverfahren berechneten Helligkeitsverteilung auf der Magenoberfläche. Es wird jeweils der interpolierte Lichtschein des Endoskops gelblich auf einem ansonsten homogen braunen Magen dargestellt, d.h. es werden keine sonstigen Beleuchtungseffekte berechnet. Auf den ersten Blick ähneln sich die Ergebnisse des Shepard-Verfahrens mit $p = 3$ und des Additionsverfahrens. Der Lichtschein, der mit dem Additionsverfahren berechnet wurde, ist jedoch glatter, während beim Shepard-Verfahren einerseits ein leichter „Durchhänger“ zwischen den beiden rechten Sensoren und andererseits ein deutlicher Helligkeitssprung zwischen den beiden rechten und dem linken Sensor sichtbar ist.

Auf den abgebildeten Momentanaufnahmen ist natürlich nicht zu sehen, wie der Lichtschein reagiert, falls das Endoskop bewegt wird. Betrachtet man diese Bewegung jedoch in der Praxis, so werden die Vorteile des Additionsverfahrens noch deutlicher: Beim Additionsverfahren wandert der Lichtschein stetig über die Magenoberfläche, während beim Shepard-Verfahren eher einzelne Bereiche des Magens langsam heller oder dunkler werden, ohne eine wirklich flüssige Bewegung dazwischen zu produzieren.

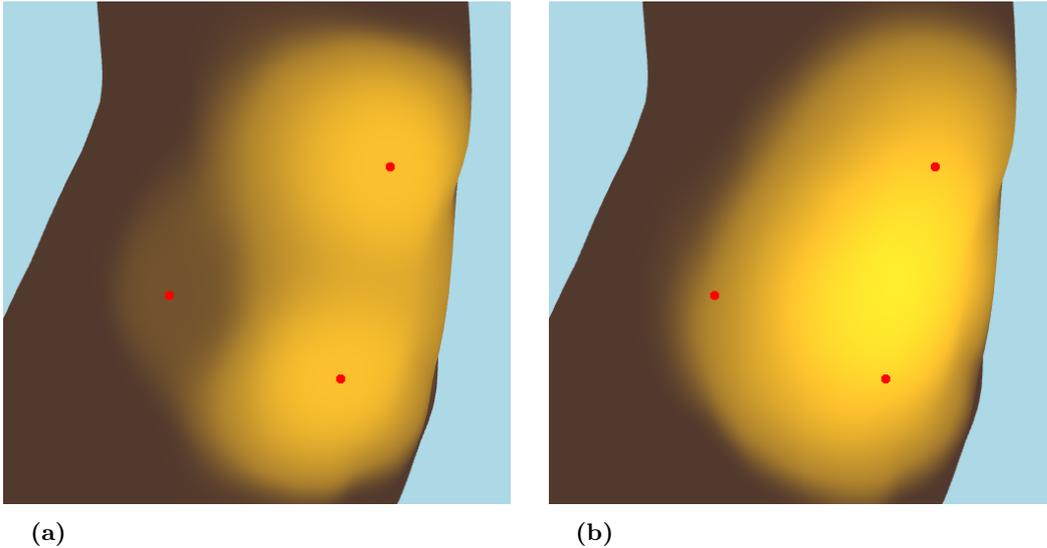


Abb. 3.12.: Gegenüberstellung der Interpolationsergebnisse von Shepard- und Additionsverfahren.
a) Shepard mit $p = 3$. b) Additionsverfahren.

Der Rechenaufwand beider Verfahren ist ungefähr derselbe. In beiden Fällen muss für jeden interpolierten Punkt eine gewichtete Summe über alle Stützstellen berechnet werden. Für moderne Computer stellt dies keinen besonderen Aufwand dar und ist damit, wie zu Beginn gefordert wurde, echtzeitfähig.

Aus diesen Gründen fiel die Entscheidung letztendlich eindeutig auf die Verwendung des Additionsverfahrens.

3.4.4. Mögliche Rekonstruktionsfehler

Da der Lichtschein letztendlich aus nur wenigen Sensorwerten berechnet wird, können bei dessen Rekonstruktion selbstverständlich Fehler auftreten. Das verwendete Modell geht davon aus, dass der Lichtkegel stets senkrecht auf einen Sensor zentriert ist. Dies entspricht natürlich nur in Ausnahmefällen der Realität; vielmehr ist der Einfallswinkel des Lichts in der Regel nicht 90° (Abb. 3.13b) und der Kegel kann so liegen, dass ein Sensor nur an dessen Rand liegt (Abb. 3.13c). Ein gekippter Einfallswinkel ist meistens unproblematisch, weil die Auswirkung nur darin besteht, dass der berechnete Lichtschein kreisförmig ist, während der reale Lichtschein die Form eines schräg projizierten Kreises hat. Die Annahme der Zentrierung des Kegels auf einen Sensor kann aber, wie die Skizze zeigt, zu relativ starken Abweichungen des berechneten Lichtscheins vom realen Lichtschein führen.

Dennoch ist diese Annahme vernünftig. In der Realität kann derselbe Sensorwert nämlich von sehr vielen verschiedenen Lichtkegeln erzeugt werden, wobei es unmöglich ist, den

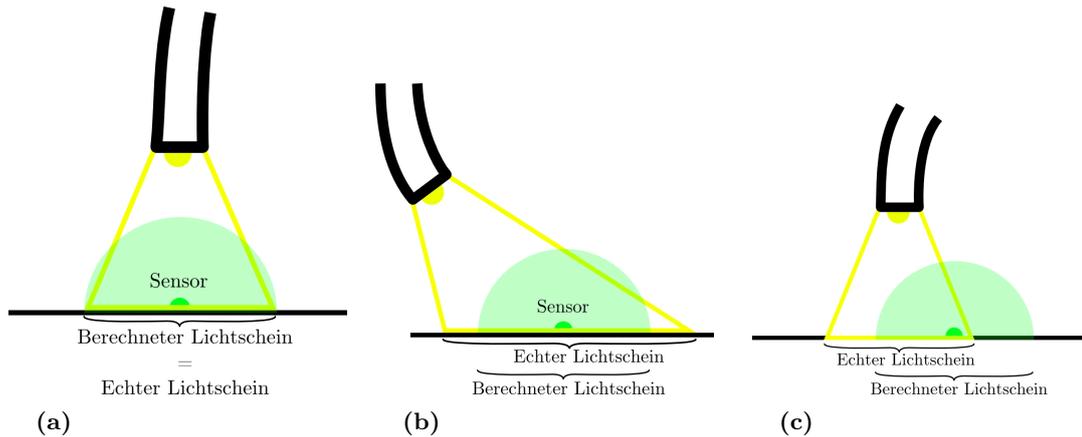


Abb. 3.13.: Korrektheit des Lichtscheins: a) Lichtschein wurde korrekt rekonstruiert. b) Lichtschein wurde bis auf Projektionsfehler korrekt rekonstruiert. c) Lichtschein wurde an der falschen Stelle rekonstruiert.

richtigen herauszufinden. Deswegen stellt die Annahme, dass der Lichtkegel stets auf den Sensor zentriert ist, einen Kompromiss dar, wobei diese Zentrierung gewissermaßen den Mittelwert über alle möglichen Lichtkegel darstellt. Damit ist der maximale Fehler bei dieser Annahme am kleinsten. Der Fehler ließe sich durch eine höhere Sensordichte deutlich verkleinern, weil in der Interpolationsformel der mittlere Abstand der Sensoren zueinander die Breite des berechneten Lichtscheins bestimmt. Es muss daher ein Kompromiss zwischen technischem Aufwand durch eine größere Sensordichte und der Präzision der Rekonstruktion des Lichtscheins eingegangen werden.

3.4.5. Definition des Abstandsbegriffs

Bisher wurde immer stillschweigend vorausgesetzt, dass der Begriff des Abstands zweier Punkte auf der Oberfläche des Magens wohldefiniert ist. Es gibt aber nicht nur eine mögliche Metrik um Abstände zu messen, sondern eine Vielzahl. Die gängige Möglichkeit, den Abstand zweier Punkte zu messen, ist die euklid'sche Metrik des \mathbb{R}^3 . Diese ignoriert die Gestalt der Oberfläche des Magens vollständig und betrachtet die direkte Verbindungslinie zwischen zwei Punkten.

Es sind aber auch andere Abstandsbegriffe denkbar, z.B. eine Metrik, die die Länge des kürzesten Weges zwischen zwei Punkten entlang einer Oberfläche berechnet. Solche kürzesten Wege werden als *Geodäten* bezeichnet und im Folgenden wird die Länge einer solchen kürzesten Verbindungsstrecke daher als der *geodätische Abstand* bezeichnet. Solange die Oberfläche nur wenig gekrümmt ist, unterscheiden sich beide Abstandsbegriffe nicht sehr voneinander, da die Geodäte zwischen zwei Punkten dann zumindest lokal annähernd der direkten Verbindungsstrecke entspricht.

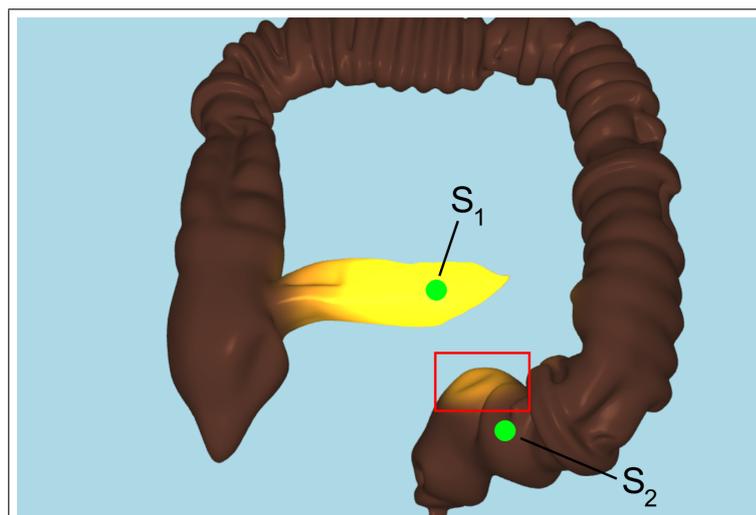


Abb. 3.14.: Beispiel für das „Auslaufen“ des Lichts im Kolon: Sensor S_1 empfängt sehr viel Licht, Sensor S_2 gar kein Licht. Durch die Interpolation der Sensorwerte ist jedoch in der Nähe von Sensor S_2 ein erhellter Bereich zu erkennen (rot umrahmt).

Bei einer starken Krümmung sieht der Fall anders aus, denn dort kann der geodätische Abstand zweier Punkte erheblich größer sein als der euklid'sche. Solche starken Krümmungen treten in der Praxis in der Regel nur auf kleinem oder sehr großem Maßstab auf, so z.B. bei der Oberfläche des Kolons, die bei näherer Betrachtung stark gefurcht ist, oder das, wenn es als Ganzes betrachtet wird, die Gestalt eines gekrümmten Schlauchs hat.

Vorteil des geodätischen Abstands

Der letztgenannte Fall kann tatsächlich ein Problem darstellen, wenn euklid'sche Distanzen betrachtet werden: Abb. 3.14 zeigt einen Fall, in dem der Sensor S_1 , der sich im Dünndarm befindet, sehr viel Licht empfängt, welches die Umgebung des Sensors S_2 beeinflusst, obwohl das Endoskop diesen Bereich in der Realität unter keinen Umständen beleuchten kann. Aufgrund der direkten Abstandsmessung ohne Berücksichtigung der Oberflächengestalt kommt es in diesem Beispiel jedoch zu einem „Auslaufen“ des Lichtscheins in Richtung S_1 . Diesen Fall würde die Verwendung von geodätischen Distanzen sofort korrigieren, da die Geodäte zwischen Anfang und Ende des Kolons im Gegensatz zur direkten Verbindungslinie eine erhebliche Länge hat und dadurch die Interpolation keine falschen Werte erzeugen würde. Wie Geodäten entlang der Oberfläche eines Dreiecksmeshs effizient approximativ berechnet werden können, wird in Anhang A.3.1 beschrieben.

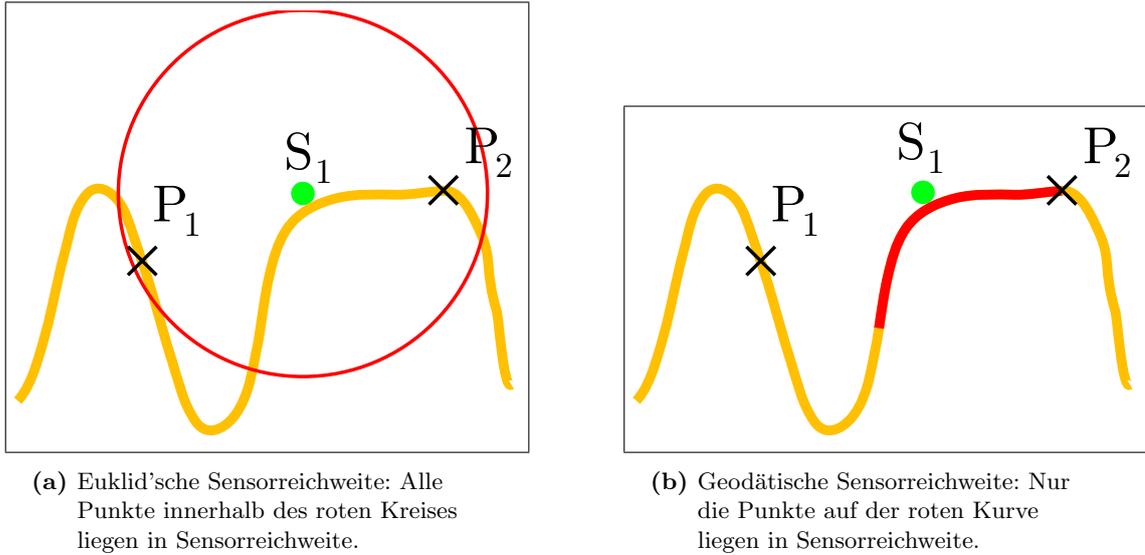


Abb. 3.15.: Gegenüberstellung der Reichweiten des Sensors S_1 in euklid'scher und geodätischer Distanz anhand eines skizzierten Schnitts durch eine Kolonfalte. Auffällig ist, dass die Punkte P_1 und P_2 in (a) beide im Einflussbereich liegen, während der Punkt P_1 in (b) außerhalb der Reichweite liegt.

Vorteil des euklid'schen Abstands

Auf der anderen Seite gibt es jedoch auch Situationen, in denen die Nutzung von geodätischen Distanzen weniger plausibel ist als die euklid'sche Distanz. Diese betreffen den kleinen Maßstab in der Größenordnung von Oberflächendetails wie z.B. den eben schon erwähnten Kolonfalten. In Abb. 3.15 ist ein solcher Fall skizziert. Die Abbildung zeigt einen Längsschnitt durch eine Kolonfalte.

Der Sensor S_1 , der einen durch den roten Kreis symbolisierten Einflussradius hat, empfängt Licht. Gemessen anhand der euklid'schen Distanz sind die beiden Punkte P_1 und P_2 gleich weit vom Sensor S_1 entfernt und es ist intuitiv auch nicht ersichtlich, wieso einer der beiden Punkte außerhalb des Einflussbereichs des Sensors liegen sollte und der andere darin, da sich offensichtlich kein Hindernis zwischen S_1 und P_1 bzw. P_2 befindet. Immerhin wird durch die Lichtquelle eines Endoskops im Allgemeinen ein gewisser Bereich erhellt und es kann dabei, zumindest in Abwesenheit von Abschattungseffekten, davon ausgegangen werden, dass der erhellte Bereich in der Realität mehr oder weniger isotrop um den Sensor liegt. Werden Abstände jedoch nach der Länge der Geodäten gemessen, liegt P_2 voll im Einflussbereich von S_1 , während P_1 außerhalb liegt, weil die Geodäte der Falte folgt und daher eine relativ große Länge aufweist. In Abb. 3.15b wird der Einflussbereich des Sensors durch den rot markierten Bereich der Kolonfalte symbolisiert.

Schlussfolgerungen aus dem Vergleich der Abstände

Obwohl sowohl für die euklid'sche als auch für die geodätische Distanz Situationen angeführt wurden, in denen das jeweilige Verfahren Schwächen aufweist, sind die Unterschiede in der Praxis eher klein. Dies liegt daran, dass in den meisten relevanten Situationen euklid'sche und geodätische Distanzen größenordnungsmäßig gleich sind. Nur sehr lokal betrachtet ergeben sich aufgrund von Falten und anderen Oberflächenstrukturen gewisse Unterschiede, die aber in der Praxis keine große Rolle spielen, weil durch die Interpolation die tatsächlichen Beleuchtungsverhältnisse ohnehin nur grob wiedergegeben werden können. Global betrachtet ergeben sich zwar manchmal große Abweichungen, wobei der euklid'sche Abstand deutlich zu klein für die Interpolation ist und daher ein „Auslaufen“ des Lichtscheins auftritt, jedoch sind diese Fälle, wie in Abb. 3.14 beispielhaft zu sehen ist, immer eindeutig zu erkennen, sodass der Benutzer keine Schwierigkeiten hat, darüber hinwegzusehen. Im Fall des Beispiels in Abb. 3.14 weiß er aufgrund der großen Helligkeit, dass sich die Endoskopspitze in der Nähe des Sensors S_1 befinden muss und daher unmöglich gleichzeitig in der Nähe des Sensors S_2 sein kann.

Die Benutzung des euklid'schen Abstands ist in der Implementierung erheblich weniger aufwändig und außerdem wesentlich effizienter zu berechnen als die geodätische Distanz, da lediglich die Norm $\|p_s - p_i\|_2$ des Differenzvektors zwischen der Stützstelle p_s und dem zu interpolierenden Punkt p_i berechnet werden muss, was ohne merkliche Verzögerung in Echtzeit für jeden dargestellten Punkt möglich ist. Für die geodätische Abstandsrechnung müsste dagegen erheblich mehr Aufwand betrieben werden, weil die Berechnung von Geodäten entlang einer durch ein Dreiecksmesh approximierten Oberfläche algorithmisch nicht trivial ist (siehe Anhang A.3.1). Damit ist eine Echtzeitberechnung ausgeschlossen.

Die Alternative besteht darin, die Geodäten bzw. die Längen der Geodäten einmalig in einem Vorverarbeitungsschritt zu berechnen und diese Informationen dann so abzuspeichern, dass sie für die Echtzeitdarstellung benutzt werden können. Diese Lösung ist durchaus praxistauglich, aber bedarf einer deutlich größeren Rechenleistung und benötigt mehr Speicher als die einfache euklid'sche Distanz, bei einem in der Praxis im Wesentlichen sehr ähnlichen Bild (vgl. Abb. 3.16). Aus diesem Grund fiel die Wahl letztendlich auf die Benutzung des einfachen euklid'schen Abstands.

Durch die Verwendung des Additionsverfahrens und des euklid'schen Abstands kann die Verteilung des Lichts auf der Oberfläche des Magens ausreichend realistisch rekonstruiert werden und damit der beschienene Oberflächenbereich gut bestimmt werden. Daher kann die erste Anforderung an das Modul als erfüllt angesehen werden.

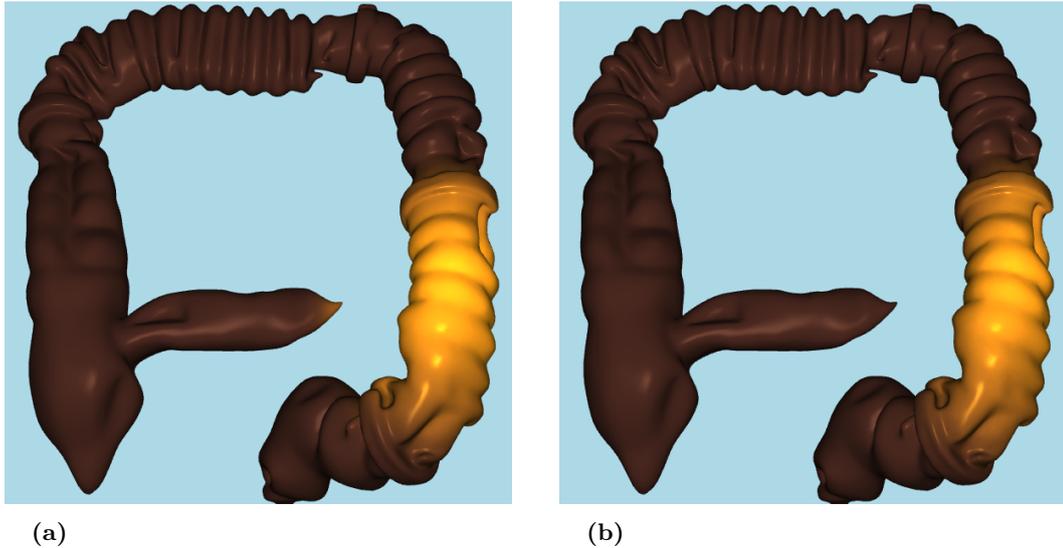


Abb. 3.16.: Ergebnisse mit a) euklid'scher und b) geodätischer Distanz in der Praxis. Die Unterschiede sind praktisch nicht zu sehen.

3.4.6. Theoretische Überlegungen zur exakten Lokalisierung

Nun soll theoretisch untersucht werden, ob eine exakte Lokalisierung der Spitze des Endoskops anhand des Lichtscheins seiner Lichtquelle auf die auf der Magenoberfläche verteilten Fotowiderstände möglich ist. Diese Lichtquelle werde zu Vereinfachungszwecken zunächst einmal als isotrope, d.h. in alle Richtungen gleichmäßig strahlende, Punktlichtquelle aufgefasst. Dann gilt nach dem bekannten $\frac{1}{r^2}$ -Abstandsgesetz sowie dem Lambert'schen Kosinusgesetz, dass die Bestrahlungsstärke I , die vom i -ten Fotowiderstand gemessen wird, einerseits vom Abstand r_i zwischen der Lichtquelle und der Detektorfläche dieses Fotowiderstandes und andererseits vom Einstrahlwinkel θ_i des Lichts auf diese Detektorfläche abhängt, wie in Abb. 3.17a skizziert ist.

Daher gilt:

$$I(r_i, \theta_i) \sim \frac{\cos(\theta_i)}{r_i^2}$$

$$\Rightarrow I(r_i, \theta_i) = I_0 \cdot \frac{\cos(\theta_i)}{r_i^2}$$

Die Proportionalitätskonstante I_0 kann durch entsprechende Kalibrierung der Sensoren relativ einfach ermittelt werden. Die Gleichung hängt daher scheinbar von zwei Unbekannten (r_i und θ_i) ab. Der Winkel θ_i muss jedoch als Winkel im Raum aufgefasst werden und lässt sich am Besten mit Hilfe des Skalarprodukts $\langle \hat{n}_i, (l - s_i) \rangle$ aus dem Einheitsnormalenvektor \hat{n}_i , der senkrecht auf dem i -ten Fotowiderstand steht, und dem normierten Vektor $\widehat{(l - s_i)}$, der vom i -ten Fotowiderstand zur Lichtquelle weist, berechnen.

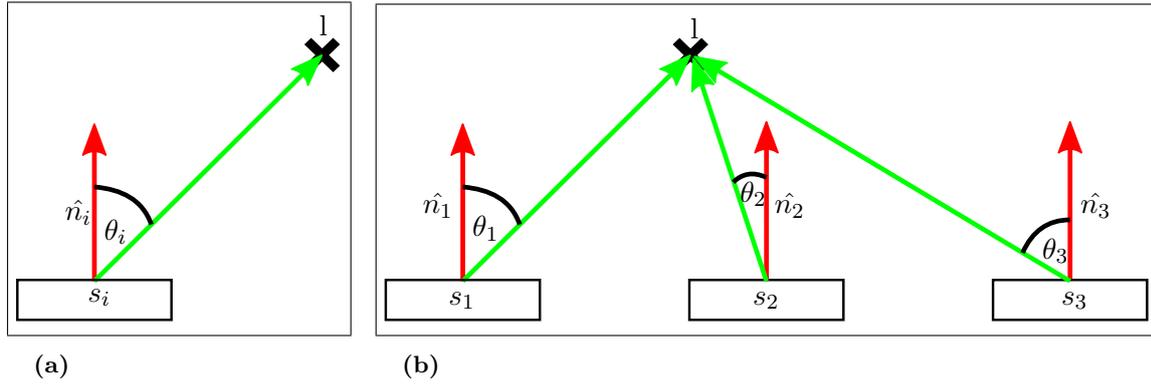


Abb. 3.17.: a) Skizze zum Lichteinfall von der Lichtquelle l auf einen einzelnen Fotosensor s_i .
b) Skizze zum Lichteinfall auf drei Fotosensoren.

Daher gilt:

$$\begin{aligned}
 I(r_i, \theta_i) &= I_0 \cdot \frac{\cos(\theta_i)}{r_i^2} \\
 \Rightarrow I(l) &= I_0 \cdot \frac{\langle \hat{n}_i, (\widehat{l - s_i}) \rangle}{\|l - s_i\|_2^2} \\
 &= I_0 \cdot \frac{\langle \hat{n}_i, (l - s_i) \rangle}{\|l - s_i\|_2^3}
 \end{aligned}$$

Nach Ausmultiplikation:

$$\begin{aligned}
 I(l) &= I_0 \cdot \frac{\langle \hat{n}_i, (l - s_i) \rangle}{\|l - s_i\|_2^3} \\
 &= I_0 \cdot \frac{(\hat{n}_{i,x} \cdot (l_x - s_{i,x}) + \hat{n}_{i,y} \cdot (l_y - s_{i,y}) + \hat{n}_{i,z} \cdot (l_z - s_{i,z}))}{((l_x - s_{i,x})^2 + (l_y - s_{i,y})^2 + (l_z - s_{i,z})^2)^{\frac{3}{2}}}
 \end{aligned}$$

Diese Gleichung hängt also sogar von drei Unbekannten ab, nämlich l_x , l_y und l_z . Daraus folgt, dass mindestens drei verschiedene solcher Gleichungen benötigt werden, damit ein Gleichungssystem aufgestellt werden kann, mit dessen Lösung die drei Komponenten l_x , l_y und l_z des Ortsvektors der Endoskopspitze eindeutig berechnet werden können. Alle anderen Größen sind theoretisch zumindest prinzipiell bekannt: Sowohl die Positionen s_i der einzelnen Sensoren als auch die Ausrichtungen der Sensoren, die sich durch ihre Normalenvektoren \hat{n}_i ausdrücken lassen, sind gegeben.

Die drei erforderlichen Gleichungen erhält man, indem man die Werte von drei Sensoren misst und jeweils in die Gleichung einsetzt. Abb. 3.17b zeigt eine Skizze der geometrischen Situation. Sofern mehr als drei Sensorwerte zur Verfügung stehen, ist das Gleichungssystem überbestimmt. In diesem Fall könnte das Ergebnis durch numerische Verfahren robuster gemacht werden als es bei nur drei vorhandenen Messwerten der Fall ist.

3.4.7. Diskussion des Ansatzes

Der oben beschriebene Ansatz ist einfach und elegant und würde unter idealisierten Bedingungen auch gut funktionieren. Es gibt jedoch in der praktischen Umsetzung an mehreren Stellen Probleme:

1. Es ist aus praktischen Gründen nicht möglich, eine genügend hohe Dichte von Sensoren zu erreichen, wobei „genügend“ hier bedeutet, dass in jeder Situation mindestens drei Sensoren überhaupt von der Lichtquelle beschienen werden.
2. Die Lichtquelle des Endoskops ist in der Realität keine isotrope Punktlichtquelle. Vielmehr ist sie flächenhaft und vor allem strahlt sie nicht in alle Richtungen gleichmäßig, sondern besitzt einen bestimmten Öffnungswinkel, der sich von Modell zu Modell unterscheidet.
3. Da der Magen aus flexiblem Latex besteht, können die Positionen sämtlicher Sensoren, die sich auf dessen Oberfläche befinden, nur mit einer großen Unsicherheit bestimmt werden. Für die Normalenvektoren gilt dies sogar in noch größerem Maße, da kleine Deformationen des Magens bereits ausreichen können, um einen Normalenvektor vollkommen zu verändern.

Vor allem Punkt 3 stellt ein gravierendes Problem dar: In den theoretischen Überlegungen wurde vorausgesetzt, dass sowohl die Positionen als auch die Normalenvektoren der Sensoren präzise bekannt sind. Dadurch, dass diese Werte in der Praxis jedoch mit großen Unsicherheiten behaftet sind, kann anhand der Sensorwerte kaum ermittelt werden, wo sich die Endoskopspitze wirklich befindet.

Weiterhin bedeutet Punkt 2, dass die Intensität am Rand des Lichtscheins geringer ist als in dessen Zentrum. Dadurch wird eine zusätzliche Ungenauigkeit in die Berechnung eingeführt. Schließlich bedeuten die Punkte 1 und 2 kombiniert, dass es durchaus Endoskoplagen geben kann, die so ungünstig sind, dass weniger als drei Sensoren überhaupt vom Lichtschein getroffen werden.

Die Probleme sind zusammengenommen derart gravierend, dass der Ansatz, die exakte Lage der Endoskopspitze mittels Lichtmessungen zu berechnen, in der Praxis nicht zum Ziel führt. Aus diesem Grund wurde wie bereits erwähnt für die konkrete Implementierung der in Abschnitt 3.4.1 beschriebene Ansatz der direkten Bestimmung und Darstellung der beobachteten Oberflächenbereiche gewählt.

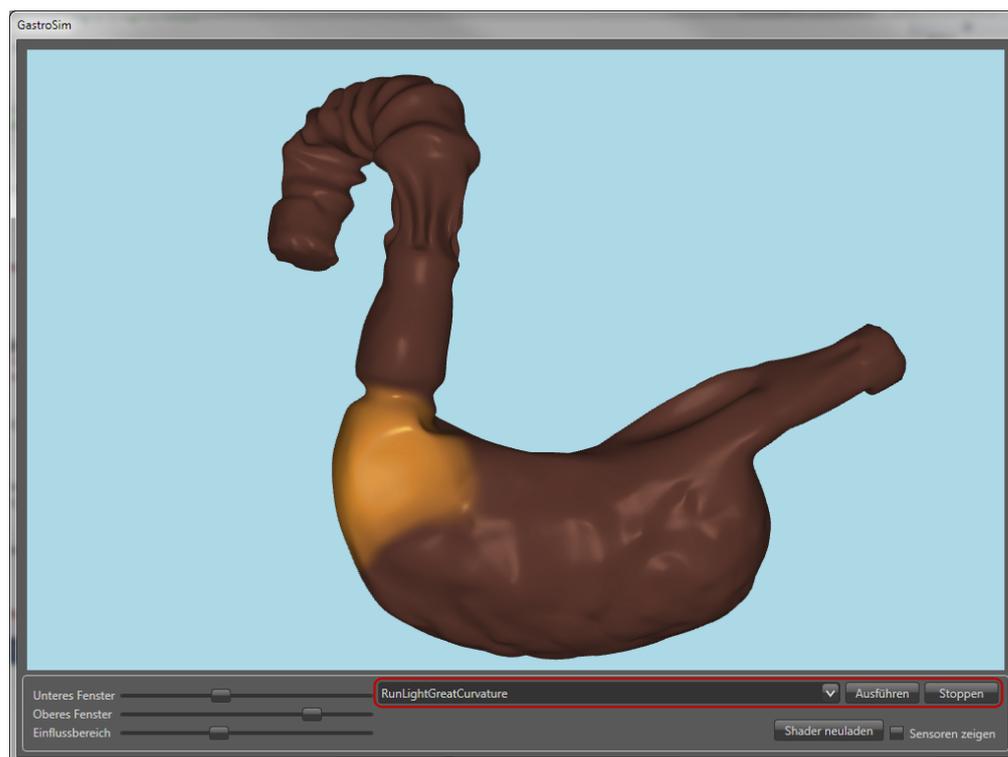


Abb. 3.18.: Hauptfenster von *GastroIntSim*. Rot umrandet ist der Bereich, in dem Skripte ausgewählt, gestartet und angehalten werden können.

3.5. Implementierung des Softwaremoduls

Das Softwaremodul, welches die Steuerung und Überwachung der Gastro- bzw. Kolophanten übernimmt, wurde *GastroInt-Sim* getauft. Die grafische Benutzeroberfläche besteht aus einem einzigen Hauptfenster mit zwei für den Endanwender relevanten Bereichen. Diese sind zum einen der Bereich zur 3D-Darstellung und zum anderen, unter diesem angeordnet, ein Steuerungsbereich. Dieser enthält eine Auswahlbox, in der voreingestellte Skripte gewählt werden können und zwei Buttons zum Starten und vorzeitigen Beenden dieser Skripte. Abb. 3.18 zeigt einen Überblick über das Hauptfenster der Anwendung.

3.5.1. 3D-Bereich

Dieser Bereich ist der bei weitem größte Teil des Hauptfensters von *GastroInt-Sim*. Zur Echtzeitdarstellung für den Trainierenden wurde eine 3D-Darstellung gewählt, die ein 3D-Modell des Magens in einem Brauntönen zeigt und den aktuell betrachteten Bereich mit einer gelblichen Schattierung versieht, die dem Schein der Lichtquelle des Endoskops nachempfunden ist. In den Abbildungen 3.19 und 3.18 ist dieser Effekt gut zu sehen.

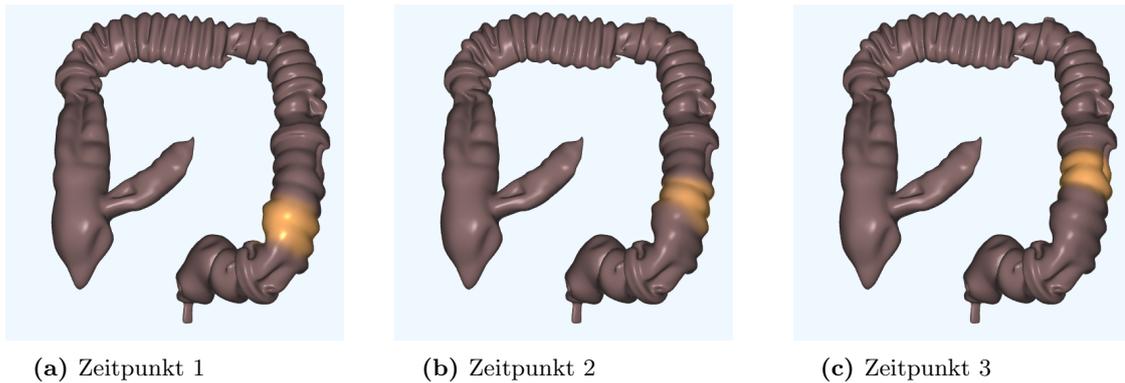


Abb. 3.19.: Darstellung des Kolons zu verschiedenen Zeitpunkten. Während der Aufnahme dieser Bilder wurde das Endoskop das Kolon entlang vorgeschoben. Der Lichtschein des Endoskops wird als gelblicher Schein dargestellt.

Idee beim Rendervorgang

Die Darstellung wird mit üblichen Methoden gerendert. Der für diesen Rendervorgang benutzte Shader (siehe A.1.1) leistet die gelbliche Einfärbung des Bereichs, den das Endoskop anstrahlt, in Abhängigkeit von den Sensorwerten nach der Methode, die in Abschnitt 3.4.1 beschrieben wurde. Die Idee hinter der Einfärbung des Lichtscheins ist im Prinzip trivial: Der Magen liegt als trianguliertes Oberflächenmesh vor. Der Shader erhält als Eingabe die Positionen aller Sensoren und deren Sensorwerte. Für jedes darzustellende Pixel des Magens muss nun lediglich die Interpolationsformel 3.1 ausgewertet werden. Die Positionen der Sensoren sind konstant, daher müssen diese nur einmal übergeben werden und können von da an verwendet werden. Die Sensorwerte dagegen ändern sich ständig und müssen daher stets neu übermittelt werden.

In dieser naiven Vorgehensweise steckt eine Menge Redundanz: Weil das Interpolationsverfahren ein *lokales* ist, sind für ein gegebenes Pixel f die Gewichte der meisten Sensoren gleich 0, weil diese zu weit von f entfernt sind. Dennoch werden bei dem naiven Ansatz stets alle Sensoren bei der Interpolation berücksichtigt, was gerade auf schwächeren Rechnern zu einer beträchtlichen Verringerung der Bildwiederholrate führen kann. Dies liegt insbesondere drin begründet, dass Speicherzugriffe innerhalb eines Shaders verglichen mit arithmetischen Operationen relativ zeitaufwändig sind, sodass diese nach Möglichkeit so selten wie möglich stattfinden sollten.

Besser wäre es, von vornherein nur die Sensoren zu berücksichtigen, deren Werte auch einen Beitrag zur Interpolation leisten. Innerhalb des *Fragmentshaders* ist diese Optimierung aber nicht durchführbar; dazu müsste für jedes Pixel bestimmt werden, welche der Sensoren nahe genug liegen, sodass sie für die Interpolation eine Rolle spielen. Diese Bestimmung benötigt aber durch die dazu notwendigen Speicherzugriffe und Berechnungen

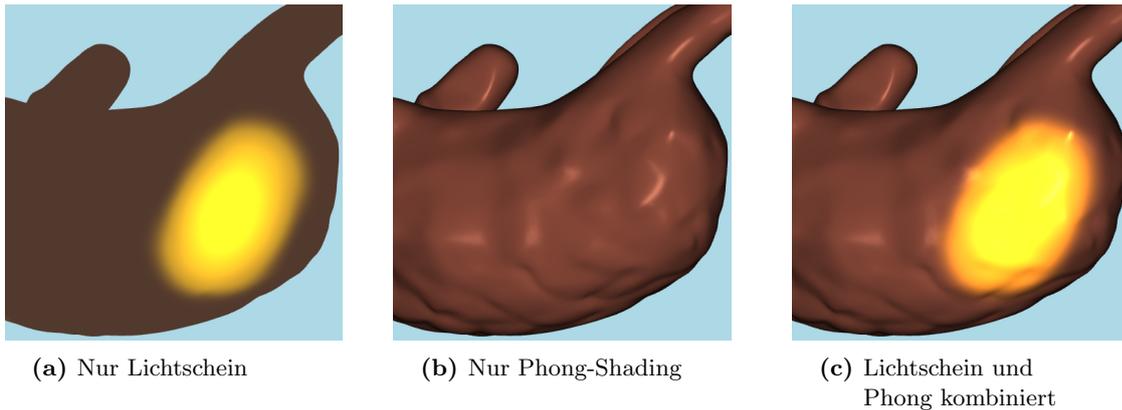


Abb. 3.20.: Endgültige Oberflächenbeleuchtung des Magens

mehr Rechenzeit als die Berechnung der Interpolation selbst, sodass sich kein Geschwindigkeitsgewinn ergäbe.

Eine andere Möglichkeit besteht darin, die Ermittlung der in Frage kommenden Sensoren nicht im *Fragment*- sondern bereits im *Geometryshader* durchzuführen, da diese Berechnungen dann auf Dreiecksbasis stattfinden. Da in aller Regel wesentlich weniger Dreiecke als Pixel berechnet werden müssen (typisch 20.000 Dreiecke gegenüber 200.000 Pixeln), werden auf diese Weise die Berechnungen der relevanten Sensoren ganz erheblich reduziert. Diese Optimierung kann noch weiter geführt werden, indem die Sensoren, die einen Einfluss auf ein einzelnes Dreieck haben, in einem Vorverarbeitungsschritt einmalig bestimmt werden. Auf diese Weise entfällt die permanente Suche nach den relevanten Sensoren, was zusätzlich Rechenzeit einspart.

Das Interpolationsresultat ist zunächst ein skalarer Wert ≥ 0 . Damit ein gelblicher Schein entsteht, wird dieser Wert im Anschluss komponentenweise mit dem Farbwert $(r, g, b) = (1, 0, 0, 8, 0, 0)$ multipliziert. Um ein plastischeres Resultat zu erzielen, wird zusätzlich zu diesem gelblichen Lichtschein noch mit einem herkömmlichen normalenvektorbasierten Phong-Shadingverfahren ein Beleuchtungswert berechnet. Beide Farbwerte werden kombiniert und dargestellt (Abb. 3.20).

3.5.2. Skripte

Das Auswahlfeld im unteren linken Bereich des Hauptfensters listet eine Anzahl von Skripten auf, die mit dem Startbutton rechts davon gestartet werden können. Eins dieser Skripte startet beispielsweise ein Laufflicht entlang der großen Krümmung des Magens, indem die dort angebrachten LEDs mit einer bestimmten Frequenz der Reihe nach ein- und ausgeschaltet werden. Dieses Laufflicht soll dem Trainierenden als Orientierung die-

nen, damit er die große Krümmung identifizieren und versuchen kann, an ihr entlang zu navigieren.

Ein anderes Skript ist dem erstgenannten sehr ähnlich, mit dem kleinen Unterschied, dass das Lauflicht nicht automatisch wandert, sondern dass eine LED so lange eingeschaltet bleibt, bis der Trainierende die durch die LED angegebene Stelle des Magens erreicht und inspiziert hat. Hat er dies geschafft, wird die LED abgeschaltet und die nächste in der Reihe wird eingeschaltet. Technisch wird dies realisiert, indem der Wert des Sensors, der sich direkt neben der LED befindet, mit einer einstellbaren Schwelle verglichen wird. Liegt der Wert für einen gewissen Zeitraum über der Schwelle, wird davon ausgegangen, dass der Trainierende die Stelle offenbar gut beobachten konnte. In diesem Fall wird mit der nächsten LED fortgefahren. Dies funktioniert, da bei der Verteilung der Sensoren stets darauf geachtet wurde, dass sich neben jeder LED zwingend ein Sensor befinden muss.

Ähnliche Skripte existieren für die kleine Krümmung des Magens und für ausgewählte oder zufällig bestimmte weitere LEDs, die nicht entlang einer der beiden Krümmungen liegen. Diese Liste ist beliebig erweiterbar, da es sich bei den Skripten um Python-Skripte handelt, die, wie in Abschnitt A.1.3 beschrieben wird, nahezu beliebig erweiterbar sind.

3.6. Weitere Arten von Sensoren

Es wurden auf Wunsch von Prof. Dr. Grund noch zwei weitere Arten von Sensoren in den Kolophanten eingebaut, die im Folgenden kurz beschrieben werden.

3.6.1. Lagesensoren

Bei einer realen Koloskopie ist es üblich, den Patienten wenn nötig „umzulagern“, d.h. ihn entlang seiner Körperachse zu drehen. Dies kann dann sinnvoll sein, wenn das Kolon aufgrund der momentanen Lage des Patienten einen für die Koloskopie ungünstigen Verlauf nimmt. Häufig kann durch Drehung des Patienten in die Seitlage eine günstigere Anordnung seines Kolons bewirkt werden.

Um dieses Manöver im Modul *GastroInt-Sim* nachstellen zu können, wurde in den Kolophanten ein Gravitationsensor eingebaut, mit dessen Hilfe die Drehung des Phantoms um seine Längsachse erkannt werden kann. Auf dem Bildschirm wird dies dadurch verdeutlicht, dass die virtuelle Darstellung des Kolophanten ebenfalls gedreht wird. Abb. 3.21a stellt diese Drehung dar.

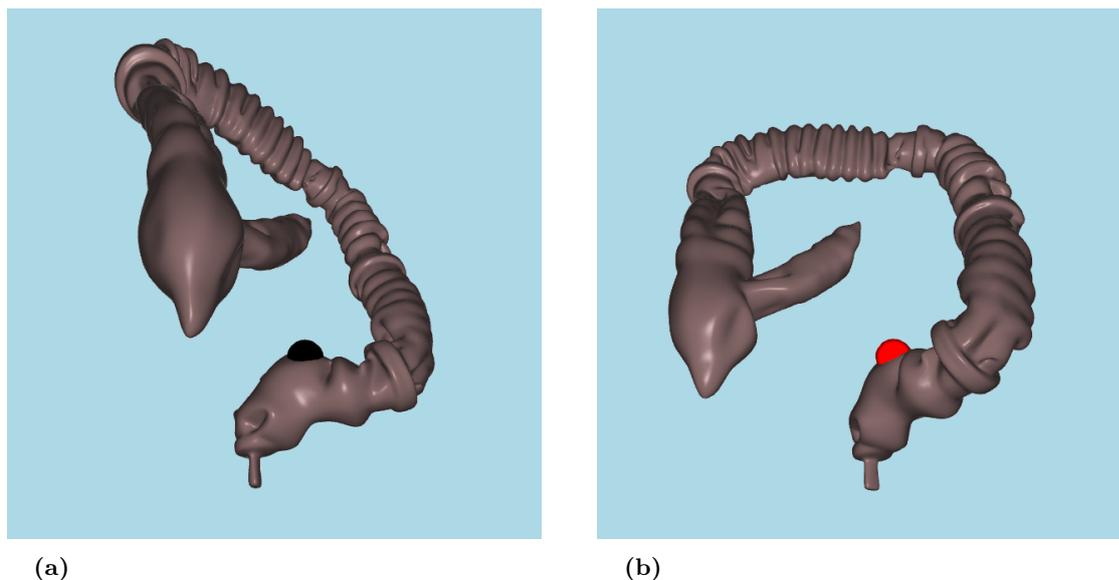


Abb. 3.21.: Darstellung der Messwerte von Lage- und Drucksensoren. a) Gekippte Darstellung des Kolophanten b) Rötliche Darstellung des Drucksensors zur Verdeutlichung von ausgeübtem mechanischem Druck auf die Prostata.

3.6.2. Drucksensoren

Bei einer Koloskopie eines männlichen Patienten wird routinemäßig eine Prostatauntersuchung vorgenommen. Mittels eines Drucksensors kann im Kolophanten überprüft werden, ob ein Trainierender diese Untersuchung durchgeführt hat. Dazu wurde direkt hinter dem Rektum ein Drucksensor eingesetzt, mit dem ein mechanischer Druck erkannt werden kann. Erst wenn der Druck, der auf diesen Sensor ausgeübt wurde, groß genug ist, wobei diese Schwelle konfigurierbar ist, gilt die Prostata als untersucht.

In der grafischen Darstellung von *GastroInt-Sim* wird der Drucksensor als einfache Kugel dargestellt, die direkt hinter dem virtuellen Rektum liegt. In Abhängigkeit von dem Druck, den der Sensor misst, wird diese Kugel eingefärbt. Dabei ist die Färbung desto rötlicher, je größer der Druck ist. In Abb. 3.21b wird diese rötliche Kugel dargestellt.

4. Gallengangssysteme

4.1. Die ERCP

„Die endoskopische retrograde Cholangiopankreatographie (ERPC) ist die Kombination einer endoskopischen und einer radiologischen Untersuchung, die die Darstellung des biliären und des pankreatischen Gangsystems durch die direkte Kanülierung der Papilla Vateri und die retrograde Kontrastmittelinjektion unter Verwendung eines Seitblickoptik-Duodenoskops ermöglicht. [...] Jedoch ist ein spezielles Training erforderlich, um die Kompetenz und die notwendigen Fertigkeiten zu erwerben und daher sollte nicht jeder Endoskopiker die ERCP durchführen, auch angesichts der Risiken dieser interventionellen Prozedur.“ [Gas03]. Nach diesem einleitenden Zitat folgt eine kurze Beschreibung des Procederes während einer ERCP. Diese Beschreibung stellt eine Zusammenfassung der Ausführungen in [Gas03] dar.

Der Eingriff beginnt mit der Intubation mit einem speziellen Endoskop, dem sog. *Seitblickduodenoskop*, welches im Gegensatz zu „normalen“ Gastroskopen (vgl. Abschnitt 3.1.1) nicht geradeaus, sondern im Winkel von 90° zur Seite blickt. Dieses wird dem Patienten analog einer Gastroskopie (vgl. Kapitel 3) oral eingeführt und durch die Speiseröhre in den Magen und von dort weiter zum *Pylorus* ins *Duodenum* gebracht. Dort wird die Spitze des Duodenoskops vorsichtig im *Absteigenden Duodenum* verhakt und das Gerät ein Stück zurückgezogen, bis es in einer möglichst begradigten Gestalt durch Magen und Duodenum verläuft, wie Abb. 4.1 zeigt. Diese Begradigung ist notwendig, um optimale Kontrolle über die Steuerung der Endoskopspitze zu erlangen und vermindert außerdem die Beschwerden für den Patienten. Durch Betätigung der Steuerräder des Duodenoskops wird die Spitze so eingestellt, dass die *Papilla Vateri*, das ist die gemeinsame Öffnung von Hauptbauchspeicheldrüsen- und -gallengang, im Folgenden kurz als „Papille“ bezeichnet, direkt und frontal angesehen wird (Abb. 4.2).

Nun kann die eigentliche ERCP durchgeführt werden. Es wird zwischen der rein *diagnostischen* und der *therapeutischen* ERCP unterschieden. Bei der *diagnostischen* ERCP wird z.B. zur Anfertigung eines sog. *Cholangiogramms* über die Papille Kontrastmittel in das Gallengangssystem injiziert, um dieses anschließend mit Röntgenstrahlung zu durchleuchten. Eine ERCP wird nach [Gas03] als *therapeutisch* definiert, wenn eine endoskopische

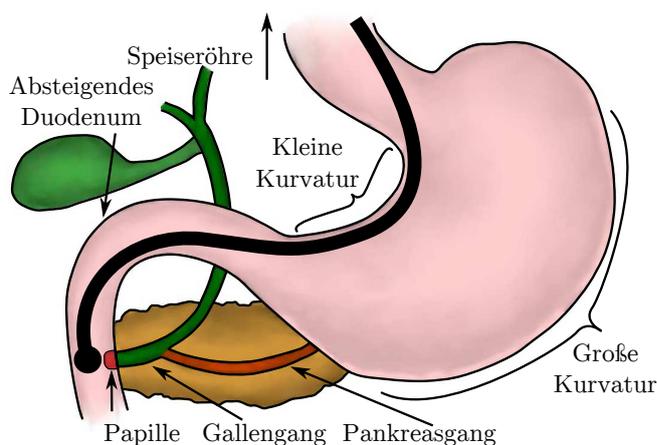


Abb. 4.1.: Begradigter Verlauf des Duodenoskops bis zur *Papilla Vateri*. Abbildung von Duygu Özmen.



Abb. 4.2.: Frontale Ansicht der *Papilla Vateri*. Quelle: J. Guntau für de.wikipedia [Public domain], aus Wikimedia Commons, entnommen am 13.05.2014.

Sphinkterotomie oder eine Drainage angestauter Gallenflüssigkeit vorgenommen wird. Bei einer Sphinkterotomie handelt es sich um die Erweiterung der in der Regel sehr schmalen natürlichen Öffnung der Papille, des sog. *Porus*, mit Hilfe eines *Papillotoms*, welches die Papille mittels Hochfrequenzchirurgie durch Lichtbögen aufschneidet, oder mit Hilfe eines sog. *Needle Knife*, mit dem die Papille mechanisch geschnitten wird. Die verwendeten Instrumente werden über den Instrumentierkanal des Duodenoskops zur Papille vorgeschoben.

Eine Sphinkterotomie wird notwendig, sobald Gallensteine entfernt werden sollen oder wenn, z.B. wegen eines Tumors, eine *Stenose*¹ der Gallenwege vorliegt, die mit einem *Stent*² behoben werden soll. Der natürliche Porus der Papille ist zu schmal, um mit den zur Steinextraktion benötigten Hilfsmitteln wie Zangen, *Dormia-Körbchen*³ oder *Ballonkathetern*⁴ oder mit *Stents* zur entsprechenden Stelle des Gallengangsystems vordringen zu können. Diese Hilfsmittel werden ebenfalls über den Instrumentierkanal des Duodenoskops vorgeschoben. Da es jenseits der Papille keine Möglichkeit gibt, die Instrumente unter direktem Sichtkontakt zu steuern, werden Eingriffe wie Stentlegung oder Steinextraktion unter permanenter Röntgenkontrolle durchgeführt.

Aufgrund ihrer Komplexität stellt die ERCP einen sehr risikoreichen Eingriff dar, wie in [Gas03] beschrieben wird: „Die ERCP hat ein größeres Potential für untersuchungsbe-

¹Verengung

²Gitterartige, röhrenförmige Gefäßstütze zur Offenhaltung von Gefäßen

³Instrument zur Extraktion von z.B. Gallensteinen

⁴Hilfsmittel zur Gefäßdilatation

dingte Komplikationen als andere endoskopische Untersuchungen des oberen Gastrointestinaltraktes. (...) Komplikationen der Endoskopie sind Nebenwirkungen intravenöser Medikationen, Sauerstoffmangel, Aspiration von Magen- oder Darminhalt und die Hämorrhagie oder Perforation bei traumatischer Passage des Endoskops. Komplikationen im Zusammenhang mit der Manipulation an der Ampulle und dem Gallengangssystem sind Pankreatitis und Sepsis“. Das Risiko bei der ERCP hängt signifikant davon ab, ob eine diagnostische oder eine therapeutische ERCP durchgeführt wird. In einer Studie von 1998 kommen Loperfido et al. zu dem Ergebnis, dass das Risiko für moderat-schwere bis sehr schwere Komplikationen bei der diagnostischen ERCP bei 1,38% liegt, während es bei der therapeutischen ERCP 5,4% beträgt [LAB⁺98]. Die Morbiditätsrate, also der Prozentsatz an tödlichen Komplikationen während des Eingriffs und bis zu 30 Tagen danach, beträgt bei der diagnostischen ERCP 0,21% und bei der therapeutischen ERCP 0,5%.

Aufgrund dieses hohen Risikos ist ein intensives Training von Arzt und Assistenz unumgänglich. Da die diagnostische ERCP ein signifikant geringeres Risiko als die therapeutische ERCP aufweist, sollten unerfahrene Ärzte nicht direkt therapeutische ERCP durchführen, sondern zunächst ihre Fähigkeiten in der diagnostischen ERCP perfektionieren. Dieser Ansatz des *Trainings am Patienten* weist jedoch erhebliche Probleme auf:

- Auch eine rein diagnostische ERCP bedeutet ein hohes Risiko für den Patienten. Das Training eines derart risikoreichen Eingriffs ist daher eindeutig als unethisch zu bezeichnen. Selbst wenn dem Anfänger ein erfahrener Endoskopiker zur Seite steht, sind die Risiken und Fehlermöglichkeiten so vielfältig, dass auch der Experte im Zweifelsfall nicht rechtzeitig eingreifen kann.
- Seit einigen Jahren steht mit der *MRCP*, der *Magnetresonanztomographie-Cholangiopankreatographie*, eine nichtinvasive, strahlungsfreie und damit vollständig risikolose Alternative bereit, die für viele Anwendungsfälle, in denen herkömmlich mit diagnostischer ERCP gearbeitet wurde, die bessere Wahl darstellt. Vor diesem Hintergrund ist die diagnostische ERCP nur noch in wenigen Fällen vertretbar.
- Aufgrund des vorherigen Punktes steht man nun also vor dem Dilemma, dass eine risikoreiche therapeutische ERCP nicht von unerfahrenen Ärzten durchgeführt werden sollte, während die deutlich weniger riskante diagnostische ERCP kaum noch durchgeführt wird. Dadurch sind die Trainingsmöglichkeiten erheblich eingeschränkt.

4.1.1. Tübinger Biliphant

Eine Alternative zum Training am Patienten stellt das Training an Phantomen und Simulatoren dar. Wie auch für die Kolo- und die Gastroskopie hat die Tübinger Arbeitsgruppe

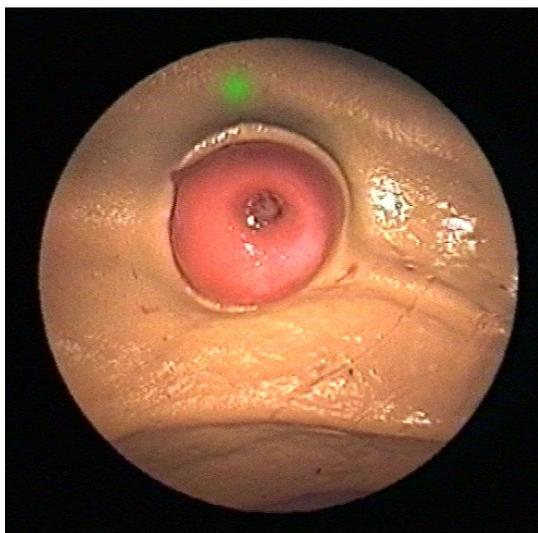


Abb. 4.3.: Frontale Ansicht der künstlichen Papille im *Biliphanten* aus Sicht des Duodenoskops. Die Papille ist die rötliche runde Struktur in der Bildmitte.

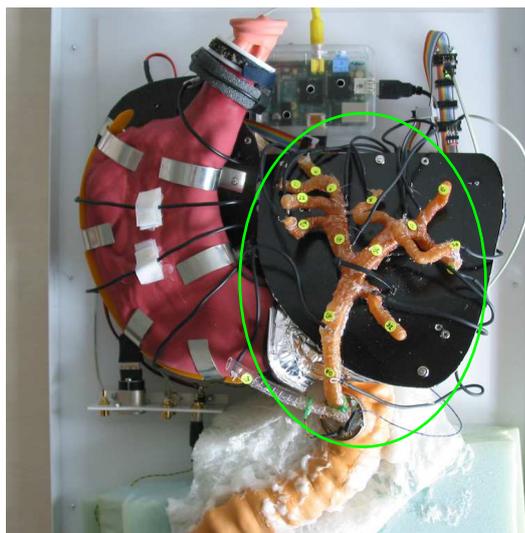


Abb. 4.4.: Blick auf den geöffneten Biliphanten. Die grün eingekreiste orangefarbene Struktur in der rechten Bildhälfte ist das künstliche Gallengangsystem.

um Prof. Dr. Grund daher im Rahmen ihres Phantomprojekts den sog. „Biliphanten“ entwickelt. Dieser besteht wie der Gastrophant (vgl. Kapitel 3) aus Kopf, Speiseröhre, Magen und Duodenum, hat aber zusätzlich in der Wand des Duodenums eine Papille und jenseits davon ein Gallengangsystem. In diesem Gallengangsystem können vielfältige therapeutische Eingriffe sehr realitätsgetreu trainiert werden [Muc05, GID⁺12, VzGA13, GIS⁺12a, VAG13b]. Abb. 4.3 zeigt ein Bild der künstlichen Papille aus Sicht eines Duodenoskops im Duodenum, Abb. 4.4 ein Bild des geöffneten Biliphanten. Die künstliche Papille besitzt in der Regel, wie die echte Papille auch, einen Porus. Dieser kann wie bei einer echten ERCP mittels Hochfrequenzchirurgie durch Lichtbögen erweitert werden, damit Instrumente in das Gallengangsystem eingebracht werden können. Analog zu den Gastro- und Kolophanten wurde an der Universität Düsseldorf zur Überwachung, Steuerung und Kontrolle des Biliphanten das Softwaremodul „X-Sim“ für die *Simtux*-Plattform entwickelt, welches im Folgenden beschrieben wird.

4.2. Anforderungen an das Softwaremodul X-Sim

Die Trainierenden am Biliphanten lassen sich im Wesentlichen in zwei Gruppen einteilen: Einerseits die Anfänger, deren Lernziel darin besteht, die durchaus komplizierte Navigation des Duodenoskops bis zur Papille, die korrekte Einstellung dieser im Bild des Duodenoskops und das anschließende Aufschneiden mit dem Papillotom zu erlernen bzw. ihre Fähigkeiten darin zu verbessern. Auf der anderen Seite sind die Fortgeschrittenen, für

die die Navigation keine Herausforderung mehr darstellt, und die daher die tatsächlichen therapeutischen Eingriffe erlernen bzw. verbessern sollen. Entsprechend dieser beiden Gruppen ergeben sich unterschiedliche Anforderungen an das Modul *X-Sim*.

Strahlenbelastung

Weil eine ERCP stets unter Röntgenkontrolle durchgeführt wird, wurde der Tübinger Biliphant so konzipiert, dass alle Bestandteile eine wohldefinierte Röntgendichte haben, die weitgehend der von menschlichem Gewebe entspricht. Während eines Trainingskurses kann die ERCP am Phantom daher wie bei einem echten Patienten unter einem C-Bogen bei permanenter Röntgenkontrolle durchgeführt werden. Dies bedeutet jedoch eine hohe Strahlenbelastung für Trainierende und Tutoren. Dies ist vor allem für junge weibliche Teilnehmerinnen, insbesondere wenn eine Schwangerschaft vorliegt, nicht vertretbar. Aktuelle Studien [GGUA13] zeigen, dass schon während einer klinischen ERCP die Strahlendosis nicht unter den Grenzwerten, die von der internationalen Strahlenschutzkommission ICRP gesetzt werden, gehalten werden kann. Insbesondere die Augenlinse wird einer stark erhöhten Strahlenbelastung ausgesetzt.

Ein Hauptmerkmal des ERCP-Trainings mit dem Biliphanten ist, dass dieser Eingriff beliebig lange und beliebig häufig trainiert werden kann. Dadurch ergibt sich aber das Problem, dass während des Trainings eine um ein Mehrfaches längere Durchleuchtungszeit als bei einer klinischen Intervention am Patienten auftritt und somit alle Grenzwerte der Strahlenexposition in nicht tolerierbarer Weise überschritten werden. Die zentrale Anforderung an das Modul *X-Sim* ist es daher, eine ERCP am Phantom einerseits so realistisch wie möglich zu gestalten, aber andererseits nicht auf Röntgenstrahlung angewiesen zu sein. Dies ist auch in logistischer Hinsicht von großem Vorteil, da dadurch die ERCP-Kurse in praktisch jeder Lokalität durchgeführt werden können und nicht auf das Vorhandensein eines C-Bogens angewiesen sind.

Um diese beiden scheinbar konträren Ziele zu erreichen, wurde in das Modul *X-Sim* als zentrale Komponente eine *virtuelle Röntgensimulation* implementiert, die am Rechner ein Röntgenbild erzeugen kann, welches dem Bild entspricht, das ein realer C-Bogen in der gegebenen Situation liefern würde. Damit haben die Trainierenden die Möglichkeit, Eingriffe ohne jede Strahlenbelastung vornehmen und gleichzeitig deren Ablauf unter simulierter Röntgendurchleuchtung überwachen und diesen daher ohne zeitliche Einschränkungen trainieren zu können. Die Röntgensimulation wird für Anfänger und Fortgeschrittene entsprechend den jeweiligen Anforderungen auf unterschiedliche Weise berechnet. Die Abschnitte 4.4 und 4.6 gehen näher auf diese Anforderungen und ihre Umsetzungen ein.

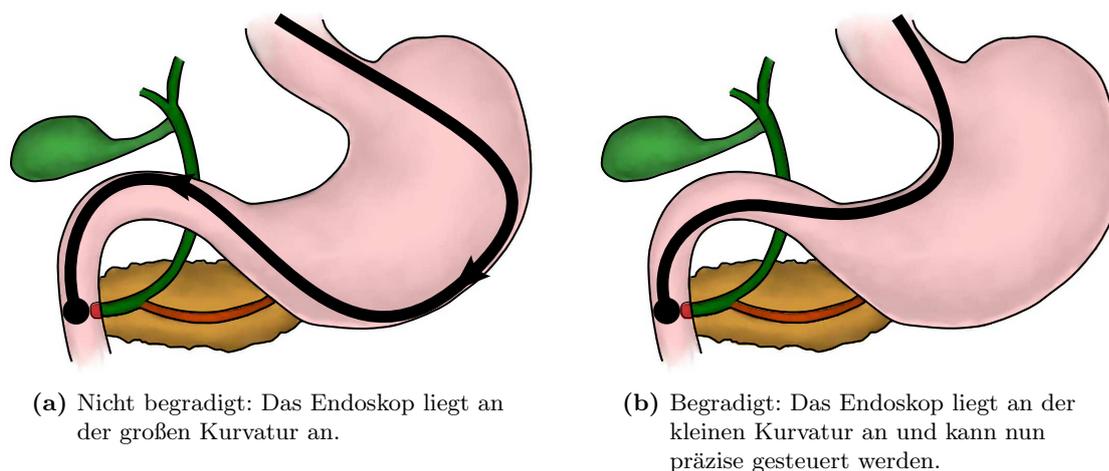


Abb. 4.5.: Begradigung. Abbildungen von Duygu Özmen.

Bedienung

Bei einer Gastro- oder Koloskopie besteht die Bedienung der Instrumente im Wesentlichen aus der korrekten Steuerung des Endoskops. Bei einer ERCP muss zusätzlich noch die Bedienung des C-Bogens erlernt werden, welche üblicherweise über ein Kontrollpult erfolgt. Dazu gehört einerseits das Erlernen der Funktionen der einzelnen Knöpfe des Kontrollpults, andererseits aber auch die Entwicklung einer räumlichen Vorstellung der Lage des C-Bogens und des Gallengangsystems und vor allem das Wissen, welche Aktionen des C-Bogens (Rotationen, Vergrößerungen, Verschiebungen, etc.) durchgeführt werden müssen, damit das Röntgenbild auf die gewünschte Art und Weise reagiert. Damit Anfänger diese Steuerung erlernen können, soll die Bedienung des Moduls *X-Sim* daher möglichst nahe an der Realität angelehnt sein. Abschnitt 4.7.2 beschreibt, wie diese Anforderung umgesetzt wurde.

Begradigungsanzeige

Die Präzision der Steuerung des Dudenoskops vor der Papille hängt stark von dessen Lage in Magen und Duodenum ab. Wurde es eben erst vor die Papille navigiert, liegt es in der Regel in einer weiten Schlaufe an der großen Kurvatur des Magens an (Abb. 4.5a). Eine präzise Steuerung der Spitze des Dudenoskops ist damit erschwert. Besser gelingt diese, wenn das Endoskop so geradlinig wie möglich durch den Magen und das Duodenum verläuft. Um dies zu erreichen, sollte der Trainierende vor allen weiteren Eingriffen das sogenannte „Begradigungsmanöver“ durchführen. Zu diesem Zweck kippt er die Spitze des Dudenoskops so ab, dass sie die Wand des Duodenums berührt und sich daran „verhakt“.

Anschließend kann das Endoskop ein Stück weit wieder aus dem Mund herausgezogen werden, wodurch es, da das Ende ja im Duodenum fixiert ist, begradigt wird. Der Magen-Darm-Trakt ist nicht beliebig flexibel, daher ist eine vollständige Begradigung im Sinne einer geraden Linie nicht möglich. Der bestmöglich erreichbare Zustand ist dann erreicht, wenn das Endoskop an der kleinen Krümmung des Magens anliegt (Abb. 4.5b). Jede weitere Begradigung müsste den Magen insgesamt verschieben, was jedoch bei einem realen Magen aufgrund der Fixierung durch Muskelstränge und im Phantom aufgrund der dort ebenfalls vorhandenen Fixierung nicht möglich ist. Das System soll dieses Begradigungsmanöver auswerten und dem Trainierenden entsprechende Rückmeldung geben können. In Abschnitt 4.8 wird die Umsetzung dieser Auswertung beschrieben.

Optimale Einstellung der Papille

Eine sehr wichtige Voraussetzung für eine erfolgreiche ERCP besteht in der richtigen Ausrichtung des Duodenoskops beim Blick auf die Papille. Eine ideale sog. *Kanülierungsposition* ist dann gegeben, wenn der Porus der Papille zentral oder geringfügig links der Mitte des Bildes gelegen ist. Abbildung 4.3 zeigt eine Einstellung der Papille, die laut Prof. Dr. Grund optimal ist. Anfänger, die noch ungeübt in der Bedienung des Duodenoskops sind, tun sich jedoch häufig schwer mit der korrekten Einstellung der Papille. Bei verschiedenen Besuchen der Trainingskurse, welche mit den Tübinger Phantomen durchgeführt wurden, konnte ich feststellen, dass auch die anwesenden Tutoren nicht immer auf eine solche gute Kanülierungsposition achten und die Trainingsteilnehmer nur selten auf wenig ideale Blickwinkel hinweisen. Am Institut für Informatik der Universität Düsseldorf wurde daher im Rahmen einer Bachelorarbeit ein Verfahren entwickelt, das ein Duodenoskopbild analysiert und in Hinblick auf die Güte der Einstellung bewertet [Özm12].

In naher Zukunft soll dieses Verfahren in das bestehende *X-Sim*-Modul integriert werden, sodass die Bewertung der Papillenanlage automatisch durchgeführt werden kann. Damit wird ermöglicht, dass jeder Trainingsteilnehmer das Einstellen der Papille so lange üben kann, bis diese optimal im Bild liegt.

4.3. Simulation eines Röntgenbildes

Wie in der Einleitung beschrieben wurde, besteht die zentrale Idee des Moduls *X-Sim* darin, am Rechner ein Röntgenbild zu simulieren. Im Folgenden werden zunächst einige physikalische Betrachtungen des realen Röntgendurchleuchtungsvorgangs skizziert und anschließend die Umsetzung der Simulation im Rechner beschrieben. Die physikalischen Grundlagen sind [SB02] und [SM13] entnommen.

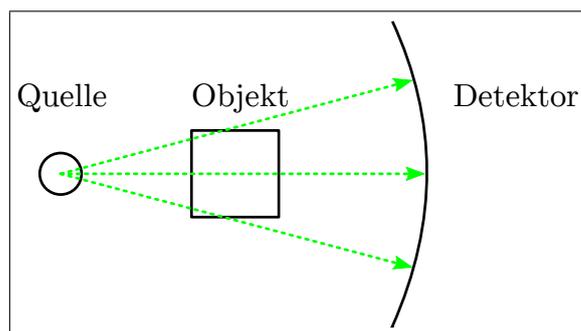


Abb. 4.6.: Röntgenvorgang: Eine Quelle sendet Röntgenstrahlung aus, die das Objekt durchdringt, dabei abgeschwächt wird und am Schluss vom Detektor gemessen wird.

4.3.1. Physikalische Betrachtungen

Ein Röntgengerät erzeugt Bilder durch Emission von Röntgenstrahlung, deren Absorption durch das durchleuchtete Objekt mittels eines Detektors gemessen wird (Abb. 4.6). Vereinfacht gesagt wird von einer Röntgenquelle Röntgenstrahlung der Intensität I_0 emittiert. Diese durchdringt das Objekt, wird dabei abgeschwächt und erreicht zuletzt die Pixel des Detektors, woraufhin in diesen die Intensität I gemessen wird. Die Abschwächung, die ein Röntgenstrahl bei der Durchdringung eines Mediums erfährt, wird durch den sog. *linearen Abschwächungskoeffizienten*, der im Folgenden mit „LAC“ abgekürzt wird, beschrieben, welcher als Maß für die Transparenz des Mediums für Röntgenstrahlung interpretiert werden kann. Nach [SM13] lässt sich der Bildgebungsprozess durch eine Kombination von primären und sekundären Effekten beschreiben. Der wesentlich für die Erzeugung eines Röntgenbildes verantwortliche Prozess ist die Abschwächung von *Primärphotonen* während der Durchdringung des Objekts. Unter *Primärphotonen* verstehen die Autoren von [SM13] solche Röntgenphotonen, die auf direktem Wege durch das Objekt zum Detektor wandern, ohne dabei gestreut und abgelenkt zu werden. Nach dem Lambert-Beerschen Gesetz gilt für die Intensität dieser Primärstrahlung, nachdem sie ein homogenes Material der Dicke d mit konstantem LAC μ durchdrungen hat:

$$I = I_0 \cdot \exp(-\mu \cdot d) \quad (4.1)$$

Reale Objekte weisen in der Regel keine homogene Röntgendichte auf und haben damit keinen konstanten LAC. Vielmehr muss für solche Medien der LAC $\mu(r)$ in Abhängigkeit des Ortes r angegeben werden, da er potentiell an jedem Punkt des durchdrungenen Materials unterschiedlich sein kann. Daher muss über die gesamte Strecke s , auf der die Photonen das Medium durchdringen, integriert werden:

$$I = I_0 \cdot \exp\left(-\int_s \mu(t) dt\right) \quad (4.2)$$

Neben diesem primären Prozess gibt es noch weitere Effekte, die die Erzeugung des Röntgenbildes beeinflussen. Diese sind:

- Einige Photonen werden auf ihrem Weg von der Quelle zum Detektor gestreut, sodass sie nicht geradlinig, sondern an einer anderen Stelle auf den Detektor treffen.
- Röntgenstrahlung wird in der Regel nicht *monochromatisch*, also mit genau einer Frequenz, sondern *polychromatisch* mit einer gewissen Frequenzverteilung emittiert. Photonen mit geringerer Energie werden bei der Durchdringung eines Objekts mit einer größeren Wahrscheinlichkeit abgeschwächt als solche mit höherer Energie, sodass sich der Mittelwert der Frequenzverteilung in Richtung höherer Energie verschiebt. Dieser Effekt wird als *Strahlaufhärtung* bezeichnet.
- Schleierbildung. Dieser Effekt beschreibt ein „Auslaufen“ des Signals eines Detektorpixels auf die umliegenden Detektorpixel.

Die Aufnahme eines klassischen zweidimensionalen Röntgenbildes lässt sich durch Gleichung 4.2 sowie der Berücksichtigung der sekundären Effekte gut beschreiben. Es ist aber nicht möglich, diese Berechnung zu invertieren: Aus einem gegebenen Röntgenbild lässt sich die räumliche Verteilung $\mu(r)$ nicht rekonstruieren, weil in einem Pixel des Röntgenbildes nur die Information des kompletten Integrals über die gesamte Durchdringungsstrecke enthalten, die genaue Verteilung $\mu(r)$ entlang des Strahls aber unbekannt ist.

Mit der *Computertomographie (CT)* steht ein Verfahren zur Verfügung, mit welchem die Verteilung $\mu(r)$ vollständig rekonstruiert werden kann. Hierbei wird das Konzept derart modifiziert, dass Röntgenquelle und -detektor um das aufzunehmende Objekt herum rotiert werden, wobei eine Serie von Bildern aus unterschiedlichen Blickwinkeln produziert wird. Aus diesen verschiedenen Aufnahmen lässt sich durch Anwendung der *Inversen Radon-Transformation* die vollständige Verteilung $\mu(r)$ rekonstruieren, sodass anstelle eines zweidimensionalen Bildes ein dreidimensionaler sog. *Volumendatensatz* berechnet werden kann. Dieser wird in *Hounsfield-Einheiten*⁵ dargestellt. Die Hounsfield-Skala stellt letztendlich eine Normierung des LAC auf die Werte von Luft und Wasser dar. Die Umrechnungsvorschrift h von der üblichen Einheit des LAC $[\frac{1}{cm}]$ zu $[HU]$ ist gegeben durch:

$$h\left(\mu_{\text{Gewebe}}[\frac{1}{cm}]\right) := \frac{\mu_{\text{Gewebe}} - \mu_{\text{Wasser}}}{\mu_{\text{Wasser}}} \cdot 1000 [HU] \quad (4.3)$$

Dadurch wird eine Normierung der Messwerte erreicht, nach der die Werte für Luft $-1000 HU$ und für Wasser $0 HU$ betragen. Diese Skala ist durch diese Normierung wei-

⁵Nach Sir Godfrey Hounsfield (1919 - 2004)

testgehend von der Empfindlichkeit der Detektoren und der Intensität I_0 der emittierten Strahlung unabhängig.

Der so erhaltene Volumendatensatz stellt eine diskretisierte und auf Hounsfield-Einheiten normierte Repräsentation der kontinuierlichen Verteilung des LAC des durchleuchteten Objekts dar. Ein typisches CT bietet eine Ortsauflösung von deutlich weniger als 1 mm und eine Wertauflösung von 12 Bit, also 4096 verschiedenen Intensitätswerten. Die einzelnen diskreten Elemente eines solchen Datensatzes werden mit dem Kunstwort *Voxel* (von *Volume Pixel*) bezeichnet.

4.3.2. Berechnung eines Röntgenbildes

Gleichung 4.2 beschreibt im Prinzip, sekundäre Effekte außen vorgelassen, wie ausgehend von einer bekannten räumlichen Verteilung $\mu(r)$ ein Röntgenbild entsteht. Nach [SM13] können die Gleichungen, die die Erzeugung eines Röntgenbildes in der Realität beschreiben, auch dazu verwendet werden, ein künstliches Röntgenbild, eine sog. *Digitally Reconstructed Radiography*, aus einem Volumendatensatz zu berechnen. Dies ist möglich, da der Volumendatensatz eine – bis auf die Einheiten – diskretisierte Repräsentation des ursprünglichen LAC ist. Mit Hilfe einer Transferfunktion lässt sich der Datensatz von $[HU]$ in $[\frac{1}{cm}]$ umwandeln. Diese Transferfunktion hängt von den Parametern des Röntgenapparats ab. Die Autoren von [SM13] ermitteln diese empirisch anhand von Messwerten von CT-Phantomen mit bekannter wohldefinierter Röntgendichte. Durch Anwendung der Transferfunktion auf den Volumendatensatz erhält man bis auf Diskretisierung die ursprüngliche Verteilung $\mu(r)$ des LAC.

Mittels *Raycasting*, welches in Abschnitt 4.3.3 beschrieben wird, lässt sich nun ausgehend von diesem Volumendatensatz ein künstliches Röntgenbild erzeugen, wie in [SM13] ausführlich beschrieben wird. Dazu werden dieselben primären und sekundären Effekte, wie sie auch in der Realität bei der Erzeugung eines Röntgenbildes auftreten, im Computer berechnet. In [SM13] werden Resultate mit und ohne Berechnung der sekundären Effekte verglichen. Der Vergleich zeigt, dass die Ergebnisse zwar im Detail unterschiedlich, aber im Wesentlichen ähnlich sind.

Für die Röntgensimulation des Biliphanten muss das Ergebnis nicht physikalisch exakt, sondern lediglich so plausibel sein, dass ein Röntgenbild täuschend echt angenähert wird. Daher kann für den vorliegenden Anwendungszweck auf die Berechnung der sekundären Effekte verzichtet werden, wodurch sich die Berechnung des Bildes auf die Lösung des Integrals

$$R(x, y) = I_0 \cdot \exp \left[- \int_{\sigma(x,y)} V'(t) dt \right] \quad (4.4)$$

reduziert. Dabei gibt t die Parametrisierung des virtuellen Röntgenstrahls σ an, der ausgehend von der Röntgenquelle durch das Pixel $R(x, y)$ der Detektorfläche „stößt“ (vgl. Abb. 4.7) und V' das mittels einer Transferfunktion von Hounsfield-Einheiten zurück in die übliche Einheit $[\frac{1}{cm}]$ des LAC gerechnete Volumen ist. Es hat sich gezeigt, dass für die Transferfunktion die einfache Abbildung

$$h(\mu_{\text{Gewebe}}[HU]) := \mu_{\text{Gewebe}} \left[\frac{1}{cm} \right] \quad (4.5)$$

verwendet werden kann, mit der $[HU]$ und $[\frac{1}{cm}]$ einfach gleichgesetzt werden. Bei der Kontrast- und Helligkeitsberechnung des endgültigen Bildes muss dann nur noch mit geeigneten, heuristisch festgelegten Konstanten multipliziert werden, damit der Dynamikbereich des berechneten Bildes dem eines realen Röntgenbildes entspricht.

4.3.3. Umsetzung im Rechner

Wie in Anhang A.1.2 beschrieben wird, können Rendervorgänge in der Engine *Taraxacum* mittels der Programmierung von Shadern definiert und kontrolliert werden. Ein solcher Shader wurde für die Erzeugung der simulierten Röntgenbilder entwickelt. Die Idee hinter diesem Rendervorgang ist nicht neu und orientiert sich stark an [Vie07]: Der CT-Volumendatensatz des Biliphanten wird als 3D-Textur in den Speicher der Grafikkarte des Rechners geladen und dann per Raycasting dargestellt.

Das Raycastingverfahren

Das Raycastingverfahren ist ein bekannter Algorithmus zur Erzeugung zweidimensionaler Bilder aus dreidimensionalen Daten. Dabei ist zunächst einmal offen, um welche Arten von Daten es sich handelt. Es können triangulierte Oberflächen genauso dargestellt werden wie Punktwolken oder Volumendatensätze in Form von Voxelmengen oder auch abstrakte, nur durch mathematische Gleichungen definierte Mengen. Im Folgenden wird das Raycastingverfahren für Voxelmengen skizziert:

Gegeben sei eine virtuelle Kamera $C = (p, \alpha, n, \rho, f, u)$. Dabei ist

- p der *Augpunkt* oder das *Projektionszentrum* der Kamera
- α der *Öffnungswinkel* der Kamera in horizontaler Richtung
- n die *Entfernung* vom Projektionszentrum zur Projektionsebene
- ρ das *Seitenverhältnis* $\frac{\text{Bildbreite}}{\text{Bildhoehe}}$

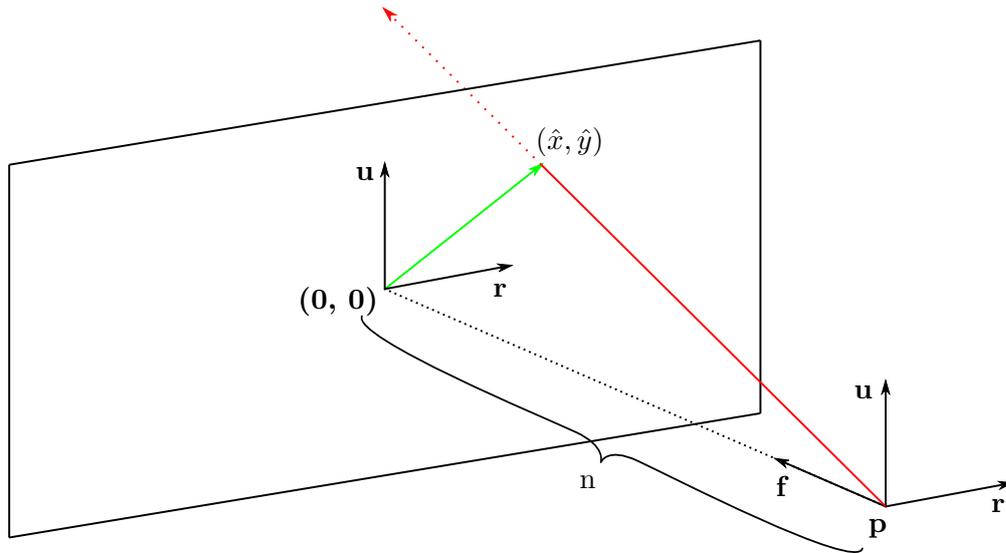


Abb. 4.7.: Skizze der geometrischen Verhältnisse beim Raycasting. Ausgehend vom Projektionszentrum p der Kamera wird ein Sehstrahl (rot) losgeschickt, der die Projektionsebene im Punkt (\hat{x}, \hat{y}) schneidet. Der Bildmittelpunkt hat die Pixelkoordinate $(0, 0)$.

- f der Vektor, der in Sichtrichtung der Kamera zeigt („Vorwärtsvektor“)
- u der Vektor, der bezüglich der Kamera nach oben zeigt („Aufwärtsvektor“).

Die Idee hinter dem Raycasting besteht darin, das zu erzeugende Bild pixelweise abzutasten und für jedes Pixel $P_{(x,y)}$ einen Farbwert zu berechnen. Dazu wird für $P_{(x,y)}$ ein sog. *Sehstrahl*, das ist eine Halbgerade, die im Projektionszentrum der Kamera beginnt und die Projektionsebene an der Stelle (x, y) „durchstößt“, betrachtet. Dieser Sehstrahl wird durch die komplette darzustellende Szene hindurch verfolgt. Abbildung 4.7 liefert eine Skizze zu diesem Verfahren. Bei jedem Schnitt des Sehstrahls mit einem Voxel wird ermittelt, wie dieses Voxel zu der Farbe, die für $P_{(x,y)}$ ermittelt werden soll, beiträgt. Dieser Sehstrahl lässt sich durchaus als Analogon zu einem Röntgenstrahl während der Aufnahme eines echten Röntgenbildes verstehen.

Mit dem Raycasting-Ansatz kann nun anhand eines gegebenen Volumens V ein simuliertes Röntgenbild berechnet werden, indem für jeden Sehstrahl das Integral aus Gleichung 4.4 berechnet wird. Da V diskretisiert ist, kann anstelle des Integrals die diskrete Summe

$$R(x, y) = I_0 \cdot \exp \left[\sum_{i=0}^n V(p + i \cdot f) \right] \quad (4.6)$$

berechnet werden, wobei p und f Augpunkt und Vorwärtsvektor der virtuellen Kamera sind. Abb. 4.8 zeigt den wesentlichen Teil des Fragmentshaders, der mittels Raycasting das simulierte Röntgenbild berechnet, als Pseudocode.

```

1 while(true) {
2     if (left_volume(pos))
3         break;
4
5     integral += fromHU(get_intensity(pos))
6     pos += ray.dir;
7 }
8
9 I = (1 - exp(-integral) * contrast;
10 color = vec4(brightness - I, brightness - I, brightness - I, 1.0);

```

Abb. 4.8.: Shadercode zur Röntgendarstellung. `contrast` und `brightness` sind Konstanten, mit denen Helligkeit und Kontrast des Röntgenbildes eingestellt werden.

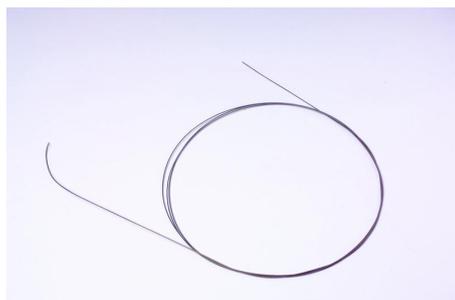


(a) Echtes Röntgenbild. Die dunkle längliche Struktur stellt das Endoskop dar und sollte hier ignoriert werden.

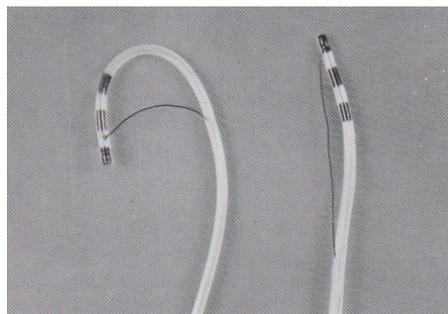
(b) Simuliertes Röntgenbild ohne eingeführtes Endoskop oder sonstige Instrumente.

Abb. 4.9.: Vergleich von echtem und simuliertem Röntgenbild

Abb. 4.9 zeigt eine Gegenüberstellung eines echten Röntgenbildes mit einem simulierten Röntgenbild. Das echte Röntgenbild (Abb. 4.9a) und der Volumendatensatz wurden nicht von demselben Phantom aufgenommen, weshalb sich „anatomische“ Unterschiede ergeben. Davon abgesehen ähneln sich beide Bilder aber qualitativ in hohem Maße. Dies zeigt zumindest exemplarisch, dass das vorgestellte Verfahren nicht nur theoretisch plausibel ist, sondern sich auch in der Praxis bewährt. Das echte Röntgenbild wirkt schärfer als das leicht verwaschen aussehende simulierte Bild, was aber kein grundsätzliches Problem der Berechnung ist. Vielmehr liegt dies schlicht an der verhältnismäßig geringen Auflösung des zugrundeliegenden Volumendatensatzes von ca. 1 mm^3 pro Voxel. Die Berechnung des Röntgenbildes beansprucht selbst auf Mittelklasse-Notebooks nicht viel Zeit und kann daher in Echtzeit laufend neu berechnet werden. Beispielsweise sind mit einem Notebook mit einer Grafikkarte des Typs *GeForce GT 555M* von NVIDIA mindestens 20 Bilder pro Sekunde möglich.



(a) ©Fraunhofer IPT



(b) Aus: Ludwig Demling,
Endoskopisch retrograde
Cholangio-Pankreatikographie:
ERCP [DKR79]

Abb. 4.10.: In den Anfängerkursen kommen nur Sondierungsdrähte (a) und Papillotome (b) zum Einsatz.

Vorder- und Hintergrund

Der überwiegende Teil des Phantoms ist relativ statisch, da Magen, Duodenum und Gallengänge zwar aus flexiblem Latex bestehen, dieses jedoch in einen nur wenig elastischen Schaumstoff eingebettet ist, sodass sich die Organe nur sehr eingeschränkt bewegen können. Es ist daher plausibel, für die Simulation dieser Organe auf einen einmalig aufgenommenen statischen CT-Datensatz zurückzugreifen. Während der Eingriffe werden jedoch Instrumente in Magen, Duodenum und Gallengänge eingeführt, deren Lage im Raum natürlich mitnichten statisch, sondern hochdynamisch ist. Daher wird die oben beschriebene Erzeugung des Röntgenbildes per Raycasting nur für die statischen Anteile des Bildes durchgeführt und für die Darstellung der dynamischen Instrumente ein anderes Verfahren gewählt. Die statischen Teile werden im weiteren Verlauf dieses Kapitels als der „Hintergrund“ bezeichnet werden. Hintergrund und Instrument werden am Schluss der Berechnungen zu einem gemeinsamen Bild zusammengefügt. Da die Lage der Instrumente prinzipbedingt nicht a priori bekannt ist, müssen diese zunächst einmal detektiert werden. Um diese Detektion zu ermöglichen, wurden zwei verschiedene Arten von Sensoren entwickelt, welche im Folgenden beschrieben werden.

4.4. Konzeption der Sensorik für die Anfängerphantome

In den Anfängerkursen kommen keine Instrumente komplexer Gestalt wie *Dormia-Körbchen*, *Biopsiezangen* oder *Ballonkatheter* zum Einsatz. Die einzigen Instrumente, die hier üblicherweise eingesetzt werden, sind *Sondierungsdrähte* und *Papillotome* (Abb. 4.10). Da sowohl Sondierungsdrähte als auch Papillotome aus einem zumindest für Wechselstrom bei $\approx 1kHz$ geringfügig leitfähigen Material bestehen, liegt es nahe, die elektri-

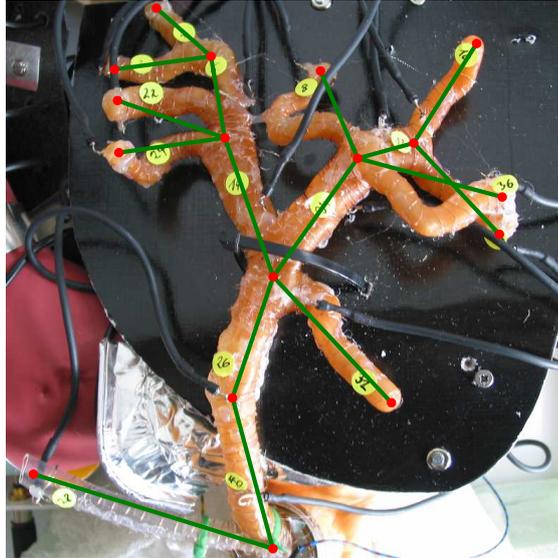


Abb. 4.11.: Skizze der Definition der Gallengangsegmente. Die grünen Linien markieren die einzelnen Segmente, die roten Punkte definieren jeweils Beginn und Ende eines neuen Segments.

sehen Eigenschaften bei der Konzeption der Sensorik auszunutzen, konkret die elektrische *Kapazität*. Dazu wurde die Oberfläche der einzelnen Gallengangsegmente auf der Außenseite mit überlappenden Elektroden überdeckt. Der Begriff „Gallengangsegment“ muss an dieser Stelle geklärt werden:

Definition 4.1. *Ein Gallengangsegment ist genau der Abschnitt des gesamten Gallenbaums, der durch eine einzelne Elektrode und damit einen Sensor abgedeckt wird. Da die Elektroden überlappend angeordnet sind, werden Anfangs- und Endpunkte eines Gallengangsegments definiert als die Punkte auf halbem Wege der Überlappungszonen an seinem Anfang bzw. Ende. In Abb. 4.11 sind die Segmente linienhaft mit grüner Farbe eingezeichnet.*

Wird nun ein Instrument, z.B. ein Sondierungsdraht, in ein Gallengangsegment eingeführt, kann die elektrotechnische Konfiguration in erster Näherung als Zylinderkondensator beschrieben werden. Voraussetzung dafür ist, dass das Gallengangsegment annähernd zylindrisch ist und keine starke Krümmung aufweist. Außerdem muss das Instrument geradlinig, d.h. ohne eine Schlaufe oder einen Knick zu bilden, sowie zentral durch das Segment hindurch laufen. Abb. 4.12 zeigt eine Skizze dieser idealisierten Situation.

Für die elektrische Kapazität eines Zylinderkondensators der Länge l , des Innenradius R_1 und des Außenradius R_2 gilt:

$$C = 2\pi\epsilon_0\epsilon_r \cdot \frac{l}{\ln\left(\frac{R_2}{R_1}\right)}$$

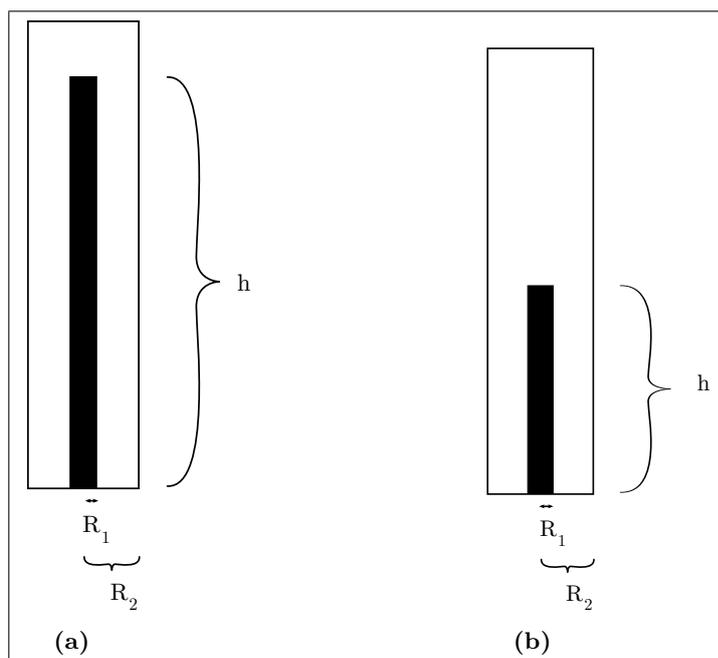


Abb. 4.12.: Idealisierter Zylinderkondensator: Der eingeführte Draht bildet die eine Elektrode, die äußere Ummantelung des Gallengangsegments die andere. Es werden zwei verschiedene effektive Längen h dargestellt.

Wird der Vorschub des Sondierungsdrahts in das Gallengangsegment vergrößert oder verkleinert, ändert sich, Randeffekte an der Spitze des Instruments außen vorgelassen, die effektive Länge l des Kondensators. Alle anderen Parameter sind konstant. Damit ist die Kapazität C direkt proportional zu der Länge l des in das Segment eingeführten Teils des Instruments.

Der Vorschub des Instruments in ein gegebenes Gallengangsegment kann also unter idealisierten Annahmen durch Messung der Kapazität bestimmt werden. Um diese zu messen, wird an das Duodenoskop ein niederfrequentes Signal angelegt, das auf das im Instrumentierkanal befindliche Instrument und von dort auf die Elektroden überkoppelt. Die Signalstärke an den einzelnen Elektroden ist proportional zur Koppelkapazität und damit, wie oben für den idealisierten Fall beschrieben, zur Länge des eingeführten Teils des Instruments.

4.4.1. Kalibrierung der Sensoren

Die kapazitiven Sensoren liefern nach der Aufbereitung der Messwerte in ihren Mikroprozessoren zunächst wenig aussagekräftige 16-Bit-Ganzzahlwerte. Es ist zunächst nicht möglich, anhand dieser die Länge des in das Gallengangsegment eingeführten Teils des Instruments zu ermitteln. Um dies zu ermöglichen, muss bekannt sein, in welchem Intervall die Sensorwerte typischerweise liegen, wenn ein bestimmter Vorschub des Instru-

ments gegeben ist. Die Sensorwerte müssen also kalibriert werden. Um die Kalibrierung vorzunehmen, wird ein Instrument vollständig in das entsprechende Gallengangsegment eingeführt und dann der Wert $v_{i,max}$ gemessen, den der Sensor S_i in dieser Situation liefert. Diese Kalibrierung wird einmalig für jedes S_i vorgenommen und fest abgespeichert, sodass bei der nächsten Benutzung des Phantoms keine Kalibrierung mehr durchgeführt werden muss. Wird dann zu einem späteren Zeitpunkt im laufenden Betrieb ein Sensorwert v_i gemessen, kann dieser durch die Division $\frac{v_i}{v_{i,max}}$ auf das Intervall $[0, 1]$ normiert werden. Dies ist wichtig, damit die Werte verschiedener Sensoren untereinander vergleichbar sind.

Normierung der Kalibrierungswerte

Sofern bei der nächsten Benutzung ein Endoskop eines anderen Herstellers oder ein Instrument eines anderen Typs eingesetzt wird, stimmen die Kalibrierungswerte im Allgemeinen nicht mehr. Grund dafür sind kleine Unterschiede im elektrischen Verhalten der Geräte. Durch diese Geräteabhängigkeit geht der Vorteil, dass die Kalibrierung nur einmal durchgeführt werden muss, verloren. Dies kann jedoch vermieden werden: Dazu wird neben der Kalibrierung und Normierung der **Sensorwerte** noch eine Normierung der **Kalibrierungswerte** $v_{i,max}$ selbst durchgeführt.

Dazu wird ein Sensor S_c , in der Regel der von der Papille aus gesehen erste, ausgewählt. Die Kalibrierungswerte $v_{i,max}$ aller Sensoren werden dann auf den Kalibrierungswert $v_{c,max}$ von S_c bezogen, indem sie so normiert werden, dass $v_{c,max}$ den Wert 1 erhält. Dazu muss nur der Faktor $f_i := \frac{v_{i,max}}{v_{c,max}}$ der Kalibrierungswerte $v_{i,max}$ aller anderen Sensoren S_i bestimmt und anstelle von $v_{i,max}$ abgespeichert werden.

Da ein eingeführtes Instrument den Sensor S_c aufgrund dessen prominenter Lage sehr schnell passiert, ist bereits kurz nach dem Einführen des Instruments in die Gallengänge dessen Maximalwert bekannt. Von da an kann für jeden weiteren Sensor S_i einfach ein dynamischer Kalibrierungswert berechnet werden, indem dessen Faktor f_i mit dem aktuell von S_c gemessenen Wert v_c multipliziert wird. Anschließend kann der aktuell gemessene Wert v_i eines Sensors S_i kalibriert und auf das Intervall $[0, 1]$ normiert werden:

$$\hat{v}_i = \frac{v_i}{f_i \cdot v_c} \quad (4.7)$$

Der große Vorteil dieser Vorgehensweise besteht darin, dass sich, wenn ein anderes Instrument mit anderem elektrischen Verhalten eingeführt wird, der Wert des Sensors S_c ändert, wodurch sich auch der Kalibrierungswert von S_i ändert. Dadurch reagiert die Kalibrierung dynamisch auf die jeweilig eingesetzten Geräte, ohne dass für jede Gerätekombination eigene Kalibrierungen durchgeführt werden müssen.

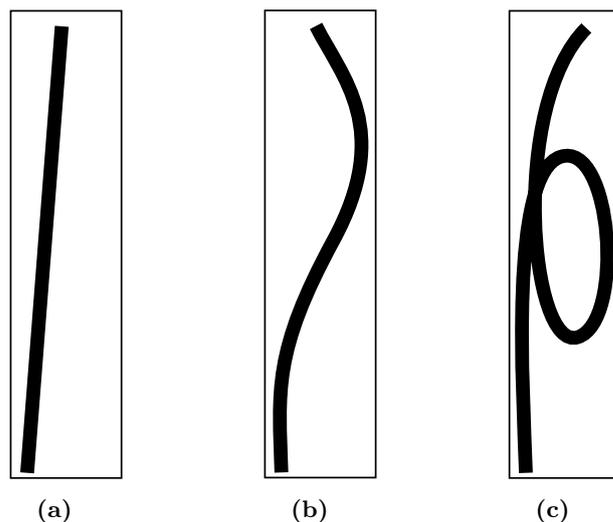


Abb. 4.13.: Die Verläufe des Instruments weichen in der Realität von der idealisierten Vorstellung des Zylinderkondensators ab.

4.4.2. Probleme in der Praxis

In der Praxis treten in Zusammenhang mit dieser Sensorik einige Probleme auf:

- In der idealisierten Modellierung liegt das Instrument stets genau zentral in dem als zylindrisch angenommenen Gallengangsegment. Das eingeführte Instrument liegt aber in der Realität in den seltensten Fällen genau auf diese Weise. Vielmehr ist der Verlauf in der Praxis eher so, wie in Abb. 4.13 skizziert wird: Es liegt in der Regel näher an einer Seitenwand des Gallengangsegments als an den anderen, kann diagonal verlaufen oder sogar „hin und her“, wie Abb. 4.13b zeigt. Die Kapazität eines nicht-konzentrischen Zylinderkondensators verhält sich jedoch anders als die eines konzentrischen Zylinderkondensators (siehe Abschnitt 4.4.3); daher ergibt sich das Dilemma, dass der Vorschub des Instruments von der Kapazität und dem Verlauf des Instruments abhängt, welcher jedoch durch reine Kapazitätsmessung nicht ermittelt werden kann.
- Schlimmer als der nicht-zentrale Verlauf des Instruments ist die in der Praxis relativ häufig auftretende Bildung von Schlaufen (Abb. 4.13c). In diesen Fällen kann aus der gemessenen Kapazität kaum auf die Länge des eingeführten Instruments geschlossen werden, schon alleine deshalb, weil im Fall einer Schlaufe der Begriff „Länge“ nicht vernünftig definiert ist.
- Die Randeffekte an der Spitze des Instruments sind in der Praxis nicht zu vernachlässigen. In Abschnitt 4.4.4 wird beschrieben, wie durch Messungen an einem Testaufbau ermittelt wurde, wie sich diese auswirken.

Die oben genannten Probleme sind glücklicherweise nicht zu gravierend, wie folgende Überlegungen zeigen.

4.4.3. Theoretische Betrachtung des Fehlers bei der Längenmessung

Wie oben bereits beschrieben wurde, beträgt die Kapazität C eines konzentrischen Zylinderkondensators (Abb. 4.14a) der Länge l , des Innenradius R_1 und des Außenradius R_2 :

$$C = 2\pi\epsilon_0\epsilon_r \cdot \frac{l}{\ln\left(\frac{R_2}{R_1}\right)}$$

Für den in Abb. 4.14b skizzierten Fall, dass das Instrument in einem Abstand $d \geq 0$ (aus technischen Gründen muss außerdem $d < R_2 - R_1$ gelten, da die beiden Zylinder sich ansonsten gegenseitig durchdringen würden) parallel zur Zylinderachse liegt, ist die Kapazität nach [Jac99] gegeben durch

$$\begin{aligned} C &= 2\pi\epsilon_0\epsilon_r \cdot \frac{l}{\cosh^{-1}\left(\frac{R_1^2 + R_2^2 - d^2}{2R_1R_2}\right)} \\ &= 2\pi\epsilon_0\epsilon_r \cdot \frac{l}{\ln\left(\frac{R_1^2 + R_2^2 - d^2}{2R_1R_2} + \sqrt{\left(\frac{R_1^2 + R_2^2 - d^2}{2R_1R_2}\right)^2 - 1}\right)} \end{aligned}$$

Wie Nachrechnen zeigt, ergibt sich daraus für den Spezialfall $d = 0$ die oben genannte Gleichung für den konzentrischen Fall. Für wachsendes d wird das Argument des Logarithmus kleiner. Da dieser streng monoton steigend ist, wächst deswegen der Wert des Bruchs und damit die Kapazität. Weil das Modell davon ausgeht, dass das Instrument stets der Mittelachse des Zylinders folgt, wird damit in der Praxis ein fehlerhafter Vorschub berechnet. Der tatsächliche Vorschub ist dabei aufgrund der größeren gemessenen Kapazität stets kleiner als der berechnete.

Um zu untersuchen, wie groß diese Abweichung ist, soll das Diagramm in Abb. 4.15 dienen. Dort wird die relative Kapazitätzunahme ΔC_{rel} in Abhängigkeit von der relativen Exzentrizität e_{rel} dargestellt. $\Delta C_{rel} = 0$ bedeutet hierbei, dass sich die Kapazität verglichen mit der Kapazität eines konzentrischen Zylinderkondensators nicht verändert hat, während $\Delta C_{rel} = 1$ bedeutet, dass sie sich gerade verdoppelt hat. Analog bedeutet $e_{rel} = 0$, dass keine Exzentrizität vorliegt, während $e_{rel} = 1$ bedeutet, dass die Achse des inneren Zylinders gerade exakt an der Außenwand des äußeren Zylinders anliegt, in obiger Formel also $d = R_2 - R_1$ gilt. Das Diagramm geht nur bis zu einer relativen Exzentrizität $e_{rel} = 0,8$, da in der Praxis die Wandstärke der Gallengangsegmente in der Regel ca. $\frac{1}{6}$ des Außendurchmessers ausmacht und der Radius des Instruments in dersel-

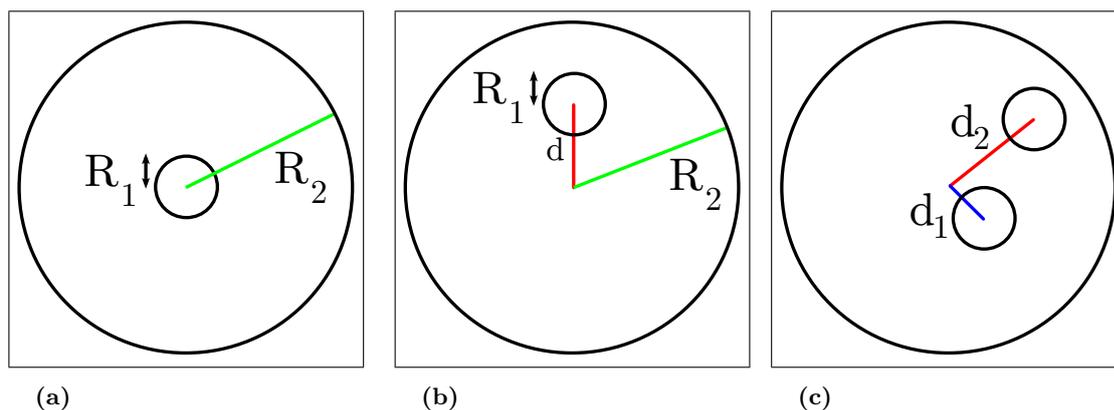


Abb. 4.14.: Zylinderkondensatoren von oben betrachtet. a) Konzentrischer Zylinderkondensator. b) Exzentrischer Zylinderkondensator. c) Windschiefer Zylinderkondensator. Die beiden Abstände d_1 und d_2 geben die Exzentrizität der Schnittpunkte des Innenleiters mit dem unteren bzw. oberen „Zylinderdeckel“ an.

ben Größenordnung liegt. Deswegen ist die absolute Exzentrizität in der Praxis maximal $\frac{4}{6}R_2$, was zu einer maximalen relativen Exzentrizität von $\frac{4}{5}$ führt.

Nichtparalleler, exzentrischer Zylinderkondensator

Nun wird die Berechnung noch weiter verallgemeinert, indem nicht mehr davon ausgegangen wird, dass die Achsen der beiden Zylinder parallel zueinander verlaufen, sondern dass die Achse des inneren Zylinders windschief zur Achse des äußeren Zylinders verläuft, wie in Abb. 4.14c von oben skizziert dargestellt ist. Nach [Tho66] kann die Berechnung der relativen Kapazitätzunahme in diesem Fall auf den parallelen exzentrischen Fall zurückgeführt werden, indem eine angenäherte äquivalente absolute Exzentrizität e_m durch Mittelwertbildung berechnet wird:

$$e_m = \frac{d_1 + d_2}{2}$$

Daraus ergibt sich sofort eine äquivalente relative Exzentrizität $e_{rel} = \frac{e_m}{R_2 - R_1}$.

Abschätzung des Fehlers durch Exzentrizität

In der Realität ist die Annahme, dass die gegebene Situation überhaupt als ein Zylinderkondensator modelliert werden kann, natürlich nicht korrekt, da, wie die Abb. 4.13b und 4.13c zeigen, der Verlauf des Instruments im Allgemeinen nicht geradlinig ist. Für die hier durchgeführten Abschätzungen zum Fehler durch Abweichungen von der idealisierten Vorstellung eines konzentrischen Zylinderkondensators soll dies jedoch nicht

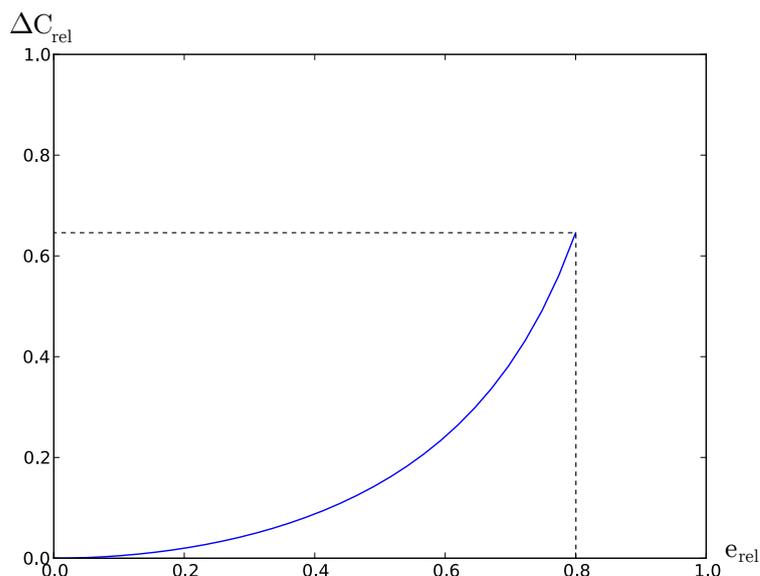


Abb. 4.15.: ΔC_{rel} in Abhängigkeit von e_{rel} aufgetragen.

mehr berücksichtigt werden, da ein geradliniger Verlauf eine in der Regel ausreichende Näherung darstellt.

Wie das Diagramm in Abb 4.15 zeigt, beträgt bei der maximal möglichen relativen Exzentrizität $e_{rel} = 0,8$ die relative Kapazitätzunahme $\Delta C_{rel} \approx 0,65$. Dies bedeutet, dass auch der Fehler bei der Längenmessung 65% betrüge, was natürlich inakzeptabel groß wäre. Es gibt jedoch zwei Faktoren, die den Fehler in der Praxis deutlich reduzieren:

1. Das Instrument verläuft in der Praxis in einem Segment nie vollständig entlang der Wand des Segments, sondern verläuft eher diagonal im Bereich zwischen 40% und 80% durch das Segment. Damit ist die gemittelte relative Exzentrizität in der Regel nicht größer als ungefähr 0,6, was einer relativen Kapazitätzunahme von ca. 25% entspricht.
2. Die Kalibrierung der Sensoren, die in Abschnitt 4.4.1 beschrieben wurde, wird ebenfalls nicht in einer idealen Zylinderkondensatorsituation durchgeführt, sondern einfach, indem das Instrument in das entsprechende Segment eingeführt wird und dann die resultierende Kapazität gemessen wird. Dadurch ist auch bei der Kalibrierung von einer gewissen Exzentrizität auszugehen. Es ist schwierig, diese genau zu beziffern, aber geschätzt dürfte sie bei etwa 40% liegen. Daraus ergibt sich, dass auch der Kalibrierungswert um ca. 10% zu groß ist.

Beide Faktoren zusammengenommen bedeuten, dass der tatsächliche resultierende Fehler in der Praxis geschätzt ungefähr $\frac{125-110}{125} = 12\%$ beträgt. Dieser Wert ist nun durch-

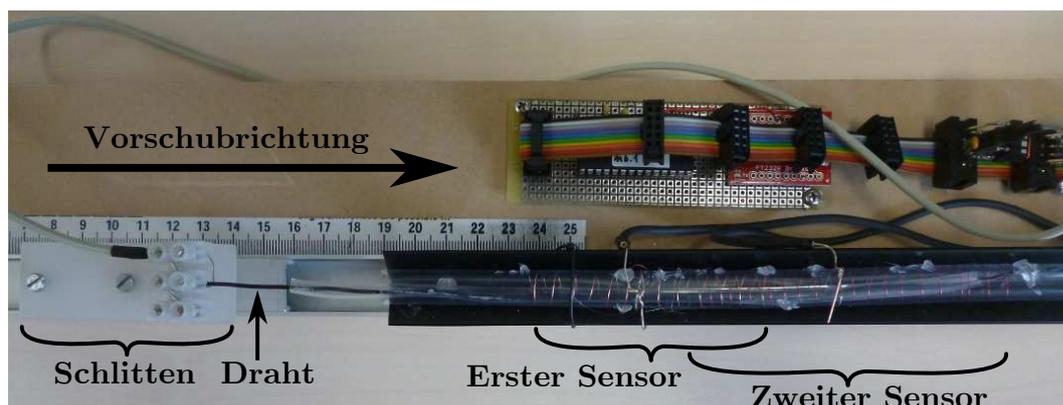


Abb. 4.16.: Testanordnung für die kapazitive Sensorik.

aus akzeptabel. Im Folgenden soll durch Messungen untersucht werden, ob sich die Längenmessung durch Kapazitätsbestimmung in der Praxis bewährt.

4.4.4. Messungen zur Kapazitiven Sensorik

Zur Messung der Verlässlichkeit der Sensorik wurde eine einfache Testanordnung konstruiert. Dazu wurde ein ca. 20 cm langer transparenter Isolierschlauch mit einem Durchmesser von 12 mm mit zwei spulenförmigen Elektroden umwickelt. Mit Hilfe eines Schlittens, der auf einer Schiene, die mit einer Millimeterskala versehen wurde, frei laufen kann, kann ein Stück Draht, in welches ein Signal eingespeist wird, sehr präzise in diesen Schlauch eingeführt werden. An der Skala kann die tatsächliche Einschubtiefe abgelesen werden, sodass eine Messreihe erstellt werden konnte, in der die Längen und die dabei gemessenen Sensorwerten ermittelt wurden. Abb. 4.16 zeigt ein Foto dieses Aufbaus. Die erste Elektrode hat eine Länge von 6,5 cm, während die zweite eine Länge von 8 cm aufweist. Die beiden Elektroden sind überlappend mit einer Überlappungszone von 2,5 cm Länge angeordnet. Die Skala ist so kalibriert, dass Länge 0 cm bedeutet, dass sich der Draht genau am Beginn der ersten Elektrode befindet. Die zweite Elektrode beginnt 4,5 cm weiter rechts.

Durchführung der Messungen

Der Schlitten mit dem Draht wurde in 5 mm-Schritten vorgeschoben und jeweils der gemessene Wert des ersten Sensors ermittelt. Nach einer Messung wurde der Draht wieder vollständig herausgezogen, ein wenig „geknetet“ und wieder begradigt und dann erneut eingeführt. Dieser Vorgang wurde viermal wiederholt, um robustere Messergebnisse zu erhalten. Dieses Verformen und anschließende wieder Begradigen des Drahts dient dem Zweck, dass er dadurch beim erneuten Einführen einen etwas anderen Verlauf nimmt. Auf

Länge [mm]	Messreihe 1	Messreihe 2	Messreihe 3	Messreihe 4	Ohne Sensor 2
0	200	150	150	200	200
5	400	400	350	450	450
10	700	700	650	680	750
15	1100	1100	1100	1030	1200
20	1500	1600	1700	1500	1700
25	2100	2300	2400	2100	2300
30	3000	2900	3000	3000	3000
35	2900	3700	3900	3550	4000
40	4600	4700	5100	4070	4600
45	5500	5400	5900	4850	5400
50	5900	6000	6300	5660	6200
55	6700	6500	6800	6530	7000
60	7200	7000	7300	7320	8000

Tabelle 4.1.: Messreihen für den ersten Sensor.

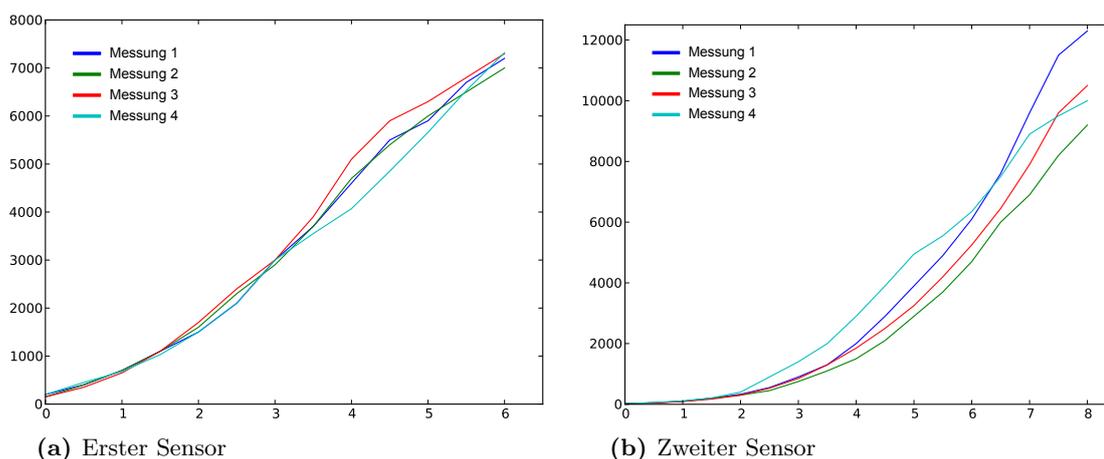


Abb. 4.17.: Die vier Messungen der beiden Sensoren.

diese Weise wurden die Messungen noch robuster. Anschließend wurde derselbe Vorgang für den zweiten Sensor durchgeführt, sodass auch hier vier Messreihen zur Verfügung stehen. Abschließend wurden noch je eine Messung des ersten und des zweiten Sensors durchgeführt, während der der jeweils andere Sensor abgekoppelt wurde, um zu ermitteln, inwieweit sich die Sensoren gegenseitig beeinflussen. Die Tabellen 4.1 und 4.2 zeigen die gemessenen Werte. Abb. 4.17 zeigt diese Werte in einem Diagramm dargestellt, in dem die Sensorwerte gegen die gemessenen Längen aufgetragen sind. In Abb. 4.18a und 4.18b wurden die punktwisen arithmetischen Mittelwerte aller vier Messungen sowie eine Regressionsgerade in ein Diagramm eingezeichnet.

Länge [mm]	Messreihe 1	Messreihe 2	Messreihe 3	Messreihe 4	Ohne Sensor 1
0	20	20	20	20	1100
5	50	50	40	60	1200
10	100	100	90	110	1600
15	200	200	170	210	1900
20	330	300	300	400	2200
25	550	450	530	900	2800
30	900	750	850	1400	3500
35	1300	1100	1300	2000	4500
40	2000	1500	1850	2900	5400
45	2900	2100	2500	3900	6600
50	3900	2900	3250	4950	8000
55	4900	3700	4200	5550	9300
60	6100	4700	5250	6350	10800
65	7600	6000	6450	7500	12200
70	9600	6900	7900	8900	13700
75	11500	8200	9600	9500	14800
80	12300	9200	10500	10000	16600

Tabelle 4.2.: Messreihen für den zweiten Sensor.

Auswertung der Messergebnisse

Auf den ersten Blick fällt auf, dass der erhoffte lineare Zusammenhang zwischen Sensorwerten, also gemessenen Kapazitäten, und der Länge des eingeführten Instruments vor allem bei dem zweiten Sensor nicht gilt. Die Kurve „hängt“ deutlich durch. Ein ähnlicher, wenn auch schwächerer Effekt, tritt auf dem Anfangsstück des ersten Sensors von 0 bis ca. 2 cm auf. Ebenfalls wird deutlich, dass beim zweiten Sensor eine Abflachung der Kurve auf dem letzten Stück geschieht. Beide Effekte hängen vermutlich mit Randeffekten zusammen, die an der Spitze des eingeführten Instruments auftreten und dadurch die Kapazität gegenüber der eines idealen Zylinderkondensators verringern. Beim zweiten Sensor ist der Effekt wesentlich größer, was möglicherweise zusätzlich daran liegen kann, dass in der Überlappungszone Beeinflussungen durch den ersten Sensor auftreten.

Diese Effekte lassen sich überraschend gut kompensieren, indem nicht die Sensorwerte V_i , sondern die Quadratwurzeln der Sensorwerte $\sqrt{V_i}$ betrachtet werden. In diesem Fall ergibt sich in brauchbarer Näherung ein linearer Zusammenhang. Die Abb. 4.18c und 4.18d zeigen diese Diagramme. Vor allem beim zweiten Sensor funktioniert diese Art der Kompensation sehr gut. Beim ersten Sensor ist der Gewinn nicht ganz so offensichtlich, aber auch bei diesem konnte das Anfangsstück bis ca. 4,5 cm Länge sehr gut linearisiert werden, während dahinter die Abflachung sichtbar wird. Da nach Definition 4.1 ein Gallengangsegment stets in der Mitte einer Überlappungszone beginnt bzw. endet, ist diese leichte Abflachung zum Ende hin nicht von Bedeutung, da dort ohnehin bereits der

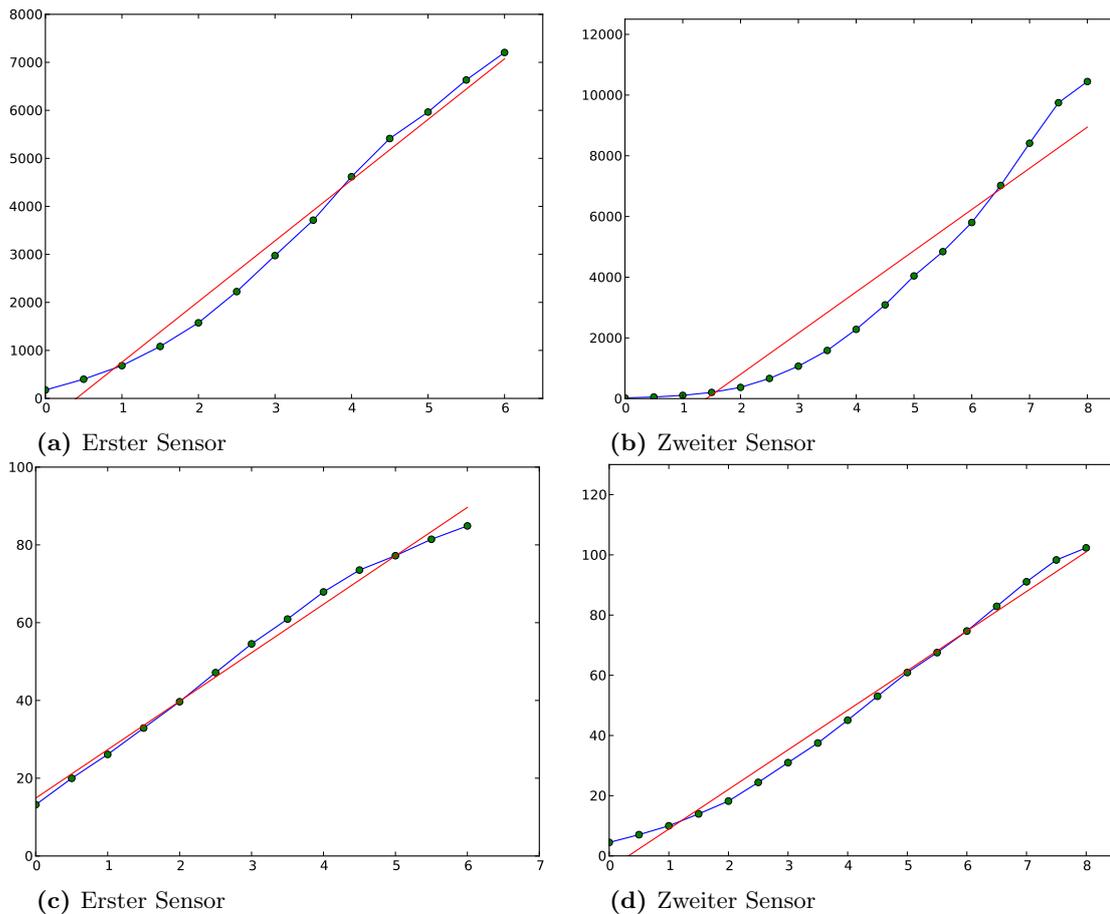


Abb. 4.18.: Durchschnittliche Sensorwerte. Oben: Werte direkt eingezeichnet. Unten: Quadratwurzel der Werte eingezeichnet.

nächste Sensor relevant wird. Die Geraden sind zwar keine Ursprungsgeraden, lassen sich aber durch geeignete Kalibrierung der Sensoren einfach in solche umwandeln.

Beeinflussung durch den jeweils anderen Sensor

Abb. 4.19 zeigt die Quadratwurzeln der Messwerte der beiden Sensoren für den Fall, dass der jeweils andere Sensor abgekoppelt wurde. Die Messwerte des ersten Sensors werden durch den zweiten offenbar kaum gestört, mit Ausnahme des letzten Stücks. Dies liegt daran, dass der eingeführte Draht den zweiten Sensor noch nicht passiert hat, wenn er gerade in dem ersten liegt. Deswegen wird er durch den zweiten Sensor auch nicht gedämpft. Lediglich in der Überlappungszone der beiden Sensoren ist eine geringe Dämpfung sichtbar.

Bei dem zweiten Sensor sieht der Fall anders aus. Hier sind die Messwerte deutlich höher, wenn der erste Sensor abgekoppelt wurde. Dies bedeutet, dass der erste Sensor einen erheblich dämpfenden Einfluss auf die Messung des zweiten Sensors hat. Wie das Diagramm

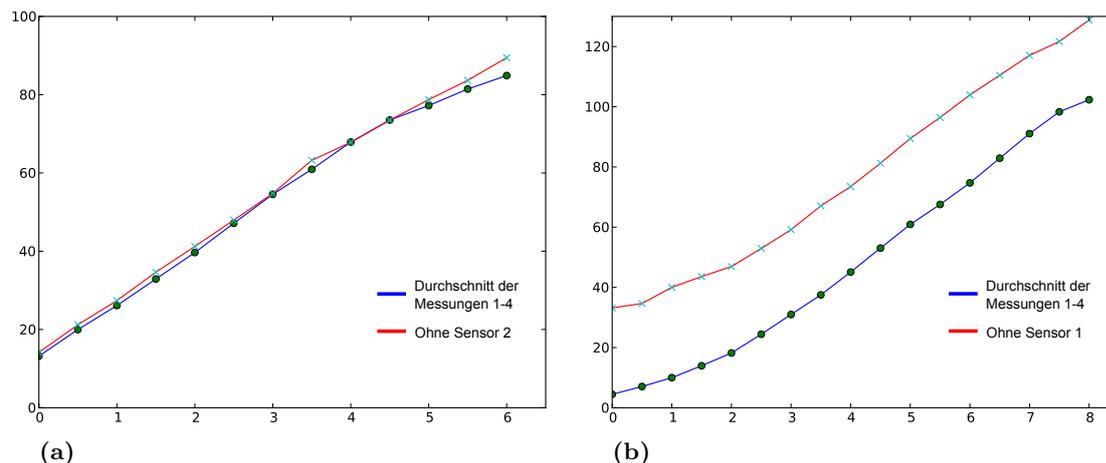


Abb. 4.19.: Darstellungen der Quadratwurzeln der Sensorwerte, wenn der jeweils andere abgekoppelt wurde. a) Erster Sensor ohne den zweiten. b) Zweiter Sensor ohne den ersten. Zum Vergleich sind die Durchschnittswerte der vorigen vier Messungen jeweils mit eingezeichnet.

zeigt, ist dieser Einfluss jedoch konstanter Natur, weil alle Messwerte um ungefähr denselben Wert verändert werden. Aus diesem Grund kann diese Konstante leicht durch entsprechende Kalibrierung neutralisiert werden.

Erkennung von Schlaufen

Nun soll noch untersucht werden, wie die Messwerte durch die Anwesenheit einer Schlaufe des Drahts beeinflusst werden. Dazu wurde der Testdraht an seinem Ende zu einer Schlaufe gebogen, die eine Länge von ca. 3 cm hat, und in den Isolierschlauch eingeführt. Wie bei den übrigen Messungen wurde dieser Draht mit Schlaufe in 5 mm-Schritten vorgeschoben und dabei die Werte des ersten Sensors gemessen. Tabelle 4.3 listet die gewonnenen Messwerte auf und Abb. 4.20 stellt die Quadratwurzeln dieser Messwerte in einem Diagramm dar, in das zum Vergleich auch der Durchschnitt über die Messwerte eines in den ersten Sensors eingeführten Drahts ohne Schlaufe eingezeichnet wurde. Man sieht, dass die Messwerte bei einer eingeführten Schlaufe deutlich über denen liegen, die der geradlinige eingeführte Draht aufweist. Da die Schlaufe eine Länge von ca. 3 cm hat, ist sie erst ab einem Vorschub von 3 cm vollständig innerhalb des Sensors. Der Messwert bei diesem Vorschub ist mit 8260 bereits größer als der eines komplett in das Segment gebrachten geradlinigen Drahts mit ≈ 7200 .

4.4.5. Fazit der Messung

Wie die Experimente zeigen, kann mit der vorgestellten Sensorik und der Anwendung der Wurzelfunktion auf die Sensorwerte die Länge des eingeführten Anteils des Instruments

Länge [mm]	Messwerte
0	280
5	760
10	1950
15	3230
20	4450
25	6370
30	8260
35	10100
40	11700
45	13200
50	15600
55	16200
60	16400

Tabelle 4.3.: Messwerte des ersten Sensors mit eingeführter Schlaufe.

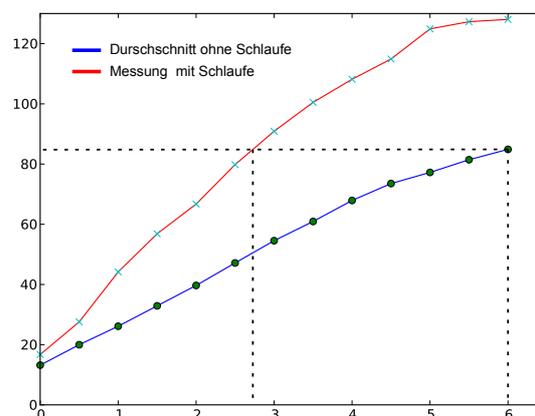


Abb. 4.20.: Darstellung der Messwerte der Schlaufe und zum Vergleich die eines geradlinigen Drahts.

gut bestimmt werden. Eine physikalische Erklärung dafür, dass gerade die Quadratwurzelfunktion gute Resultate erzielt, wird an dieser Stelle nicht versucht. Stattdessen wird dies einfach als empirisch gefundenes Ergebnis hingenommen. Im weiteren Verlauf wird die Quadratwurzelfunktion daher stets angewendet, sodass immer dann, wenn von einem Sensorwert die Rede ist, die Quadratwurzel des eigentlich gemessenen Werts gemeint ist.

Wie bereits gesagt wurde, kann mit der kapazitiven Sensorik die genaue Lage des Instruments innerhalb des Querschnitts der Gallengänge nicht erkannt werden. Die Bildung einer Schlaufe kann aber durch die dabei auftretenden sehr großen Sensorwerte immerhin grundsätzlich erkannt werden. Es kann allerdings lediglich das Gallengangsegment, in dem diese auftritt, bestimmt werden und nicht, an welcher genauen Stelle sich die Schlaufe gebildet hat.

Trotz dieser Einschränkungen kommt diese Sensorik im aktuellen Biliphanten mit großem Erfolg zum Einsatz. Für die Anfängerkurse fallen die Nachteile nicht ins Gewicht, da hier nicht die genaue Lage der Instrumente interessiert, sondern es nur wichtig ist, zu erkennen, dass die Teilnehmer die Instrumente in die korrekten Gallengangsegmente eingeführt haben und wie groß deren Vorschub ist. Die Resonanz bei Trainingsteilnehmern und Experten auf dem Gebiet der ERCP war so positiv, dass zwei verschiedene Posterpräsentationen mit Preisen prämiert wurden:

Das Poster [GIS⁺12b] wurde auf dem 23. Kongress der Südwestdeutschen Gesellschaft für Gastroenterologie in Freiburg präsentiert. Es beschreibt den Biliphanten inklusive der Röntgensimulation mit *X-Sim* und wurde mit dem **Stiftungspreis der Südwestdeutschen Gesellschaft für Gastroenterologie** ausgezeichnet. Weiterhin wurde das

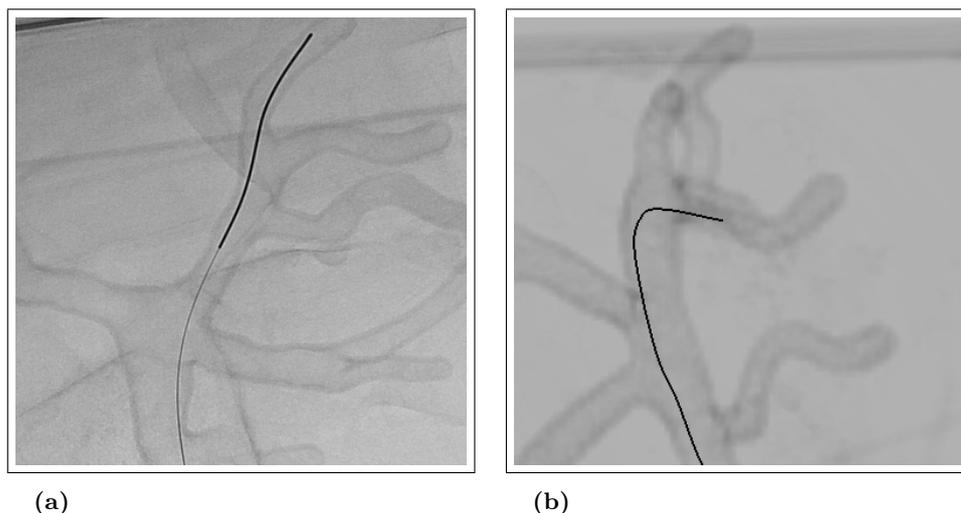


Abb. 4.21.: Vergleich von a) echtem und b) simuliertem Röntgenbild eines Sondierungsdrahts.

Poster [VAG13a], das die kapazitive Sensorik von *X-Sim* und die Begradigungsanzeige (vgl. Abschnitt 4.8) vorstellt, auf der DGE BV 2013 mit dem **Boston Scientific Innovationspreis** ausgezeichnet.

Für die Kurse mit erfahrenen Teilnehmern reicht die vorgestellte kapazitive Sensorik jedoch nicht mehr aus, da es für die Techniken, die üblicherweise in den Fortgeschrittenkursen gelehrt werden, unerlässlich ist, die exakte Lage und vor allem auch Gestalt der verwendeten Instrumente zu erkennen. Aus diesem Grund wurde noch eine andere Sensorik konzipiert, die diese Anforderungen erfüllt. In Abschnitt 4.6 wird diese genauer beschrieben.

4.5. Darstellung des Instruments in den Anfängerphantomen

Für die Entwicklung eines Verfahrens zur Darstellung der Instrumente wird zunächst einmal vereinfachend angenommen, dass es sich bei dem Gallengangssystem um einen entarteten Baum handelt, der keinerlei Verzweigungen aufweist. Er besteht nur aus einem einzigen Pfad von seiner Wurzel, also der Papille, bis zu seinem einzigen Blatt. Später in Abschnitt 4.5.4 wird diese Vereinfachung dann aufgehoben und ein Verfahren präsentiert, mit dem das Instrument in beliebigen baumstrukturierten Gefäßsystemen dargestellt werden kann.

4.5.1. Definition der Instrumente mittels Splines

Wie eingangs beschrieben sind die einzigen Instrumente, die in den Anfängerkursen zum Einsatz kommen, Sondierungsdrähte und Papillotome (vgl. Abb. 4.10). Aufgrund ihrer in

erster Näherung eindimensionalen Gestalt lassen sich beide durch einfache parametrisierte Kurven $\gamma : [0, L(\gamma)] \rightarrow \mathbb{R}^3$ beschreiben. Hierbei ist $L(\gamma)$ die Länge der Kurve γ . Eine solche Kurve wird für den Verlauf des Instruments definiert, bei dem es vollständig in den Gallengang eingeführt wurde, d.h. γ beschreibt den vollständigen möglichen Verlauf des Instruments von der Papille bis zum Ende des Gallengangs. Um einen beliebigen anderen Vorschub des Instruments darzustellen, muss γ lediglich in Abhängigkeit von den Sensorwerten nur auf einem Teilintervall $[0, x]$ ausgewertet und eingezeichnet werden, wie in Abschnitt 4.5.3 verdeutlicht wird. Weil die Sensorik, die in Abschnitt 4.4 beschrieben wurde, nicht erkennen kann, wo das Instrument innerhalb der Gallengänge exakt verläuft, ist es plausibel, den Verlauf des Instruments willkürlich durch eine fest vorgegebene Kurve zu beschreiben, sodass es möglichst realistisch aussieht.

Im Allgemeinen lassen sich sowohl Papillotome als auch Sondierungsdrähte nicht beliebig stark krümmen, sondern liegen zumeist relativ glatt in den Gallengängen. Eine Ausnahme tritt nur dann auf, wenn der Sondierungsdraht enge Schleifen bildet; weil diese Situation jedoch nicht genau erkannt werden kann, muss die Darstellung einer solchen Schleifenbildung nicht berücksichtigt werden. Abb. 4.21 zeigt eine Gegenüberstellung eines echten Röntgenbildes eines Sondierungsdrahts und einem simulierten Röntgenbild, das mit den im Folgenden beschriebenen Methoden erzeugt wurde. Aufgrund des glatten Verlaufs der Instrumente eignen sich zur Definition der Kurven sog. *Splines*. Die Ausführungen und Definitionen im folgenden Abschnitt lehnen sich stark an [Sal06, S. 161-211] an:

Definition 4.2. Eine Bézierkurve β vom Grad n zu gegebenen $n + 1$ Kontrollpunkten P_0, P_1, \dots, P_n , die das sogenannte Kontrollpolygon bilden, ist für $t \in [0, 1]$ definiert als

$$\beta(t) = \sum_{i=0}^n B_{i,n}(t)P_i, \text{ wobei}$$

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \text{ das } i\text{-te Bernsteinpolynom } n\text{-ten Grades ist.}$$

Die *Bézierkurve* zeichnet sich dadurch aus, dass sie stets innerhalb der konvexen Hülle des Kontrollpolygons verläuft. Im Fall eines Grades von $n = 3$ lassen sich die vier Punkte des Kontrollpolygons interpretieren als Start- und Endpunkte der Tangenten in den Anfangs- bzw. Endpunkten der Kurve (Abb. 4.22a).

Definition 4.3. Unter einem Spline n -ten Grades versteht man eine Kurve, die stückweise durch Polynome vom Grad n definiert ist. In den Punkten, wo zwei dieser Polynome aneinanderstoßen (den sog. Knoten des Splines) werden an die Kurve Bedingungen hinsichtlich ihrer Differenzierbarkeit gestellt, damit der Spline dort einen glatten Verlauf aufweist. Häufig werden Splines aus Bézierkurven konstruiert, wobei der Grad oft $n = 3$ ist.

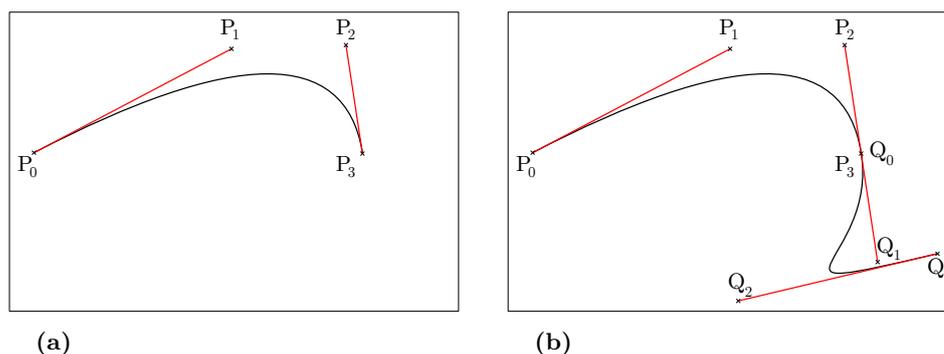


Abb. 4.22.: a) Béziersegment, das die Knotenpunkte P_0 und P_3 interpoliert. Durch P_1 und P_2 werden die Tangenten definiert.
 b) Aus zwei Segmenten zusammengesetzter Spline. Damit die Kurve stetig differenzierbar ist, müssen sowohl P_3 und Q_0 als auch die Tangenten in diesen Punkten übereinstimmen.

Um bei der Zusammensetzung des Splines aus einzelnen Bézierkurven einen glatten Übergang zwischen den einzelnen Kurvensegmenten zu erreichen, müssen in den Knotenpunkten die Tangenten der beiden dort zusammenstoßenden Kurven identisch sein, sodass die komplette Kurve zweimal stetig differenzierbar ist (Abb 4.22b). Über die Wahl der Knotenpunkte und Tangenten lässt sich der Verlauf des Splines gut und intuitiv steuern.

Wahl der Kontrollpunkte

Da der genaue Verlauf des Instruments nicht bestimmt werden kann, muss der Spline γ willkürlich festgelegt werden. Diese Festlegung kann einmalig für den vollständigen möglichen Verlauf des Instruments von der Wurzel bis zum Blatt erfolgen und dann fest abgespeichert werden, um so für jede Benutzung von *X-Sim* denselben Verlauf des Instruments zu erhalten. Dieser sollte so gewählt werden, dass das Instrument im simulierten Röntgenbild möglichst plausibel innerhalb der Gallengänge liegt.

Die Wahl, den Spline in der Nähe der Mittellinie der Gallengangsegmente verlaufen zu lassen, liefert in der Regel die besten Resultate. Weil ein Spline seine Knotenpunkte interpoliert, müssen daher auch diese zumeist auf der Mittellinie liegen. Es kann allerdings bei manchen Abschnitten des Gallengangsystems auch vorkommen, dass eine optisch ansprechendere Wahl der Knotenpunkte darin besteht, von der Mittellinie abzuweichen. Abb. 4.23 skizziert einen solchen Spline innerhalb des vereinfachten linearen Gallengangsystems. Hier wurde der Knotenpunkt N am Ende des dritten Segments deutlich von der Mittellinie weg hin zum unteren Rand verschoben, weil ein Instrument, das von der linken Seite her eingeführt wird, in der Realität in der Nähe des Punktes N zunächst einmal an die untere rechte Wand des Gefäßes stoßen würde, bevor es nach oben rechts abgelenkt wird.

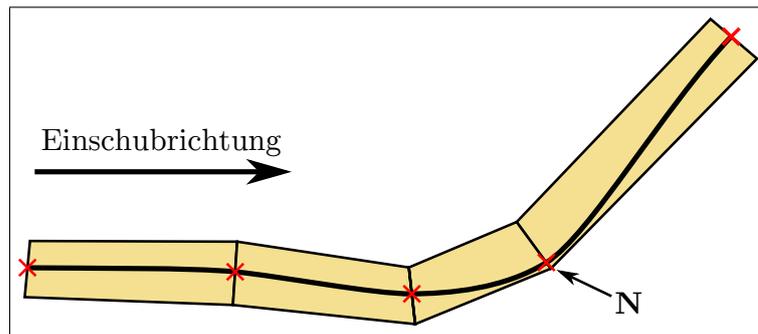


Abb. 4.23.: Definition des Splines innerhalb des (vereinfachten) linearen Gallengangsystems, das in diesem Fall aus vier Segmenten besteht. Die roten Punkte stellen die Knotenpunkte des Splines dar, während die Kontrollpunkte oder Tangenten nicht eingezeichnet sind.

Definition der Tangenten

Damit ein Bézierspline eine gegebene Menge von Punkten interpolieren kann, müssen für jeden Abschnitt die vier Kontrollpunkte p_0 bis p_3 festgelegt werden. p_0 und p_3 sind leicht zu finden: Es sind gerade die Knotenpunkte des Splines, also die zu interpolierenden Punkte. Die Punkte p_1 und p_2 jedoch, also die Endpunkte der Tangenten, sind nicht so offensichtlich gegeben und müssen entweder manuell oder heuristisch automatisch festgelegt werden. Eine gute Möglichkeit, diese Tangenten automatisch festzulegen, ist analog zu sog. *Cardinal-Splines* vorzugehen [Sal06, S. 161]. Diese erzielen optisch ansprechende Resultate, indem die Tangente T_i in einem Knotenpunkt P_i definiert wird als $T_i := \tau \cdot (P_{i+1} - P_{i-1})$ mit üblicherweise $\tau \in [0,1]$. τ wird als *Spannung* (engl. *tension*) bezeichnet.

Anschaulich wird die Tangente in einem Punkt P_i damit festgelegt als der Vektor, der vom vorherigen Knotenpunkt P_{i-1} zum nächsten Knotenpunkt P_{i+1} zeigt, multipliziert mit dem *Spannungsfaktor* τ . Je kleiner τ ist, desto „schärfer“ ist der Verlauf der Kurve in den Knotenpunkten, während ein größerer Wert einen viel „gemächlicheren“ Verlauf der Kurve in den Knotenpunkten, dafür jedoch einen gewundeneren Verlauf dazwischen zur Folge hat (Siehe Abb. 4.24). Im Extremfall $\tau = 0$ entspricht die resultierende Kurve gerade dem Polygonzug, der die Knotenpunkte verbindet. Empirisch hat sich gezeigt, dass $\tau = \frac{1}{3}$ den optisch ansprechendsten Kompromiss aus einerseits glattem und andererseits wenig unnötig gewundenem Verlauf darstellt.

4.5.2. Zuordnung der Sensoren zu Kurvenabschnitten

Zu Beginn des vorigen Abschnitts 4.5.1 wurde gesagt, dass die Kurve γ , die das Instrument repräsentiert, in Abhängigkeit von den Sensorwerten mehr oder weniger weit eingezeichnet werden soll. Es muss also eine Zuordnung zwischen den einzelnen Abschnitten

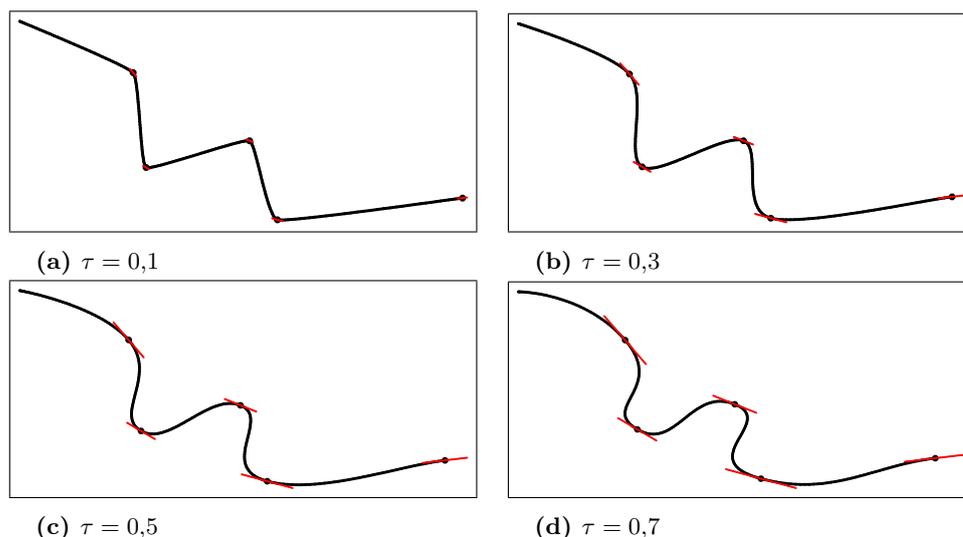


Abb. 4.24.: Spline mit verschiedenen *Spannungsfaktoren* τ . Die Tangenten sind in rot eingezeichnet.

von γ und den Sensoren geben, anhand derer bestimmt wird, auf welche Weise welcher Sensor welche Kurvenabschnitte beeinflusst. Nach Definition 4.1 wird ein Gallengangsegment durch genau einen Sensor definiert. Eine erste Idee der Zuordnung der Gallengangsegmente zu den Splineabschnitten ist daher, dass jedem Gallengangsegment genau ein Splineabschnitt zugeordnet wird, dessen Start- und Endpunkte genau am Anfang bzw. Ende des Gallengangsegments liegen. Dadurch liegen die Knotenpunkte des Splines immer genau an den Grenzen zwischen zwei Gallengangsegmenten. Für jedes Gallengangsegment könnte dessen Sensorwert bestimmen, wie weit der Splineabschnitt, der in diesem Gallengangsegment liegt, eingezeichnet werden muss.

Es kann jedoch bei besonders gewundenen Gallengangsegmenten vorkommen, dass eine einzelne Bézierkurve vom Grad $n = 3$ nicht ausreicht, um einen Verlauf des Instruments in diesen Segmenten zu erreichen, der die Wände des Gallengangsystems nicht schneidet. Abb. 4.25 zeigt eine zu Illustrationszwecken übertrieben dargestellte Skizze eines solchen Falls: Es ist unmöglich, in diesem Beispiel eine Bézierkurve dritten Grades so einzupassen, dass sie die Wände des Segments nicht schneidet. Die Kurve muss in diesem Beispiel zwei Wendestellen aufweisen, damit sie „kollisionsfrei um die Ecke kommt“. Wegen des Grades $n = 3$ hat aber jeder Splineabschnitt höchstens eine Wendestelle.

Eine Möglichkeit der Verbesserung besteht in der Wahl eines höheren Grades. Dadurch wird jedoch die Kontrolle über den Verlauf der Kurve weniger intuitiv. Vor allem aber reicht der Grad $n = 3$ für die meisten Gallengangsegmente aus, was entweder bedeuten würde, dass für diese mehr Aufwand als nötig betrieben wird oder dass für die unterschiedlichen Gallensegmente verschiedene Grade verwendet werden, was bei der Zusammensetzung der einzelnen Bézierkurven komplizierter wäre.

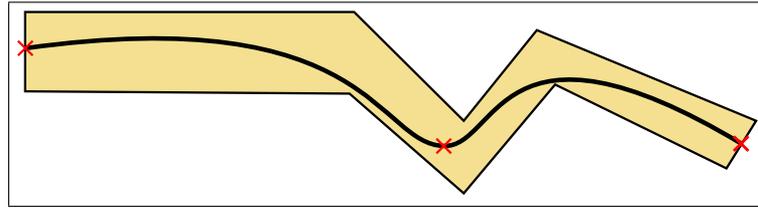


Abb. 4.25.: Unter Umständen ist ein Béziersegment nicht ausreichend. Erst wenn ein zweiter Knotenpunkt hinzugefügt wird, kann der Spline das komplette Gallengangsegment ohne Kollisionen mit dessen Wand durchlaufen.

Eine andere Lösung besteht darin, die Hardware des Biliphanten selbst zu verändern und einen Sensor zusätzlich zu verwenden. Damit könnte das fragliche Gallengangsegment in zwei Segmente unterteilt werden, für die jeweils eine Kurve vom Grad $n = 3$ wieder ausreichend wäre. Diese Lösung würde jedoch ebenfalls einen unnötig erhöhten Aufwand bedeuten.

Eine dritte Lösung, welche letztendlich auch zum Einsatz kommt, besteht darin, dass einem Gallengangsegment mehr als eine Bézierkurve zugeordnet werden kann. Abb. 4.25 zeigt diese Lösung: Dort wird ein aus zwei Bézierkurven zusammengesetzter Spline verwendet. Diese Lösung bedeutet den geringsten Aufwand sowohl bei der Bestückung des Biliphanten mit Elektroden als auch bei der Implementierung der Programmfunktionen zur Zuordnung von Splineabschnitten und Sensoren.

Diese Zuordnung wird für jedes Exemplar des Biliphanten einmalig vorgenommen und dann in dessen Beschreibungsdatei (siehe Abschnitt 2.4.3) fest abgespeichert. Bei der initialen Zuordnung wird nach dem folgenden Ablauf vorgegangen:

1. Bevor die Sensoren festgelegt werden, muss der Spline definiert werden, welcher den vollständigen möglichen Weg des Instruments vom Anfang, also der Papille, bis zum Ende, also dem in der momentanen Vereinfachung einzigen Blatt des entarteten Gefäßbaums, beschreibt. Die Knotenpunkte sollten so gewählt werden, dass der resultierende Spline einen Verlauf nimmt, der in dieser Form auch bei einem realen Instrument plausibel wäre.
2. Nun können auf Grundlage dieses Splines Gallengangsegmente, also Elektrodenbereiche, definiert werden. Diese sollten natürlich genauso definiert werden, wie sie auch tatsächlich um das reale Gallensystem im Biliphanten gelegt wurden, damit der simulierte Gallenbaum mit dem realen übereinstimmt. Durch das System wird erzwungen, dass ein solches Gallengangsegment stets an einem Knotenpunkt des Splines beginnt und endet. Dadurch wird erreicht, dass einem Gallengangsegment genau ganzzahlig viele Bézierkurven zugeordnet werden. Durch das System wird außerdem auch erzwungen, dass sich keine zwei Gallengangsegmente überschneiden.

3. Nachdem alle Gallengangsegmente definiert wurden, wird für jedes Segment S_i dessen Teilspline γ_i definiert. γ_i ist ein Bézierspline, der aus den einzelnen Bézierkurven zusammengesetzt wird, die S_i in Schritt 2 zugeordnet wurden. Die disjunkte Vereinigung aller γ_i ist gerade der Spline γ . γ_i wird nun derart umparametrisiert, dass $\gamma_i(0)$ gerade am Beginn von S_i und $\gamma_i(1)$ genau am Ende von S_i liegt. Weiterhin soll für die *Geschwindigkeit* $v_i(t)$ von γ_i gelten, dass $v_i(t) := \left\| \frac{d\gamma_i(t)}{dt} \right\| = \text{const.}$

Umparametrisierung der Kurvenabschnitte

Eine solche Parametrisierung kann in der Praxis hinreichend genau erreicht werden, indem der Spline γ_i durch einen Polygonzug approximiert wird. Dazu gibt man sich eine kleine Schwelle θ vor. Nun wird γ_i in regelmäßigen Abständen $\delta \ll \theta$ ausgewertet: $P_{neu} = \gamma_i(k * \delta)$ mit $k \in \mathbb{N}$. Sobald der berechnete Punkt P_{neu} vom letzten Punkt des Polygonzugs einen Abstand $> \theta$ hat, wird P_{neu} in den Polygonzug aufgenommen. Dies wird so lange wiederholt, bis γ_i bis zum Ende ausgewertet wurde. Der so entstandene Polygonzug entspricht in beliebig guter Näherung dem Verlauf des Splines γ . Er ist an seinen Eckpunkten nicht differenzierbar, weshalb der Begriff der „Geschwindigkeit“ nicht im Kontinuierlichen, sondern als diskretes Analogon $v_i(t) := \left\| \frac{\Delta\gamma_i(t)}{\Delta t} \right\|$ verstanden werden sollte. In der Praxis spielt dies jedoch keine Rolle.

4.5.3. Auswerten und Zeichnen der Splines

Wie bereits beschrieben wurde, besteht die dem Darstellungsverfahren zugrunde liegende Idee darin, den Verlauf des Instruments in den Gallengängen a priori festzulegen, d.h. den Spline γ , der das Instrument darstellen soll, vollständig vorzuberechnen. In Echtzeit muss nur noch anhand der Sensorwerte entschieden werden, wie **weit** das Instrument eingeführt wurde und dementsprechend, welcher Teil der vordefinierten Kurve gezeichnet werden muss und welcher unsichtbar bleibt. Diese Entscheidung wird anhand des folgenden Algorithmus getroffen:

1. Falls ein Sensorwert v_i über einer kleinen Schwelle ε liegt, wird davon ausgegangen, dass im Gallengangsegment S_i ein Instrument liegt. Daher wird in einem ersten Schritt eine geordnete Liste aller Gallengangsegmente erstellt, deren Sensoren einen Wert von mindestens ε liefern. Die Ordnung wird dabei gerade von der Reihenfolge vorgegeben, in der die Segmente von dem Spline γ durchlaufen werden. Die Schwelle ε wird eingeführt, damit das Verfahren robust gegenüber Sensorrauschen wird.
2. Nun wird aus dieser Liste das Segment S_h mit größtem Index (also das letzte in der Liste) ausgewählt. S_h ist das Segment, in dem die Spitze des eingeführten Instruments liegen muss. An welcher Stelle innerhalb von S_h diese aber genau liegt,

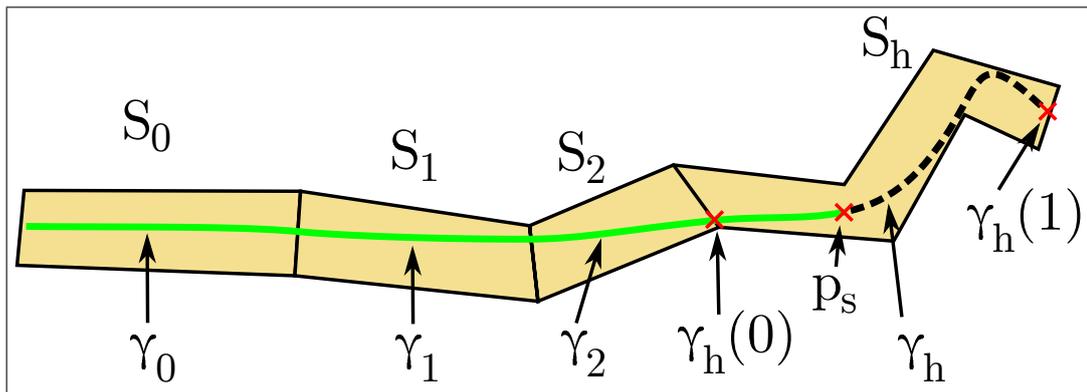


Abb. 4.26.: Skizze der Splineauswertung: Der Gallenbaum besteht in diesem Beispiel aus den vier Segmenten S_0 , S_1 , S_2 und S_h . Die Teilsplines dieser Segmente sind γ_0 , γ_1 , γ_2 und γ_h . Der Punkt $p_s = \gamma_h(v_h)$ ist gerade die Spitze des Instruments.

gilt es nun zu ermitteln. Im letzten Abschnitt 4.5.2 wurde die Umparametrisierung beschrieben, nach deren Durchführung für den Teilspline γ_h von S_h gilt, dass seine Geschwindigkeit konstant ist und $\gamma_h(0)$ bzw. $\gamma_h(1)$ gerade Anfang bzw. Ende von S_h sind.

Daher kann der mittels Gleichung 4.7 auf das Intervall $[0, 1]$ normierte Sensorwert v_h des Segments S_h direkt dazu benutzt werden, die Position der Spitze des Instruments zu berechnen. Für den Punkt p_s an der Spitze gilt $p_s = \gamma_h(v_h)$ (Siehe Abb. 4.26). Alle Punkte $\gamma_h(t)$ mit $t \leq v_h$ gehören zu dem Instrument und sind daher einzuzeichnen, während alle Punkte $\gamma_h(t)$ mit $t > v_h$ nicht mehr zu dem Instrument gehören und daher unsichtbar bleiben müssen. In der Skizze sind diese durch einen gestrichelten Verlauf angedeutet.

3. Alle übrigen Segmente S_i mit $0 \leq i < h$ werden von dem Instrument auf jeden Fall durchlaufen. Daher können alle Splineabschnitte, die diesen Segmenten zugeordnet sind, vollständig eingezeichnet werden.

4.5.4. Splinebäume

Bisher wurde von der vereinfachenden Annahme ausgegangen, dass kein mehrfach verzweigter Gallenbaum vorliegt, sondern nur ein einziger linearer Gang. In der Realität ist das Gallengangsystem aber natürlich verzweigt. Aus diesem Grund muss die Idee, ein eingeführtes Instrument durch einen einzigen fest definierten Spline darzustellen, modifiziert werden, damit auch Instrumente in baumstrukturierten Gefäßsystemen dargestellt werden können. Ein erster Ansatz besteht darin, nicht nur einen Spline, sondern eine Menge von n separaten Splines, die im folgenden *Pfade* genannt werden, zu definieren, wobei n die Anzahl der Blätter des Gallengangbaums ist. Jeder dieser Pfade beschreibt

einen vollständigen möglichen Verlauf eines eingeführten Instruments von der Wurzel des Baums, also der Papille, bis zu „seinem“ Blatt.

Wie in dem einfachen, unverzweigten Fall können nun die mittels Gleichung 4.7 auf das Intervall $[0, 1]$ normierten Werte der Sensoren, eine Parametrisierung der Pfade mit konstanter Geschwindigkeit vorausgesetzt, direkt dazu benutzt werden, den Pfad eines Instruments einzuzichnen. Damit nicht mehrere Instrumentenpfade gleichzeitig eingezeichnet werden, muss aus allen existierenden Pfaden derjenige ausgewählt werden, welcher dem Verlauf des Instruments in der Realität am besten entspricht. Diese Auswahl kann anhand der Sensorwerte getroffen werden. Details zu diesem Auswahlverfahren werden später in 4.5.5 beschrieben.

Bei der Definition der einzelnen Pfade sollte darauf geachtet werden, dass Pfade verschiedener Blätter, die durch dasselbe Gallengangsegment laufen, in diesem Segment exakt denselben Weg nehmen. Ist dies, wie in in Abb. 4.27a beispielhaft (und übertrieben) skizziert, nicht der Fall, ereignet sich ein optisch störender Sprung in der Darstellung des Instruments, sobald das Instrument durch Manipulationen durch den Trainierenden von einem Gallengang in einen anderen gerät. In diesem Fall wird der Spline, der dieses Instrument beschreibt, in seinem kompletten Verlauf einen Sprung machen, nicht nur in dem Bereich an der Spitze des Instruments ab der letzten Verzweigung, wo sich der Verlauf tatsächlich geändert hat.

Verlaufen die einzelnen Splines jedoch, wie in Abb. 4.27b skizziert, stets bis zu der Verzweigung, ab der sie in unterschiedlichen Segmenten weiterlaufen, identisch, tritt dieser optisch störende Sprung des kompletten Instruments nicht auf. Um dies zu erreichen, müssen nicht nur die Knotenpunkte, sondern auch die Tangenten der Einzelsplines bis zur Verzweigung identisch sein.

Es bedeutet für den Benutzer einen erheblichen Aufwand, wenn er bei der Definition der Splines manuell darauf achten muss, dass Knotenpunkte und Tangenten identisch sind. Um diesen Aufwand zu minimieren, erzwingt die Software dies automatisch, indem anstelle von separaten Einzelsplines sog. *Splinebäume* zum Einsatz kommen. Obwohl das dahinterstehende Konzept an sich nicht überraschend ist und vermutlich in der Vergangenheit schon verwendet wurde, konnte trotz intensiver Recherchen kein Hinweis auf frühere Verwendungen gefunden werden. Die Idee ist die folgende:

Wie oben beschrieben gibt es bei einem gewöhnlichen Bézierspline eine geordnete Folge von Knotenpunkten, die durch diesen Spline interpoliert werden. Insbesondere hat dabei jeder Knotenpunkt, mit Ausnahme des ersten und des letzten, genau einen Vorgänger und einen Nachfolger. Bei einem *Splinebaum* hat nun jeder Knoten außer dem Wurzel- und den Blattknoten genau einen Vorgänger, aber beliebig viele Nachfolger. Dabei kann

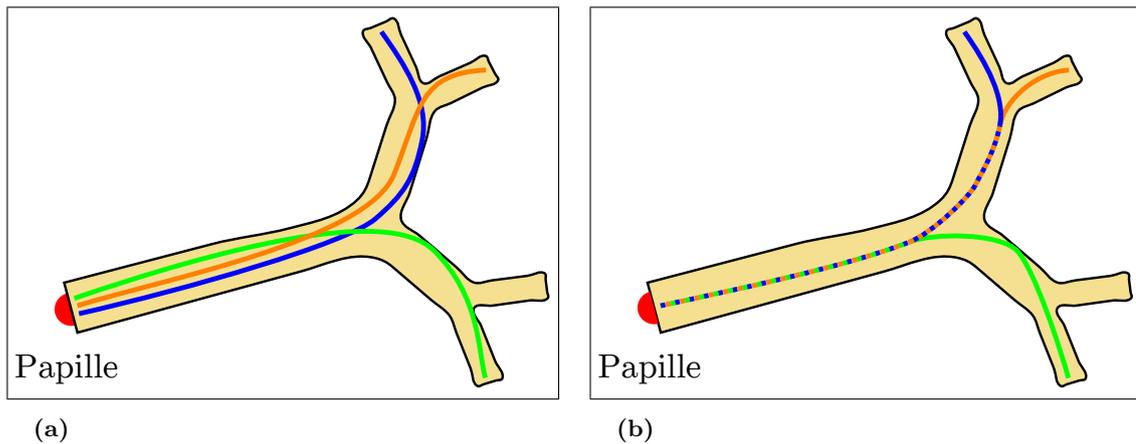


Abb. 4.27.: a) Drei einzelne Splines. b) Splinebaum mit drei Blättern.

jeder der Einzelpfade des Splinebaums als eigener Spline betrachtet werden (Abb. 4.28). Seine Knotenpunkte und Tangenten müssen gemäß obiger Forderung immer so festgelegt werden, dass alle Einzelsplines bis zu einer Verzweigung denselben Verlauf nehmen. Die Software erzwingt diese Festlegung automatisch.

Im Folgenden werden Kanten des Splinebaums als *Splinekanten* und Knoten, aus denen mehr als eine Splinekante hinausläuft, als *Verzweigungsknoten* bezeichnet. Wie bei einem gewöhnlichen Bézierspline hat jeder Knoten genau eine Tangente, welche bei jeder Verschiebung eines Knotens automatisch neu berechnet wird. Eine Verschiebung von Knoten kommt bei zwei Gelegenheiten vor:

1. Während der anfänglichen Definition des Splinebaums müssen die Knotenpunkte in der Regel oft verschoben werden, bis der Verlauf der Pfade gut aussieht.
2. Obwohl der Splinebaum statisch vorgegeben ist, kann dem Verlauf des Instruments eine gewisse Dynamik verschafft werden, indem Knotenpositionen in Abhängigkeit von Sensorwerten geringfügig verschoben werden. In Abschnitt 4.5.6 wird dies genauer beschrieben.

Bestimmung der Tangenten in den Verzweigungsknoten

Da jeder Knotenpunkt des Splinebaums genau eine Tangente haben soll, muss für einen Verzweigungsknoten auf plausible Weise eine Tangente festgelegt werden. Dazu wird zunächst für jede ausgehende Splinekante unabhängig von den übrigen Kanten eine Tangente gemäß des in 4.5.1 beschriebenen Verfahrens ermittelt. Nachdem auf diese Weise für jede auslaufende Splinekante eine individuelle Tangente bestimmt wurde, wird anschließend die Tangente des Verzweigungsknotens durch Bildung eines gewichteten Mittelwerts über die Einzeltangenten berechnet. Dieser kann im einfachsten Fall das arithmetische

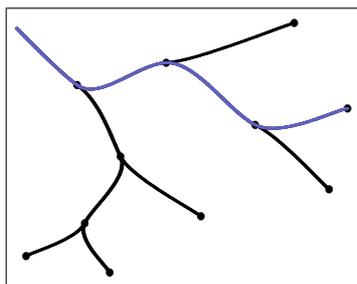


Abb. 4.28.: Ein einzelner Pfad innerhalb des Splinebaums.

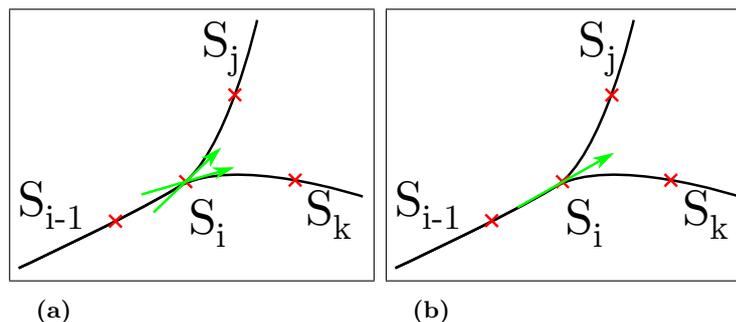


Abb. 4.29.: Berechnung der Tangenten in den Verzweigungsknoten eines Splinebaums. a) Berechne zunächst alle Tangenten einzeln. b) Bilde nun den Mittelwert über diese Tangenten.

Mittel sein, es können jedoch auch für jede Kante unterschiedliche Gewichte verwendet werden. Der Sinn einer unterschiedlichen Gewichtung wird später in Abschnitt 4.5.6 deutlich werden. Abb. 4.29 skizziert die Tangentenberechnung.

4.5.5. Auswahl des richtigen Pfades

Obwohl der Splinebaum stets jede mögliche Instrumentenlage innerhalb des Gallenbaums enthält, repräsentiert natürlich zu jedem Zeitpunkt nur ein bestimmter Pfad des Splinebaums die tatsächliche Lage des Instruments. Es ist also notwendig, zu jedem Zeitpunkt genau den Pfad auszuwählen, dessen Verlauf am besten mit dem Verlauf des realen Instruments übereinstimmt. Diese Auswahl muss aufgrund der Sensorwerte getroffen werden. Sobald der richtige Pfad ermittelt wurde, kann dieser, wie in Abschnitt 4.5.3 für einen entarteten Gallenbaum mit nur einem Pfad beschrieben wurde, in Abhängigkeit von den Sensorwerten eingezeichnet werden. Ein erster Ansatz für die Auswahl besteht darin, den richtigen Pfad einfach anhand des Sensors mit dem größten Signalpegel auszuwählen:

1. Um unempfindlich gegenüber Sensorrauschen zu sein, wird zunächst eine Liste M_P der Pfade erstellt, deren Sensoren einen Pegel über einer kleinen Schwelle ε haben.
2. Wenn in M_P zwei Pfade P_a und P_b existieren, wobei P_a Teilpfad von P_b ist, kann P_a nicht der gesuchte Pfad sein. Daher werden aus der Menge M_P alle Pfade aussortiert, die Teilpfad eines anderen Pfades in M_P sind. Hierbei bedeutet „Teilpfad sein“, dass die Menge der Knoten von P_a eine echte Untermenge der Knotenmenge von P_b ist.
3. Unter den übrig gebliebenen Pfaden wird jetzt derjenige ausgewählt, dessen Sensor den größten Wert liefert und als richtiger Pfad definiert.

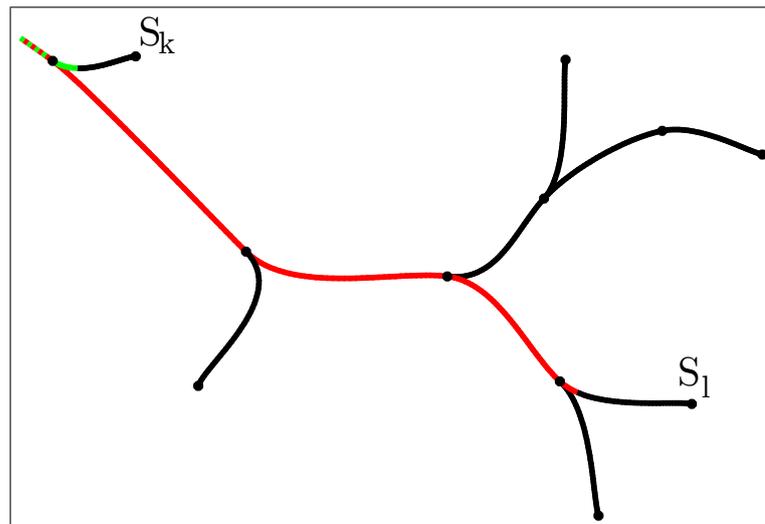


Abb. 4.30.: Unsicherheiten bei der Auswahl des richtigen Pfades. Winzige Schwankungen der Sensorwerte S_k und S_l bewirken, dass entweder der grüne oder der rote Pfad eingezeichnet werden.

Dieser erste Ansatz funktioniert in der Praxis in den meisten Fällen recht gut, bereitet jedoch in manchen Situationen Probleme.

Probleme des ersten Ansatzes

Weil in Schritt 3 nur der Sensor mit dem größten Pegel für die Auswahl des richtigen Pfades betrachtet und nicht der komplette Verlauf berücksichtigt wird, kann es zu starken Sprüngen in der Darstellung kommen. Folgende in Abb. 4.30 skizzierte Beispielsituation soll dies verdeutlichen: Die beiden Sensoren S_k und S_l haben beide denselben, niedrigen Sensorwert. Gemessen anhand der Anzahl an Knoten, die bis dorthin durchlaufen werden, liegt der Sensor S_l jedoch von der Wurzel, also der Papille, aus gesehen, weit weg, während S_k direkt an der Wurzel liegt. Im Phantom könnte S_k zum Beispiel dem Sensor des Pankreasgangs, der direkt hinter der Papille abzweigt, entsprechen. Keiner der beiden Pfade, die von der Wurzel zu diesen Sensoren laufen, ist Teilpfad des anderen, sodass beide bis zu Schritt 3 übrig bleiben.

Da beide Sensoren nach Voraussetzung denselben Sensorwert liefern, ist das Ergebnis der Auswahl des richtigen Pfades zunächst undefiniert. Hier hängt die Wahl von der konkreten Implementierung ab. Dies ist aber noch nicht problematisch. Problematisch wird es, wenn Schwankungen in den Sensorwerten aufgrund von Sensorrauschen oder Bewegungen des Instruments durch den Trainierenden ins Spiel kommen. Durch solche Schwankungen kann die Wahl des Pfades plötzlich wechseln, was sich durch sehr störende Sprünge und letztendlich falsche Darstellung des Instruments ausdrückt. In Abb. 4.30

würde die Darstellung ständig zwischen dem roten und dem grünen Pfad springen, wobei der rote Pfad höchstwahrscheinlich der korrekte wäre.

Diese Situationen kommen in der Praxis immer dann vor, wenn das Instrument soweit vorgeschoben wurde, dass es gerade einen neuen Sensorabschnitt betreten hat und die Sensorwerte dieses neuen Abschnitts dementsprechend sehr niedrig sind. Wurde das Instrument bereits ein Stück in das Segment eingeschoben, sind diese Sprünge in der Regel selten, können aber weiterhin auftreten, wenn z.B. durch Schlaufenbildung das Instrument ein Stück in Richtung des Pankreasgangs gedrückt wird und dadurch dessen Sensor einen erhöhten Wert liefert.

Ein verbessertes Auswahlverfahren

Die störenden Sprünge im oben beschriebenen Ansatz rühren vor allem daher, dass in Schritt 3 unter allen bis dahin übrig gebliebenen Pfaden nur nach dem Pfad mit dem größten Sensorwert gesucht wird. Tatsächlich sollten aber auch die Sensoren der Teilpfade eines Pfades, die in Schritt 2 aussortiert wurden, eine Rolle bei der Auswahl spielen. Im Fall des korrekten Pfades liefern diese ebenfalls große Sensorwerte, während es im Fall eines falschen Pfades welche gibt, die nur sehr kleine Werte liefern. Aus diesem Grund wurde der Algorithmus verbessert, indem ein Punktesystem zur Bewertung einzelner Pfade integriert wurde. Die Schritte 1 und 2 sind identisch mit denen des ersten Algorithmus:

1. Um unempfindlich gegenüber Sensorrauschen zu sein, wird zunächst eine Liste M_P der Pfade erstellt, deren Sensoren einen Pegel über einer kleinen Schwelle ε haben.
2. Existieren in M_P zwei Pfade P_a und P_b , wobei P_a Teilpfad von P_b ist, so kann P_a nicht der gesuchte Pfad sein. Daher werden aus der Menge M_P alle Pfade aussortiert, die Teilpfad eines anderen Pfades in M_P sind.
3. Zur Ermittlung des „Siegerpfades“ wird jetzt für jeden übrig gebliebenen Pfad P_i aus M_P ein Punktestand berechnet. Dazu werden für jedes P_i die Sensorwerte aller Teilpfade von P_i betrachtet. Wenn der Sensorwert eines solchen Teilpfades über einer Schwelle σ liegt, wird der Punktestand inkrementiert, liegt er darunter, wird er dekrementiert.
4. Nun wird unter allen Pfaden in M_P der Pfad ausgewählt, der den höchsten Punktestand hat. Erst bei Punktegleichheit wird der tatsächliche Sensorwert als sekundäres Vergleichskriterium benutzt. Haben alle Pfade einen Punktestand ≤ 0 , wird kein Pfad ausgewählt, weil dies bedeutet, dass kein Instrument eingeführt wurde.

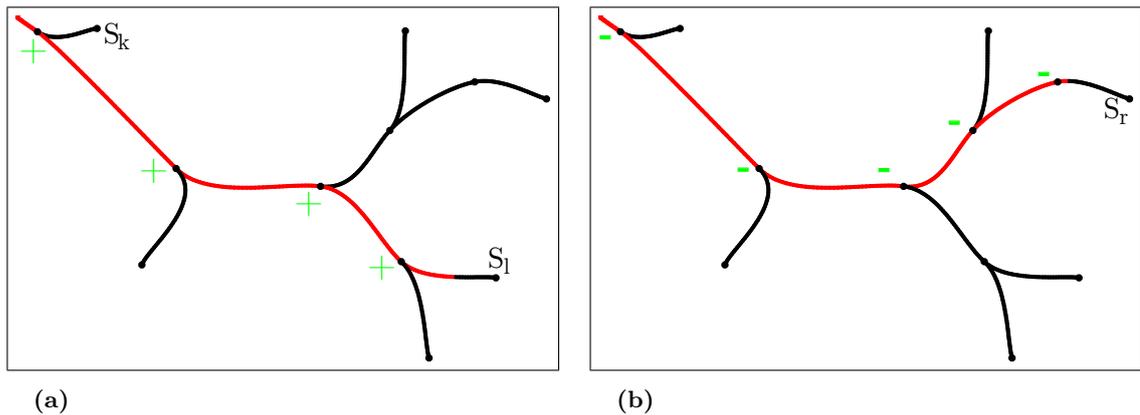


Abb. 4.31.: Auswahl der korrekten Pfade über ihre Punktstände. a) Der Pfad zu S_l wird gegenüber dem zu S_k klar bevorzugt, da er mehr „Zeugen“ hat. b) Der Pfad zu S_r wird nicht gewählt, da er einen negativen Punktstand hat.

Durch die Auswahl per Punktstand können Teilpfade sozusagen als „Zeugen“ dafür dienen, dass ein Pfad korrekt ist, oder auch im Gegenteil „dagegen aussagen“. Abb. 4.31a skizziert eine solche Situation, in der ein Instrument in das Gallengangsegment eingebracht wurde, das zu dem Sensor S_l gehört, während es natürlich nicht gleichzeitig in dem Pfad liegen kann, der zu S_k führt. Alle Sensoren auf dem Weg von der Papille bis zu S_l haben in dieser Situation natürlich ebenfalls ein Signal, welches über der Schwelle liegt, sodass sie alle „bestätigen“ können, dass in ihrem Segment ein Instrument liegt. Daher hat S_l einen Punktstand von vier, während der Knoten S_k nur einen Punkt hat, der durch das kurze Segment geliefert wird, auf dem die beiden Pfade zu S_k und S_l identisch sind.

Mit diesem Ansatz können auch durch Rauschen oder Fehlkalibrierung verursachte Fehler elegant behoben werden. In Abb. 4.31b wird eine Situation skizziert, in der kein Instrument in das Gallengangsystem eingeführt wurde, aber durch zufälliges Sensorrauschen der Sensor S_r kurzzeitig einen Pegel oberhalb des notwendigen Schwellwerts ε erreicht. Mit dem zuerst skizzierten einfachen Algorithmus würde der in der Abbildung rot dargestellte Pfad fälschlicherweise kurzzeitig eingezeichnet werden. Mit dem verbesserten Ansatz jedoch hat der Pfad, an dessen Ende S_r liegt, einen Punktstand von -5 , da es keine „Zeugen“ auf dem Weg zu S_r gibt, die „bestätigen“, dass ein Instrument vorliegt, während es fünf Zeugen gibt, die dies „widerlegen“. Dieser negative Punktstand führt dazu, dass der Pfad nicht eingezeichnet wird.

4.5.6. Verschiebung von Knotenpunkten in Abhängigkeit von Sensorwerten

Bei der Konstruktion der Splinebäume wurde darauf geachtet, dass alle Knotenpunkte jederzeit verschoben werden können und die Tangenten automatisch neu berechnet wer-

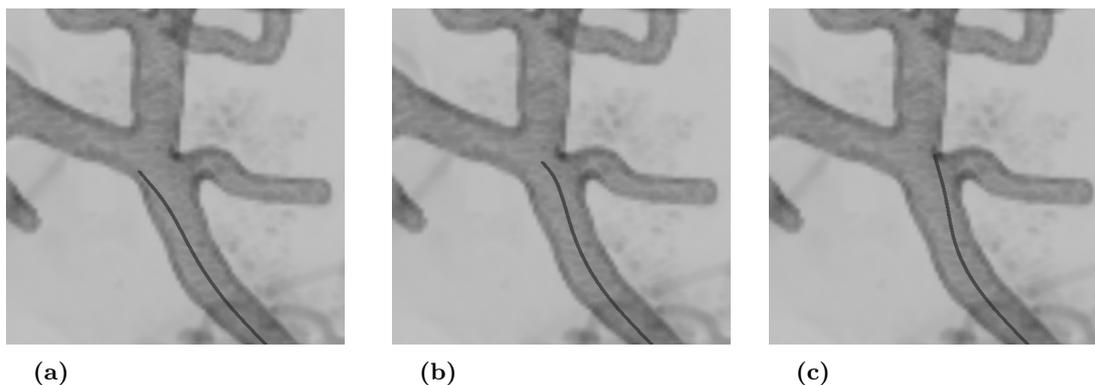


Abb. 4.32.: Pseudo-dynamischer Verlauf des Instruments. a)+b) Das Instrument wird in den linken Zweig eingeführt. Dabei ist a) mit Verschiebung des Knotenpunkts in Richtung des linken Zweigs und b) ohne Verschiebung des Knotenpunkts. In c) wird das Instrument in den mittleren Zweig eingeführt und der Knotenpunkt in Richtung dieses Zweigs verschoben.

den. Diese Eigenschaft kann dazu benutzt werden, dem prinzipiell statisch vorgegebenen Verlauf des Instruments eine gewisse Dynamik zu verschaffen, sodass sich der Verlauf innerhalb gewisser Grenzen verändern kann und so zumindest an Verzweigungen in plausibler Art und Weise dem Verlauf des Instruments in der Realität nachempfunden werden kann. Dazu folgende Überlegung:

Verschiebung der Knotenpunkte

Benachbarte Elektroden auf dem Gallengangsystem sind stets überlappend angebracht. Wird das Instrument eingeführt und nähert sich einer Verzweigung, bekommen daher alle Zweige ein gewisses Signal ab. Dieses ist für einen Zweig umso stärker, je stärker der Verlauf des Instruments eine Tendenz in Richtung dieses Zweigs hat. Es ist daher plausibel, den Verzweigungsknoten ein wenig in Richtung desjenigen Zweiges zu verschieben, dessen Signalpegel unter allen Zweigen der größte ist. Um bei der Auswahl des stärksten Signalpegels keine harten Ja/Nein-Entscheidungen treffen zu müssen, die gerade in Anwesenheit des unvermeidlichen Sensorrauschens zu Sprüngen bei der Verschiebung des Verzweigungsknotens führen können, geschieht dies auf eine sanfte Art und Weise: Der Verzweigungsknoten wird von jedem Zweig angezogen, wobei die Stärke der Anziehung vom Sensorwert des Zweigs relativ zur Summe der Sensorwerte aller Zweige abhängt.

Neigung der Tangente

Um einen realitätsnahen Verlauf des Splines zu erreichen, sollte die Tangente ebenfalls ein wenig in Richtung des Zweiges mit dem stärksten Signalpegel geneigt werden, was leicht möglich ist, da, wie in Abschnitt 4.5.4 beschrieben wurde, bei der Berechnung der

endgültigen Tangente eines Verzweigungsknotens ein gewichteter Mittelwert über alle Einzeltangenten berechnet wird. Es müssen also lediglich die Gewichte in Abhängigkeit von den Signalpegeln gewählt werden. Die Neigung der Tangente ergibt sich dann automatisch. Die Abb. 4.32a und 4.32b zeigen, wie sich die Verschiebung des Knotenpunktes und die Neigung der Tangente subtil auf den Verlauf des Splines auswirkt.

Obwohl dieser pseudo-dynamische Verlauf des Instruments durchaus plausibel ist, kann natürlich keineswegs garantiert werden, dass das Instrument tatsächlich den dargestellten Verlauf hat. Jedoch bietet die Dynamik dem Benutzer in jedem Fall eine gute Hilfestellung, da er während des Vorschubs schon recht gut abschätzen kann, in welche Richtung sich das Instrument bewegen wird, was bei einem statischen Kontrollpunkt nicht möglich wäre.

4.6. Optische Erfassung von Instrumenten

Die zweite Ausbaustufe des Biliphanten richtet sich in ihrem Funktionsumfang an diejenigen Trainingsteilnehmer, die über das Anfängerstadium hinaus gelangt sind und denen die Navigation des Duodenoskops keine größeren Probleme mehr bereitet. Diese Teilnehmer sollen am Biliphanten ihre Fähigkeiten, tatsächliche therapeutische Eingriffe durchzuführen, verfeinern und entsprechende Verfahren trainieren können. Wie in Abschnitt 4.4 erläutert wurde, ist die kapazitive Sensorik, die in den Anfängerphantomen zum Einsatz kommt, nicht präzise genug, um feinere Details oder die genaue Lage von eingeführten Instrumenten zu erfassen. Für das Training fortgeschrittener Methoden sind eine exakte Lokalisierung und Darstellung der Instrumente aber unerlässlich. Es wurde daher für die Fortgeschrittenen-Phantome ergänzend zur kapazitiven eine andersartige Sensorik entwickelt, mit der feinste Details erfasst werden können.

Die Grundidee beruht auf der einfachen Überlegung, dass sich die Bilderzeugung eines realen Röntgenbildes z.B. durch einen C-Bogen in vielen wesentlichen Eigenschaften kaum von der Erzeugung eines Schattenbildes durch eine optische Kamera unterscheidet, bei dem ein Objekt von der einen Seite her beleuchtet wird und die Kamera von der anderen Seite auf dieses Objekt schaut. Sowohl Röntgengerät als auch optische Kamera liefern eine Projektion des aufgenommenen dreidimensionalen Objekts auf eine zweidimensionale Bildebene.

Es stellt sich nun die Frage, ob man diese Ähnlichkeit der Aufnahmeverfahren ausnutzen kann, um mittels einer Kamera ein röntgenähnliches Bild zu erzeugen. Dabei ergibt sich sofort ein Problem.

Problem

Röntgenstrahlen sind in der Lage, viele Materialien zu durchdringen. Es ergeben sich zwar Abschwächungseffekte (vgl. Abschnitt 4.3.1), jedoch kommt in der Regel am Detektor noch messbare Strahlung an. Elektromagnetische Strahlung im optischen Bereich reagiert jedoch anders auf die meisten Materialien. Zwar ist z.B. Latex auch für diesen Frequenzbereich transparent, jedoch nur bei sehr geringen Materialstärken. Die Gallengänge des Biliphanten weisen eine Wandstärke von $\geq 1 \text{ mm}$ auf und sind damit bereits fast vollständig opak. Aus diesem Grund ist es nicht ohne Weiteres möglich, simulierte Röntgenbilder der Gallengänge mit einer optischen Kamera zu erzeugen. Zur Lösung dieses Problems mussten einige Modifikationen am Aufbau des Biliphanten vorgenommen werden.

4.6.1. Modifikationen am Aufbau

In den Prototypen des Biliphanten mit optischer Sensorik wurde eine Kamera mit Blick auf das Gallengangssystem eingebaut. Damit die Kamera ein eingeführtes Instrument überhaupt „sehen“ kann, muss der Teil des Gallengangsystems, der unter optischer Sensorikontrolle stehen soll, für sichtbares Licht transparent sein. Im ersten Prototypen wurde dies dadurch erreicht, dass ein Teil des Gallengangsystems durch transparente PVC-Schläuche ersetzt wurde (siehe Abschnitt 4.9.1). Dieser ersetzte Teil beinhaltet eine Y-Verzweigung und hat eine planare Gestalt, was bedeutet dass alle drei Gangabschnitte in einer Ebene liegen.

Diese Einschränkung wird voraussichtlich auch für zukünftig zu entwickelnde, verbesserte Gangsysteme gelten, da es ansonsten sehr schwierig wird, die Kamera so einzubauen, dass das komplette Gangsystem gut gesehen werden kann. Die Kamera wurde so montiert, dass sie genau senkrecht auf die Ebene blickt, in der die Verzweigung liegt. Auf der der Kamera entgegengesetzten Seite dieser Ebene befindet sich eine flächenhafte Lichtquelle, deren Licht für die Bildung des Schattenbildes der eingeführten Instrumente gebraucht wird. In Abschnitt 4.6.2 wird näher auf dieses Schattenbild und seine Verwendung eingegangen. Abbildung 4.33 zeigt diesen Prototypen. Bei den roten Kügelchen, die an transparenten Stielen angebracht wurden, handelt es sich um Kalibrierungsmarker. Ihr genauer Zweck wird in den folgenden Abschnitten geklärt.

Unglücklicherweise haben die PVC-Schläuche nicht dieselben Flexibilitätseigenschaften wie Latex, sondern sind deutlich starrer. Deshalb ist die Erforschung und Erprobung von Materialien und Herstellungsmethoden, mit denen sich Gallengänge mit ähnlicher Flexibilität wie Latex und gleichzeitiger Transparenz herstellen lassen, ein aktuelles Forschungsthema. In Abschnitt 4.9 wird näher darauf eingegangen. Für die Entwicklung der

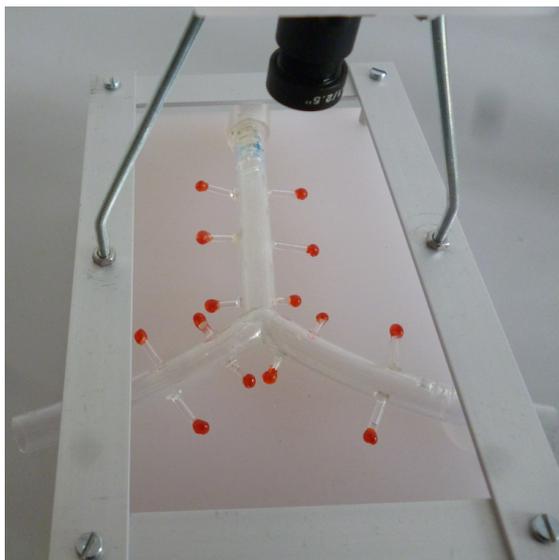


Abb. 4.33.: Modifiziertes, transparentes Gallengangsystem. Dieser einfache Prototyp besteht lediglich aus einer Y-Verzweigung. Die roten Kügelchen dienen als Marker.

im Folgenden vorgestellten Algorithmen spielt dies aber keine wesentliche Rolle; vielmehr soll die grundsätzliche Machbarkeit und der daraus zu gewinnende Nutzen gezeigt werden, wofür sich der Prototyp aus PVC-Schläuchen grundsätzlich eignet.

Die Elektroden, die in den Anfängerphantomen die kapazitive Erfassung der Instrumentposition ermöglichen, sind bei einer optischen Erfassung sehr hinderlich, da sie nicht durch transparente Gegenstücke ersetzt werden können und damit den ungehinderten Blick der Kamera auf das darunterliegende Gallengangsystem verhindern. Diese Sensoren sind aber in dem unter optischer Kontrolle stehenden Bereich gar nicht notwendig, da die optische Erfassung ohnehin wesentlich genauere Ergebnisse liefert. Daher können die kapazitiven Sensoren an den transparenten Gängen weggelassen werden. An den intransparenten Bereichen, die weiterhin aus Latex bestehen, können die kapazitiven Sensoren weiterhin, wie in Abschnitt 4.4 beschrieben wurde, eingesetzt werden. Die Übergangsstellen zwischen dem Originalbaum aus Latex und dem transparenten Bereich müssen sehr sorgfältig gearbeitet werden, damit dort keine nennenswerten Grate entstehen, an denen eingeführte Instrumente hängenbleiben könnten.

Je nach verwendeter Kameraoptik ist ein Mindestabstand zwischen Kamera und Gallengangsystem einzuhalten, der durch die Brennweite der Kamera vorgegeben wird. Ist die Kamera zu nah an den Gängen, kann nicht mehr der komplette gewünschte Bereich beobachtet werden und das Bild wird unscharf. Für die Optik, die im Prototypen eingesetzt wird, beträgt diese Entfernung ca. 5 cm. Dazu kommt die Höhe der Kamera, die ca. 3 cm beträgt, sodass der Gesamtaufbau ca. 8 cm benötigt. Dadurch wird das Phantom ein wenig dicker als üblich, was bei der Konstruktion des Chassis berücksichtigt werden muss. Die verwendete Linse hat eine relativ geringe Brennweite von 3,6 mm, wodurch

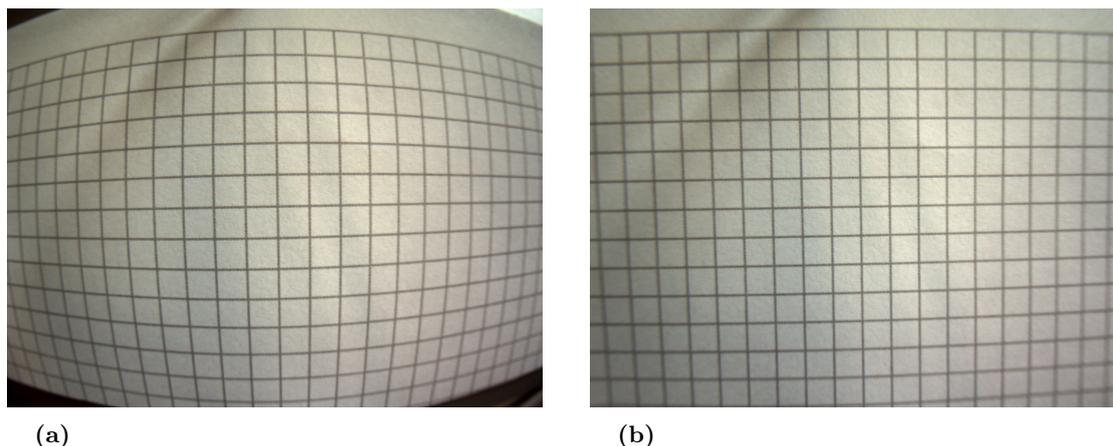


Abb. 4.34.: Korrektur des „Fischaugen“-Effekts. a) ohne und b) mit dieser Korrektur.

sich ein deutlicher „Fischaugen“-Effekt einstellt. Dabei wird das Bild zu den Rändern hin „tonnenförmig“ verzerrt.

Dieser Effekt lässt sich aber durch Anwendung einer entsprechenden Transformation weitestgehend rückgängig machen. Für die Methoden der Bildverarbeitung und -aufbereitung wurde in diesem Modul die Bibliothek OpenCV eingesetzt. Diese bietet die Möglichkeit, die Verzerrungsparameter einer gegebenen Linse mittels eines Kalibrierungsbildes zu ermitteln und kann diese Verzerrung anschließend anhand der ermittelten Parameter rückgängig machen. Abb. 4.34 stellt ein auf diese Weise korrigiertes Bild einem nicht korrigierten Bild gegenüber. Durch die Korrektur gehen zwangsläufig Bereiche des Bildrandes verloren, was jedoch für die beschriebene Anwendung nicht von Bedeutung ist.

4.6.2. Die Erkennung der eingeführten Instrumente

Das Bild der Kamera wird mit derselben Frequenz eingelesen, mit der das virtuelle Röntgenbild aktualisiert werden soll (in der Regel 2 Hz oder 10 Hz). Abbildung 4.35 zeigt ein typisches Bild des Y-förmigen Prototypen, wie es durch die Kamera gesehen wird. 4.35a zeigt nur den leeren Gang, während 4.35b ein eingeführtes *Dormia-Körbchen* zeigt. Die Erkennungsaufgabe besteht nun im Wesentlichen darin, die Pixel im Bild zu finden, die zu dem eingeführten Instrument gehören. Diesen Vorgang bezeichnet man als *Segmentierung*.

Globale Schwellwertentscheidung

Zur Lösung dieser Aufgabe bieten sich verschiedene Ansätze an. Eine naive Vorgehensweise besteht darin, das Bild mittels einer globalen Schwellwertentscheidung zu binarisieren,

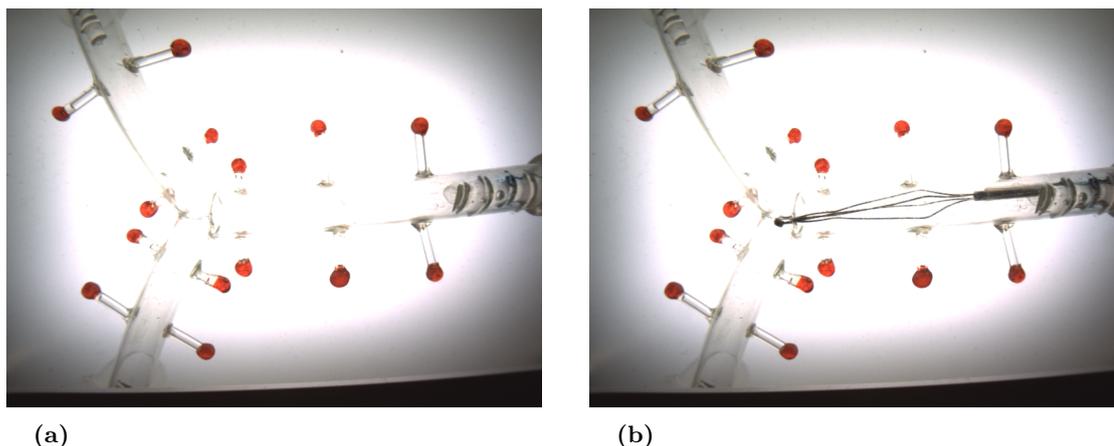


Abb. 4.35.: Rohdaten der optischen Erkennung. a) ohne und b) mit eingeführtem Instrument.

indem man sich eine Schwelle vorgibt und jedes Pixel im Bild, dessen Wert unter der Schwelle liegt, als dem Instrument zugehörig definiert. Im Fall von RGB-Farbbildern muss die Schwellwertentscheidung geeignet definiert werden, z.B. indem man das Bild zunächst vom RGB- ins HSV-System umwandelt und die Schwelle dann dort auf den *Value*-Kanal, der die Helligkeit eines Pixels angibt, anwendet.

Eine globale Schwellwertentscheidung funktioniert jedoch nur unzureichend. Betrachtet man Abbildung 4.36a, so entsteht zunächst der Eindruck, dass dieses Verfahren gut funktionieren müsste, weil die Pixel des Instruments scheinbar dunkler sind als die übrigen Pixel. Betrachtet man jedoch die Ausschnittsvergrößerung in Abbildung 4.36b, so fällt auf, dass durchaus viele der Pixel des Instruments eindeutig dunkler sind als andere Pixel, die nicht zum Instrument gehören. Zwischendurch gibt es jedoch auch viele Pixel, die ein menschlicher Betrachter eindeutig dem Instrument zuweisen würde, z.B. die Pixel innerhalb der grünen Umrandung, die jedoch tatsächlich heller sind als manche der Pixel in der roten Umrandung, obwohl diese eindeutig nicht zum Instrument gehören.

Dieses Beispiel zeigt, dass eine naive Schwellwertentscheidung, bei der jedes Pixel separat von allen anderen Pixeln betrachtet wird, nicht zum Erfolg führen kann. Ebenso würde eine adaptive Schwelle, bei der der Schwellwert für jedes Pixel anhand der Pixelwerte in seiner Umgebung ermittelt wird, nicht funktionieren. Einerseits würden viele Artefakte wie z.B. die Pixel innerhalb des roten Rahmens in Abb. 4.36b auftreten, andererseits würden aber Pixel des Instruments, die deutlich heller sind als andere Pixel des Instruments in ihrer Umgebung, durch diese Schwelle möglicherweise übersehen.

Verbesserte Schwellwertentscheidung

Das Verfahren wurde daher modifiziert: Das Bild wird zunächst mittels einer sehr niedrigen Schwelle s_{res} , die nur wenige Pixel als dem Instrument zugehörig erkennt, segmentiert.

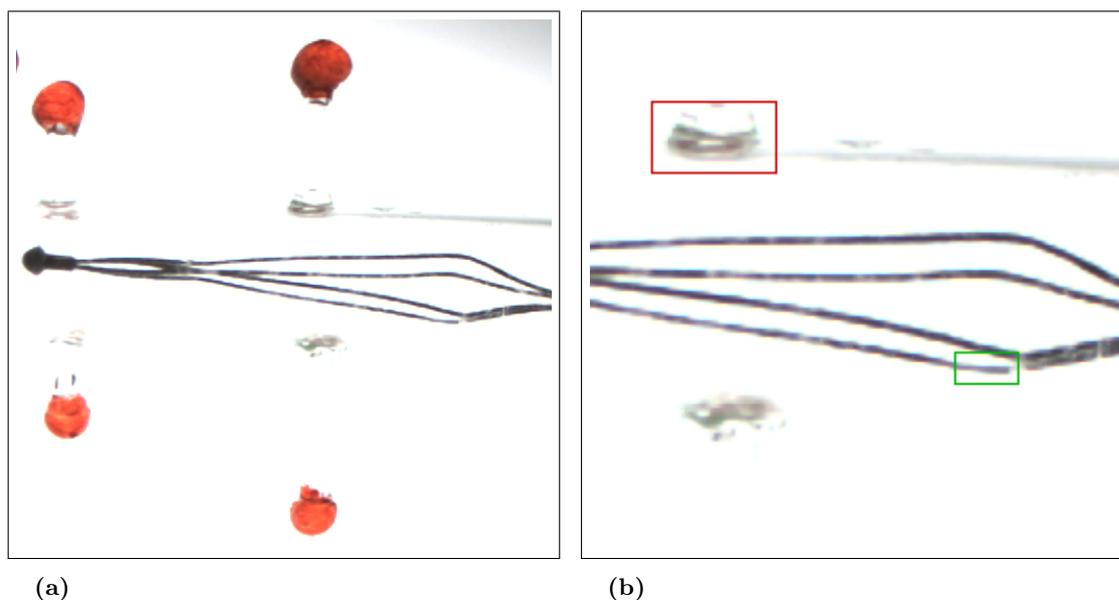


Abb. 4.36.: Globale Schwellwerte sind zur Segmentierung der Instrumente nicht geeignet. a) Bild eines Dormia-Körbchens. b) Nahaufnahme dieses Bildes.

Durch die restriktive Schwelle sind in diesem Schritt *false positives*, also fälschlicherweise als dem Instrument zugehörig erkannte Pixel, praktisch ausgeschlossen. Anschließend wird in der Nachbarschaft der so gefundenen Pixel mit einer weniger restriktiven Schwelle s_{lax} nach weiteren Instrumentpixeln gesucht. In der Praxis wird die Nachbarschaft bestimmt, indem die ursprünglich erkannte Pixelmenge mittels Dilatation „aufgebläht“ wird. Diese vergrößerte Pixelmenge dient dann als Maske, mit deren Hilfe entschieden wird, in welchen Bereichen des Bildes die weniger restriktive Schwelle angewendet wird. Damit das Bild an den Rändern von Strukturen keine abrupten Übergänge zeigt, wird diese zweite Schwelle nicht einfach als binäre ja/nein-Schwelle benutzt, sondern es wird eine weiche Entscheidung durchgeführt. Dazu wird die ursprünglich binäre Maske mittels eines einfachen Gauß-Filters geglättet und die so erhaltenen Zwischenwerte s_{smooth} als Gewichtungsfaktoren der Pixelwerte benutzt: Auf ein Pixel innerhalb der Maske wird zunächst die Schwelle s_{lax} angewendet und das binäre Ergebnis wird anschließend mit s_{smooth} multipliziert, um am Rand der Maske einen weichen Übergang zu erzeugen.

In der Praxis funktioniert dieses Verfahren recht gut; es gibt allerdings auch hier einige Probleme:

- Verunreinigungen des transparenten Gallengangs werden mit diesem Verfahren ebenfalls als zu dem Instrument gehörig klassifiziert.
- In der Praxis ist die Hintergrundbeleuchtung nie vollkommen gleichmäßig. Eine ungleichmäßige Beleuchtung bereitet aber stets Probleme, wenn Schwellwertentscheidungen getroffen werden sollen.

- Nicht alle Arten von Instrumenten erzeugen Schattenbilder, die dunkel genug sind, um sie mittels Schwellwertentscheidungen erkennen zu können. Beispielsweise beinhalten *Dormia-Körbchen* sehr dünne Strukturen, die dadurch auch im Schattenbild mittel- bis hellgrau erscheinen, z.B. in Abb. 4.36b im rot umrandeten Bereich. Ein zweites Beispiel sind Dilatationsballons, die halbtransparent sind und daher gar keine richtigen Schatten werfen.

Anderer Ansatz

Zur Lösung der verbleibenden Probleme wurde zusätzlich noch ein komplett anderer Ansatz entwickelt. Dieser beruht auf der Beobachtung, dass sich ein Bild von dem Gangsystem, in das kein Instrument eingeführt wurde (Abb. 4.35a), von einem Bild mit eingeführtem Instrument (Abb. 4.35b) im Idealfall genau in den Pixeln unterscheidet, die zum Instrument gehören. Daraus lässt sich ein Verfahren ableiten, das ähnlich funktioniert wie eine *Digitale Subtraktionsangiographie* zur radiologischen Darstellung von Gefäßen:

1. Vor dem eigentlichen Beginn der Simulation und ohne dass ein Instrument eingeführt wurde, wird ein *Referenzbild* B_{ref} aufgenommen und in ein Graustufenbild umgewandelt. Die Umwandlung eines Farbtripels (r, g, b) in einen Grauwert v geschieht wie bei der Berechnung des Value-Kanals im HSV-Farbsystem mittels der Vorschrift $v(r, g, b) := \max\{r, g, b\}$. Weil in einem Schattenbild die Unterschiede zwischen den einzelnen RGB-Farbkanälen in der Regel sehr klein sind, gehen dadurch nahezu keine Informationen verloren.
2. Für jedes zu simulierende Röntgenbild wird das aktuelle Kamerabild B_a aufgenommen und wie in Schritt 1 in ein Graustufenbild umgewandelt.
3. Nun wird pixelweise das Differenzbild $B_d(x, y) := B_{ref}(x, y) - B_a(x, y)$ zwischen dem Referenzbild und dem aktuellen Bild berechnet. Das Abbild des Instruments ist stets dunkler als der entsprechende Bereich des Referenzbildes, weil die Beleuchtung von der Rückseite her stattfindet und daher jedes zusätzliche Hindernis, in diesem Fall also das Instrument, bedeutet, dass die Kamera weniger Licht von der Lichtquelle empfängt. Es gilt also $\forall(x, y) : B_a(x, y) \leq B_{ref}(x, y) + \varepsilon$ für ein $\varepsilon > 0$, das addiert wird, um unempfindlich gegenüber dem stets vorhandenen Kamerarauschen zu sein. An den Stellen, an denen auch im aktuellen Bild B_a kein Instrument vorliegt, sind die Pixelwerte abgesehen vom Kamerarauschen nahezu gleich und ihre Differenzen daher nahezu 0.
4. Alle Pixel (x, y) mit $B_d(x, y) > \varepsilon$ können jetzt als dem Instrument zugehörig klassifiziert werden.

4.6.3. Korrektur bei elastischen Deformationen

Ein Problem, welches bei der Berechnung von Differenzbildern immer auftreten kann, besteht darin, dass sowohl die Kamera als auch die gesamte Anordnung der aufgenommenen Szene absolut starr sein müssen, damit es bei der Differenzbildung nicht zu unerwünschten Artefakten kommt. Die Position der Kamera im Phantom kann zwar gut fixiert werden, jedoch ist das Gallengangssystem bis zu einem gewissen Grad elastisch. Artefakte treten selbst bei geringen Verschiebungen der Strukturen zwischen dem Referenzbild und dem aktuellen Bild auf. Erfahrungsgemäß reicht eine Verschiebung einer Struktur im Bild um mehr als fünf Pixel aus, um einen störenden Rand um diese Struktur herum zu bewirken. Kleinere Verschiebungen führen aufgrund von Unschärfefeffekten bei der Aufnahme und der Schwelle ε bei der Differenzbildung noch nicht zu sichtbaren Artefakten.

Bei einer Auflösung von 1280×960 Pixeln, einem Öffnungswinkel nach der Entzerrung des Fischaugen-Effekts von ca. 90° und einem Abstand zwischen Projektionszentrum und Bildebene von 5 cm wird eine 5-Pixel-Abweichung bereits von einer Verschiebung des Gallengangs um $0,4\text{ mm}$ bewirkt. Eine solche Abweichung kommt in der Praxis regelmäßig vor. Aus diesem Grund empfiehlt es sich, die naive Erzeugung des Differenzbildes, wie sie im vorigen Abschnitt 4.6.2 beschrieben wurde, in der Praxis nicht zu verwenden und stattdessen vor der Differenzbildung die im Folgenden beschriebene Korrektur durchzuführen. Abb. 4.37 zeigt exemplarisch, wie sich solche Artefakte äußern können und wie die Bilder aussehen, wenn eine Korrektur durchgeführt wird. Die Artefakte wurden im Differenzbild rot markiert, um sie deutlich hervorzuheben.

Zur Behebung der Artefakte müssen die beiden Bilder B_{ref} und B_a vor der Differenzbildung aufbereitet werden. Dabei wird die geringe Veränderung des Gangsystems im Bild B_a , die letztendlich zu den Artefakten führt, als eine elastische Deformation modelliert. Diese elastische Deformation muss korrigiert werden, bevor eine Differenzbildung sinnvoll durchgeführt werden kann. Obwohl die Deformation streng genommen in drei Dimensionen stattfinden kann, hat sich gezeigt, dass der überwiegende Anteil der Deformation in der Ebene stattfindet, in der das Gallensystem liegt. Daher reicht es aus, die Deformationskorrektur mit Hilfe einer geeigneten elastischen zweidimensionalen Transformation im Kamerabild B_a durchzuführen, bevor das Differenzbild B_d berechnet wird. Die Deformation wird folgendermaßen modelliert:

Sei B_i das (prinzipiell unbekannte) ideale Bild eines nicht deformierten Gallengangs und $R \subset \mathbb{N}^2$ der Definitionsbereich von B_i . Weiter sei B_a das tatsächlich aufgenommene Bild des deformierten Gangsystems. Dann gilt $B_i(x, y) = B_a(T(x, y))$ mit einem stetig differenzierbaren Vektorfeld $T : R \rightarrow \mathbb{R}^2$, das die Deformation in jedem Punkt beschreibt.

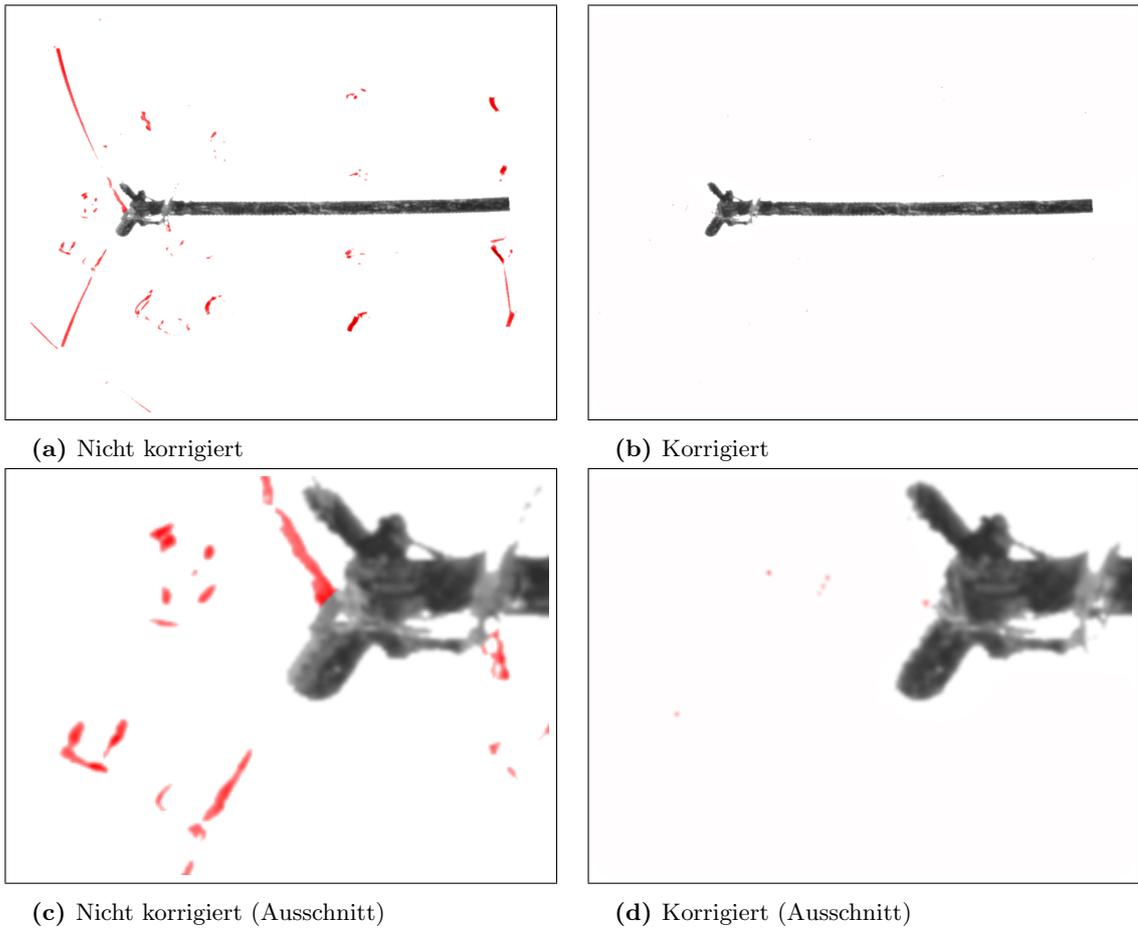


Abb. 4.37.: Bewegungsartefakte im Differenzbild mit einer eingeführten Zange und ihre Korrektur. Zur besseren Vergleichbarkeit mit realen Röntgenbildern wurden die Farben der Bilder invertiert. Die Artefakte sind in a) und c) rot markiert.

Das Differenzbild B_d berechnet sich demnach mittels

$$\begin{aligned} B_d(x, y) &:= B_{ref}(x, y) - B_i(x, y) \\ &= B_{ref}(x, y) - B_a(T(x, y)) \end{aligned} \quad (4.8)$$

B_a ist ein Bild mit diskretem Definitionsbereich. Der Punkt $T(x, y) \in \mathbb{R}^2$ kann jedoch durchaus zwischen vier Pixeln liegen. In diesem Fall muss der Wert des Pixels $B_a(T(x, y))$ z.B. durch bilineare Interpolation aus den Werten der vier umliegenden Pixel berechnet werden.

Bestimmung der Deformation

Es kann also bei bekanntem T die Deformation jedes aufgenommenen Bildes B_a korrigiert werden. Das Problem besteht darin, dass T im Allgemeinen nicht *a priori* bekannt ist,

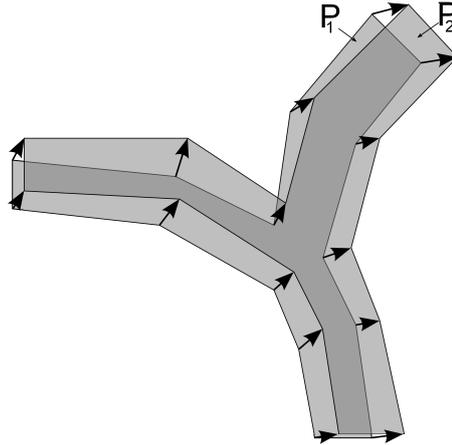


Abb. 4.38.: Verschiebung zwischen P_1 und P_2 .

weil die Deformation des Gallengangsystems von der Interaktion des Trainierenden mit dem Biliphanten abhängt und sich ständig ändern kann. T muss also *ad hoc* in Echtzeit bestimmt werden. Unter anderem um diese Bestimmung zu ermöglichen, dienen die Marker, die von außen an den Gallengang angebracht wurden (siehe Abb. 4.35). Diese haben eine rötliche Färbung, sind aber dennoch transparent, sodass sie bei einer Beleuchtung von der Rückseite keinen echten Schatten werfen, sondern vielmehr als intensive rötliche Bereiche sichtbar sind (siehe Abb. 4.35, 4.36a). Anhand ihrer charakteristischen Farbe können diese Markierungen im Bild mit einfachen Standardverfahren der Bildverarbeitung wie globalen Schwellwertentscheidungen sicher gefunden werden. Der Algorithmus arbeitet nun wie folgt:

1. Vor der eigentlichen Simulation wird das Referenzbild B_{ref} aufgenommen.
2. Die Marker im Referenzbild werden gesucht.
3. In jedem neuen Bild B_a werden die Marker ebenfalls gesucht.
4. Zwischen jedem Marker M_i in B_{ref} und dem entsprechenden, möglicherweise verschobenen, Marker M'_i in B_a wird eine Zuordnung gesucht.

Damit ist die Abbildung T an den Markerpositionen M_i bereits durch $T(M_i) := M'_i$ definiert. Mittels eines Interpolationsverfahrens lässt sich $T(x, y)$ für alle $(x, y) \in R$ ermitteln. Dazu wird die Folge der Markerpunkte (M_i) als Polygonzug P aufgefasst, der auf den Polygonzug P' mit den Eckpunkten (M'_i) abgebildet werden soll (Abb. 4.38). Jeder Punkt p im Inneren von P soll dabei auf einen Punkt p' im Inneren von P' abgebildet werden, sodass das resultierende Vektorfeld stetig differenzierbar und die Norm des Gradienten $\|\nabla T\|_2$ überall möglichst klein ist, damit keine starken lokalen Verzerrungen auftreten. Für diese Interpolationsaufgabe eignet sich die Verwendung verallgemeinerter

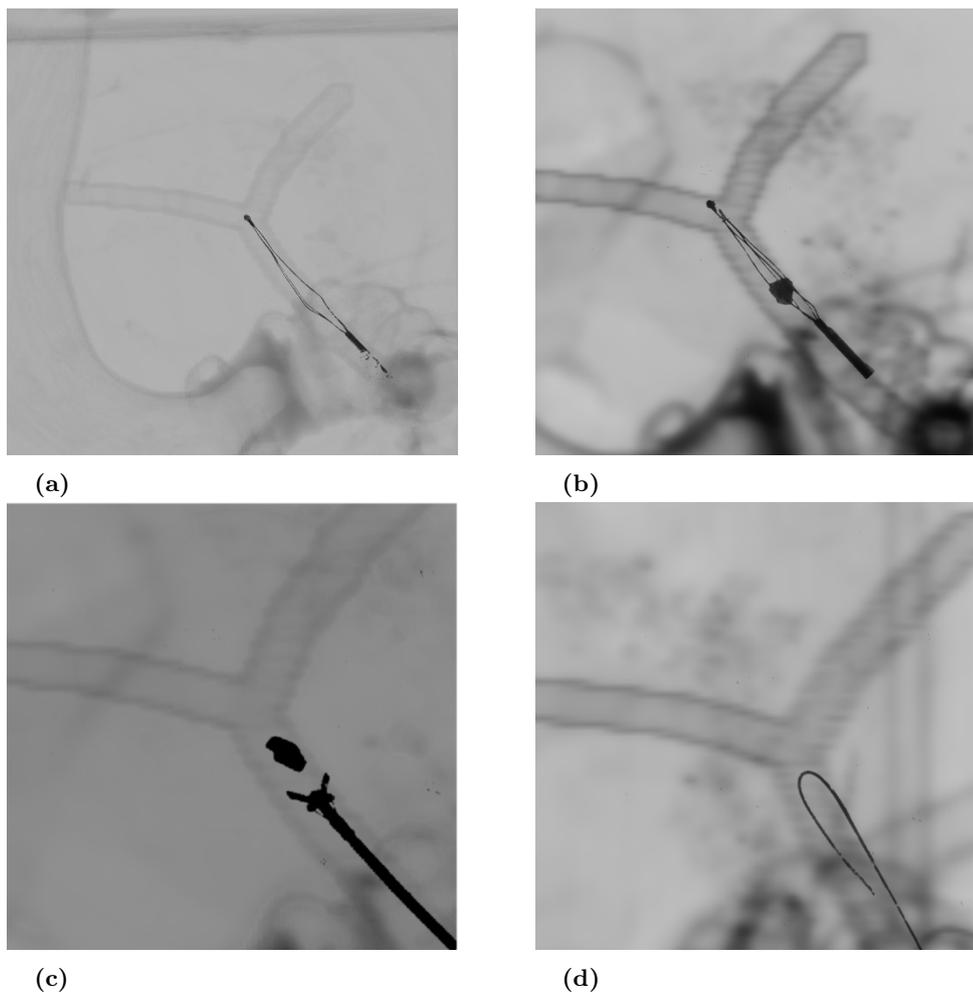


Abb. 4.39.: Beispiele für die optische Erkennung. a) Dormia-Körbchen b) Dormia-Körbchen mit eingefangenen Gallenstein c) Biopsiezange, die sich einem Gallenstein nähert d) Sondierungsdraht, dessen Schlingenbildung deutlich zu sehen ist.

Baryzentrischer Koordinaten als Gewichte. Es gibt diverse Verallgemeinerungen für Baryzentrische Koordinaten auf n -Ecke. Für die Verwendung in *X-Sim* wurden *Mean Value Coordinates [DM]* verwendet.

Mit dieser Korrektur der Deformation funktioniert die Differenzmethode nach Gleichung 4.8 sehr stabil und liefert ausgezeichnete Differenzbilder. Abb. 4.39 zeigt einige Beispiele.

4.6.4. Darstellung des Instruments

Nachdem die Aufgabe der Erkennung der zum Instrument gehörigen Pixel des Kamerabildes gelöst wurde, sollen diese nun ähnlich wie Sondierungsdraht und Papillotom in den Anfängerphantomen als Teil des simulierten Röntgenbildes grafisch dargestellt werden.

Ein sehr einfacher, naiver Lösungsansatz besteht darin, das Bild des Hintergrundes zu berechnen, wie in 4.3.3 beschrieben wurde. Anschließend werden die Pixel des Kamerabildes, die dem Instrument zugeordnet wurden, ähnlich wie beim Anfängerphantom in Abschnitt 4.5.4 eingeblendet. Dabei ergeben sich aber zwei Probleme:

1. Bei der auf Splines basierenden Darstellung des Sondierungsdrahts in den Anfängerphantomen liegen die Informationen über die Punkte des Splines als dreidimensionale Informationen vor. Es ist daher nicht weiter schwierig, diese zusammen mit dem Hintergrund geometrisch korrekt in das Röntgenbild einzufügen. Bei der optischen Erkennung wird durch das Kamerabild aber lediglich ein zweidimensionales Abbild des Instruments aus der Sicht der eingebauten Kamera geliefert, während das simulierte Röntgenbild zunächst einmal aus einem dreidimensionalen Datensatz aus Sicht einer virtuellen Kamera berechnet wird. Die virtuelle Kamera ist aber analog zu einem realen Röntgenapparat dreh- und schwenkbar, sodass es prinzipiell nicht möglich ist, die Bilder der realen Kamera und die aus Sicht der virtuellen Kamera in jeder Situation deckungsgleich übereinander zu legen.
2. Selbst wenn die virtuelle Kamera in ihrem Blickwinkel starr festgelegt wäre, sodass ihr Blickwinkel weitestgehend dem der realen Kamera entspräche, wäre es aus mechanischen und handwerklichen Gründen unmöglich, das Gallengangssystem stets exakt so zu positionieren, wie es zum Zeitpunkt der Erzeugung des 3D-Datensatzes lag. Vielmehr ist eine gewisse Variabilität der Positionierung unvermeidlich.

Aus diesen Gründen kann das Abbild des erkannten Instruments nicht einfach ohne weitere Maßnahmen in das Röntgenbild eingeblendet werden. Der gewählte Lösungsansatz besteht darin, einen „Umweg“ von 2D über 3D zu gehen. Dazu wird zunächst ein Rechteck R definiert, das schräg in den Volumendatensatz V eingepasst wird, sodass es relativ zu V genau so liegt, wie das transparente Stück des Gallengangs, das, wie in Abschnitt 4.6.1 beschrieben wurde, planar ist, relativ zum realen Phantom liegt. Auf dieses Rechteck wird anschließend das Differenzbild B_d , welches gemäß Abschnitt 4.6.2 berechnet wurde, als Textur (vgl. anhang A.1.2) abgebildet. Das Rechteck stellt also gewissermaßen eine „Leinwand“ dar, auf die das Instrument abgebildet wird.

Bestimmung der Lage des Rechtecks relativ zum Phantom

Um die Lage des Rechtecks relativ zum Phantom zu bestimmen, wurden im Volumendatensatz mit Hilfe des Softwarepakets *Eccet* von Dr. A. Beck [Bec03] einige Voxel mit einer eindeutigen ID markiert. Diese sind relativ zum Koordinatensystem des Volumendatensatzes genau an den Positionen, an denen sich die in Abschnitt 4.6.3 beschriebenen roten Marker relativ zum realen Phantom befinden. Anhand dieser Marker lässt sich nun

eine einfache affin-lineare Abbildungsvorschrift ermitteln, mit der sich das Rechteck R geometrisch korrekt in das Koordinatensystem des Volumendatensatzes transformieren lässt.

Die Vorschrift besteht aus der Verkettung von drei elementaren Transformationen: Das Rechteck muss an die richtige Stelle im Volumendatensatz verschoben werden, es muss in die Ebene hineingedreht werden, die durch die Markerpunkte des Volumendatensatzes aufgespannt wird und anschließend muss es noch innerhalb dieser Ebene so ausgerichtet werden, dass jeweils die korrespondierenden Marker aufeinander liegen. Es ist aus Gründen der Einfachheit der Berechnungen zweckmäßig, zunächst die beiden Rotationen anzuwenden und die Ebene erst am Schluss an die richtige Stelle zu verschieben. Damit ergibt sich die Gesamttransformation:

$$M = T \circ R_{outer} \circ R_{inner}$$

Aus der Kenntnis der Markerpositionen im Kamerabild und im Volumendatensatz ist es einfach möglich T , R_{outer} und R_{inner} zu berechnen. Mit der Transformation M kann das Rechteck jetzt an der korrekten Stelle im Raum relativ zum Volumendatensatz positioniert werden und als „Leinwand“ für das Bild dienen, das die Pixel des zuvor erkannten Instruments enthält.

Es bleibt noch eine letzte Korrektur durchzuführen, bevor die Berechnung abgeschlossen werden kann. Es kann sich nämlich noch ein Problem ergeben, das oben bereits angesprochen wurde: Der Volumendatensatz wurde zu irgendeinem Zeitpunkt in der Vergangenheit aufgenommen und ist von da an statisch gegeben. Das reale Phantom ist jedoch bis zu einem gewissen Grad flexibel. Mit Hilfe der oben beschriebenen Transformation M lassen sich zwar alle starren Veränderungen des transparenten Teils des Gallengangsystems relativ zum Volumen ausgleichen, aber es bleibt noch das Problem bestehen, das auch schon bei der Berechnung der Differenzbilder in Abschnitt 4.6.3 auftauchte:

Der Gallengang ist, obgleich relativ starr, bis zu einem gewissen Grad flexibel. Die in 4.6.3 beschriebene Korrektur dieser elastischen Deformation kann sehr gut **relative** Deformationen, d.h. Deformationen, die zwischen der Aufnahme des Referenzbildes B_r und dem jeweils aktuellen Bild B_a aufgetreten sind, beheben, sodass sie keine Artefakte im Differenzbild erzeugen. Es lassen sich damit jedoch keine **absoluten** Deformationen, d.h. Deformationen, die sich über einen längeren Zeitraum seit der Erzeugung des Volumendatensatzes ereignet haben, beheben, da diese in der Regel im Referenzbild und dem jeweils aktuellen Bild gleichermaßen auftreten.

Um diese Deformationen zu beheben, wird die Textur auf dem Rechteck R , nachdem es an die richtige Lage im Volumendatensatz gebracht wurde, analog zur Deformationskorrektur in 4.6.3 korrigiert. Es werden wieder zwei Polygonzüge, diesmal derjenigen, der die

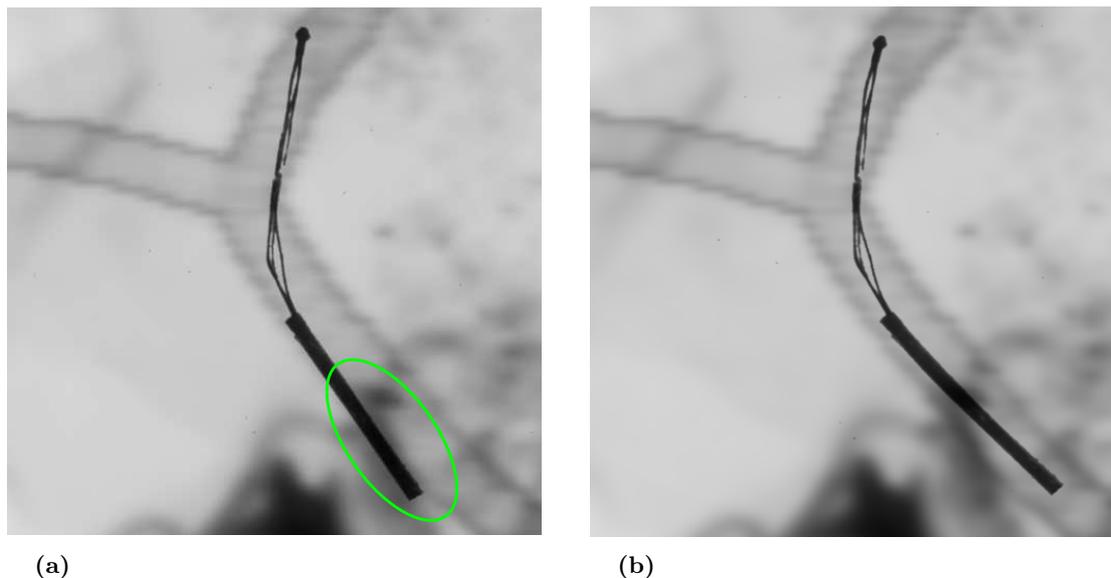


Abb. 4.40.: Gegenüberstellung eines Bildes a) ohne und b) mit der endgültigen Deformationskorrektur. Bei näherer Betrachtung sieht man, dass das Instrument bei nicht durchgeführter Deformationskorrektur innerhalb des markierten Bereichs teilweise außerhalb des Gallengangs liegt.

Marker im Volumendatensatz verbindet und derjenigen, der die Marker im Kamerabild verbindet, aufeinander abgebildet und damit die Textur mittels Interpolation so verzerrt, dass die Marker im Kamerabild und die im Volumendatensatz exakt aufeinander liegen. Damit ist sichergestellt, dass sich die Instrumente, die im Kamerabild erkannt werden, stets im korrekten Bereich des Röntgenbildes befinden und nicht etwa z.B. scheinbar eine Wand des Gallengangs durchstoßen. Abb. 4.40 stellt exemplarisch eine Aufnahme ohne und mit dieser letzten Deformationskorrektur gegenüber.

Schwäche des optischen Verfahrens

Das von der Kamera erzeugte Bild ist natürlich nur eine zweidimensionale Projektion unter einem fixierten Blickwinkel. Das simulierte Röntgenbild lässt sich jedoch aufgrund der Erzeugung aus einem dreidimensionalen Datensatz aus jedem beliebigen Blickwinkel betrachten. Wie oben beschrieben wird das zweidimensionale Abbild des Instruments zwar geometrisch korrekt in das Volumen eingepasst, jedoch bleibt es dabei natürlich flach. Das führt dazu, dass bei einem zu stark aus der Ausgangslage gekippten Betrachtungswinkel diese Flachheit deutlich sichtbar ist. Abb. 4.41 zeigt drei Bilder eines Dormia-Körbchens mit eingefangenem Gallenstein unter Drehwinkeln von 25° , 45° bzw. 65° . Bei einer Drehung von 25° ist der Projektionsfehler noch unmerklich, während er bei 45° bereits eine störende „Flachheit“ des Körbchens bewirkt. Bei 65° ist schließlich deutlich sichtbar, dass das Körbchen zweidimensional ist. In einem Bild, das unter einem Drehwinkel von

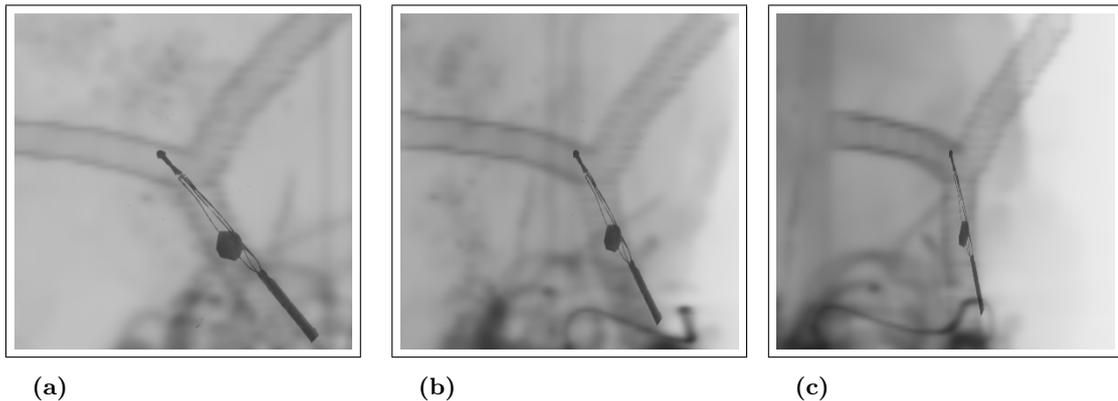


Abb. 4.41.: Gedrehte Ansichten. a) 25° , b) 45° und c) 65° . Der durch die Projektion entstehende Fehler wird bei wachsendem Rotationswinkel immer deutlicher.

90° aufgenommen würde, wäre das Körbchen sogar nur noch als einzelne Linie zu sehen. Dies bedeutet, dass der Drehwinkel des simulierten Röntgenbildes in der Praxis auf $\pm 30^\circ$ eingeschränkt werden sollte, damit diese Fehler nicht sichtbar werden. Dies stellt keine große Einschränkung dar, weil der transparente Gallengang nach Konstruktion eine relativ planare Gestalt aufweist, bei der es nicht notwendig ist, den virtuellen C-Bogen überhaupt zu rotieren.

4.7. Implementierung des Softwaremoduls

Die oben beschriebenen Algorithmen und Ideen wurden im Softwaremodul *X-Sim* implementiert, welches im Folgenden beschrieben wird. Die Nutzeroberfläche dieses Moduls umfasst zwei Fenster. Das erste zeigt das simulierte Röntgenbild, während das zweite ein Kontrollpult für die Steuerung der Simulation enthält. Es wurde eine solche Zweifenster-Darstellung gewählt, da auf diese Weise leicht mehrere Monitore ausgenutzt werden können, indem die beiden Fenster auf verschiedenen Monitoren dargestellt werden. Typischerweise wird das Röntgenbild auf einem großen Monitor oder Beamer dargestellt, während das Kontrollpult auf einem Touchscreen angezeigt wird (siehe Abschnitt 4.7.2).

4.7.1. Fenster für das Röntgenbild

An einem echten C-Bogen sind in der Regel zwei Monitore angeschlossen, wie Abb. 4.42a zeigt. Auf dem einen ist stets das aktuelle Röntgenbild dargestellt, während der andere Monitor die Möglichkeit bietet, Bilder aus der Bildhistorie auszuwählen und anzuzeigen. Dazu kann ein Bild jederzeit gespeichert und damit in diese Historie aufgenommen werden. Dies ist beispielsweise dann nützlich, wenn eine Vorher-/Nachher-Ansicht für einen



(a) Reale Anordnung der Monitore und des Bedienfeldes.



(b) Vollständiger Aufbau von Simtux mit Röntgenmonitor und Kontrollpult.

Abb. 4.42.: Monitoranordnungen

Eingriff gewünscht wird. Analog zu diesem Zwei-Monitor-Aufbau ist das Fenster, in dem das simulierte Röntgenbild in *X-Sim* dargestellt wird, zweigeteilt (siehe Abb. 4.42b und Abb. 4.43). In der linken Hälfte ist das aktuelle Röntgenbild zu sehen, während die rechte Hälfte ein Bild aus der Bildhistorie zeigt. Bilder können jederzeit in diese Historie aufgenommen werden, indem die entsprechende Funktion auf dem Kontrollpult gewählt wird.

4.7.2. Bedienung

In der Realität wird ein C-Bogen mittels eines Bedienfeldes, wie z.B. des in Abb. 4.44a abgebildeten, gesteuert. Damit Anfänger ein Gefühl dafür bekommen, wie sich bei einem realen C-Bogen die Betätigung des Bedienfeldes auf die Veränderung z.B. der Drehung des C-Bogens auswirkt, wurde auf Wunsch von Prof. Dr. Grund ein simuliertes Bedienfeld in das *X-Sim*-Modul integriert, welches an ein reales Bedienfeld angelehnt ist (Abb. 4.44b). Im oberen linken Bereich des Bedienfeldes ist außerdem eine schematische Ansicht des virtuellen C-Bogens abgebildet, damit der Benutzer eine Vorstellung davon bekommt, wie ein realer C-Bogen eingestellt sein müsste, um ein Bild wie das simulierte zu erhalten. Zu einer vollausgestatteten *X-Sim*-Installation gehört auch ein berührungsempfindlicher Bildschirm, auf dem dieses virtuelle Bedienfeld dargestellt wird. Über diesen Bildschirm lässt sich somit die Darstellung des virtuellen Röntgenbildes ähnlich wie mit dem Bedienfeld eines echten C-Bogens steuern.

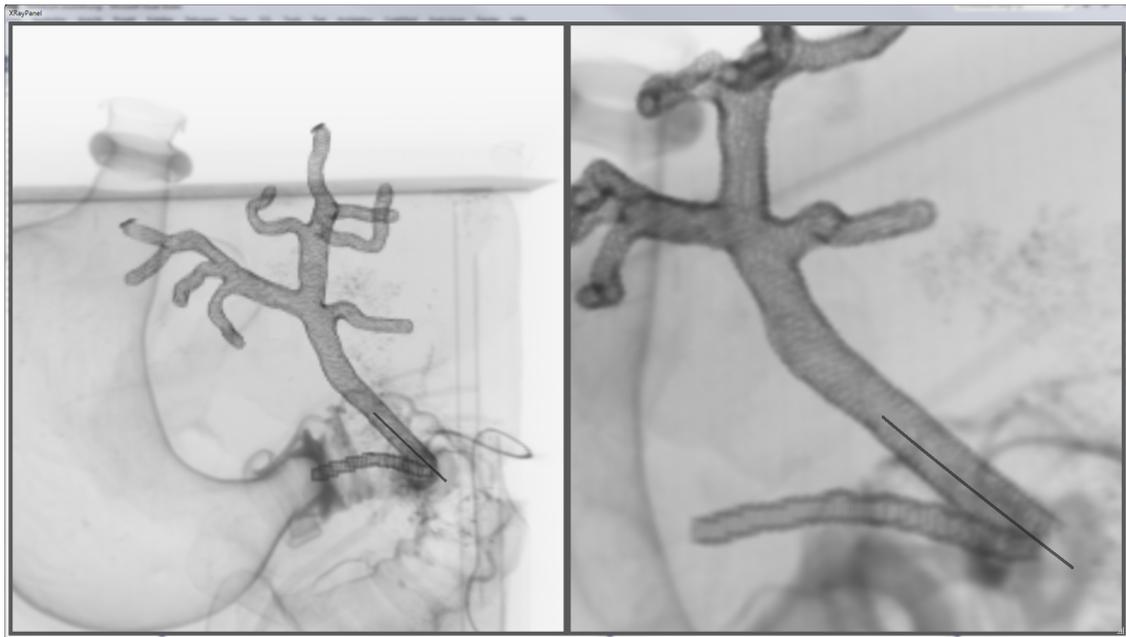
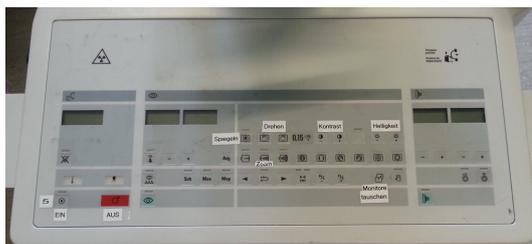


Abb. 4.43.: Das Röntgenfenster von X-Sim



(a) Bedienfeld eines echten C-Bogens.



(b) Virtuelles Bedienfeld, darstellt auf einem Touchscreen.

Abb. 4.44.: Bedienfelder

Zukünftige Erweiterung des Bedienkonzepts

Die Verwendung von zwei separaten Monitoren ist für einen modernen Desktop-Rechner kein Problem, da diese praktisch immer den Anschluss mehrerer Monitore unterstützen. Aus praktischen Gründen sollen aber für die Trainingskurse ausschließlich Notebooks eingesetzt werden. Diese besitzen stets ein integriertes Display, das jedoch in den allermeisten Fällen nicht berührungsempfindlich ist (in Zukunft mag sich das durchaus ändern), wodurch sich dieses nicht für die Darstellung des Bedienfeldes eignet. Für die Darstellung des simulierten Röntgenbildes eignet sich das integrierte Display ebenfalls nur bedingt, da ein Notebook-Monitor erstens in der Regel zu klein ist und es zweitens nicht immer möglich ist, das Notebook so aufzustellen, dass es gut gesehen werden kann.



Abb. 4.45.: Fußpedal zur Auslösung des virtuellen Röntgenvorgangs.

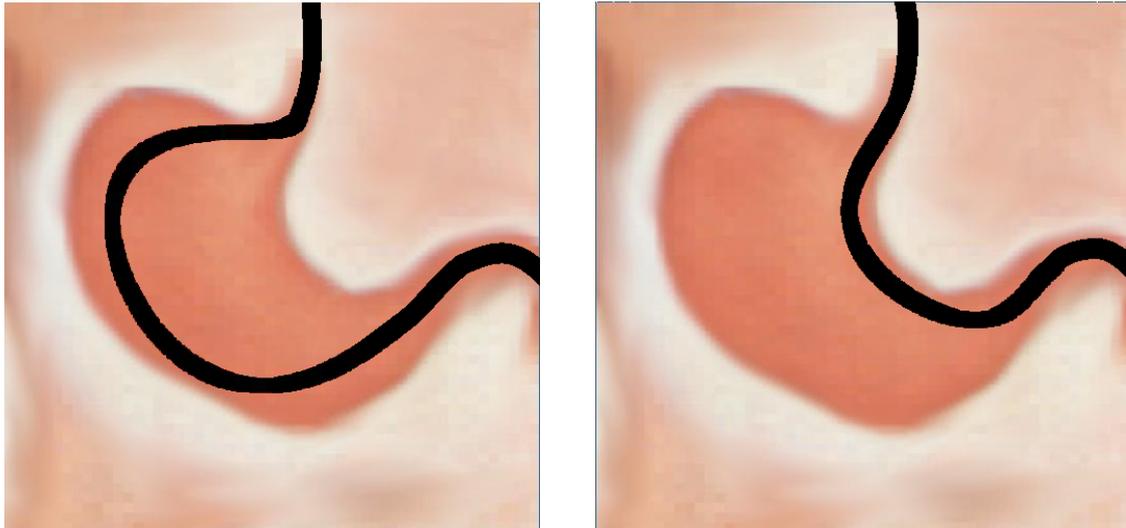
Aus diesem Grund müssen an das Notebook zusätzlich zum integrierten Display zwei weitere Monitore angeschlossen werden können. Während der Entwicklung von *X-Sim* hat dies durchaus funktioniert, allerdings nicht sehr stabil.

Häufig war das Monitor-Setup des Betriebssystems nach einem Neustart durcheinander und sehr häufig ging die Kalibrierung des Touchscreens aus unerfindlichen Gründen verloren, sodass die Inbetriebnahme nur für einen Experten durchführbar ist. Aus diesem Grund kam der Gedanke auf, das Bedienfeld unabhängig von dem Rechner zu machen, auf dem *Simtux* ausgeführt wird. Die Idee ist, dass *Simtux* wie vorher auch zwar das Bedienfeld berechnet und auf Eingaben reagiert, der Inhalt des Fensters, welches das Bedienfeld enthält, jedoch gleichzeitig per HTTP-Stream zu beliebigen Webbrowsern gesendet wird. Zum Beispiel könnte ein beliebiges Tablet über eine W-Lan-Verbindung mit dem *Simtux*-Rechner verbunden werden und per Browser das aktuelle Bild des Bedienfeldes empfangen und alle Benutzereingaben an *Simtux* zurücksenden, wo diese an das Fenster mit dem Bedienfeld weitergeleitet werden. Auf diese Weise müsste ein Monitor weniger angeschlossen werden, da der Touchscreen nicht mehr benötigt wird.

Diese Lösung ist insofern elegant, als dass sie den bisherigen Aufbau praktisch als Spezialfall enthält. Es kann nämlich selbstverständlich auch der Rechner, auf dem *Simtux* ausgeführt wird, selbst per Webbrowser auf das Bedienfeld zugreifen, sodass sich in diesem Fall im Vergleich zur aktuellen Variante äußerlich nur sehr wenig ändert.

Fußpedal

Bei einem echten C-Bogen werden im Sinne des Strahlenschutzes nicht etwa kontinuierlich Röntgenbilder erzeugt, sondern nur dann, wenn der Bediener es wünscht. Ausgelöst wird ein Röntgenvorgang, indem der Bediener ein Fußpedal betätigt. Dieser Mechanismus wurde in *X-Sim* nachgebildet, indem an das Phantom ein einfaches handelsübliches Fußpedal angeschlossen wurde, wie es in Abb. 4.45 abgebildet ist. Technisch wird dieses einfach als Sensor behandelt, welcher nur zwei Werte („an“ und „aus“) liefert, die wie alle



(a) Nicht begradigt: Das Endoskop liegt an der großen Krümmung an.

(b) Begradigt: Das Endoskop liegt an der kleinen Krümmung an und kann nun präzise gesteuert werden.

Abb. 4.46.: Schematische Begradigungsanzeige.

anderen Sensoren auch über den Sensorbus an den Busmaster und von dort an den Sensorserver gesendet werden (vgl. Abschnitt 2.3.2). Ein simuliertes Röntgenbild wird nur dann berechnet, wenn dieser Sensor den Wert „an“ liefert. Wird das Pedal kontinuierlich gedrückt, werden gemäß der Busfrequenz ca. 30 Sensorwerte pro Sekunde empfangen, was deutlich über der Bildwiederholfrequenz eines üblichen C-Bogens liegt. Auch ist auf schwächeren Rechnern nicht immer garantiert, dass ein Röntgenbild in $\frac{1}{30}$ Sekunde simuliert werden kann. Aus diesen Gründen wird die Bildwiederholrate künstlich gedrosselt. Diese Beschränkung beträgt dabei wahlweise 2 oder 10 Bilder pro Sekunde, was den beiden Durchleuchtungsmodi „gepulst“ und „kontinuierlich“ bei einem realen C-Bogen nachempfunden ist.

4.8. Erkennung der Begradigung

Um das Begradigungsmanöver (vgl. Abschnitt 4.2) auswerten zu können, wurden kapazitive Sensoren ähnlich denen, die in Abschnitt 4.4 beschrieben wurden, an Magen und Duodenum des Biliphanten angebracht. Die Elektroden am Magen sind dabei entlang der beiden Krümmungen angeordnet, wie Abb. 4.47 zeigt. Anhand der Messwerte der Sensoren kann auf den Abstand des Duodenoskops zu den Sensoren geschlossen werden. Dadurch kann die Lage des Duodenoskops in groben Zügen erfasst werden.

Zur Darstellung wird in eine schematische Abbildung des Magens (Abb. 4.46) eine schwarze Kurve eingezeichnet, die das Endoskop symbolisieren soll. Diese Kurve ist ein Bézier-



Abb. 4.47.: Elektroden auf der Magenoberfläche, vier an der großen und zwei an der kleinen Krümmung.

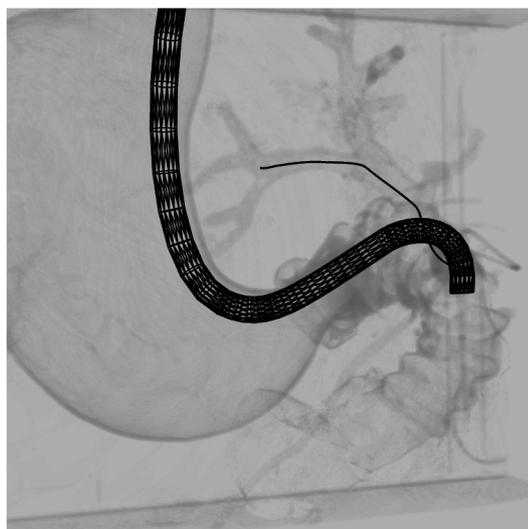


Abb. 4.48.: Darstellung des Duodenoskops für die Begradigungsanzeige direkt im simulierten Röntgenbild.

spline vom Grad 6, dessen Kontrollpunkte in Abhängigkeit der Sensorwerte in Richtung der großen bzw. kleinen Krümmung verschoben werden, sodass er genau dann an der großen Krümmung des skizzierten Magens anliegt (Abb. 4.46a), wenn er auch in der Realität dort anliegt und entsprechend an der kleinen Krümmung (Abb. 4.46b), wenn der Verlauf des realen Duodenoskops der kleinen Krümmung folgt.

Die ursprüngliche Überlegung bestand darin, auch das Begradigungsmanöver mittels der Röntgensimulation darzustellen (Abb. 4.48). Da jedoch der Verlauf des Duodenoskops mit der kapazitiven Sensorik nur qualitativ, nicht jedoch quantitativ bestimmt werden kann (vgl. Abschnitt 4.4), wurde diese Idee zugunsten einer schematischen Ansicht verworfen. Die Erfahrung zeigt, dass eine zu realistische Anzeige den Benutzer eher verwirrt als ihm dient, weil der dargestellte Verlauf des Duodenoskops nicht der Realität entspricht. Eine schematische Darstellung wird in dieser Hinsicht leichter akzeptiert, da sie offensichtlich gar nicht den Anspruch erhebt, realistisch zu sein und daher Ungenauigkeiten viel eher toleriert werden.

4.9. Herstellung der transparenten Gallengänge

Die Herstellung der transparenten Gallengänge erweist sich als nicht-triviale Aufgabe. Wie in Abschnitt 4.6 erwähnt wurde, ist das Latex, welches für den Biliphanten zum Einsatz kommt, bei den verwendeten Wandstärken nicht mehr transparent. Es wurden daher

verschiedene Ansätze erdacht, aus anderen Materialien ein transparentes Gefäßsystem zu konstruieren.

4.9.1. PVC-Isolierschläuche

Der in dieser Dissertation vorgestellte Prototyp des transparenten Gallengangsystems wurde aus transparenten PVC-Isolierschläuchen hergestellt. Diese sind einfache Schläuche aus einem transparenten PVC mit einer Wandstärke von ca. 1 mm, die normalerweise dazu benutzt werden, Kabel und Leitungen zu isolieren und zu schützen. Die verwendeten Schläuche weisen hervorragende Transparenzeigenschaften auf und sind sehr ebenmäßig beschaffen, sodass es zu wenigen Lichtbrechungen kommt. Es wurden drei Stücke mit einer Länge von jeweils ca. 10 cm zusammengeklebt. Dazu wurden diese im passenden Winkel zurechtgeschnitten und dann an den Schnittkanten mit transparentem Kleber Y-förmig verklebt. Schnitt und Klebung wurden sehr sorgfältig durchgeführt, da ansonsten an der Klebefläche Unregelmäßigkeiten entstanden wären. Diese würden sich durch verstärkte Lichtbrechungen bemerkbar machen und letztendlich die Transparenz negativ beeinflussen, sodass in diesem Bereich das simulierte Röntgenbild Fehler aufwies. Außerdem mussten Grate vermieden werden, da diese das Einführen von Instrumenten behindern können.

Nachteile dieser Konstruktion sind:

- Die Herstellung von Systemen mit mehr als einer Verzweigung ist sehr aufwändig.
- Klebeflächen sind oft nicht richtig transparent.
- Das Material weist im Vergleich zu realen Gallengängen eine geringere Flexibilität und damit unrealistische Haptik auf.

4.9.2. Glas

Weil Glas ebenfalls hervorragende Transparenzeigenschaften aufweist und sich, zumindest für erfahrene Glasbläser, relativ gut verarbeiten lässt, wurde kurzzeitig darüber nachgedacht, ein Gallengangsystem aus Glas herstellen zu lassen. Zum Test wurde ein Verbindungsstück in Y-Form ähnlich dem PVC-Prototypen im Laborbedarf gekauft, welches in Abb. 4.49 dargestellt ist. Nach kurzen Tests wurde diese Idee aus folgenden Gründen schnell wieder verworfen:

- Aufgrund des Herstellungsprozesses, bei dem das Glas geschmolzen wird, treten an den Übergängen zwischen einzelnen Segmenten Unregelmäßigkeiten auf, die auf den ersten Blick nicht auffallen, die aber zu erheblichen Lichtbrechungen führen, sodass



Abb. 4.49.: Foto des Y-Stücks aus Glas.

das Abbild des Instruments in diesen Bereichen stark verzerrt ist. Auch abseits der Übergänge treten leichte Brechungseffekte auf, weil herstellungsbedingt, anders als bei den PVC-Schläuchen, die Wandstärke der Glasröhren nicht vollkommen konstant ist.

- Glas ist sehr hart und unflexibel. Dadurch ist die Haptik bei einem Eingriff mit einem Instrument deutlich anders als in der Realität. Beispielsweise kann sich ein Stent in der Verzweigung nicht richtig entfalten, weil die Wand des Gefäßes schlicht zu unnachgiebig ist.

4.9.3. Kunstharz

Aktuell wird die Erstellung von Gallengangssystemen aus transparentem Kunstharz erforscht. Kunstharz gibt es in verschiedenen Härte- und Flexibilitätsgraden. Die Grundidee bei der Konstruktion ist, zunächst ein Positivmodell der Gänge zu erstellen, dieses in einen Kunstharzblock einzugießen und anschließend aus dem Block wieder zu entfernen. Dadurch erhält man einen kompakten Block, in dem die Gallengänge sozusagen „Tunnel“ darstellen. Diese Art der Konstruktion hat den Vorteil, dass es wenige Flächen gibt, an denen problematische Lichtbrechungen auftreten können, weil die Oberfläche des Blocks durch Polieren spiegelglatt gemacht werden kann. Es können nur noch an den Wänden der „Tunnel“ Brechungen auftreten, die deshalb möglichst glatt und eben sein sollten.

Wachs

Wichtig für die Herstellung des Positivmodells ist die Wahl eines geeigneten Materials. Dieses muss sich sowohl gut formen als auch gut wieder aus dem Harzblock entfernen

lassen. Die erste Idee bestand darin, Kerzenwachs zu verwenden. Es lässt sich recht gut durch wiederholtes Eintauchen einer Seele aus z.B. Draht formen und durch Schmelzen wieder entfernen. Unglücklicherweise hat Wachs jedoch keine besonders glatte Oberfläche und lässt sich vor allem weniger gut als erhofft durch Schmelzen aus dem Harzblock entfernen. Selbst bei Spülung mit kochendem Wasser bleiben Reste des Wachses sehr hartnäckig an dem Harz kleben. Dadurch leidet die Transparenz erheblich. Aufgrund der nicht besonders glatten Oberfläche haben außerdem auch die Gänge in dem Harzblock eine milchige Oberfläche, wodurch sich das so hergestellte Gefäßsystem nicht gut zur optischen Erkennung von Instrumenten eignet.

Karamell

Ein Material, dessen Erprobung zum Zeitpunkt des Drucks dieser Arbeit noch nicht abgeschlossen ist, ist Karamell. Es lässt sich leicht in beliebigen Mengen aus Zucker herstellen und ähnlich wie Wachs durch wiederholtes Tauchen relativ gut in Form bringen. Vor allem bildet Karamell eine sehr glatte Oberfläche aus und lässt sich ausgesprochen gut wieder aus dem Harzblock entfernen. Es reicht nämlich aus, den Block für eine gewisse Zeit in heißes Wasser zu legen, weil das Karamell durch das Wasser gelöst wird und sich so vollständig rückstandsfrei entfernen lässt. Die ersten Erfahrungen mit Karamell sind sehr vielversprechend und es wird in Zukunft weiter in diese Richtung geforscht werden.

5. Fazit und Ausblick

Mit der vorgestellten Kombination aus Hard- und Software *Simtux* wurde im Rahmen dieser Dissertation ein modulares System entwickelt, mit dem das Training endoskopischer Eingriffe mit den Endoskopiephantomen, die von der Arbeitsgruppe um Prof. Dr. Grund in der Abteilung für Experimentelle Chirurgische Endoskopie der Eberhard-Karls-Universität Tübingen entwickelt wurden, effektiv überwacht und gesteuert werden kann. Dem Benutzer werden dabei in Echtzeit Informationen über seinen aktuellen Trainingsverlauf geboten.

Das Modul *GastroInt-Sim* zeigt einem Trainierenden während einer Gastro- oder Koloskopie den Lichtschein des Endoskops in einer 3D-Darstellung, damit er sehen kann, welche Bereiche der Oberfläche des Phantoms er aktuell betrachtet. Zusätzlich kann die Navigation des Gastroskops trainiert werden, indem dem Benutzer bestimmte Aufgaben zur Navigation innerhalb des Magens gestellt werden können, die er lösen muss. Dazu markieren LEDs bestimmte Punkte auf der Oberfläche des Magens, die angesteuert werden müssen.

Das Modul *X-Sim* bietet als entscheidende Neuerung eine virtuelle Röntgensimulation, die eine ERCP am *Biliphanten* ohne jede Strahlenbelastung ermöglicht und so einen wichtigen Beitrag zum Strahlenschutz leistet. Diese Röntgensimulation ist sowohl für Anfängerkurse als auch für Fortgeschrittenenkurse verfügbar. In den Anfängerkursen müssen die Teilnehmer lediglich wissen, welcher Gallengang mit einem Instrument kanüliert wird und wie groß dessen Vorschub ist. In den Fortgeschrittenenkursen können die Teilnehmer tatsächliche therapeutische Eingriffe erlernen bzw. üben. Für beide Arten von Kursen werden die Röntgenbilder entsprechend der jeweiligen Anforderungen auf verschiedene Weisen berechnet. Ermöglicht wird die Simulation mit Hilfe einer kapazitiven bzw. optischen Sensorik, kombiniert mit der Erzeugung eines simulierten Röntgenbildes aus einem CT-Volumendatensatz mit Hilfe eines Computers.

Die Steuerung des Biliphanten wurde so konzipiert, dass sich die Bedienung so natürlich wie möglich in den üblichen Ablauf einer ERCP einfügt. Dazu wird vollständig auf Maus- und Tastatursteuerung verzichtet und stattdessen auf einen Touchscreen und ein Fußpedal zurückgegriffen.

Die Resonanz bei Trainingsteilnehmern und Experten auf dem Gebiet der ERCP war sehr positiv. Dementsprechend wurden zwei Poster auf zwei Fachkonferenzen in den Jahren 2012 und 2013 mit Preisen prämiert: Auf dem 23. Kongress der Südwestdeutschen Gesellschaft für Gastroenterologie in Freiburg wurde das Poster [GIS⁺12b] vorgestellt, das unter anderem die Röntgensimulation von *X-Sim* beschreibt. Dieses Poster wurde mit dem **Stiftungspreis der Südwestdeutschen Gesellschaft für Gastroenterologie** ausgezeichnet. Weiterhin wurde auf der DGE-BV 2013 das Poster [VAG13a] vorgestellt, das die Röntgensimulation von *X-Sim* für die Anfängerphantome beschreibt. Dieses Poster wurde mit dem **Boston Scientific Innovationspreis** ausgezeichnet.

Diese Auszeichnungen und die viele positiven Rückmeldungen zeigen die Bedeutung und Relevanz des vorgestellten Systems.

5.1. Bronchophant

Dank der strengen Modularisierung und des ausdrucksstarken Skriptsystems kann das System in Zukunft leicht für weitere Phantomtypen erweitert werden, wie z.B. das aktuell in Entwicklung befindliche Modul *Broncho-Sim*, mit dem das Phantom des Bronchien-systems, an dessen Entwicklung die Tübinger Arbeitsgruppe zur Zeit arbeitet, gesteuert werden soll. Hierbei geht es im Kern um das Training der *Bronchoskopie*, bei der mit einem sog. *Bronchoskop* die einzelnen Äste des Bronchiensystem, das, ähnlich wie das Gallengangsystem, eine baumförmige Gestalt hat, untersucht werden. Im Gegensatz zu diesem ist das menschliche Bronchiensystem jedoch von Mensch zu Mensch weit weniger variabel, sodass ein standardisierter Atlas existiert. Bei der Bronchoskopie muss jeder Zweig des Bronchiensystems bis mindestens zur dritten Verzweigungstiefe inspiziert werden, damit dort keine Pathologien übersehen werden. Noch tiefere Verzweigungsebenen können aufgrund ihres geringen Gefäßdurchmessers in der Regel nicht mehr mit einem Bronchoskop untersucht werden. Die Herausforderung besteht darin, den Überblick über die bereits untersuchten Gänge zu behalten. Dies erfordert sowohl ein gutes Gedächtnis als auch ein gutes räumliches Vorstellungsvermögen.

5.1.1. Ideen zur Entwicklung eines Moduls für Simtux

Eine Firma, die sich auf die Herstellung dreidimensionaler Prototypen mittels 3D-Druck spezialisiert hat, hat uns ein Modell des menschlichen Bronchiensystems erstellt. Dieses entspricht zwar in keiner Weise den medizinischen Anforderungen an ein realistisches Phantom, konnte jedoch dazu verwendet werden, einen ersten Prototypen für das *Simtux*-Modul *Broncho-Sim* zu entwickeln. Die Grundidee für dieses Modul besteht darin, die bei



Abb. 5.1.: Prototyp der Hardware des *Broncho-Sim*-Moduls.

allen anatomischen und konzeptionellen Unterschieden zwischen ERCP und Bronchoskopie bestehenden Gemeinsamkeiten auszunutzen: In beiden Fällen wird ein Instrument in ein baumstrukturiertes Gefäßsystem eingeführt. Es liegt also nahe, eine ähnliche Sensorik wie bei den ERCP-Anfängerphantomen einzusetzen.

Im Fall der Bronchien ist die Aufgabe sogar einfacher zu lösen, weil weder die Lage des Bronchoskops noch dessen genauer Vorschub von wesentlicher Bedeutung sind. Es ist ausschließlich wichtig zu wissen, welches Segment des Bronchienbaums betreten und durchquert wurde und welches nicht. Es reicht also vollkommen aus, jeweils Anfang und Ende der einzelnen Segmente mit einer ringförmigen Elektrode zu umwickeln. Sobald durch einen solchen Anfangssensor ein Signal gemessen wird, das über einer bestimmten, für jede Elektrode zu kalibrierenden, Schwelle liegt, kann dieses Segment als betreten betrachtet werden. Sobald der Sensor am Ende ein Signal über seiner Schwelle misst, kann das ganze Segment als inspiziert angesehen werden. Da ein Bronchoskop einen erheblich größeren Durchmesser aufweist als die typischerweise bei einer ERCP eingesetzten Instrumente, ist auch die Kapazität und damit der Signalpegel wesentlich größer. Dadurch wird die Messung deutlich weniger rauschempfindlich. Abb. 5.1 zeigt ein Foto des bisher entwickelten sehr experimentellen Aufbaus.

5.1.2. Implementierungsideen für das Softwaremodul *Broncho-Sim*

Das Softwaremodul *Broncho-Sim* existiert zum Zeitpunkt des Drucks dieser Arbeit nur prototypisch als Machbarkeitstudie. Der Grundaufbau sieht vor, dass dem Benutzer eine

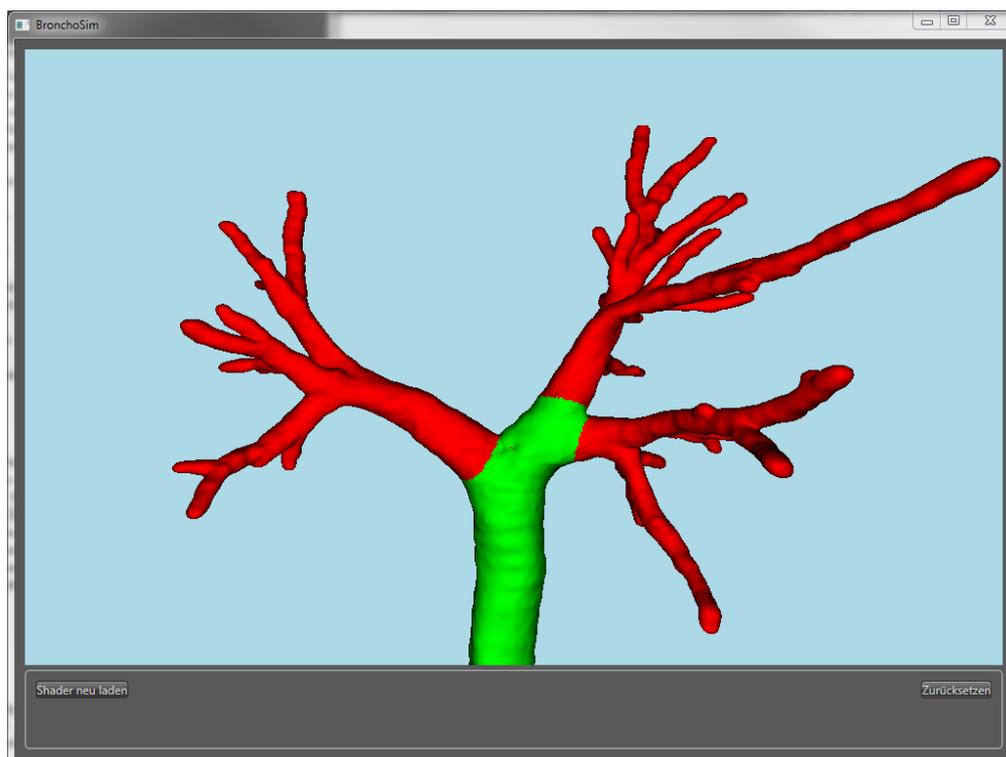


Abb. 5.2.: Benutzeroberfläche des Prototyps des *BronchoSim*-Moduls.

grafische Echtzeit-Repräsentation geboten wird, damit er sein dreidimensionales Vorstellungsvermögen schulen kann. Abb. 5.2 zeigt das Fenster, in dem diese grafische Repräsentation der Bronchoskopie dargestellt wird. Dabei sind die rot gefärbten Bereiche des Bronchienbaums uninspiziert, während diejenigen Segmente, die bereits untersucht wurden, grün gefärbt sind. Diese Färbung dient dem Trainierenden als Gedächtnishilfe, welche Segmente noch aufzusuchen sind. Er kann damit jedoch nicht sehen, in welchem Segment er sich aktuell befindet. Nicht implementiert, aber für die Zukunft angedacht, ist daher eine zusätzliche schematische Darstellung des Bronchoskops auf Basis von Splines, ähnlich wie die des Instruments bei den ERCP-Anfängerphantomen.

A. Anhänge

A.1. Das Programmframework Taraxacum

Bei dem Framework *Taraxacum* handelt es sich um eine Bibliothek, die im Kern eine Grafikengine und ein leistungsfähiges Skriptsystem umfasst. Die Grafikengine stellt eine Abstraktionsschicht für den Zugriff auf die Grafikkarte des Rechnersystems bereit, mit deren Hilfe zwei- oder dreidimensionale Grafiken auf einfache Weise erzeugt werden können. Der Zugriff geschieht dabei mit Hilfe der relativ hardwarenahen Bibliothek *OpenGL*. Die Grafikengine deckt eine große Teilmenge des Funktionsumfangs der zum Zeitpunkt des Drucks dieser Arbeit aktuellen OpenGL-Version 4.4 ab.

A.1.1. OpenGL

Die Engine *Taraxacum* wurde für das .Net-Framework von Microsoft entwickelt. OpenGL jedoch ist eine reine C-Bibliothek. Um den Zugriff von .Net heraus auf die Funktionen von OpenGL zu ermöglichen, wird als Zwischenschicht die Open-Source-Bibliothek *OpenTK* (kurz für Open ToolKit) eingesetzt. Diese stellt einen einfachen Wrapper für die OpenGL-Bibliothek dar, der Methoden, die innerhalb des .Net-Frameworks aufgerufen werden, auf die entsprechenden C-Funktionen abbildet. Einige der in dieser Arbeit vorgestellten Konzepte setzen einen gewissen Einblick in die Funktionsweise von OpenGL voraus. Deshalb soll an dieser Stelle eine kurze Einführung in OpenGL gegeben werden. Für eine erschöpfendere Beschreibung sei an diverse Fachliteratur wie z.B. [RED] oder [CR12] sowie die offizielle Internetseite www.opengl.org verwiesen. Diese Kurzbeschreibung folgt weitgehend der in [Vie07].

Die Grundlage der Erzeugung aller Grafiken, des sog. *Renderings*, mittels OpenGL ist das sog. *Primitiv*, die kleinste darstellbare geometrische Einheit. Primitive sind in aller Regel Punkte, Strecken oder Dreiecke. Diese Primitive werden durch ihre Eckpunkte definiert. Diese Eckpunkte werden durch die Datenstruktur *Vertex* beschrieben, welche eine Ansammlung verschiedener sog. *Attribute* a_i ist, die entweder skalar- oder vektorwertig sind (also $a_i \in D^n$ mit $n \in \{1,2,3,4\}$, wobei D der Wertebereich des Datentyps des Attributs a_i ist. In der Regel gilt $D = float$, häufig kommt aber auch $D = int$ zum Einsatz).

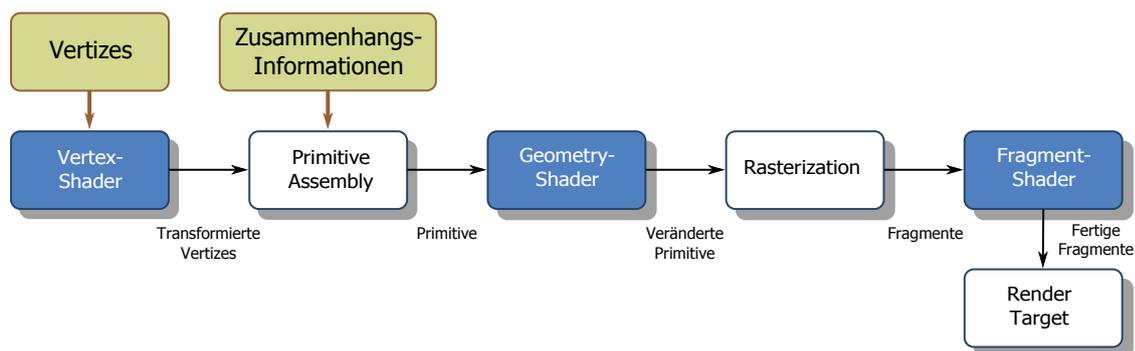


Abb. A.1.: OpenGL-Pipeline. Die Blau hinterlegten Abschnitte lassen sich mittels *Shaderprogrammen* programmieren.

Alle darzustellenden Primitive werden mittels eines Pipeline-Prinzips, das in Abb. A.1 skizziert wird, sehr effizient von der Grafikkarte des Computersystems verarbeitet. Sehr grob betrachtet wird in die Pipeline eine Menge von Vertices hineingegeben, auf die verschiedene Koordinatentransformationen angewendet werden. Aus den transformierten Vertices werden anhand vorher definierter Regeln Primitive konstruiert. Diese Primitive werden gemäß des Pixelrasters des sog. *Renderziels* (z.B. der Bildschirm) *gerastert*, wobei für jedes Pixel ein sog. *Fragment* erzeugt wird, für das dann die Farbe berechnet wird, die am Schluss in das dem Fragment zugehörige Pixel des Renderziels kopiert wird.

Um den Rendervorgang kontrollieren zu können, sind einige Schritte in der Pipeline nicht fest vorgegeben, sondern können durch den Programmierer definiert werden. In Abb. A.1 sind diese Schritte blau hinterlegt. Dazu dienen sog. *Shaderprogramme*. Das sind Programme, die direkt von den Prozessoren der GPU¹ der Grafikkarte des Computers ausgeführt werden. Diese Prozessoren sind massiv parallel ausgelegt, wodurch die GPU ihre enorme Rechengeschwindigkeit entfalten kann. Programmiert werden diese Shader üblicherweise in der stark an C angelehnten Programmiersprache *GLSL* (*GL Shading Language*). Konkret bedeutet dies im einfachsten Fall, dass die Transformationen, die auf die Vertices angewendet werden sollen, sowie die Berechnung der Farben der Fragmente frei programmierbar sind. Man spricht hier von *Vertex-* bzw. *Fragmentshadern*.

In der Pipeline existieren neben Vertex- und Fragmentshadern noch einige weitere durch Shader definierbare Schritte. Erwähnt werden sollte an dieser Stelle der *Geometryshader*. Dieser wird auf bereits fertig konstruierte Primitive angewendet und kann dazu benutzt werden, diese Primitive nachträglich zu verändern oder sogar komplett zu verwerfen. In der Pipeline sind noch einige andere Shadertypen angesiedelt, die jedoch für diese Arbeit keine Rolle spielen und deshalb nicht näher erwähnt werden.

Mit der Entwicklung von OpenGL 4.3 wurde zusätzlich noch ein neuer Shadertyp, der sog. *Compute Shader*, eingeführt, der anders als alle anderen Shader nicht Teil der Pipe-

¹*Graphics Processing Unit*, analog zur CPU des Computersystems

line ist, sondern für allgemeine Rechenaufgaben abseits der üblichen Grafikkberechnungen genutzt werden kann. Dies kann unter manchen Umständen sehr interessant sein, da die zentralen Recheneinheiten (GPUs) moderner Grafikkarten über eine Rechenleistung verfügen, die z.T. um Größenordnungen über der einer CPU liegt. Diese Leistung kommt vor allem durch eine enorme Parallelisierung der Rechenoperationen zustande, sodass Algorithmen, die sich gut parallelisieren lassen, stark von einer Portierung auf die Grafikkarte profitieren können.

Mit der *CUDA*-Technologie von *NVIDIA* sowie der konkurrierenden offenen *OpenCL*-Bibliothek standen bereits vorher durchaus mächtigere Werkzeuge bereit, um die Rechengeschwindigkeit von modernen GPUs für allgemeine Berechnungen nutzen zu können. Compute Shader sind jedoch viel enger an die üblichen Prozeduren von *OpenGL* gekoppelt, wodurch eine sehr reibungslose Integration in die Grafikkengine erreicht werden konnte. Insbesondere für das Modul *X-Sim*, welches in Kapitel 4 beschrieben wurde, kommen Compute Shader zum regen Einsatz.

Konzeptionell ist *OpenGL* als Zustandsautomat realisiert. Im Wesentlichen bedeutet dies, dass es zwei Kategorien von Funktionen gibt: Funktionen der einen Kategorie ändern den internen Zustand dieses Zustandsautomaten, während Funktionen der zweiten Kategorie Befehle an *OpenGL* sind, Aktionen auf Grundlage des momentanen Zustands auszuführen. Ein Beispiel für eine Funktion der ersten Kategorie ist `glUseProgram(...)`, mit der das aktuell auszuführende Shaderprogramm gesetzt wird, ein Beispiel für die zweite Kategorie ist `glDrawElements(...)`, mit der eine vorher festgelegte Menge von Primitiven gerendert wird. Dabei definiert der aktuelle Zustand das Resultat des Rendervorgangs. Der Vorteil dieses zustandsbasierten Designs der Bibliothek ist der Wegfall der Notwendigkeit, für jeden Renderbefehl stets alle diesen definierenden Parameter anzugeben. Stattdessen gilt die Devise: „Benutze die aktuellen Parameter so lange, bis sie sich ändern“.

A.1.2. Grafisches Subsystem

Die Engine *Taraxacum* beinhaltet ein Grafik-Subsystem. Dieses stellt im Kern einen Wrapper dar, der die Methoden von *OpenTK* und damit die dahinterstehenden *OpenGL*-Funktionen objektorientiert kapselt. Das im Kern zustandsbasierte *OpenGL* wird durch diese objektorientierte Abstraktion für den Programmierer komfortabler zu verwenden. Es existieren Wrapperobjekte für die verschiedensten Aufgaben. Es folgt eine Auswahl der Wichtigsten unter diesen, die auch für die Module, die in den einzelnen Kapiteln vorgestellt wurden, von Bedeutung sind.

- **Bufferobjekte:** Ein *Buffer* ist im Prinzip ein Array im Grafikkartenspeicher, auf das nicht direkt, sondern nur mittels spezieller Funktionen zugegriffen werden kann.
- **Shaderobjekte:** Ein *Shader* ist wie oben beschrieben ein Programm, das von der GPU ausgeführt wird. Das Hauptprogramm kann an einen Shader Daten senden, die dann aus Sicht des Shaders als Programmkonstanten wahrgenommen werden. Diese Daten werden als *uniforme Variablen* bezeichnet. Das Wrapperobjekt bietet Methoden an, um die Werte dieser uniformen Variablen zu manipulieren.
- **Texturobjekte:** Eine *Textur* ist, ähnlich wie ein Buffer, ein Array im Grafikkartenspeicher. Im Unterschied zu jenem kann eine Textur auch zwei- oder dreidimensional sein und ist vor allem formatiert, d.h. jedes Element der Textur (ein sog. *Texel*) hat ein bestimmtes, strukturiertes Format.

A.1.3. Skripting-Schnittstelle

Häufig wird es in der Softwareentwicklung als sehr wünschenswert angesehen, wenn eine einfache Erweiterbarkeit der Software ohne die Notwendigkeit, die komplette Software stets neu kompilieren zu müssen, gegeben ist. Diese Erweiterbarkeit bietet verschiedene Vorteile:

- *Rapid Prototyping:* Neue Ideen und Algorithmen können schnell und ohne viel Extraaufwand ausprobiert werden.
- Erfahrene Anwender können, auch ohne im Besitz des Quellcodes zu sein, Anpassungen und Erweiterungen implementieren.

Solche Erweiterbarkeiten werden in der Regel durch die Einbettung einer Skriptsprache in die Anwendung ermöglicht. Ein Skript hat dann über klar definierte Schnittstellen Zugriff auf Kernfunktionen. Für die .Net-Umgebung existiert eine Implementierung der beliebten Skriptsprache *Python*, die *IronPython* genannt wird. Diese Sprache läuft direkt in der .Net-Laufzeitumgebung und hat somit wie alle .Net-Sprachen Zugriff auf sämtliche vorhandenen Klassen und Bibliotheken. Diese Eigenschaft ist im Rahmen der Verwendung als Skriptsprache für *Simtux* sehr wünschenswert. Daher wurde in die *Taraxacum*-Engine ein Skripting-Subsystem auf Basis von Python integriert, das damit für die Benutzung in *Simtux* zur Verfügung steht.

Damit ein fehlerhaftes Skript nicht die komplette Anwendung in einen ungültigen oder inkonsistenten Zustand bringen kann („zum Abstürzen bringen kann“), wird einem Skript kein direkter Zugriff auf Objekte der Engine oder des *Simtux*-Kerns ermöglicht; vielmehr wurde eine Schnittstelle definiert, über die dieser Zugriff geregelt wird. Dazu gibt es die Klasse `PythonInterface`, die eine Sammlung öffentlicher, statischer Methoden enthält,

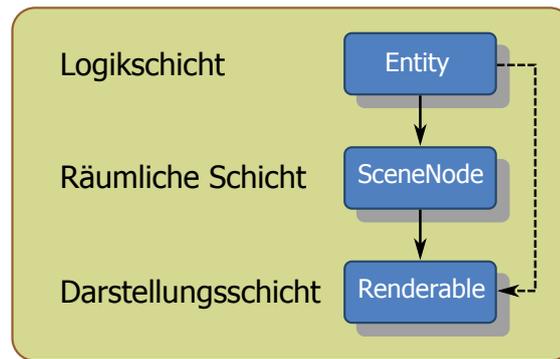


Abb. A.2.: Hierarchie innerhalb eines Szenenobjekts. Eine *Entity* besitzt einen *SceneNode*, an welchen wiederum beliebig viele *Renderables* angehängt werden können. Eine *Entity* hat außerdem direkten Zugriff auf die *Renderables* ihres Szenenknotens.

die für den Aufruf aus Skripten heraus zur Verfügung stehen. Das `PythonInterface` ist mit dem modularen Ansatz von *Simtux* voll kompatibel, da *Simtux*-Module eigene Methoden definieren können, um die das `PythonInterface` erweitert wird.

Alle Skripte werden in einer abgeschotteten *Sandbox*-Umgebung ausgeführt, innerhalb derer alle Skripte volle Flexibilität haben, jedoch nach außen hin, also mit dem Hauptprogramm, nur über diese wohldefinierte Schnittstelle kommunizieren können. Innerhalb des .Net-Frameworks gibt es das Konzept der sogenannten *AppDomains*, die genau dies ermöglichen. Eine *AppDomain* ist eine isolierte Ausführungsumgebung, deren Objekte keinen Zugriff auf andere *AppDomains* haben. Jegliche Kommunikation zwischen *AppDomains* muss über einen sog. „Marshalling“-Mechanismus geschehen, wodurch in Zusammenhang mit der umfangreichen Rechteverwaltung von *AppDomains* gewährleistet wird, dass keine unerwünschten Zugriffe auftreten können.

A.1.4. Szenenbeschreibung

Definition A.1. *Mit dem Begriff Szene sei im Folgenden die Menge aller grafischen Objekte, die durch die 3D-Engine dargestellt werden sollen sowie alle Klassen, die zur Darstellung, Steuerung oder andersweitiger Definition benötigt werden, bezeichnet.*

Diese Objekte lassen sich einteilen in die drei Schichten Logik/Steuerung, räumliche Lagebeschreibung und grafische Darstellung. Diese drei Schichten sind hierarchisch angeordnet, wobei die Logik die oberste Ebene darstellt, die Lagebeschreibung darunter liegt und die Darstellung die unterste Ebene einnimmt (Abb. A.2).

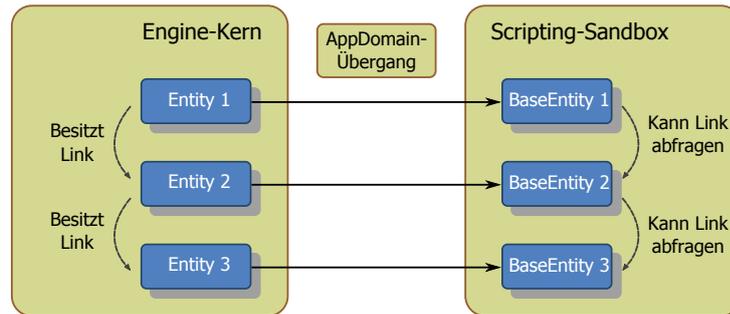


Abb. A.3.: Abhängigkeiten zwischen *Entities* und *BaseEntities* im Skript-System.

Entitäten

Eine Entität oder *Entity* ist ein Objekt, welches in der Engine einen Gegenstand der gedanklichen Anschauung repräsentiert. Es kann sich dabei z.B. um einen Sensor, eine Lichtquelle, eine Kamera oder wie im Modul *GastroInt-Sim* (Kapitel 3) um den Magen oder das Kolon des Phantoms handeln. Sollte die Engine für andere Zwecke eingesetzt werden, wie z.B. ein Computerspiel, so würde etwa die Spielfigur ebenfalls als *Entity* realisiert werden. Dabei ist zu beachten, dass das Objekt *Entity* selbst keinerlei Informationen darüber besitzt, *wie* der Gegenstand durch die 3D-Engine dargestellt wird. Dies ist Aufgabe der Klassen der Darstellungsschicht, insbesondere des *Renderables*. Stattdessen definiert die *Entity* nur das *Verhalten* des Objekts.

Entities können zu einem gerichteten Graphen verbunden werden, der logische Abhängigkeiten zwischen einzelnen *Entities* ausdrückt. Dieser Graph muss zyklensfrei sein. Existiert also eine Kante (A, B) zwischen einer *Entity A* und einer *Entity B*, so darf keine Kante (B, A) hinzugefügt werden, da sonst ein unendlicher Regress von logischen Abhängigkeiten entstünde. Die Kanten werden innerhalb der Engine als *Links* bezeichnet. Eine logische Abhängigkeit (A, B) bedeutet insbesondere, dass die *Entity A* der *Entity B* räumlich folgt, d.h. alle ihre Bewegungen im Raum ebenso durchführt.

Das Verhalten einer *Entity* auf Ereignisse wie z.B. Nutzereingaben, aktualisierte Sensorwerte oder solche Ereignisse, die durch andere *Entities* ausgelöst wurden, kann durch Skripte bestimmt werden. Um dies zu ermöglichen, besitzt jede *Entity* eine Referenz auf ein Objekt vom Typ *BaseEntity* oder eines davon abgeleiteten Typs, welches, wie in A.1.3 beschrieben, innerhalb der Sandbox des Skripting-Subsystems existiert und dort das Gegenstück der *Entity* darstellt. Falls eine *Entity E_a* mit einer anderen *Entity E_b* verlinkt ist, kann die *E_a* zugeordnete *BaseEntity B_a* den Link von *E_a* abfragen, woraufhin sie automatisch eine Referenz auf die *BaseEntity B_b* von *E_b* erhält (Abb. A.3). Hierdurch werden Interaktionen zwischen verlinkten *Entities* mit Hilfe von Skripten sehr vielseitig programmierbar.

Der Szenengraph

Entities lassen sich frei in der virtuellen, dreidimensionalen Welt positionieren und rotieren. Um diese Lageinformationen beschreiben und effizient verwalten zu können, wurde in der *Räumlichen Schicht* das Konzept des *Szenengraphen* implementiert. In der *Taraxacum*-Engine ist ein Szenengraph ein Wurzelbaum, dessen Knoten die sog. *Szenenknoten* oder *Scene-Nodes* sind und dessen Kanten räumliche Abhängigkeiten analog zu den logischen Abhängigkeiten der Entities beschreiben. Ein Szenenknoten besitzt im Wesentlichen ein lokales Koordinatensystem, welches durch eine 4×4 -Transformationsmatrix T definiert wird, eine Liste von Objekten vom Typ *Renderable*, die für die Definition der grafischen Darstellung verantwortlich sind und eine sog. *Bounding Sphere*, die die räumliche Ausdehnung dieses Szenenknotens beschreibt.

Die Matrix T eines Knotens N ist stets regulär und beschreibt das für N lokale Koordinatensystem, indem sie eine Abbildungsvorschrift angibt, auf welche Weise Vektoren, die bezüglich des Koordinatensystems von N definiert wurden, in das Koordinatensystem des in der Baumhierarchie nächsthöheren Knotens transformiert werden können. Das lokale Koordinatensystem des Wurzelknotens wird dabei als das sog. *Weltkoordinatensystem* bezeichnet, da es in dem Sinne ausgezeichnet ist, dass es von dort aus kein übergeordnetes Koordinatensystem mehr gibt. Die Transformationsmatrix des Weltkoordinatensystems ist zwingend die Identität; seine Basis ist daher gerade die Standardbasis $\{e_1, e_2, e_3, e_4\}$. Da jeder Knoten seine eigene Transformationsmatrix besitzt, ist es möglich, einen Vektor, der in einem beliebigen der lokalen Koordinatensysteme definiert wurde, in jedes andere der lokalen Koordinatensysteme des Szenengraphen zu transformieren, indem er mit den passenden Matrizen multipliziert wird. Insbesondere die Transformation eines Vektors v_i , der bezüglich des Koordinatensystems des Knotens N_i definiert wurde, in das Weltkoordinatensystem ist leicht durchführbar. Dazu muss nur das Produkt aller Matrizen vom Knoten N_i bis zur Wurzel berechnet werden. Abb. A.4 zeigt dies exemplarisch für einen Knoten der Ebene 3.

Während die Transformationsmatrix also die räumliche Lage des Szenenknotens beschreibt, dient dessen *Bounding Sphere* dazu, seine räumliche Ausdehnung zu definieren. Sie ist eine Kugel, deren Mittelpunkt im Ursprung des lokalen Koordinatensystems des Knotens liegt. Ihr Radius wird gerade so gewählt, dass von der Kugel sämtliche sichtbaren Teile aller an den Szenenknoten angehängten *Renderables* vollständig umschlossen werden. Mit Hilfe der *Bounding Sphere* ist ein effizientes sog. *Frustum Culling* möglich, bei dem vor einem Rendervorgang der 3D-Engine entschieden wird, ob das darzustellende Objekt überhaupt im Sichtfeld der Kamera liegt. Diese Entscheidung lässt sich für Kugeln besonders effizient durchführen, weil lediglich der Abstand des Kugelzentrums zu den sechs Begrenzungsebenen des Kamerasichtfeldes berechnet und mit dem Radius der

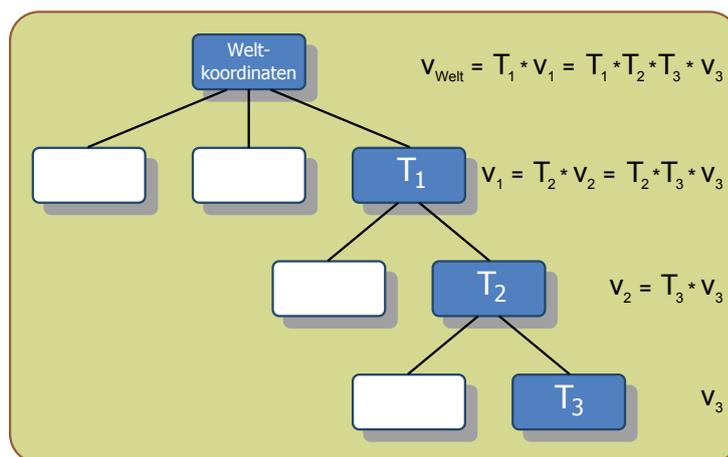


Abb. A.4.: Ausschnitt aus einem Szenengraphen. Die Matrizen T_1 bis T_3 beschreiben die relative Lage der Knoten untereinander und geben die Vorschrift an, mit der Vektoren zwischen den Koordinatensystemen transformiert werden können, hier exemplarisch für v_3 dargestellt. Die weißen Knoten stehen für beliebige Teilbäume, die hier nicht relevant sind.

Kugel verglichen werden muss. Auf Grundlage des Frustum Cullings bietet der Szenengraph eine Methode an, die eine Kamera als Eingabewert annimmt und eine Liste aller Szenenknoten zurückliefert, die im Sichtbereich dieser Kamera liegen.

Renderables

Mit dem Begriff *Renderable* wird ein Objekt bezeichnet, welches für die grafische Repräsentation einer Entity (oder eines Teils der Entity) verantwortlich ist. Aus softwarearchitektonischer Sicht ist dies jedes Objekt, das das Interface `IRenderable` implementiert. Zur Zeit existieren zwei verschiedene Implementierungen dieses Interfaces: `MeshRenderable` und `VolumeRenderable`, wobei es von `MeshRenderable` für verschiedene Aufgabenbereiche noch verschiedene Unterklassen gibt. Beispiele hierfür sind `GasterRenderable` und `ColonRenderable` aus dem Modul *GastroInt-Sim* sowie `InstrumentRenderable` aus dem Modul *X-Sim*. Die Erzeugung der grafischen Repräsentation basiert im Prinzip bei allen Renderables auf den drei in A.1.2 beschriebenen Objekten *Shader*, *Buffer* und *Textur*.

Durch das Interface werden die beiden Methoden `CheckRenderingConditions` und `Render` vorgegeben. `CheckRenderingConditions` überprüft unabhängig vom Frustum Culling, welches wie oben beschrieben auf Szenenknotenebene stattfindet, ob für das Rendern des Renderables alle notwendigen Bedingungen erfüllt sind (beispielsweise könnte ein Renderable unter bestimmten Umständen unsichtbar sein und würde in diesem Fall nicht dargestellt). Falls diese Überprüfung ein positives Resultat erzielt, wird die Methode `Render` aufgerufen, die den eigentlichen Zeichenvorgang einleitet. Die Einzelheiten

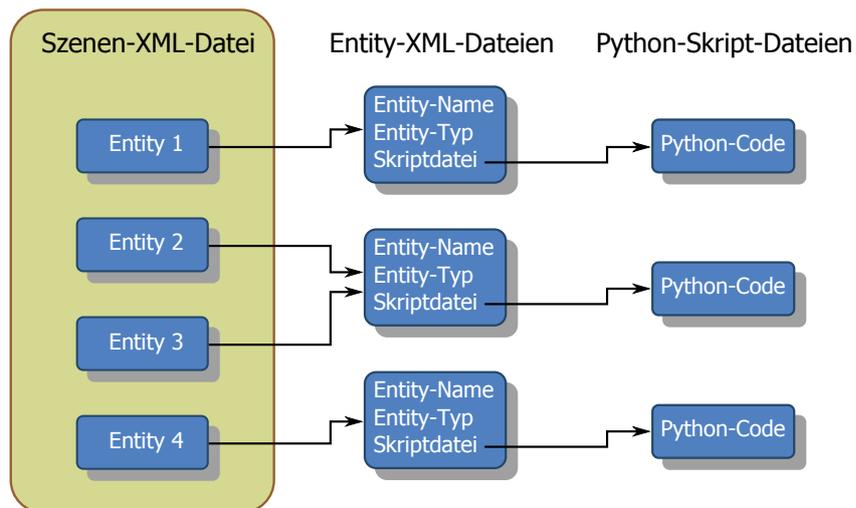


Abb. A.5.: Die einzelnen Dateien, die zusammen eine Szene beschreiben.

unterscheiden sich je nach Implementierung. Das Aussehen eines `MeshRenderable` wird durch ein Dreiecksmesh definiert, also im Wesentlichen eine Menge von Dreiecken, die die Oberfläche des Objekts darstellen. Ein `VolumeRenderable` hingegen wird durch einen Volumendatensatz definiert, bei dem nicht bloß Oberflächeninformationen, sondern echte räumliche Informationen für jeden Punkt des Volumens in Form von sog. *Voxeln* abgespeichert sind und dargestellt werden können. An einen Szeneknoten kann mehr als nur ein Renderable angehängt werden; daher kann die grafische Darstellung einer Entity aus mehreren Renderables zusammengesetzt sein.

Zusammenfassung

Zusammengefasst werden durch die Entities einer Szene die logischen Objekte und ihr Verhalten sowie ihre Interaktionen untereinander definiert, durch die Szeneknoten die Lage der Objekte im Raum sowie ihre Ausdehnung festgelegt und durch die Renderables das Aussehen der Objekte bestimmt, oder, um es ganz plakativ auszudrücken: Entities definieren „Wer macht was und mit wem?“, Szeneknoten „Wo und wie groß?“ und Renderables „Wie sieht es aus?“.

A.2. Dateiformate

A.2.1. Szenenbeschreibungsdateiformat

Die *Taraxacum*-Engine definiert einen Mechanismus, mit dem Szenen auf der Festplatte abgespeichert und von dort wieder geladen werden können. Eine solche Szene wird

durch eine XML-Datei beschrieben, in der alle Entities mit ihren Eigenschaften und etwaigen Links auf andere Entities gespeichert sind. Zu jeder Entity ist dort außerdem ein Entity-Typ gespeichert, dessen Definition in einer separaten XML-Datei erfolgt. In dieser wiederum sind relevante Informationen über die Entity abgelegt, wie z.B. welches Dreiecksmesh oder welcher Volumendatensatz der grafischen Repräsentation dient und welches Python-Skript das Verhalten der Entity festlegt. Eine solche Definitionsdatei eines *Entity-Typs* kann von mehreren Entities in einer Szene gemeinsam benutzt werden, sodass der Typ in diesem Fall nur einmal definiert werden muss. Abb. A.5 zeigt eine Übersichtsskizze der verschiedenen beteiligten Dateien.

Mit dem Skript kann unter anderem das Verhalten der Entity beim Laden und Speichern der Szene festgelegt werden. Im einfachsten Fall wird eine Entity nach dem Laden ohne weitere Aktionen der Liste der vorhandenen Entities hinzugefügt und ihr Szeneknoten wird in den Szenengraphen eingefügt. Von da an ist sie Teil der Szene. Oft ist es jedoch unerlässlich, dass bestimmte Eigenschaften der Entity nach dem Ladevorgang festgelegt werden müssen. Diese Eigenschaften sind unter anderem eine eindeutige Entity-ID, ein eventueller Verweis auf eine verlinkte Entity und eine Transformation, die die Lage der Entity im Raum beschreibt.

Bei einigen Entities müssen nach dem Laden sogar noch weitere Aktionen durchgeführt werden. Beispielsweise muss ein Lichtsensor nicht einfach nur der Szene hinzugefügt werden, sondern darüber hinaus müssen bestimmte Initialisierungsroutinen ausgeführt werden, bevor er ordnungsgemäß funktioniert. Zu diesem Zweck definiert der Typ *BaseEntity* die beiden Methoden *Loading* und *Loaded*, die zum Zeitpunkt des Ladens dieser Entity bzw., nachdem die komplette Szene geladen wurde, aufgerufen werden.

Anhand des folgenden Ausschnitts aus der Szenenbeschreibungsdatei eines Gastrophanthen (Abb. A.6) soll exemplarisch verdeutlicht werden, wie die einzelnen Entities in der XML-Datei konkret gespeichert sind. Der Ausschnitt zeigt das XML-Element, das für den Lichtsensor S_0 steht. Der *Tag* des XML-Elements, „Entity“, signalisiert dem Sze-

```
1 <Entity EntityType="LightPointSensor" Id="4" EntityName="S0" LinkedTo="3">
2   <Property Name="LowerCalibrationValue" Value="0.0" />
3   <Property Name="UpperCalibrationValue" Value="255.0" />
4   <Property Name="SensorId" Value="1" />
5   <Property Name="GroupId" Value="0" />
6   <Transformation Translation="(158.84,-10.62,-28.74)"
7     Rotation="(0.00,0.00,0.00)" Scale="(1.00,1.00,1.00)" />
8 </Entity>
```

Abb. A.6.: XML-Element, das einen Lichtsensor beschreibt.

nenimportmodul von *Taraxacum*, dass das Element eine Entity-Beschreibung darstellt. Das Attribut „EntityType“ ist zwingend notwendig, damit entschieden werden kann, um

welche Art von Entity es sich handelt. In diesem Fall ist der Typ „LightPointSensor“. Daraus folgt, dass eine Datei mit Namen *LightPointSensor.xml* existieren muss, in der nähere Details zu diesem Entity-Typ festgelegt sind. Weiterhin ist der Name der Entity „S0“, ihre ID ist 4 und sie ist mit einer anderen Entity verlinkt, deren ID 3 ist.

A.2.2. Volumendateiformat

Das Dateiformat, in dem die Volumendaten gespeichert sind, wurde vom *Eccet*-Softwarepaket [Bec03] von Dr. Beck übernommen. Es ist verhältnismäßig einfach aufgebaut und besteht aus einem kurzen Header in ASCII-Codierung, der von einer Leerzeile abgeschlossen wird, gefolgt von den sequenziellen Daten der einzelnen Voxel in Binärcodierung:

```

1 3D32
2 Volumenbreite Volumenhöhe Volumentiefe
3 Kommentarzeilen
4
5 Voxel0 Voxel1 Voxel2 Voxel3... VoxelBreite*Hoehe*Tiefe-1

```

} Header
} Daten

Abb. A.7.: Volumendateiformat von *Eccet*.

A.3. Algorithmen und Implementierungen

A.3.1. Berechnung von Geodäten

Für die Berechnung von Geodäten auf Polyedern existieren verschiedene Algorithmen (z.B. [CH90, MMP87, SSK⁺05]). Diese lassen sich grob einteilen in exakte und approximative Algorithmen. Für die Anwendung in *Simtux* reichen approximative Geodäten aus, daher werden wir uns hier auf diese beschränken. Die meisten Algorithmen zur Geodätenbestimmung operieren auf dem gegebenen Polyeder und verfeinern diesen im Bedarfsfall, bis eine exakte oder approximative Geodäte gefunden wurde. Diese Algorithmen sind in der Implementierung nicht trivial. Der in [SSK⁺05] vorgeschlagene, approximative Algorithmus hat eine Laufzeit in $\mathcal{O}(n \cdot \log(n))$. Im Folgenden soll ein Algorithmus vorgestellt werden, der ebenfalls in $\mathcal{O}(n \cdot \log(n))$ arbeitet, aber trivial zu implementieren ist. Er ist stark an [DL13] angelehnt, operiert aber nicht auf einem soliden Voxelvolumen, sondern nur auf der Voxelrepräsentation der Oberfläche des Polyeders, welcher in der vorliegenden Anwendung immer ein *Dreiecksmesh*, also eine zusammenhängende Menge von Dreiecken, ist.

Erzeugung der Voxelrepräsentation

Die in Simtux verwendeten Dreiecksmeshes sind zu einem großen Teil direkt mit Hilfe von *Eccet* aus Volumendaten erzeugt worden. Demzufolge existiert zu diesen Meshes bereits eine akkurate Voxelrepräsentation. Es sind aber auch Fälle denkbar, in denen die Meshes direkt in einem CAD-Programm erstellt oder zumindest nach dem Export aus *Eccet* nachbearbeitet wurden, sodass sie nicht mehr mit dem ursprünglichen Voxelmodell übereinstimmen. In diesen Fällen muss zunächst eine geeignete *Voxelisierung* der Dreiecksmeshes durchgeführt werden.

Definition A.2. Eine Voxelisierung eines Dreiecksmeshes M ist eine Abbildung $v : B \rightarrow V$, wobei V das durch die Bounding Box B von M umschlossene diskrete Volumen ist. Dabei gilt

$$v(p) = \begin{cases} 1 & \text{falls } \exists D \in M : p \in D \\ 0 & \text{sonst.} \end{cases}$$

Es wird also im Volumen V jedes Voxel mit dem Wert 1 markiert, in dem ein Punkt liegt, der zu einem Dreieck D des Meshes gehört. Alle übrigen Voxel werden mit 0 markiert.

Da man nicht an einem soliden Voxelvolumen interessiert ist, sondern lediglich eine Repräsentation der Meshoberfläche benötigt, ist diese Voxelisierung einfach durchzuführen. Der Ansatz beruht auf der diskreten Abtastung der Dreiecke: Die Diskretisierung eines Dreiecks D mit den Eckpunkten v_0, v_1 und v_2 und den Kanten $e_0 = v_0 - v_1$, $e_1 = v_1 - v_0$ und $e_2 = v_2 - v_0$ wird durchgeführt, indem die zu D gehörigen Punkte durch die Gleichung $p = v_0 + \lambda \cdot e_1 + \mu \cdot e_2$ unter der Nebenbedingung $\lambda + \mu \leq 1$ beschrieben werden. Indem λ und μ in diskreten Schritten durchlaufen werden, können auf diese Weise Punkte berechnet werden, die in dem Dreieck liegen. Die Schrittweite hängt dabei von der gewünschten Auflösung des Voxelgitters ab.

Dieser Algorithmus wird in einem einfachen sequenziellen Durchlauf durch die Liste aller Dreiecke realisiert. Die Berechnungen der Dreiecke sind voneinander komplett unabhängig. Im Volumen werden zwar Schreiboperationen durchgeführt, die prinzipiell zu Synchronisationsproblemen führen können, da aber immer derselbe Wert, nämlich 1, zur Markierung verwendet wird, sind diese irrelevant. Daher kann der Algorithmus einfach und ohne jede Synchronisation parallelisiert werden und damit von Mehrkernprozessoren deutlich profitieren.

Geodätenberechnung

Das Volumen wird nun als Graph interpretiert, wobei die markierten Voxel die Knoten und die Nachbarschaftsbeziehungen zwischen den Voxeln als Kanten interpretiert werden.

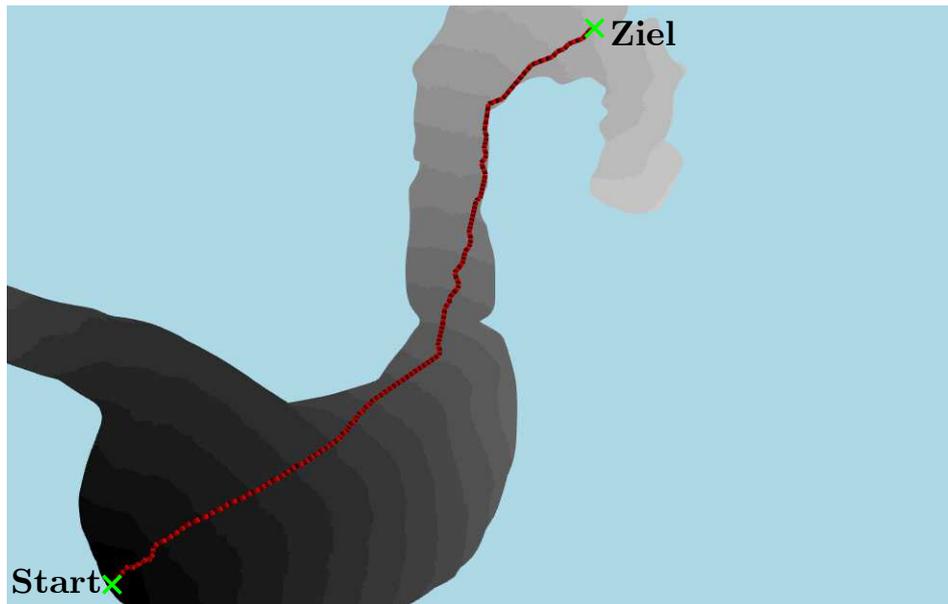


Abb. A.8.: Helligkeitscodierte geodätische Abstände zum gegebenen Startpunkt und eingezeichnete Geodäte zwischen Start und Ziel. Die Abstände werden zur besseren Sichtbarkeit grob quantisiert dargestellt.

Ein Voxel V hat dabei bis zu 26 Nachbarn, also bis zu 26 ausgehende Kanten, nämlich zu den 6 Voxeln, die mit einer Fläche an V grenzen, zu den 12 Voxeln, die mit einer Voxelkante an V grenzen und zu den 8 Voxeln, die mit einer Ecke an V grenzen. In diesem Graphen kann nun von einem gegebenen Startknoten k_s aus die Distanz zu jedem anderen Knoten k_i berechnet werden. Die Distanz zwischen zwei Knoten k_i und k_j sei definiert als die Summe der Kosten aller Kanten des kostengünstigsten Pfades von k_i zu k_j . Die Kosten werden dabei folgendermaßen gewählt: Eine Kante, die zwei über ihre Flächen benachbarte Voxel verbindet, hat Kosten 1, eine Kante, die zwei über ihre Seitenkanten benachbarte Voxel verbindet, hat Kosten $\sqrt{2}$ und eine Kante, die zwei über ihre Ecken benachbarte Voxel verbindet, hat Kosten $\sqrt{3}$. Mit dieser Kantengewichtung wird zumindest lokal die euklid'sche Metrik des \mathbb{R}^3 gut angenähert.

Diese Distanz zwischen k_s und allen n Voxeln kann nun mittels des Dijkstra-Algorithmus effizient in $\mathcal{O}(n \cdot \log(n))$ berechnet werden. Während der Algorithmus durchlaufen wird, wird in jedem Voxel gespeichert, von welchem Voxel aus es besucht wurde. Da es nur 26 verschiedene Möglichkeiten gibt, reichen für die Speicherung dieser Information 5 Bit ($2^5 = 32 > 26$) aus, wodurch sich ein Platzverbrauch von nur einem Byte pro Voxel ergibt. Abb. A.8 zeigt exemplarisch helligkeitscodiert die Distanzen jedes Punktes zum Startpunkt sowie eine auf diese Weise gefundene Geodäte.

Abbildungsverzeichnis

1.1. Flexibles Endoskop	2
1.2. Typisches Chassis eines Tübinger Phantoms mit montiertem Kopf.	3
2.1. Sizzle des Simtux-Systems.	9
2.2. Aufbau der Softwareseite des Simtux-Systems	13
3.1. Gastrophant mit montiertem Kopf.	17
3.2. Kolophant	17
3.3. Gastroskopie: Vorschub des Gastroskops bis ins Duodenum.	18
3.4. Koloskopie: Vorschub des Koloskops bis zur Dünndarmmündung.	19
3.5. Einfall des Lichtscheins	22
3.6. Skizze der Ringsensoren für das Kolon	23
3.7. Mit Shepard-Interpolation berechnete Helligkeiten	25
3.8. Überlappende Kreisscheiben.	27
3.9. Graph der Funktion $h_1(d)$	27
3.10. Funktionsgraphen des Additionsverfahrens im Eindimensionalen	28
3.11. Funktionsgraphen des Additionsverfahrens im Zweidimensionalen	29
3.12. Gegenüberstellung der Interpolationsergebnisse	30
3.13. Korrektheit des Lichtscheins	31
3.14. Beispiel für das „Auslaufen“ des Lichts im Kolon	32
3.15. Gegenüberstellung der Reichweiten in euklid’scher und geodätischer Distanz	33
3.16. Ergebnisse mit euklid’scher und geodätischer Distanz in der Praxis	35
3.18. Hauptfenster von <i>GastroIntSim</i>	38
3.19. Darstellung des Kolons zu verschiedenen Zeitpunkten	39
3.20. Endgültige Oberflächenbeleuchtung des Magens	40
3.21. Darstellung der Messwerte von Lage- und Drucksensoren	42
4.1. Begradigter Verlauf des Duodenoskops	44
4.2. Frontale Ansicht der <i>Papilla Vateri</i>	44
4.3. Frontale Ansicht der künstlichen Papille im <i>Biliphanten</i>	46
4.4. Blick auf den geöffneten Biliphanten	46
4.5. Begradigung. Abbildungen von Duygu Özmen.	48
4.6. Röntgenvorgang	50

4.7. Skizze der geometrischen Verhältnisse beim Raycasting	54
4.8. Shadercode zur Röntgendarstellung	55
4.9. Vergleich von echtem und simuliertem Röntgenbild	55
4.10. Nur Sondierungsdrähte und Papillotome in den Anfängerkursen	56
4.11. Skizze der Definition der Gallengangsegmente	57
4.12. Idealisierter Zylinderkondensator	58
4.13. Die Verläufe des Instruments in der Realität	60
4.14. Zylinderkondensatoren von oben betrachtet	62
4.15. ΔC_{rel} in Abhängigkeit von e_{rel} aufgetragen.	63
4.16. Testanordnung für die kapazitive Sensorik.	64
4.17. Die vier Messungen der beiden Sensoren.	65
4.18. Durchschnittliche Sensorwerte	67
4.19. Darstellungen der Messwerte der Sensoren, ohne den jeweils anderen	68
4.20. Darstellung der Messwerte der Schlaufe	69
4.21. Vergleich von echtem und simuliertem Röntgenbild eines Sondierungsdrahts	70
4.22. Zusammengesetzter Spline	72
4.23. Definition des Splines innerhalb des linearen Gallengangsystems	73
4.24. Spline mit verschiedenen <i>Spannungsfaktoren</i> τ	74
4.25. Unter Umständen ist ein Béziersegment nicht ausreichend	75
4.26. Skizze der Splineauswertung	77
4.27. a) Drei einzelne Splines. b) Splinebaum mit drei Blättern.	79
4.28. Ein einzelner Pfad innerhalb des Splinebaums.	80
4.29. Berechnung der Tangenten in den Verzweigungsknoten eines Splinebaums	80
4.30. Unsicherheiten bei der Auswahl des richtigen Pfades	81
4.31. Auswahl der korrekten Pfade über ihre Punktstände	83
4.32. Pseudo-dynamischer Verlauf des Instruments	84
4.33. Modifiziertes, transparentes Gallengangsystem	87
4.34. Korrektur des „Fischaugen“-Effekts	88
4.35. Rohdaten der optischen Erkennung	89
4.36. Globale Schwellwerte sind zur Segmentierung der Instrumente nicht geeignet	90
4.37. Bewegungsartefakte im Differenzbild	93
4.38. Verschiebung zwischen P_1 und P_2	94
4.39. Beispiele für die optische Erkennung	95
4.40. Gegenüberstellung: ohne und mit der endgültigen Deformationskorrektur	98
4.41. Gedrehte Ansichten	99
4.42. Monitoranordnungen	100
4.43. Das Röntgenfenster von X-Sim	101
4.44. Bedienfelder	101
4.45. Fußpedal zur Auslösung des virtuellen Röntgenvorgangs.	102
4.46. Schematische Begradigungsanzeige.	103

4.47. Elektroden auf der Magenoberfläche	104
4.48. Darstellung der Begradigungsanzeige direkt im simulierten Röntgenbild .	104
4.49. Foto des Y-Stücks aus Glas.	106
5.1. Prototyp der Hardware des <i>Broncho-Sim</i> -Moduls.	111
5.2. Benutzeroberfläche des Prototyps des <i>BronchoSim</i> -Moduls.	112
A.1. OpenGL-Pipeline	114
A.2. Hierarchie innerhalb eines Szenenobjekts	117
A.3. Abhängigkeiten zwischen <i>Entities</i> und <i>BaseEntities</i> im Skript-System. . .	118
A.4. Ausschnitt aus einem Szenengraphen	120
A.5. Die einzelnen Dateien, die zusammen eine Szene beschreiben.	121
A.6. XML-Element, das einen Lichtsensor beschreibt.	122
A.7. Volumendateiformat von <i>Eccet</i>	123
A.8. Helligkeitscodierte geodätische Abstände	125

Tabellenverzeichnis

2.1. Alle bisher verfügbaren Phantomtypen und ihre Codierungen.	11
4.1. Messreihen für den ersten Sensor.	65
4.2. Messreihen für den zweiten Sensor.	66
4.3. Messwerte des ersten Sensors mit eingeführter Schlaufe.	69

Literaturverzeichnis

- [Bec03] BECK, Andreas: *Ein System zur Verarbeitung und Visualisierung von Voxel-daten.*, Heinrich-Heine-Universität Düsseldorf, Diss., 2003
- [CH90] CHEN, Jindong ; HAN, Yijie: Shortest Paths on a Polyhedron. In: *Proceedings of the Sixth Annual Symposium on Computational Geometry*. New York, NY, USA : ACM, 1990 (SCG '90). – ISBN 0–89791–362–0, 360–369
- [CR12] COZZI, Patrick ; RICCIO, Christophe: *OpenGL Insights*. CRC Press, 2012. – ISBN 978–1439893760
- [CWW12] CRANE, Keenan ; WEISCHEDEL, Clarisse ; WARDETZKY, Max: Geodesics in Heat. In: *CoRR* abs/1204.6216 (2012)
- [DKR79] DEMLING, Ludwig ; KOCH, H ; RÖSCH, W: *Endoskopisch retrograde Cholangio-Pankreatikographie: ERCP*. Schattauer GmbH, 1979. – ISBN 3794506456
- [DL13] DIONNE, Olivier ; LASA, Martin de: Geodesic Voxel Binding for Production Character Meshes. In: *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA : ACM, 2013 (SCA '13). – ISBN 978–1–4503–2132–7, 173–180
- [DM] DEROSE, T ; MEYER, M: *Harmonic Coordinates*. Pixar Studios,
- [Fri08] FRIMBERGER, E.: *Ercp trainingsmodell (ercp-tm) für endoskopische diagnostik und therapie im bereich des pankreo-biliären systems*. <http://www.google.com/patents/W02008101477A2?c1=de>. Version: August 28 2008. – WO Patent App. PCT/DE2008/000,283
- [Gas03] *Gastroenterologische Endoskopie : das Referenzwerk zur endoskopischen Diagnostik und Therapie*. Thieme, Stuttgart, 2003
- [GGUA13] GALSTER, M ; GUHL, C ; UDER, M ; ADAMUS, R: *Exposition der Augenlinse des Untersuchers und Effizienz der Strahlenschutzmittel bei fluoroskopischen Interventionen*. In: *Rofa* (2013), S. 474–481

- [GID⁺12] GRUND, K.-E. ; INGENPASS, R ; DURST, F ; SCHWEIZER, U ; VIETZ, M ; AURICH, V: Neuartiges Hands-on-Phantom für das realistische Training der gesamten diagnostischen und therapeutischen ERCP. In: *Endo heute* (2012), S. 14–17
- [GIS⁺12a] GRUND, K.-E. ; INGENPASS, R ; SCHWEIZER, U ; VIETZ, M ; AURICH, V: Neues ERCP-Trainings-Modell für alle diagnostischen und therapeutischen Eingriffe. In: *Z Gastroenterol* (50) (2012), S. 4–5
- [GIS⁺12b] GRUND, K.E. ; INGENPASS, R ; SCHWEITZER, U ; AURICH, V ; VIETZ, M: *Neues ERCP-Trainings-Modell für alle diagnostischen und therapeutischen Eingriffe*. 2012. – Poster, präsentiert auf dem XXIII. Kongress der Südwestdeutschen GEsellschaft für Gastroenterologie in Freiburg
- [GSFa] GRUND, K.-E. ; STAUB, T ; FARIN, G: Integriertes Ausbildungskonzept für die diagnostische und interventionelle flexible Endoskopie. In: *Z Gastroenterol* 37
- [GSFb] GRUND, K.-E. ; STAUB, T ; FARIN, G: New modalities for training in flexible endoscopy with special regard to interventions. In: *Gastrointest Endosc* 49
- [Hea14] HEALTHCARE, CAE: *EndoVR*. <http://www.caehealthcare.com/eng/surgical-simulators/endovr>, 2014. – [Online; Version vom 11. Mai 2014]
- [Jac99] JACKSON, John D.: *Classical electrodynamics*. 3rd ed. New York, NY : Wiley, 1999 <http://cdsweb.cern.ch/record/490457>. – ISBN 9780471309321
- [LAB⁺98] LOPERFIDO, S ; ANGELIN, G ; BENEDETTI, G ; CHILOVI, F ; COSTAN, F ; DE BERARDINIS, F ; DE BERNARDIN, M ; EDERLE, A ; FINA, P ; FRATTON, A.: Major early complications from diagnostic and therapeutic ERCP: a prospective multicenter study. In: *Gastrointest Endoscopy* 48 (1998), S. 1–10
- [MMP87] MITCHELL, Joseph S. B. ; MOUNT, David M. ; PAPADIMITRIOU, Christos H.: The Discrete Geodesic Problem. In: *SIAM J. Comput.* 16 (1987), August, Nr. 4, 647–668. <http://dx.doi.org/10.1137/0216045>. – DOI 10.1137/0216045. – ISSN 0097–5397
- [Muc05] MUCKENHAUPT, Tina: *Training in der flexiblen Endoskopie: Entwicklung eines Simulators für die ERCP und Integration in ein modular aufgebautes Gesamtsystem.*, Eberhard-Karls-Universität zu Tübingen, Diss., 2005
- [NME⁺00] NEUMANN, M ; MAYER, G ; ELL, C ; FELZMANN, T ; REINGRUBER, B ; HORBACH, T ; HOHENBERGER, W: The Erlangen Endo-Trainer: life-like simulation for diagnostic and interventional endoscopic retrograde cholangiography.

- In: *Endoscopy* 32 (2000), Nr. 11, 906-10. <http://www.biomedsearch.com/nih/Erlangen-Endo-Trainer-life-like/11085482.html>. – ISSN 0013-726X
- [Noa95] NOAR, M D.: An established porcine model for animate training in diagnostic and therapeutic ERCP. In: *Endoscopy* 27 (1995), Nr. 1, 77-80. <http://www.biomedsearch.com/nih/established-porcine-model-animate-training/7601042.html>. – ISSN 0013-726X
- [RED] *OpenGL Programming Guide: The Official Guide to Learning OpenGL (Red Book)*. Addison-Wesley
- [Sal06] SALOMON, David: *Curves and Surfaces for Computer Graphics*. 2006. – ISBN 0-387-24196-5 , 0-387-28452-4 (e-book)
- [SB02] SCHLEGEL, W ; BILLE, J: *Medizinische Physik Bd. 2 - Medizinische Strahlenphysik*. Springer, 2002. – ISBN 3-540-65254-X
- [She68] SHEPARD, Donald: A Two-dimensional Interpolation Function for Irregularly-spaced Data. In: *Proceedings of the 1968 23rd ACM National Conference*. New York, NY, USA : ACM, 1968 (ACM '68), 517-524
- [Sim14] SIMBIONIX: *GI Mentor*. <http://symbionix.com/>, 2014. – [Online; Version vom 29. März 2014]
- [SM13] STAUB, David ; MURPHY, Martin J.: A digitally reconstructed radiograph algorithm calculated from first principles. In: *Med Phys* 40 (2013), Nr. 1, 011902. <http://www.biomedsearch.com/nih/digitally-reconstructed-radiograph-algorithm-calculated/23298093.html>. – ISSN 0094-2405
- [SSK⁺05] SURAZHSKY, Vitaly ; SURAZHSKY, Tatiana ; KIRSANOV, Danil ; GORTLER, Steven J. ; HOPPE, Hugues: Fast Exact and Approximate Geodesics on Meshes. In: *ACM SIGGRAPH 2005 Papers*. New York, NY, USA : ACM, 2005 (SIGGRAPH '05), 553-560
- [Tho66] THOMA, P.: Kapazität und Wärmefluß zwischen ineinanderliegenden Zylindern mit nichtparallelen Achsen. In: *Archiv für Elektrotechnik* 50 (1966), Nr. 6, 395-400. <http://dx.doi.org/10.1007/BF01407272>. – DOI 10.1007/BF01407272. – ISSN 0003-9039

- [TLD14] TRIANTAFYLLOU, Konstantinos ; LAZARIDIS, Lazaros D. ; DIMITRIADIS, George D.: Virtual reality simulators for gastrointestinal endoscopy training. In: *World J Gastrointest Endosc* 6 (2014), Nr. 1, 6-12. <http://www.biomedsearch.com/nih/Virtual-reality-simulators-gastrointestinal-endoscopy/24527175.html>. – ISSN 1948–5190
- [VAG13a] VIETZ, M ; AURICH, V ; GRUND, K.-E.: *Neuartige virtuelle Simulation des Röntgenbildes und der Endoskop-Position für das phantomgestützte Hands-on-ERCP-Training ohne Strahlenbelastung*. 2013. – Poster, präsentiert auf der DGE-BV 2013 in München
- [VAG13b] VIETZ, M. ; AURICH, V. ; GRUND, K.-E.: Neuartiges Hands-On-Trainings-Phantom für die ERCP mit Röntgensimulation. In: *Journal* (2013), S. 666–667
- [Vie07] VIETZ, Matthias: *Schnelles Volumenrendering unter Verwendung von 3D-Consumerhardware.*, Heinrich-Heine-Universität Düsseldorf, Masterarbeit, 2007
- [VzGA13] VIETZ, M ; ÖZMEN, D ; GRUND, K.-E. ; AURICH, V: Neuartige virtuelle Simulation des Röntgenbildes und der Endoskop-Position für das phantomgestützte Hands-on-Training ohne Strahlenbelastung. In: *Endo heute* 26 (2013), S. 17
- [Özm12] ÖZMEN, Duygu: *Detektion der Papilla Duodeni Maior im Bild eines Duodenoskops.*, Heinrich-Heine-Universität Düsseldorf, Bachelorarbeit, 2012