

EINSATZ VON PRÄ- UND
INTRAOPERATIVER BILDGEBUNG ZUR
UNTERSTÜTZUNG VON
OPERATIONSPLANUNG UND
-DURCHFÜHRUNG IN DER
ROBOTERASSISTIERTEN CHIRURGIE

Dissertation

zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften

vorgelegt von:

Erwin Gerz

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen

Siegen, Dezember 2020

Betreuer und erster Gutachter
Prof. Dr.-Ing. Dr. h.c. Hubert Roth
Universität Siegen

Zweiter Gutachter
Prof. Dr. Kaspar Althoefer
Queen Mary University of London

Tag der mündlichen Prüfung
02.12.2021

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Regelungs- und Steuerungstechnik der Universität Siegen im Rahmen des EU-Forschungsprojektes „STIFF-FLOP“ und dem vom BMBF geförderten Projekt „S//SPINE“. Zuerst möchte ich mich bei Herrn Prof. Dr.-Ing. Dr. h.c. Hubert Roth für die Möglichkeit bedanken, an seinem Lehrstuhl Forschen und Promovieren zu können. Ich bedanke mich für die Betreuung meiner Arbeit und die Übernahme der Rolle des Erstgutachters. Durch Sie hatte ich viel Freiheit beim Forschen und die Möglichkeit viele Kontakte in verschiedenen Projekten zu knüpfen. Neben den fachlichen Erfahrungen konnte ich auch viele persönliche Eindrücke im In- und Ausland sammeln, die mich bereichert haben.

Bei Prof. Kaspar Althoefler bedanke ich mich für die konstruktive und erfolgreiche Koordination des „STIFF-FLOP“ Projektes, sowie für die Übernahme der Rolle des zweiten Gutachters.

Mein weiterer Dank gilt Herrn Dr. Jürgen Wahrburg für die Möglichkeit, als Teil der MODICAS Arbeitsgruppe des ZESS, das Projekt „S//SPINE“ voranzubringen. Besonders bedanke ich mich für die hilfreichen Gespräche die zur Fertigstellung dieser Dissertation beigetragen haben.

Persönlich bedanke ich mich bei meinen Kollegen des Lehrstuhls RST und den Kollegen der MODICAS Arbeitsgruppe des ZESS für die freundschaftliche Arbeitsatmosphäre, die hilfsbereite Unterstützung bei der Arbeit und für die Diskussionsrunden in denen viele Anregungen entstanden.

Herzlich bedanke ich mich bei meiner Frau Elvira und bei meinen Eltern, die mich während dieser Zeit persönlich unterstützt und motiviert haben.

Inhaltsverzeichnis

Kurzfassung	IV
Abstract	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
2 Entwicklung der computer- und roboterassistierten Chirurgie	3
3 Betrachtung von zwei Anwendungsgebieten	8
4 Komponenten der roboterassistierten Chirurgie	10
4.1 Medizinische Bildgebung	10
4.2 3D-Lokalisierverfahren zur intraoperativen Navigation	15
4.3 Systeme in der roboterassistierten Chirurgie	19
4.3.1 Assistenzsysteme für die knöchernen Chirurgie	19
4.3.2 Flexible Robotersysteme für die Weichteilchirurgie	23
5 Problemstellung und Ziel	26
5.1 Medizinische Aufgabenstellung	26
5.2 Technische Umsetzung mit Assistenzrobotersystemen	30
6 Operationsplanung und -durchführung in der knöchernen Chirurgie	33
6.1 Systemübersicht des verwendeten Assistenzrobotersystems	34
6.2 Entwicklung einer Software zur Operationsplanung mit dem Software Framework Mevislab	37
6.2.1 Visualisierung medizinischer Bilddaten	38
6.2.2 Planung von Pedikelschrauben	46
6.2.3 Anbindung der Stereokamera	54
6.2.4 Anbindung an das Robotersystem	56
6.2.5 Bestimmung der Zielposition	59
6.2.6 Visualisierung von Komponenten im Bilddatensatz	60
6.2.7 Entwurf einer Benutzeroberfläche	62
6.3 Registrierung der intraoperativen Patientenposition	66
6.3.1 Einsatz eines Kalibrierkörpers zur Registrierung	68

6.3.2	Detektion des Kalibrierkörpers im Volumen der Bildgebungsmodalität	70
6.3.3	Registrierung der detektierten Markerpositionen	73
6.3.4	Benutzeroberfläche zur Registrierung des Kalibrierkörpers . .	78
6.3.5	Akquise eines registrierten 3D Bilddatensatzes	82
6.4	Präoperative Operationsplanung	83
6.4.1	Matchingstrategien für multimodale Wirbelaufnahmen	84
6.4.2	Erweiterung der Software zur präoperativen Operationsplanung	89
6.5	Ergebnis Betrachtung des Gesamtsystems	95
6.5.1	Messung der Planungsgenauigkeit am Modell	95
6.5.2	Positioniergenauigkeit des Systems am Beispiel einer intraoperativen Planung	99
6.5.3	Positioniergenauigkeit des Systems am Beispiel einer präoperativen Planung	102
7	Operationsplanung und -durchführung in der Weichteilchirurgie	106
7.1	Beschreibung des verwendeten flexiblen Roboterarmes	107
7.2	Aufbau des optischen Positionsbestimmungssystems	109
7.2.1	Lokalisation der intraoperativ verwendeten Komponenten . . .	113
7.2.2	Bilderfassung und Bildvorverarbeitung	115
7.3	Erkennung des flexiblen Roboterarmes in der Endoskop Aufnahme . .	118
7.3.1	Anwendung der generalisierten Hough Transformation zur Detektion	119
7.3.2	Anwendung der merkmalsbasierten Detektion	122
7.3.3	Detektion des flexiblen Roboterarmes unter Verwendung von maschinellem Lernen	124
7.3.4	Training des Support Vector Machine Algorithmus	129
7.3.5	Ergebnisse der Detektion des flexiblen Roboterarmes	134
7.3.6	Implementierung der Online Detektion	140
7.4	Weiterführende Arbeiten zur Erkennung der Verbindungspunkte des STIFF-FLOP Armes	143
8	Diskussion der Ergebnisse für die knöchernen und die Weichteilchirurgie	146
9	Zusammenfassung und Ausblick	149
	Literaturverzeichnis	152
	Abbildungsverzeichnis	162

Tabellenverzeichnis	171
A Mevislab Funktionen der Planungssoftware	i
B Mevislab Modul zur Registrierung von Punktdatensätzen	v
C Mevislab Makro Modul zur Registrierung von multimodalen Bilddatensätzen	vii
D Funktionen zur Detektion des STIFF-FLOP Armes durch den Einsatz von Support Vector Machines	xii

Kurzfassung

Diese Dissertation beschäftigt sich mit der Entwicklung computergestützter Verfahren zur Verarbeitung medizinischer Bilddaten, die von verschiedenen prä- und intraoperativen Bildgebungsmodalitäten stammen. Der Schwerpunkt liegt insbesondere darin, die Durchführung minimalinvasiver chirurgischer Operationsverfahren mit robotischen Assistenzsystemen zu unterstützen. Diese Systeme bieten einerseits das Potenzial, die Genauigkeit und Reproduzierbarkeit der Operationsergebnisse deutlich zu verbessern. Andererseits werden sie sich nur durchsetzen, wenn sie für die Operateure einfach und sicher zu handhaben sind. Daraus resultieren besondere Anforderungen an die Software zur Operationsplanung und -durchführung, die im Rahmen dieser Arbeit analysiert und in Lösungsvorschlägen umgesetzt werden.

Den medizinischen Hintergrund für die Arbeit bilden zwei verschiedene Anwendungsszenarien. Das erste Szenario für diese Arbeit bietet die Weiterentwicklung der Software für die Operationsplanung und -durchführung bei Eingriffen an der Wirbelsäule mit einem Assistenzroboter. Im ersten Schritt müssen die im Operationsraum verwendeten Bildgebungskomponenten registriert werden. Hierzu wird ein Verfahren basierend auf der Verwendung von Markern, die sowohl von optischen Stereokameras als auch von intraoperativ eingesetzten C-Bögen erkannt werden, entwickelt. Um dem Operateur eine präoperative Planung zu ermöglichen, werden präoperative Bilddaten verwendet. Zur Einbindung der präoperativen Planung bei der Operationsdurchführung muss diese, über die intraoperative Bildgebungsmodalität, auf die aktuelle Patientenposition bezogen werden. Hierfür wird ein geeignetes Matchingverfahren implementiert.

Das zweite Anwendungsszenario beschreibt die Entwicklung eines neuartigen flexiblen Roboterarms für minimalinvasive Eingriffe, bei dem die aktuelle Position überwacht werden muss. Hierfür wird ein optisches Positionsbestimmungssystem, bestehend aus einem Endoskop, das von einer Stereokamera überwacht wird, aufgebaut. Die größte Herausforderung liegt in der schnellen und genauen Detektion des flexiblen Roboterarmes durch die Verarbeitung von Bildern der Endoskopkamera. Um die Position des flexiblen Roboterarmes zu bestimmen, soll im Rahmen dieser Arbeit eine geeignete Methode ausgewählt und implementiert werden. Mit dem Ziel das System anpassungsfähig und flexibel umzusetzen, sollen auch Algorithmen, die Maschinelles Lernen zur Lösung einsetzen, herangezogen werden.

Abstract

This dissertation describes the development of computer-aided processes, for processing medical images from different pre- and intraoperative imaging modalities. The main focus lies on supporting the execution of minimal invasive surgical procedures with robotic assistance systems. Those systems offer the potential to enhance the accuracy and the reproducibility of surgical procedures significantly. On the other hand, such systems will only become established, if they are easy and safe to use for operators. The needed special requirements for the software, used for planning and execution, are analyzed within this review and implemented in a proposal for solution.

The medical background for this review were two projects applied in practice. The first project describes the further development of a software for planning and execution of a spine surgical intervention, using a robotic assistance system. In the first step, the intraoperative imaging modality has to be registered with the medical image of the patient. Therefore, a procedure is developed, which uses markers that are detected by an optical stereo camera system as well as by the intraoperatively used x-ray system. In order to enable the surgeon to plan the intervention beforehand, preoperative images are used. To execute the preoperative planning, it has to be transferred to the actual patient position on the operating table. Therefore, a suitable method is implemented to match the preoperative image data with intraoperative images.

The second project addresses the operation of a new flexible arm for minimal invasive surgery. The optical positioning system, required to operate such device safely, consists of an endoscopic camera that is monitored by a stereo camera system. The biggest challenge lies in the fast and precise detection of the flexible arm using the images of the endoscopic camera. To determine the position of the flexible arm, a suitable method has to be selected and implemented. With the aim of making the system flexible and adaptable, algorithms that use machine learning are applied to solve the individual problem.

Abkürzungsverzeichnis

DRB	Dynamic Reference Base - Referenzkörper für Stereokamera-system
ARB	Aim Rigid Body - Referenzkörper am Patienten
CAS	Computer Assisted Surgery
CBVol	Koordinatenursprung eines C-Bogen 3D-Bilddatensatzes
CRB	C-Bogen Rigid Body - Referenzkörper am C-Bogen
CT	Computertomographie
CTVol	Koordinatenursprung eines CT 3D-Bilddatensatzes
DICOM	Digital Imaging and Communications in Medicine
HNI	Host-Network-Interface
IMP	Frame der Implantatposition
KRB	Kalibrierkörper Rigid Body - Referenzkörper am Kalibrierkörper
LoG	Laplacian of Gaussian
MERIT	MEVIS Image Registration Toolkit
MIS	Minimalinvasive Chirurgie (minimal invasive surgery)
MODICAS	modular interactive Computer Assisted Surgery
MPR	Multi Planar Reconstruction
NMI	Normalized Mutual Information
OTS	Optisches Tracking System
PACS	picture archiving and communication system
PRB	Pointer Rigid Body - Referenzkörper am Zeigelinstrument
ROS	Robot Operating System
RRB	Robot Rigid Body - Referenzkörper am Roboter
SVM	Support Vector Machines
TCP	Robot Toolcenterpoint - Zentrum des Roboterflansches
TTP	Robot Tooltip - Werkzeugspitze am Roboter

1 Einleitung

Durch den Einsatz neuer Technologien können chirurgische Eingriffe zunehmend minimalinvasiv durchgeführt werden. Der Vorteil der minimalen Inzision stellt gleichzeitig auch neue Anforderungen an die Technologie. Eine der Herausforderungen bei minimalinvasiven chirurgischen Eingriffen ist die limitierte oder teilweise nicht vorhandene Sicht auf das Operationsgebiet. Um Instrumente zielgenau zu positionieren werden während der Operation verschiedene Bildgebungsmodalitäten und Navigationssysteme eingesetzt.

In dieser Arbeit werden computergestützte Verfahren zur Verarbeitung medizinischer Bilddaten, die von verschiedenen prä- und intraoperativen Bildgebungsmodalitäten stammen, entwickelt. Der Schwerpunkt liegt insbesondere darin, die Durchführung minimalinvasiver chirurgischer Operationsverfahren mit robotischen Assistenzsystemen zu unterstützen. Diese Systeme bieten einerseits das Potenzial, die Genauigkeit und Reproduzierbarkeit der Operationsergebnisse deutlich zu verbessern. Andererseits werden sie sich nur durchsetzen, wenn sie für die Operateure einfach und sicher zu handhaben sind. Daraus resultieren besondere Anforderungen an die Software zur Operationsplanung und -durchführung, die im Rahmen dieser Arbeit analysiert und in Lösungsvorschlägen umgesetzt werden.

Die Anforderungen an chirurgische Assistenzsysteme hängen von der jeweiligen Anwendung ab und unterscheiden sich erheblich. Bei Anwendungen in der knöchernen Chirurgie wie zum Beispiel der Hüftendoprothetik oder Operationen an der Wirbelsäule können robotische Assistenzsysteme den Chirurgen durch eine genaue Platzierung von Instrumenten unterstützen. Als Bildgebungsmodalität werden hier meist intraoperative Röntgengeräte eingesetzt. Bei Anwendungen in der Weichteilchirurgie hingegen sind starre Operationsrobotersysteme wenig hilfreich wenn Organen ausgewichen werden muss um das Operationsgebiet zu erreichen. Hierfür erschließt der im Rahmen des STIFF-FLOP Projektes entwickelte flexible Roboterarm, mit der Möglichkeit sich zu versteifen, neue Einsatzmöglichkeiten.

Im Rahmen dieser Dissertation werden zwei Anwendungsgebiete mit unterschiedlichen Anforderungen bearbeitet.

Als Beitrag zur Unterstützung und Weiterentwicklung des am ZESS entwickelten Assistenzrobotersystems MODICAS wird an einer Lösung gearbeitet, die es dem Chir-

urgen ermöglicht, Operationen an der Wirbelsäule roboterassistiert durchzuführen. Das Hauptaugenmerk der Entwicklung liegt auf der Schließung der Kette von der präoperativen Operationsplanung über die intraoperative Registrierung der Bildaufnahmen mit der Patientenposition, bis hin zur robotergestützten Durchführung der Operation. Der eigene Beitrag hierfür wird in Kapitel 6 vorgestellt und umfasst die Einführung eines neuen Software Frameworks und den Aufbau einer Planungssoftware für chirurgische Eingriffe, was im Abschnitt 6.2 beschrieben ist. Als neuer Beitrag wird im Abschnitt 6.3 ein Verfahren vorgestellt, in dem die Patientenposition mit dem Bildvolumen der Bildgebung registriert wird. Durch den Einsatz eines Kalibrierkörpers können Bilddaten von verschiedenen Bildgebungsmodalitäten für die Planung und Durchführung verwendet werden. Im Abschnitt 6.4 wird zusätzlich ein Verfahren vorgestellt mit dem eine präoperative Operationsplanung in einem 3D-Bilddatensatz mit den im Operationsraum aufgenommenen Bilddaten überlagert wird.

Als weiteres Anwendungsgebiet wird ein optisches System zur Detektion des, im Rahmen des STIFF-FLOP Projektes entwickelten, flexiblen Roboterarmes aufgebaut. Kapitel 7 stellt die eigenen Beiträge zum Aufbau des optischen Lageüberwachungssystems für den flexiblen STIFF-FLOP Arm vor. Der Schwerpunkt liegt auf der Detektion des flexiblen Roboterarmes im Videobild der Endoskopkamera. Hierfür wurden Methoden ausgewählt und implementiert um anschließend das Ergebnis der Erkennung zu untersuchen.

Abschließend werden die Ergebnisse für beide Anwendungsbereiche in Kapitel 8 diskutiert und in Kapitel 9 zusammengefasst, sowie ein Ausblick auf zukünftige Arbeiten gegeben.

2 Entwicklung der computer- und roboterassistierten Chirurgie

Als computerassistierte Chirurgie (CAS) werden Systeme bezeichnet, die chirurgische Eingriffe bei der Planung und der Ausführung unterstützen. Computergestützte und roboterassistierte Assistenzsysteme haben sich in den meisten Operationsräumen etabliert. Ziel dieser Systeme ist es, den Chirurgen zu unterstützen um Operationen genauer, sicherer und schneller durchführen zu können. Durch eine Planung von zu platzierenden Schrauben oder Implantaten in einem Bilddatensatz der zu operierenden Region, kann ein Eingriff vorbereitet werden und festgelegt werden, wo Inzisionen gemacht werden sollen oder welche sensiblen Bereiche besondere Aufmerksamkeit erfordern. Während der Operation können chirurgische Instrumente im Bilddatensatz eingeblendet werden und dadurch eine chirurgische Navigation ermöglichen. Durch den Einsatz eines Robotersystems kann die Ausführung von chirurgischen Eingriffen zusätzlich unterstützt werden.

Eines der ersten Robotersysteme in der Chirurgie war der im Jahr 1984 eingesetzte Arthrobot (Vancouver, BC, Canada) [57]. Sprachgesteuert unterstützte der Arthrobot den Chirurgen bei Arthroskopien durch eine Positionierung der unteren Extremitäten.

Im Jahr 1985 wurde der Unimation PUMA 200 Roboter in der Neurochirurgie eingesetzt. Unterstützt durch einen Computertomographen diente der Roboter als Positioniersystem für Biopsie Nadeln, mit denen Hirntumore behandelt werden konnten, die zuvor als inoperabel galten [56].

Am Imperial College London wurde das PROBOT System speziell für die Entfernung von Prostata Gewebe entwickelt. Durch Ultraschallbildgebung wurde ein Modell erzeugt, in dem die zu entfernenden Bereiche ausgewählt wurden. Anschließend führte der Roboter die angelernten Schnitte aus. Im Jahr 1988 wurde der Einsatz des Systems demonstriert [45].

Eines der ersten, für orthopädische Eingriffe zugelassenes Robotersystem, war der 1998 von Integrated Surgical Systems (ISS) vorgestellte ROBODOC, der zum ausfräsen von Hüftprothesen eingesetzt wurde. Zur Bestimmung der intraoperativen Lage wurde der zu operierende Knochen fest an dem Robotersystem fixiert bevor

operiert werden konnte [17] [9]. Nach einer Planung in einer CT-Aufnahme des Patienten, erfolgte die Ausfräsung des Femur Schaftes vollautomatisiert. Daraus resultierte ein umfangreicher Aufwand zur Operationsvorbereitung und auch das entwickelte Robotersystem brauchte viel Platz im Operationsraum. Die Abbildung 2.1 zeigt das ROBODOC System.



Abbildung 2.1: Das von Integrated Surgical Systems entwickelte ROBODOC System zum Ausfräsen des Femurschaftes für Hüftprothesen [67]

Der ROBODOC erlebte gemeinsam mit dem CASPAR System einen Höhenflug in den Medien, bevor der tiefe Fall und die darauf folgende Skepsis über den Einsatz von Operationsrobotern folgte [27]. Auch wenn die Ausfräsungen präzise ausgeführt wurden, häuften sich Beschwerden und Schadenersatzforderungen von Patienten, bei denen Nerven und Muskeln durch das System beschädigt wurden, mit dem Ergebnis, dass ISS 2005 alle Mitarbeiter entließ und die Produktion eingestellt wurde [28] [60].

Das CASPAR-System (Computer Assisted Surgical Planning and Robotics) wurde von orto Maquet (Rastatt, Deutschland) entwickelt und erstmals 1997 für die Hüftprothetik und später auch für Kreuzbandoperationen sowie für Eingriffe am Kniegelenk eingesetzt. Die Erfassung der Position des Patienten auf dem Operationstisch erfolgte über Metallpins, die vor der Bildaufnahme am Patienten angebracht und zu Beginn der Operation mit dem Roboter angetastet wurden [48]. Nach der Planung des Eingriffes im Bilddatensatz wurde der Knochen von dem Robotersystem ausgefräst. Aber auch das CASPAR-System konnte sich nicht durchsetzen und wurde 2001 eingestellt.

In den ersten robotischen Systemen wurden industrielle Roboter verwendet, die eine gute Positioniergenauigkeit besitzen. Allerdings mussten hier noch Strategien entwickelt werden, die einen sicheren Einsatz für den Patienten und die operierenden Chirurgen gewährleisten. Dies erschwerte die Zertifizierung der ersten robotischen Systeme [85]. Insbesondere der hohe Platzbedarf und die komplizierte Einrichtung führte dazu, kompaktere Systeme zu entwickeln, die eine einfachere Handhabung ermöglichen. Ein wichtiger Schritt war der Einsatz von Navigationssystemen zur Überwachung der Patientenposition. Hierdurch kann das traumatisierende und aufwendige Einspannen von Gliedmaßen entfallen.

Von 1996-1999 wurde in dem von der EU geförderten Projekt IGOS (Image Guided Orthopedic Surgery) ein optisches Navigationssystem mit einer Stereokamera für die Orthopädie entwickelt. Hieraus entstand das Orthopilot System, das 1999 eine Zertifizierung für Europa und 2001 eine Zulassung für den amerikanischen Markt erhielt. Das System dient zur Unterstützung bei Knie und Hüftoperationen ohne den Einsatz von radiologischer Bildgebung. Hierfür wurden am Patienten Sensoren angebracht, mit denen die Bewegungen der Gliedmaßen am Bildschirm überwacht wurden. Durch die Beinbewegung wurde das Zentrum des Hüftgelenkes bestimmt. Bei Knie- und Hüftprothesen konnte der Orthopilot die Beinposition über die angebrachten Sensoren überwachen und so anzeigen, in welcher Position das Bein seine korrekte Lage erreicht [58]. Seit der Einführung wurde das System stetig weiterentwickelt und wird heute erfolgreich von dem Unternehmen Aesculap (Tuttlingen, Germany) vertrieben.

Ein großer Durchbruch in dem Einsatz von robotischen Assistenzsystemen gelang mit dem von Intuitive Surgical Inc. (Sunnyvale, CA, USA) entwickelten Da Vinci Surgical System. Das System wird seit 1995 entwickelt und erhielt erstmals im Jahr 2000 eine Zulassung für den Einsatz bei laparoskopischen Eingriffen. Das System ist als Telemanipulator aufgebaut, sodass der Chirurg nicht direkt am Patienten stehen muss, sondern das System über eine Bedienkonsole steuert, die auch weiter entfernt positioniert werden kann. Das in der Abbildung 2.2 gezeigte System setzt sich aus 3 Komponenten zusammen. Der „Patient Cart“ besteht aus 3 bis 4 robotischen Armen, von denen einer eine endoskopische Kamera für die Sicht auf das Operationsgebiet führt. Die beiden weiteren Arme dienen als verlängerte Arme des Chirurgen und können mit verschiedenen Werkzeugen zum Greifen, Schneiden und auch zum Vernähen ausgestattet werden. Die zweite Komponente des Systems enthält das Vision System. Neben den Recheneinheiten für das System ist hier auch ein Bildschirm angebracht, mit dem die Operation verfolgt werden kann. Die dritte Komponente ist

die chirurgische Bedienkonsole, die dem Chirurgen eine hochauflösende 3D Sicht auf das Operationsgebiet ermöglicht. Die Konsole dient als Fernsteuerungseinheit, mit der der Chirurg die robotischen Arme während des Eingriffes steuert, ohne direkt am Patienten zu stehen. Zu Beginn wurde das System für laparoskopische Eingriffe,

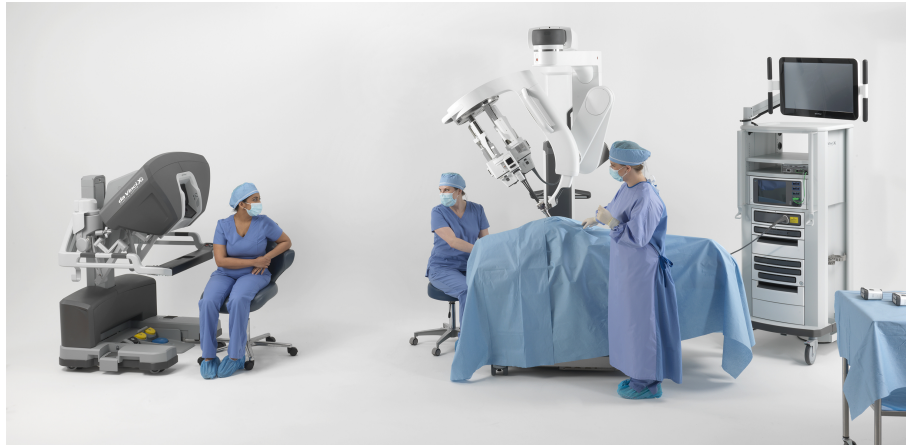


Abbildung 2.2: Das DaVinci Surgical System, (c) [2022] Intuitive Surgical Operations, Inc. — reproduced by courtesy of the manufacturer

wie der Entfernung der Gallenblase (Cholezystektomie), eingesetzt. Weiter folgten Eingriffe in der Lunge sowie der Speiseröhre und Bypass Operationen. Seit 2001 wurde das System zur laparoskopischen Entfernung der Prostata zertifiziert. Seit der Markteinführung wurde das System stetig weiterentwickelt und wird nun in der vierten Produktgeneration vermarktet. Aktuell sind mehr als 5000 Systeme weltweit installiert [51]. Das System wird zur Behandlung von Krebserkrankungen an der Prostata, bei Eingriffen an den Nieren so wie zur Reparatur von Blasenabnormalitäten eingesetzt [6][83].

Eine gängige Praxis zur Behandlung von Tumoren ist die Strahlentherapie, bei der Tumorzellen durch eine hohe Strahlendosis zerstört werden. Mit dem von der Firma Accuray Inc. (Sunnyvale, CA, USA) entwickelten Cyberknife können verschiedene Tumore im gesamten Körper behandelt werden. Seit 2005 wird das System in Deutschland zur Therapie eingesetzt. Die Bestrahlung erfolgt über einen Linearbeschleuniger der an einem Roboterarm angebracht ist. Hierdurch kann die Bestrahlung aus unterschiedlichen Richtungen erfolgen, wodurch sich auch komplexe Tumorgeometrien behandeln lassen, während die Strahlung im umliegenden Gewebe möglichst gering gehalten wird. Die Planung erfolgt anhand von 3D oder zeitaufgelösten 4D Aufnahmen. Die Positionierung erfolgt mit der Unterstützung von 2 orthogonal angeordneten Röntgenröhren, die die aktuelle Position des Patienten mit den Planungsdaten abgleichen bevor die Bestrahlung erfolgt.



Abbildung 2.3: Das Cyberknife System, bestehend aus dem Linearbeschleuniger am Roboterarm, zwei Röntgenröhren zur Positionsbestimmung, einer Stereokamera für die Bewegungsnachverfolgung, sowie der robotischen Behandlungsliege. Quelle: [1], Image courtesy of Accuray Incorporated - (c) [2022] Accuray Incorporated. All Rights Reserved.

Aktuell wird die in der Abbildung 2.3 gezeigte siebte Systemgeneration vermarktet. Durch die Integration eines Synchronisationssystems, das aus einer Stereokamera besteht, werden Patientenbewegungen, die zum Beispiel durch die Atmung entstehen, nachverfolgt. Durch eine Synchronisation der Bestrahlung mit der Patientenbewegung lassen sich auch Lungentumore behandeln die durch Ein- und Ausatmung in Bewegung sind [99].

Eine umfangreiche Übersicht der Entwicklung chirurgischer Robotersysteme ist in den Quellen [85], [87] und der Webseite von [allaboutroboticsurgery](http://allaboutroboticsurgery.com) [6] zu finden.

3 Betrachtung von zwei Anwendungsgebieten

Den medizinischen Hintergrund für diese Arbeit bieten zwei verschiedene Anwendungsszenarien. Als Anwendungsszenario in der knöchernen Chirurgie wird der chirurgische Eingriff zur Wirbelsäulenversteifung betrachtet. Der Schwerpunkt wird hierbei auf das Setzen von Pedikelschrauben gelegt. Die anatomische Struktur der Wirbelkörper lässt sich gut in den Bilddaten erkennen und kann zur Operationsplanung verwendet werden. Hierdurch kann auf intraoperative und präoperative Bilddaten zur Operationsplanung und -durchführung zurückgegriffen werden. Eine Grundlage für die Anwendung der computer- und roboterassistierten Chirurgie ist die Verknüpfung der Patientenlage mit dem verwendeten Assistenzsystem.

Durch den Einsatz von intraoperativen Lokalisiersystemen ist es möglich, die verwendeten Komponenten zu lokalisieren. Hierzu werden an den eingesetzten Robotersystemen und chirurgischen Instrumenten, als auch an den Knochen des Operationsgebietes, Referenzkörper (Dynamic Reference Base - DRB) angebracht. Ein Beispiel hierfür zeigt die Abbildung 3.1. Damit ist es möglich Robotersysteme intraoperativ zu lokalisieren und über DRB mit der Patientenlage zu verknüpfen. Verschiedene Werkzeuge können so verwendet und nach erfolgter Registrierung in den Planungsdaten eingeblendet werden. Dies erschließt weitere Anwendungsgebiete und ermöglicht eine vielseitige Anwendbarkeit der CAS. Mit der Weiterentwicklung der computergestützten Assistenzsysteme wurden auch neu entwickelte Robotersysteme kompakter und anwenderfreundlicher.

Als weiteres medizinisches Anwendungsszenario werden chirurgische Eingriffe in der Weichteilchirurgie betrachtet. Mit neuen Technologien und kompakteren Systemen ist es möglich, chirurgische Eingriffe in der Weichteilchirurgie computerassistiert zu unterstützen. Im Gegensatz zur knöchernen Chirurgie ist das Weichteilgewebe oft empfindlich und von sensiblen Regionen umgeben. Auch die präoperative Bildgebung ist bei der Operationsplanung und -durchführung nur bedingt hilfreich, da die Strukturen nicht rigide sind und sich während der Operation verschieben können, sodass eine klassische Navigation durch anbringen von DRB nicht möglich ist. Durch das Einbringen von Operationswerkzeugen wird das Gewebe verschoben, sodass es nötig ist, das Operationsgebiet intraoperativ zu überwachen. Üblicherweise erfolgt



Abbildung 3.1: Mit einem Referenzkörper ausgestattetes Zeigelinstrument und ein am Knochen befestigter Referenzkörper für eine intraoperative Navigation [101]

daher die intraoperative Navigation und Operationsüberwachung unter Verwendung von Endoskopkameras, durch die das Operationsgebiet in einem Livebild überwacht werden.

Für dieses Anwendungsszenario erschließt der Einsatz des flexiblen Roboterarms, der im Rahmen des STIFF-FLOP Projektes entwickelt wurde, neue Operationsmöglichkeiten. Bei Eingriffen im Weichteilgewebe ergeben sich besondere Anforderungen um einen sicheren Einsatz des Roboters zu gewährleisten. Hierfür werden die endoskopischen Livebilder des Operationsgebietes ausgewertet um die Lage des STIFF-FLOP Armes im Operationsgebiet zu überwachen.

4 Komponenten der roboterassistierten Chirurgie

Zur Einführung in diese Arbeit werden in diesem Kapitel die in der Medizin eingesetzten Bildgebungsmodalitäten mit ihren relevanten Charakteristika beschrieben. Danach werden Systeme zur intraoperativen Navigation vorgestellt und ein Überblick über Systeme in der roboterassistierten Chirurgie gegeben.

4.1 Medizinische Bildgebung

Den Ausgangspunkt für die computergestützte Durchführung chirurgischer Eingriffe bilden medizinische Bilddaten. Hierbei werden zur medizinischen Diagnose je nach Anwendungsszenario 2 oder 3-dimensionale Bilddaten des Operationsgebietes aufgenommen. Anhand dieser Bilddaten kann ein Eingriff und mögliche Zugangspunkte geplant werden und sensible Bereiche wie Nervenbahnen, Blutgefäße sowie anderes empfindliches Gewebe, das nicht beschädigt werden darf, identifiziert werden.

Abhängig von dem Anwendungsszenario werden unterschiedliche Bildgebungsmodalitäten eingesetzt. Nachfolgend wird ein Überblick über Geräte gegeben, die für diese Arbeit relevant sind.

Computertomograph

Bei der Computertomographie (CT) werden in der Regel 3-dimensionale Bildaufnahmen des Patienten angefertigt. Das Prinzip beruht auf der Röntgenstrahlabschwächung. Hierbei rotiert ein Röntgenprojektionssystem um den Patienten, während viele Projektionsaufnahmen erfasst werden. Aus der gefilterten Rückprojektion (Radontransformation) wird die Abschwächung in den einzelnen Bildbereichen, den sogenannten Bildvoxeln bestimmt [13]. Die Intensität der Abschwächung wird anschließend für jeden einzelnen Bildpunkt dargestellt.

Die Voxel definieren die Auflösung der Aufnahme und haben eine quaderförmige Form. Sofern nicht anders angegeben, beschreibt eine Schichtaufnahme eine axiale Ebene der 3-dimensionalen Aufnahmen. Die axiale Ebene liegt orthogonal zur Patientenachse, welche von den Füßen zum Kopf zeigt, was in der Abbildung 4.3

dargestellt ist. Innerhalb einer Schichtaufnahme beschreibt der Pixelabstand die axiale Auflösung. Hier bieten aktuelle Geräte eine Auflösung von bis zu $0,2\text{ mm}$. Der Abstand zwischen den Schichtebenen definiert die Länge eines Bildvoxels. Mit modernen Geräten werden bis zu 256 Schichtaufnahmen gleichzeitig erfasst. Um das gesamte Operationsgebiet abzubilden, wird der Patient schrittweise auf der Patientenliege durch das Gerät gefahren, während Schichtaufnahmen in einem Abstand von minimal $0,2\text{ mm}$ aufgenommen werden.

Da jede Aufnahme auch eine Strahlenbelastung für den Patienten bedeutet, wird möglichst nur das relevante Gebiet durchleuchtet. Auch die Auflösung und der Schichtabstand wird so gewählt, dass die untersuchten Strukturen ausreichend genau dargestellt werden während die Strahlenbelastung für den Patienten so gering wie möglich bleibt.

CT-Aufnahmen eignen sich besonders zur Darstellung von knöchernen Strukturen. Durch eine Zugabe von Kontrastmitteln können auch Stoffwechselprozesse im Körper erfasst werden. Der Kontrast von Weichteilgewebe hingegen ist sehr gering, weshalb CT-Aufnahmen dafür nur bedingt geeignet sind.

Die in dieser Arbeit verwendeten CT Aufnahmen von knöchernen Strukturen werden mit einer Auflösung von 512×512 Pixeln und einem Schichtabstand von 5 mm aufgenommen.

Magnetresonanztomographie

Die Magnetresonanztomographie (MRT) basiert auf der Messung von Kernspin-Effekten der Moleküle im Gewebe. Hierfür wird der Patient einem starken, homogenen elektromagnetischen Feld ausgesetzt, sodass sich die Atomkerne aufgrund von Kernspin-Effekten entlang der Feldlinien ausrichten. Durch zusätzliche Magnetspulen werden kurze Anregungsimpulse ausgesendet, die die Ausrichtung der Atomkerne anregen. Nach dem Impuls richten sich die Kerne wieder entlang der Feldlinien aus. Diese Relaxation des angeregten Areals wird gemessen. Die ermittelten Daten werden aufbereitet und in Volumendaten dargestellt. Hierin sind Rückschlüsse auf die Dichte, die Menge und die Art des Gewebes möglich.

Der Einsatz der MRT erfordert besondere Voraussetzungen für die Erfassung von Aufnahmen. So darf insbesondere das homogene Magnetfeld nicht durch magnetische Metalle wie Implantate oder auch konventionelle chirurgische Werkzeuge gestört werden. Durch unterschiedliche Spulen und Anregungssequenzen können vielfältig

gewichtete Bilder erstellt werden, die zum Beispiel besonders gut anatomische Areale und Gefäße abbilden können. Durch anders gewichtete Bilder lassen sich besonders gut Flüssigkeiten wie Liquor oder Ödeme darstellen. Häufig wird die MRT zur Diagnostik und Untersuchung von Weichteilgewebe eingesetzt. Hier ist der Kontrast besonders gut und verschiedene Areale lassen sich gezielt untersuchen. Der Kontrast für knöcherne Strukturen fällt allerdings gering aus, weshalb hierfür in der Regel auf CT-Aufnahmen zurückgegriffen wird. Ein besonderer Vorteil der MRT-Bildgebung ist, dass keine radioaktive Strahlung bei der Aufnahme entsteht.

Aufgrund der hohen technischen Anforderungen sind MRT-Geräte in der Regel in eigenen, abgeschirmten Räumen aufgestellt. Es gibt aber auch Geräte, die für einen intraoperativen Einsatz bei chirurgischen Eingriffen ausgelegt sind. Hierfür müssen die im Operationsraum verwendeten Geräte und Instrumente MRT-kompatibel sein, sodass die Magnetfelder des MRT nicht beeinflusst werden.

C-Bogen

Insbesondere bei minimalinvasiven Eingriffen ist eine intraoperative Bildgebung nötig, um Instrumente und Implantate korrekt zu positionieren oder um anatomische Strukturen zu lokalisieren. Hierfür ist der Einsatz von intraoperativen Röntengeräten die gängige Praxis. Häufig eingesetzt werden 3D C-Bögen, die mobil am Patiententisch positioniert werden können, um so beliebig orientierte Projektionsaufnahmen oder auch, wie bei der CT, 3D-Aufnahmen des Operationsgebietes aufzunehmen. Dafür kann der C-Arm um den Patiententisch geschwenkt und rotiert werden.

Durch den kompakten Aufbau kann allerdings Streustrahlung nicht so präzise unterdrückt werden, wie bei fest installierten CT Geräten, wodurch es zu Einbußen bei der Bildqualität kommt. Ausserdem kann, bedingt durch den Aufbau, keine vollständige Rotation um den Patienten ausgeführt werden. Hierdurch entstehen Bildartefakte und einige Bereiche des Aufnahmevolumens können nicht vollständig abgebildet werden. Auch das Aufnahmevolumen ist begrenzt. Da es lediglich eine Detektoreinheit gibt, sind 3D Aufnahmen zudem zeitintensiv da einzelne Projektionsaufnahmen aus verschiedenen Winkelstellungen angefertigt und anschließend zu einem 3D Volumen transformiert werden müssen. Typisch sind hier Aufnahmezeiten von 30 – 90 Sekunden. Die Abbildung 4.1 zeigt den im Rahmen dieser Arbeit eingesetzten 3D C-Bogen (Ziehm, Vario 3D).

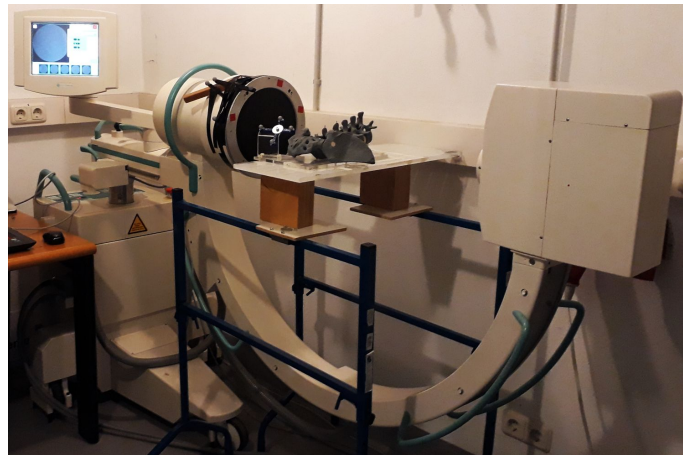


Abbildung 4.1: 3D C-Bogen der Firma Ziehm - Vario 3D

Neben der eingeschränkten Bildqualität kommt es während der Aufnahme, durch den Einsatz von Röntgenstrahlung, zu einer Strahlenbelastung für den Patienten und das gesamte Personal im Operationsraum.

Neben dem abgebildeten C-Bogen gibt es für die intraoperative Bildgebung auch neuere Geräte mit Flachdetektoren, die eine höhere Bildqualität erzielen. Weitere Modelle sind der O-Arm (Medtronic), der vollwertige 3D Aufnahmen liefern kann, oder im Operationsraum fest installierte, robotisch positionierte Röntgensysteme (z.B. Siemens Artis pheno).

Optische Verfahren (Endoskopie)

In der minimalinvasiven Weichteilchirurgie werden meist optische Verfahren zur Bildgebung eingesetzt. In der Regel kommen hier endoskopische Kamerasysteme zum Einsatz, da diese dem Chirurgen eine Sicht auf das Operationsgebiet über minimalinvasive Zugänge ermöglichen. Neben der Navigation im Weichteilgewebe können so sensible Regionen erkannt und Zielgebiete identifiziert werden. Zur Ausleuchtung des Operationsgebietes verfügen endoskopische Kameras über eine Lichtquelle. Abhängig vom Anwendungsgebiet werden flexible oder starre Endoskopkameras eingesetzt. Starre Laparoskope bieten den Vorteil, dass Sie von mechanischen oder robotischen Haltearmen positioniert werden können.

Die Abbildung 4.2 zeigt das im Rahmen dieser Arbeit eingesetzte endoskopische Kamerasystem (Endocam HD 5509, Richard Wolf) mit der zugehörigen Xenon-Lichtquelle.

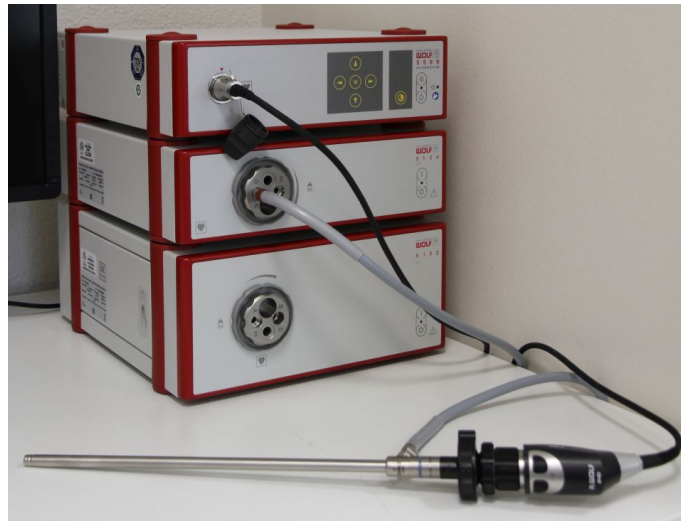


Abbildung 4.2: Endoskopkamera Endocam HD 5509 (Richard Wolf) mit zugehöriger Xenon-Lichtquelle

Verarbeitung medizinischer Bilddaten

Zur korrekten Verarbeitung und der anschließenden Visualisierung medizinischer Bilddaten müssen die Parameter der Aufnahme bekannt sein. Hierfür wurde der DICOM-Standard (Digital Imaging and Communication in Medicine) eingeführt, in dem Bilddaten medizinischer Bildgebungsgeräte wie CT oder MRT mit umfangreichen weiteren Parametern, wie Patientendaten und der Lage des Patienten bei der Aufnahme, gespeichert werden [70]. Für 3D Aufnahmen beinhaltet dies unter anderem die Anzahl und Größe der Bildvoxel einer Schichtaufnahme, sowie den Schichtabstand und die Anzahl an aufgenommenen Schichten. Neben dem mit dem Bildvolumen festgelegten Koordinatensystem, muss zur korrekten Visualisierung auch die Lage des Patienten mit dem zugehörigen Patienten Koordinatensystem berücksichtigt werden. Die Abbildung 4.3 zeigt die im medizinischen Umfeld verwendeten Schichtansichten des Patienten Koordinatensystems.

Abhängig von der Lage des Patienten bei der Aufnahme kann es nötig sein die Bilddaten zu transformieren um diese in der verwendeten Ausrichtung darzustellen. Zur Transformation von Bilddaten eignen sich affine Transformationen. Hierfür werden homogene Koordinaten verwendet. Die Formel 4.1 zeigt die affine Transformation eines Punktes $(v_x, v_y, v_z, 1)$ der um die Komponente 1 erweitert wurde.

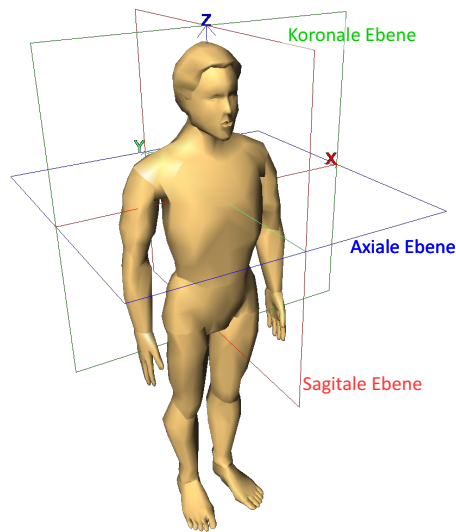


Abbildung 4.3: Patientenkoordinatensystem mit der axialen Ebene (blau), der koronalen Ebene (grün) und der sagittalen Ebene (rot)

$$\begin{pmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{pmatrix} = \begin{pmatrix} & & t_x \\ \underline{M} & & t_y \\ & & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix} \quad (4.1)$$

Die affine Transformation wird durch eine 4×4 Matrix beschrieben. Die Komponenten t_x, t_y, t_z erlauben eine Translation entlang der X-, Y-, und Z-Achse. In der 3×3 großen Matrix \underline{M} können Rotation oder auch Skalierungen beschrieben werden. Als Ergebnis ergibt sich der Punkt $(v'_x, v'_y, v'_z, 1)$, dessen letzte Komponente 1 nicht weiter berücksichtigt wird. Durch den Einsatz von homogenen Koordinaten können affine Transformationen durch Matrizenmultiplikation nacheinander ausgeführt werden.

4.2 3D-Lokalisierverfahren zur intraoperativen Navigation

Um dem Chirurgen eine intraoperative Navigation zu ermöglichen, müssen die Position und die Orientierung der verwendeten Instrumente, sowie die Lage des Patienten überwacht werden. Am weitesten verbreitet ist der Einsatz optischer 3D-Lokalisiersysteme. Solche Systeme bieten den Vorteil einer berührungslosen Messung mit einer hohen Genauigkeit im Submillimeterbereich. Nachteil dieser Systeme ist die aufwendige Einrichtung sowie die benötigte direkte Sichtverbindung zu den

überwachten Komponenten. Bei einer Sichtlinienunterbrechung, zum Beispiel durch die operierende Hand eines Chirurgen, stehen keine Messwerte zur Verfügung, was als „Line of sight“- Problem bezeichnet wird. Im Rahmen dieser Arbeit wurden verschiedene Systeme mit einer Zulassung für den Einsatz im Operationsraum eingesetzt. Beispiele hierfür sind die Polaris Systeme der Firma NDI [73], das Cambar System der Firma Axios [7], sowie die fusionTrack Serie der Firma Atracsys [5].

Diese Systeme bestehen aus einem Stereokameramodul für den infraroten Wellenlängenbereich. Detektiert werden kugelförmige Marker, von denen aktive, also selbst strahlende oder passive, sogenannte retroreflektierende Markerkugeln verfügbar sind. In dieser Arbeit wurden ausschließlich retroreflektierende Marker verwendet, da diese kompakter und somit leichter zu platzieren sind. Zur besseren Detektion werden die Markerkugeln mit infraroten Lichtblitzen von dem Stereokameramodul beleuchtet.

Die Abbildung 4.4 veranschaulicht die Detektion von Markerkugeln durch ein optisches Lokalisiersystem.

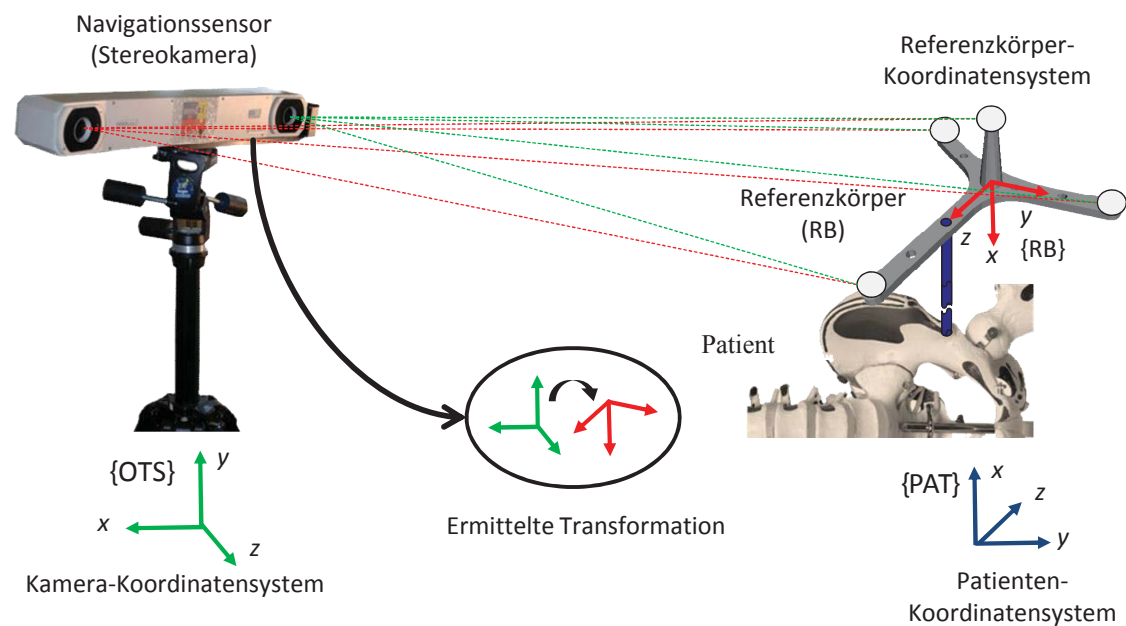


Abbildung 4.4: Funktionsprinzip zur Detektion eines Referenzkörpers mit 4 retroreflektierenden Markerkugeln mit einer Stereokamera [53]

Die Markerkugeln werden von beiden Kameras erfasst und im Kamerabild segmentiert. Anschließend müssen die korrespondierenden Markerpunkte in beiden Kamerabildern gefunden werden, was als Korrespondenzproblem bezeichnet wird. Das Problem kann durch den Einsatz der Epipolargeometrie vereinfacht und schneller

gelöst werden. Eine umfangreiche Erläuterung ist in der Quelle [46] beschrieben. Durch Stereotriangulation kann anschließend, unter Kenntnis der Disparität, die Position von korrespondierenden Markern aus den Bildpunkten der beiden Kameraaufnahmen bestimmt werden. Umfangreiche Erklärungen zur Funktionsweise von Stereokameras, sowie zur Entwicklung von Stereokamerasystemen sind in vorhergehenden Arbeiten [53], [64] und der Quelle [47] zu finden.

Die Detektion einer Markerkugel liefert lediglich deren Positionsinformation. Um die Position und Orientierung, im weiteren als Pose bezeichnet, zu erfassen, werden Referenzkörper (auch als Lokatoren oder Rigid body bezeichnet) verwendet. Ein Referenzkörper besteht aus einer festen Anordnung von mehreren Markerkugeln. Für eine präzise und zuverlässige Detektion wird der Einsatz von mindestens 4 Markerkugeln empfohlen. Bei der Erstellung der Referenzkörper ist die Beachtung von Designvorgaben ausschlaggebend für die erzielbare Genauigkeit [90]. Für den Einsatz in der Chirurgie sind möglichst kompakte Referenzkörper gewünscht, da der Platz sehr begrenzt ist. Für die Detektion sind größere Marker Abstände von Vorteil um die Präzision zu erhöhen. Genauere Vorgaben sind in den Designrichtlinien von NDI [75], sowie in den Arbeiten zur experimentellen Bestimmung von Referenzkörpergeometrien [90] und den entstehenden Fehlern [33], zu finden.

Damit die Position und Orientierung eines Referenzkörpers detektiert werden kann, muss die Anordnung der Markerpunkte relativ zu einem Referenzkoordinatensystem des Referenzkörpers definiert werden. Hierfür bietet der Hersteller NDI ein eigenständiges Tool „6D-Architect“ zum Einlernen neuer Referenzkörper [74]. Ebenso bietet Axios in seinem Softwarepaket zur Ansteuerung der Kamera ein Tool zum Einlernen neuer Referenzkörper an. Beim Einlernen wird ein Referenzkörper-Definitionsdocument (ROM-File) erzeugt, welches zur Detektion der Pose des Referenzkörpers verwendet wird.

Das Kamerasystem Cambar B2 der Firma Axios, als auch die Polaris Systeme der Firma NDI übernehmen die Schritte zur Detektion und Positionsbestimmung von Referenzkörpern auf einer eigenen Recheneinheit. Hierfür muss das Referenzkörper-Definitionsdocument an das Kamerasystem übertragen werden. Das Kamerasystem detektiert die Markerpunkte und ermittelt die Pose des geladenen Referenzkörpers im Koordinatensystem des Kamerasystems. Die detektierte Pose wird als Transformation des Referenzkörperkoordinatensystems im Koordinatensystem des Kamerasystems übertragen.

Weiterführende Arbeiten und Genauigkeitsuntersuchungen der Systeme sind in den Arbeiten [53] und [90] zu finden. Zum Monitoring der Patientenposition können Referenzkörper rigide am Patienten angebracht werden, was in der Abbildung 4.4 veranschaulicht ist. Abhängig von dem Einsatzszenario bieten sich unterschiedliche Optionen an. Für das Anwendungsgebiet der knöchernen Chirurgie bietet sich eine Anbringung an der relevanten knöchernen Struktur an. Hierfür sind angepasste Halteklammern von Herstellern chirurgischer Instrumente verfügbar, an denen ein Referenzkörper angebracht werden kann. An größeren Strukturen, wie dem Beckenboden, können auch Halteelemente durch eine Verschraubung am Knochen befestigt werden. Nach dem Anbringen des Referenzkörpers kann das Patientenkoordinatensystem mit dem Koordinatensystem des Referenzkörpers verknüpft werden und Bewegungen erfasst werden.

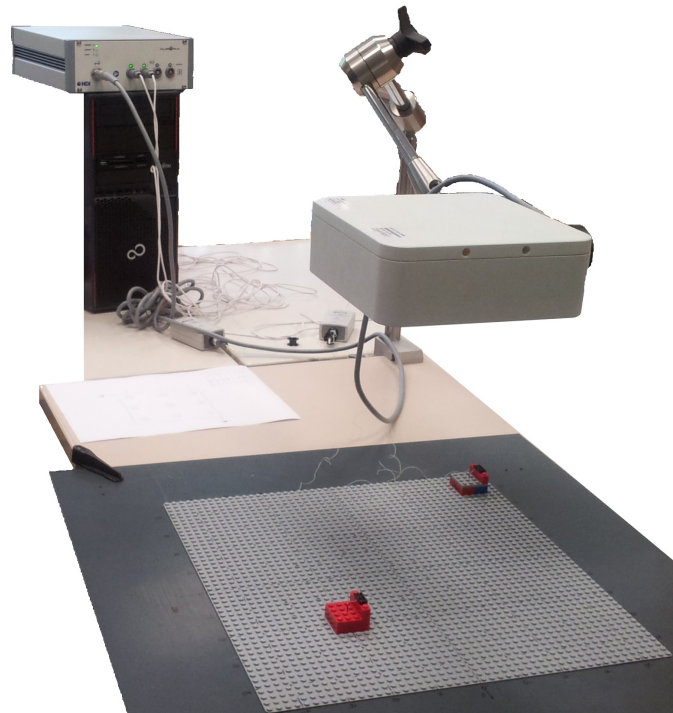


Abbildung 4.5: Das elektromagnetische Tracking System NDI Aurora bei Messungen zur Untersuchung von Störeinflüssen

Neben optischen Systemen zur intraoperativen Lokalisierung sind auch magnetische Trackingsysteme verfügbar. Diese bestehen aus einem Feldgenerator, der ein elektromagnetisches Feld generiert und elektromagnetischen Empfangsspulen. Durch Messungen in den Empfangsspulen kann deren Position innerhalb des elektromagnetischen Feldes bestimmt werden. Ein kommerzielles System ist das Aurora System der Firma NDI. Hierfür sind besonders kompakte Empfangsspulen mit einem Durchmesser von $< 1 \text{ mm}$ verfügbar. Die erzielbare Positionsgenauigkeit wird von dem

Hersteller mit $< 0,48 \text{ mm}$ RMS angegeben [72]. Die Abbildung 4.5 zeigt das elektromagnetische Tracking System NDI Aurora bei der Untersuchung von äusseren Störeinflüssen.

Der größte Vorteil dieser Systeme liegt in den kompakten Sensoren, die nicht auf eine optische Verbindung zum Tracking System angewiesen sind („Line-of-sight“ Problem). Allerdings stellt der Einsatz solcher Systeme besondere Herausforderungen an die Operationsumgebung, da metallische Gegenstände die Messergebnisse beeinflussen, was auch eigene Untersuchungen bestätigten [66] [65].

4.3 Systeme in der roboterassistierten Chirurgie

In der computer- und roboterassistierten Chirurgie wird stetig an der Entwicklung neuer Systeme gearbeitet, um Eingriffe zuverlässiger, sicherer und mit einem besseren Ergebnis für den Patienten durchführen zu können. Zur Einordnung dieser Arbeit werden in diesem Abschnitt bereits verfügbare chirurgische Assistenzsysteme für die knöcherne Chirurgie, mit dem Schwerpunkt auf dem Anwendungsgebiet der Wirbelsäulenchirurgie, vorgestellt. Für die Weichteilchirurgie wird ein Überblick über neuartige Ansätze zur Durchführung von Operationen beschrieben.

4.3.1 Assistenzsysteme für die knöcherne Chirurgie

Eingriffe an der Wirbelsäule sind laut dem statistischen Bundesamt zweimal unter den 10 häufigsten chirurgischen Eingriffen gelistet [94, Stand 2018]. Bei minimalinvasiven Eingriffen ist der Chirurg auf den Einsatz von Systemen zur intraoperativen Navigation angewiesen, um im Operationsgebiet sicher und zielgenau zu operieren. Das bedeutet der Chirurg muss die Lage und Position seiner Instrumente während der Operation jederzeit kennen und das Operationsgebiet überwachen. Neben der klassischen Bildgebung können Navigationssysteme die Lage von chirurgischen Instrumenten im Bilddatensatz visualisieren. Mit der zunehmenden Nachfrage nach intraoperativen Navigationslösungen gibt es einige Hersteller, die die zuvor im Abschnitt 4.2 vorgestellten Systeme zu intraoperativen Navigationslösungen weiterentwickelt haben und damit bereits Lösungen für gezielte Anwendungen anbieten [22].

Ein etabliertes System ist das von der Brainlab AG (München, Deutschland) entwickelte Navigationssystem für Wirbelsäulenoperationen, dass in der Abbildung 4.6



Abbildung 4.6: Das „Curve Navigation“ System der Firma Brainlab, bestehend aus einer Stereokamera und einem Monitor zur Visualisierung von Instrumenten. copyright: Brainlab AG, Quelle: [21]

dargestellt ist. Dieses besteht aus einer Stereokamera und einem Bildschirm zur visuellen Darstellung. Das System wurde für verschiedene Anwendungsgebiete entwickelt und unterstützt auch eine Navigation bei Eingriffen an der Wirbelsäule [22]. Mit einer zugehörigen Software kann der Eingriff geplant werden und Instrumente und Schrauben im Bilddatensatz visualisiert werden. Zusätzlich vermarktet Brainlab auch robotische Bildgebungsgeräte (Loop-X, Brainlab AG) zur Unterstützung von Operationen. Auf Messen wurde bereits eine robotische Positionierunterstützung vorgestellt [20]. Diese besteht aus einem passiven Arm (Cirq, Brainlab AG) der als Führung zur Implantation von Pedikelschrauben ausgerichtet werden kann. Hiermit kann der Chirurg, zusätzlich zur Navigationsunterstützung, auch mit dem positionierten Arm unterstützt werden, um Instrumente zielgenau auszurichten.

Speziell für die Wirbelsäulenchirurgie entwickelte die Firma Mazor Robotics (Caesarea, Israel) das Renaissance System. Das in der Abbildung 4.7 dargestellte Produkt besteht aus einem kompakten Robotersystem, welches sich über Schrauben direkt am Patienten fixieren lässt.

Im Operationsablauf erfolgt im ersten Schritt eine Planung mit einer darauffolgenden Bildregistrierung. Hiernach wird der Hexapod Roboter mit Schrauben am Patienten fixiert. Anschließend wird eine Werkzeugführung angebracht, mit der ein Führungsdraht in den Wirbel eingebracht wird. Über diesen wird die Pedikelschraube eingedreht. Für die Platzierung der nächsten Schraube muss der Hexapod Roboter umgebaut werden.

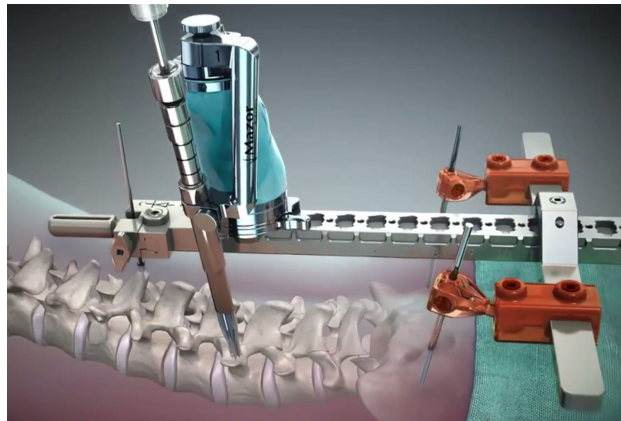


Abbildung 4.7: Das Renaissance System auf der Wirbelsäule mit einer Führung zur Unterstützung der Platzierung von Pedikelschrauben. Quelle [63] copyright: Mazor Robotics

Ein weiteres Produkt ist das ROSA surgical system, dass ursprünglich von der Firma Medtech (Montpellier, Frankreich) für die Hirnchirurgie entworfen wurde und seit 2012 entwickelt wird. Dieses besteht aus einem Robotersystem und einer Stereokamera, mit der die Patientenposition und das Robotersystem überwacht wird. Auch hier erfolgt die Implantierung der Pedikelschrauben über Führungsdrähte. Diese können mit dem Robotersystem positioniert werden. Das in der Abbildung 4.8 vorgestellte System befand sich zunächst noch in der Entwicklungsphase und erhielt als ROSA One System 2016 eine Zertifizierung.



Abbildung 4.8: Das ROSA Spine System der Firma Zimmer Biomet, vorgestellt auf dem Deutschen Wirbelsäulenkongress 2016

Während der Bearbeitungszeit dieser Arbeit haben auch weitere Hersteller Systeme für die Wirbelsäulenchirurgie vorgestellt. Die Tatsache, dass große Hersteller hohe Summen für die Entwicklung investieren, bestätigt die Relevanz der Entwicklung von robotischen Systemen für die Wirbelsäulenchirurgie. Nachfolgend aufgeführte Systeme waren zu Beginn der eigenen Arbeiten noch nicht bekannt und wurden während der Bearbeitungszeit vorgestellt.

Zur Ablösung des Renaissance Systems hat Mazor robotics das neue Mazor X System vorgestellt. Im Dezember 2018 wurde Mazor robotics für 1,7 Mrd \$ von dem Hersteller Medtronic übernommen. Medtronic vertreibt das System als Mazor X stealth Edition, zusammen mit der eigenen Wirbelsäulenimplantat-, Navigations- und 3D-Bildgebungstechnologie. Das robotische System beinhaltet neben dem Operationsroboter auch ein integriertes Navigationssystem. Damit kann die Implantation von Schrauben unterstützt werden.

Ein weiteres System ist die von Globus medical Inc. (Audubon, USA) vorgestellte ExcelsiusGPS Robotic Navigation Plattform. Der technische Aufbau ist vergleichbar mit dem ROSA System und soll eine Echtzeitplanung und Implantation von Pedikelschrauben ermöglichen. Die Abbildung 4.9 gibt einen Einblick auf das ExcelsiusGPS System, das im Dezember 2016 auf dem Deutschen Wirbelsäulenkongress (DWG) vorgestellt wurde.



Abbildung 4.9: Das ExcelsiusGPS System von Globus Medical, welches auf dem Deutschen Wirbelsäulen Kongress 2016 vorgestellt wurde

4.3.2 Flexible Robotersysteme für die Weichteilchirurgie

Verschiedene Hersteller und Forschungsprojekte befassen sich mit der Entwicklung von neuen flexiblen Operationsrobotern, die einen sicheren Einsatz im Weichteilgewebe ermöglichen. Einen neuartigen Ansatz hierfür bietet der in dieser Arbeit eingesetzte flexible Roboterarm mit der Möglichkeit, sich zu versteifen. Dieser wurde im Rahmen des STIFF-FLOP Projektes entwickelt und wird im Kapitel 7.1 vorgestellt. In diesem Abschnitt werden weitere Systeme vorgestellt, die Gegenstand der aktuellen Entwicklung sind.

Bekannt sind flexible Systeme aus perkutanen Koronarinterventionen, bei denen Stents zur Behandlung von verengten Arterien eingebracht werden, die die verengte Stelle weiten. Hierfür kann das von Corindus (Siemens Healthineers) entwickelte CorPath GRX System eingesetzt werden. Über eine Bedienkonsole wird ein robotisch gesteuerter Katheter durch die Arterie zur verengten Stelle geführt, an der Stents zielgenau über den Katheter platziert werden. Durch eine höhere Genauigkeit bei der robotischen Platzierung sollen die Ergebnisse für den Patienten verbessert werden [92]. Das System ist sehr flexibel und kann die Spitze beugen um den Stent exakt zu positionieren. Der Rest des Katheters ist flexibel und wird durch die Ader im Körper geleitet.

Einen eigenen Ansatz verfolgt die von der Vanderbilt University (Tennessee, USA) entwickelte Active Cannula [97]. Diese bestehen aus ineinander liegenden Röhrchen, die sich teleskopisch ausfahren lassen. Durch Rotation der unterschiedlich vorgebogenen Röhrchen können diese verschiedene Biegungen annehmen und so gesteuert werden, um Hindernissen auszuweichen. Die Abbildung 4.10 zeigt einen Prototyp der Active Cannula.

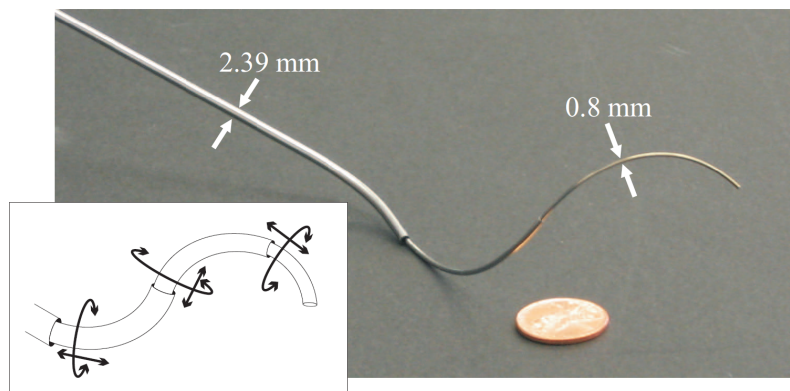


Abbildung 4.10: Prototyp der Active Cannula, aufgebaut aus ineinander liegenden vorgebogenen superelastischen Nitinol Röhrchen [97]

An der Carnegie Mellon University (Pittsburgh, Pennsylvania, USA) wird der Einsatz eines schlangenförmigen Roboters für die Herzchirurgie erforscht. Der entwickelte CardioARM ist 30 *cm* lang und hat einen Durchmesser von 12 *mm*. Er besteht aus 50 zylinderförmigen, starren Verbindungsstücken. Diese sind über Kugelgelenke mit 2 Freiheitsgraden verbunden und werden über Kabel von einer Basisstation gesteuert [80]. Kontrolliert wird die Spitze des Armes durch einen Joystick, wobei die nachfolgenden Elemente dem Pfad der gesteuerten Spitze folgen und die Biegung beibehalten. Somit kann der Einsatz am Herzen durch einen minimalinvasiven Zugang unterhalb des Brustkorbes erfolgen. Innerhalb des CardioArms befindet sich ein Kanal, durch den verschiedene Katheter oder Ablationswerkzeuge in das Zielgebiet eingeführt werden. So können auch Operationsgebiete die hinter dem Herzen liegen erreicht werden. Die Abbildung 4.11 zeigt den CardioArm.

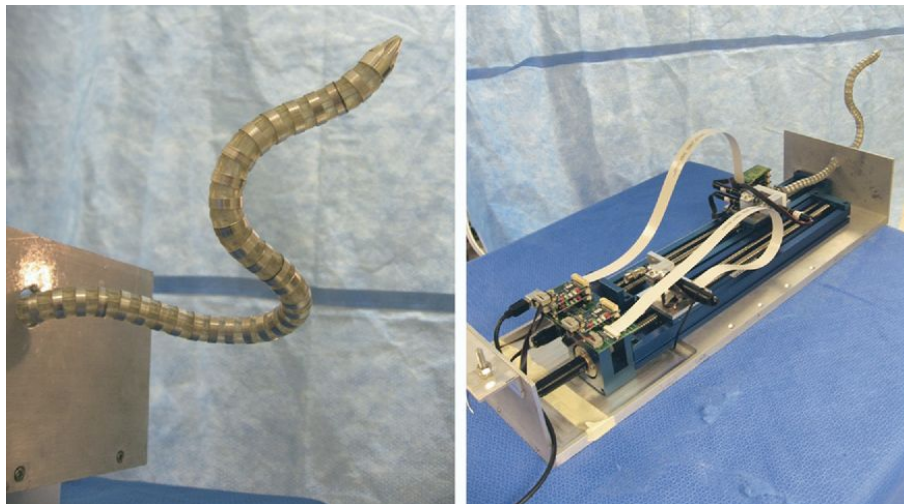


Abbildung 4.11: Der CardioARM, bestehend aus zylinderförmigen Verbindungsstücken mit Kugelgelenken (links) und die Basisstation mit Kabelzügen zur Steuerung des Armes [80]

Neben Kabelzug gesteuerten Robotern gibt es auch Ansätze, die Motoren zur Steuerung direkt in einen Roboterarm zu integrieren. Ein Beispiel hierfür ist das I-SNAKE System, das am Imperial College (London, GB) entwickelt wurde. Das System hat einen Durchmesser von 12,5 *mm* und besteht aus Segmenten die über Gelenke verbundenen sind. Die ersten beiden Gelenke bieten jeweils 2 Freiheitsgrade, während die letzten 3 Gelenke jeweils einen Freiheitsgrad besitzen. Durch die Montage an einen steifen Schaft, kann der Roboter in das Operationsgebiet eingebracht werden. Integriert sind 2 innere Kanäle, von denen einer eine Kamera und die elektrischen Leitungen beinhalten, während durch den zweiten Kanal chirurgische Werkzeuge eingebracht werden können. Die Herausforderungen liegen in der Kontrolle der ein-

zelenen Gelenke sowie der begrenzten Kraft, die durch die kompakten Motoren zur Verfügung steht [91].

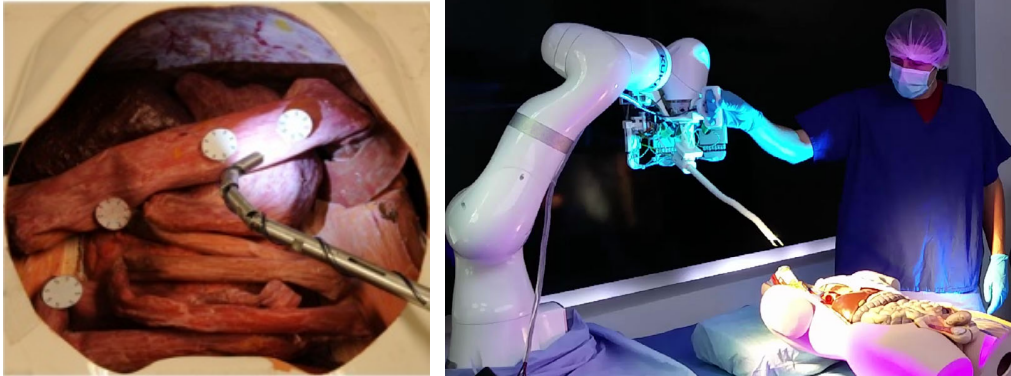


Abbildung 4.12: Das I-SNAKE System mit integrierten Motoren (links) Quelle:[91], sowie das weiterentwickelte I2-SNAKE System das über Kabelzüge aktiviert wird, montiert an einem Roboterarm mit 2 Greifwerkzeugen an der Spitze (rechts) Quelle:[11]

Zwischenzeitlich wurde das in der Abbildung 4.12 dargestellte System weiterentwickelt und wird in der I2-SNAKE Variante über Kabelzüge gesteuert. In der I2-SNAKE Variante beinhaltet der schlangenförmige Roboterarm 12 Gelenke, die von 24 Kabelzüge gesteuert werden. Der Arm mit einem Durchmesser von 16 *mm* beinhaltet 4 Kanäle, durch die eine Kamera und chirurgische Werkzeuge positioniert werden können. Das System wurde an einem KUKA Roboter angebracht, um es bei Operationen im minimalinvasiven Zugang am Patienten zu positionieren [11].

Eine umfangreiche Übersicht von Systemen und Forschungsprojekten ist in den Proceedings des ICRA Workshops über Snakes, Worms und Kathetern für die MIS zu finden [30].

5 Problemstellung und Ziel

Zur Erarbeitung einer Problemlösung muss der Operationsablauf, der sogenannte Workflow berücksichtigt werden. Hierfür wird zunächst in die medizinische Aufgabenstellung eingeführt, mit dem Fokus auf die für die Problemlösung relevanten Schritte.

Für das Aufgabengebiet in der knöchernen Chirurgie wird die medizinische Aufgabenstellung zur Wirbelsäulenversteifung mit dem Schwerpunkt auf der Planung und der Implantierung von Pedikelschrauben erläutert. In der technischen Umsetzung wird das Ziel der eigenen Beiträge zur Umsetzung des Eingriffes, unter Einsatz eines robotischen Assistenzsystems, beschrieben. Das Ziel liegt in der Planung der Pedikelschrauben in den Bilddaten der Bildgebung mit einer submillimeter genauen roboterassistierten intraoperativen Ausführung.

Im Anwendungsszenario der Weichteilchirurgie wird auf die Besonderheiten bei Eingriffen in Weichteilgewebe eingegangen und mögliche Einsatzszenarien für den im Rahmen des STIFF-FLOP Projektes entwickelten Manipulator vorgestellt. Aufgrund der flexiblen und beweglichen Strukturen muss eine Überwachung im Livebild erfolgen. Das Ziel der eigenen Beiträge zur Entwicklung eines optischen Lageüberwachungssystems für den STIFF-FLOP Manipulator wird in der technischen Umsetzung skizziert.

5.1 Medizinische Aufgabenstellung

Den medizinischen Hintergrund für die Arbeit bieten zwei verschiedene Anwendungsszenarien. Im Anwendungsbereich der knöchernen Chirurgie wird als Operationseingriff die Wirbelsäulenversteifung durch das Einbringen von Pedikelschrauben betrachtet.

In der Medizin wird der Eingriff zur Wirbelsäulenversteifung als Spondylodese bezeichnet. Die medizinische Indikation bilden vorliegende Wirbelsäulenverletzungen oder auch degenerative Veränderungen der Wirbelsäule die durch konservative Behandlungen nicht therapiert werden können. Klassische Beispiele sind Brüche des

Wirbelkörpers, Fehlstellungen, Instabilitäten im Wirbelkörper oder auch verschlissene Wirbelkörper. Die Abbildung 5.1 zeigt eine Wirbelfehlstellung vor dem chirurgischen Eingriff.

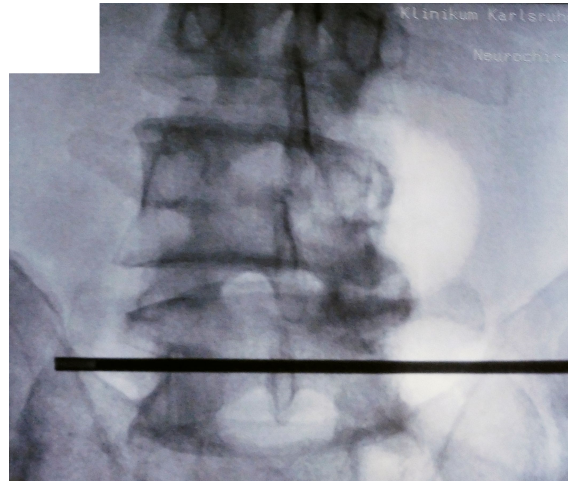


Abbildung 5.1: Präoperative Röntgenaufnahme einer gekrümmten Wirbelsäule. Der zweite Wirbel (Lendenwirbel 4) berührt den unteren Wirbel rechtsseitig

Bei der Spondylodese erfolgt die Stabilisierung der Wirbelsäule durch das Einbringen von Pedikelschrauben in die benachbarten Wirbel. Die polyaxialen Schraubenköpfe werden anschließend durch Metallstäbe verbunden und stabilisieren dadurch die Wirbelsäule. Die Abbildung 5.2 zeigt die an einem Wirbelmodell eingebrachten Pedikelschrauben, welche durch einen Metallstab verbunden sind.

Zur Stabilisierung der Wirbelsäule müssen Pedikelschrauben präzise in die Pedikel der Wirbelsäule eingebracht werden, da diese den Schrauben den nötigen halt geben. Des Weiteren verlaufen im und um den Wirbelkörper Hauptnervenbahnen, deren Verletzung bis zu einer Lähmung der unterhalb liegenden Extremitäten führen kann. Die Abbildung 5.3 zeigt den Aufbau des Wirbelkörperquerschnitts mit einer schematischen Darstellung der Schraubenplatzierung.

Das zielgenaue Einbringen von Pedikelschrauben stellt hohe Ansprüche an den operierenden Chirurgen. Da das Operationsgebiet, insbesondere der Pedikelkanal, nicht von dem Chirurgen eingesehen werden kann, ist dieser auf eine Unterstützung angewiesen. Zum präzisen Einbringen der Pedikelschrauben werden in der Regel spitze Hohladeln, sogenannte Jamshidi-Nadeln, verwendet um den Wirbelkörper zu punktieren. Anschließend werden lange Metalldrähte, auch Kirschner-Drähte genannt, durch die kanülierte Jamshidi Nadel in den Pedikel geschoben um die Trajektorie für die Pedikelschrauben festzulegen. Die Positionierung der Kirschner-Drähte wird



Abbildung 5.2: Wirbelmodell der Lendenwirbelsäule mit eingebrachten Pedikelschrauben die mit einem Metallstab verbunden sind

durch wiederholte Röntgenaufnahmen von der Patientenrückenseite in Richtung des Bauchs (AP-Aufnahme) sowie aus seitlicher Perspektive (Laterale Aufnahme) kontrolliert. Nachdem für jede Schraubenposition ein Draht erfolgreich positioniert ist, können kanülierte Schrauben über die Drähte eingedreht werden. In der Regel werden hierfür selbstschneidende Schrauben verwendet. Abhängig vom Anwendungsfall können vor dem Einbringen der Schraube die Löcher über kanülierte Hohlbohrer vorgebohrt werden.

Die für Patienten und den operierenden Chirurgen entstehende Strahlenbelastung ist abhängig von der Anzahl der Positionierungsversuche der Drähte und den damit verbundenen Kontrollaufnahmen. Während Kontrollaufnahmen werden der Draht oder die Jamshidi Nadel von dem Chirurgen festgehalten. Insbesondere für die ungeschützten Hände des Chirurgen entsteht dabei eine erhebliche Strahlenbelastung. Auch die langen Drähte bringen Gefahren mit sich, da diese zum einen das Operationsgebiet beeinträchtigen, zum anderen auch darauf geachtet werden muss, dass diese in ihrer Position bleiben. Beim Eindrehen der Schrauben muss beachtet wer-

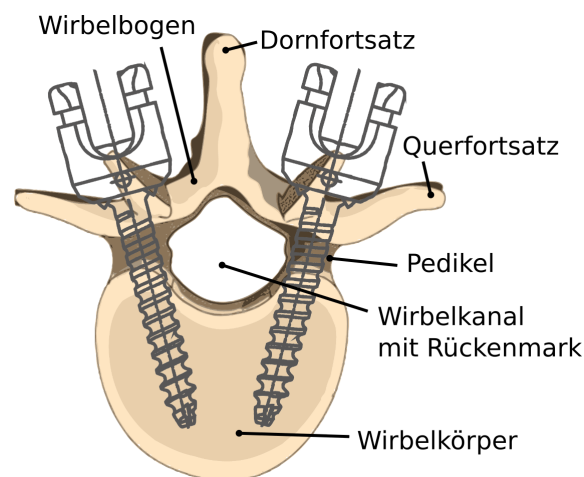


Abbildung 5.3: Aufbau des Wirbelkörpers mit schematischer Darstellung der Pedikelschraubenplatzierung

den, dass der Draht nicht von der eingeführten Schraube mitbewegt wird und den Wirbelkörper durchstößt, da dies zu kritischen Verletzungen führen kann.

Das zweite Anwendungsgebiet hat das Ziel, den Chirurgen bei minimalinvasiven Eingriffen in der Weichteilchirurgie zu unterstützen. Anders als bei der knöchernen Chirurgie kann sich der Chirurg nicht an festen Strukturen orientieren, sondern muss mit den Operationsinstrumenten in dem Operationsgebiet navigieren um Organe nicht zu beschädigen und eine günstige Position für den Eingriff zu finden. Zur bildgebenden Unterstützung werden in der Regel endoskopische Kameras eingesetzt, die ein Videobild von dem Operationsgebiet übertragen.

Der minimalinvasive Zugang zum Operationsgebiet erfolgt durch einen Trokar der in den Patienten eingeführt wird und eine innere Öffnung von $0,5 - 24\text{ mm}$ für die eingesetzten Instrumente bereitstellt. Abhängig von dem durchzuführenden Eingriff und der Lage des Operationsgebietes können mehrere Trokar Zugänge gelegt werden um mit verschiedenen Werkzeugen und Kameras das Zielgebiet zu erreichen.

Durch die Flexibilität des entwickelten STIFF-FLOP Manipulators können Operationsgebiete erreicht werden, die mit verfügbaren Operationsinstrumenten sonst nicht erreicht werden können. Dies erschließt neue Einsatzmöglichkeiten und bietet die Perspektive, chirurgische Eingriffe, die bisher konventionell durchgeführt werden, minimalinvasiv durchzuführen. Eine mögliche medizinische Aufgabenstellung für den entwickelten STIFF-FLOP Arm besteht in der minimalinvasiven Durchführung einer totalen mesorektalen Exzision. Hierbei werden zur Behandlung einer Krebserkrankung Teile des Enddarms entfernt. Einen Einblick in einen Teil des Eingriffes illustriert die Abbildung 5.4.

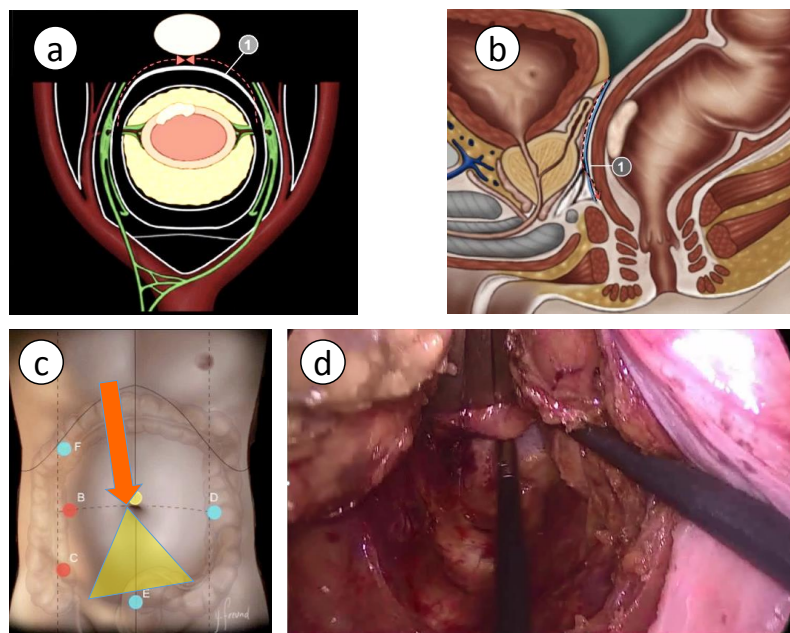


Abbildung 5.4: a) und b) zeigen die schematische Querschnittsansicht des Enddarmbereiches, wo der Enddarm vom angrenzenden Gewebe gelöst werden muss. Bild c) zeigt die Position von Trokarzugängen für Werkzeuge und Kameras. Am Bauchnabel wird ein Laparoskop eingeführt, das den gelben Bereich überwacht. Die endoskopische Ansicht dazu zeigt d). Quelle: STIFF-FLOP Projektdokumentation

Zur Freistellung des zu entfernenden Enddarmes muss dieser von dem umliegenden Gewebe gelöst werden. Zum Erreichen des Operationsgebietes werden Trokarzugänge über weiter entfernte Positionen, wie in Abbildung 5.4 c) abgebildet, gelegt. Auf Grund des schwer erreichbaren Operationsgebietes und den sensiblen umliegenden Regionen, die nicht verletzt werden dürfen, werden solche Eingriffe in der Regel nicht minimalinvasiv durchgeführt. Der Einsatz des STIFF-FLOP Armes wurde im Rahmen einer Machbarkeitsstudie untersucht [3].

5.2 Technische Umsetzung mit Assistenzrobotersystemen

Die unterschiedlichen medizinischen Anwendungen erfordern verschiedene technische Umsetzungen.

Mit dem Ziel, den Chirurgen bei der Platzierung von Pedikelschrauben zu unterstützen und die Strahlenbelastung für den Patienten und den operierenden Chir-

urgen zu minimieren, wurde das am ZESS entwickelte robotische Assistenzsystem MODICAS aufgebaut. Den Ausgangspunkt für diese Arbeit bietet die Weiterentwicklung der Software für die Operationsplanung und -durchführung bei Eingriffen an der Wirbelsäule zur Anbindung eines neu entwickelten Assistenzroboters mit medizinischer Zulassung.

Zu Beginn des chirurgischen Eingriffes werden in der Regel Aufnahmen des Patienten mit der intraoperativen Bildgebungsmodalität angefertigt. Diese Daten können verwendet werden um den Eingriff zu planen und festzulegen, an welcher Stelle Inzisionen gemacht werden und an welcher Position Pedikelschrauben platziert werden sollen. Hierfür wird im Rahmen dieser Arbeit eine Software entwickelt, die eine Planung der Schraubenposition in den aufgenommenen Bilddaten ermöglicht. Um eine robotische Platzierung der Schrauben zu ermöglichen, muss die Software an das bestehende Robotersystem und die zur Navigation verwendete Stereokamera angebunden werden.

Bei der Operationsdurchführung muss die erstellte Planung auf die intraoperative Position des Patienten übertragen werden. Hierfür muss das aufgenommene Bildvolumen mit der intraoperativen Lage des Patienten verknüpft werden. Dieser Schritt wird als intraoperative Registrierung bezeichnet. Hierfür gibt es bereits Lösungen einzelner Hersteller, wie die ZIEHM Naviport Schnittstelle [102] und die Navigationslösung der Firma Brainlab [22]. Diese sind aber nur in Verbindung mit herstellereigenen Komponenten einsetzbar. Da in Kliniken unterschiedliche Bildgebungsmodalitäten von verschiedenen Herstellern eingesetzt werden, soll eine Lösung entwickelt werden die eine herstellerunabhängige Verwendung erlaubt. Hierzu wird ein eigenes Verfahren, basierend auf der Verwendung von Markern die sowohl von optischen Stereokameras als auch von intraoperativ eingesetzten C-Bögen erkannt werden, entwickelt.

Zur Diagnostik werden in der Regel vor dem operativen Eingriff CT-Aufnahmen von der Wirbelsäule angefertigt, die eine höhere Bildqualität als die intraoperativen Bilddaten bieten. Um dem Operateur eine präoperative Planung in den qualitativ besseren CT-Bilddaten zu ermöglichen, muss diese über die intraoperative Bildgebungsmodalität mit der aktuellen Patientenposition überlagert werden. Hierfür soll eine geeignete Methode zum Matching der präoperativen Bilddaten mit den intraoperativen Bilddaten implementiert werden.

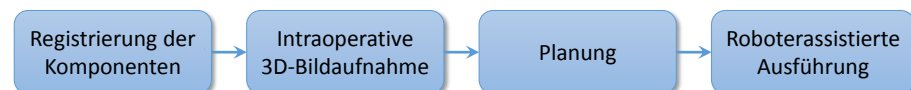
Im Anwendungsgebiet der Weichteilchirurgie soll die Lage des, im Rahmen des STIFF-FLOP Projektes entwickelten, flexiblen Manipulators im Operationsgebiet

überwacht werden. Hierfür wird ein optisches Positionsbestimmungssystem, bestehend aus einer Endoskopkamera, die von einer Stereokamera überwacht wird, aufgebaut. Der Schwerpunkt des eigenen Beitrags liegt in der Detektion des flexiblen Roboterarmes in den Bildern der Endoskopkamera. Die größte Herausforderung stellt dabei die flexible und veränderliche Form des STIFF-FLOP Manipulators dar. Auch lässt sich vorher nicht abschätzen, ob der ganze Manipulator oder nur Teile davon, von der Endoskopkamera im Operationsgebiet erfasst werden. Da der Manipulator während der Projektlaufzeit weiterentwickelt wurde, wodurch sich auch die für die optische Erkennung relevanten äusseren Merkmale änderten, sollten lernfähige Algorithmen ausgewählt werden, die auch eine Detektion von weiterentwickelten Varianten des STIFF-FLOP Manipulators erlauben. Mit dem Ziel das System anpassungsfähig und flexibel umzusetzen, sollen auch Algorithmen, die maschinelles Lernen zur Lösung verwenden, herangezogen werden.

6 Operationsplanung und -durchführung in der knöchernen Chirurgie

In diesem Kapitel wird die Umsetzung des Eingriffes zur Implantierung von Pedikelschrauben mit dem am ZESS entwickelten, robotischen Assistenzsystem MODI-CAS beschrieben. In Diskussionen mit kooperierenden Chirurgen wurde hierfür ein Workflow zum Setzen von Pedikelschrauben mit einem robotischen Assistenzsystem erarbeitet. Im ersten Schritt muss die Bildgebungsmodalität mit der intraoperativen Lage des Patienten verknüpft werden. Dieser Schritt wird als Registrierung der intraoperativen Komponenten bezeichnet. Hierfür wird am Patienten ein Referenzkörper angebracht, der von dem Navigationssystem beobachtet wird. Dieser dient als Referenzpunkt um mit dem Robotersystem die geplanten Schrauben positionieren zu können. Daraufhin wird mit der intraoperativen Bildgebungsmodalität eine 3D-Aufnahme der zu operierenden Wirbelsäulenpartie des Patienten aufgenommen. Dieser Datensatz wird verwendet um die Schraubenpositionen zu planen und anschließend robotergestützt einzubringen.

Intraoperativer Workflow (Variante 1)



Intraoperativer Workflow mit präoperativer Planung (Variante 2)

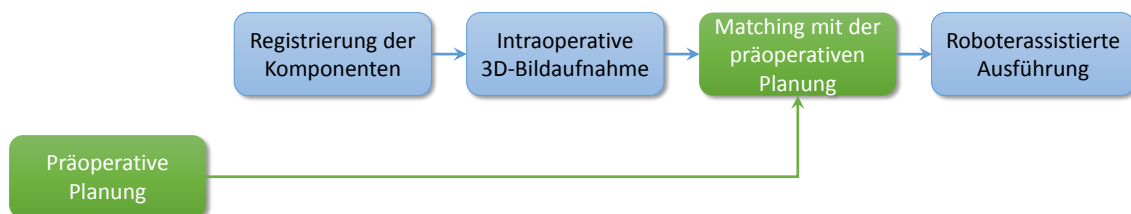


Abbildung 6.1: Workflow zum Setzen von Pedikelschrauben mit der Option einer präoperativen Planung

In einer zweiten Variante des Workflows wird dieser durch eine präoperative Planung erweitert. Diese ermöglicht es dem Chirurgen die Planung ohne Zeitdruck in bereits vorab aufgenommenen, präziseren CT-Aufnahmen durchzuführen. Intraoperativ muss dann die präoperative Planung mit der intraoperativen Bildaufnahme gematcht werden. Die Abbildung 6.1 skizziert die beiden Workflow Varianten.

Im Abschnitt 6.1 wird das bestehende Robotersystem vorgestellt. In einer Systemübersicht werden die angebundenen Komponenten aufgezeigt und die Systemarchitektur skizziert. Im Abschnitt 6.2 wird als eigener Beitrag die Implementierung und die Funktionalität der neu entwickelten Planungssoftware dargestellt. Darauf folgend wird im Abschnitt 6.3 ein eigenes Verfahren zur Registrierung der intraoperativen Bildgebungsmodalität mit dem Navigationssystem vorgestellt, das den ersten Schritt des intraoperativen Workflows umsetzt. Zur Implementierung der zweiten Workflow Variante wird im Abschnitt 6.4 eine Lösung zum Matching von präoperativen und intraoperativen 3D-Bilddatensätzen beschrieben. Abschließend werden die Ergebnisse im Abschnitt 6.5 untersucht.

6.1 Systemübersicht des verwendeten Assistenzrobotersystems

Am Zentrum für Sensorsysteme (ZESS) wird seit über 10 Jahren an dem Entwurf, der Entwicklung und der Erprobung von Komponenten des chirurgischen Assistenzsystems MODICAS - modular interactive Computer Assisted Surgery, gearbeitet. Das Ziel des aufgebauten Systems ist eine umfassende Lösung, die dem operierenden Chirurgen alle Schritte zur Ausführung des vorgestellten Workflows ermöglicht.

In vorhergehenden Arbeiten wurde am ZESS hierfür ein Prototyp des chirurgischen Assistenzsystems, bestehend aus dem 6-achsigen Industrieroboter der Firma Adept (Viper S850) aufgebaut [82] [26] [54] [89] [88] [53]. Am Roboterflansch ist ein Kraft-Momentsensor mit einem Führungsriff angebracht. Zu Beginn des Eingriffes kann der Chirurg das Robotersystem über angebrachte Betätigungsknöpfe aktivieren und den Roboter mit dem Führungsriff über dem Operationsgebiet positionieren. In dem aufgebauten System ist als Endeffektor eine Führung für chirurgische Instrumente integriert. Diese ermöglicht es dem Chirurgen, einen chirurgischen Bohrer und ein chirurgisches Werkzeug zum Eindrehen der Pedikelschrauben einzusetzen. Die Abbildung 6.2 zeigt das eingesetzte Robotersystem.

Zur intraoperativen Navigation wird ein Stereokamerasystem verwendet, das die Position von optischen Referenzkörpern erfasst. Hierfür müssen die intraoperativ eingesetzten Komponenten mit eindeutig unterscheidbaren Referenzkörpern ausgestattet werden. Hierfür wurden in der Arbeitsgruppe für Medizintechnik am ZESS bereits verschiedene Referenzkörper entworfen. Für jeden Referenzkörper wird ein eigenes



Abbildung 6.2: Das chirurgische Assistenzsystem MODICAS, mit einer Führung für chirurgische Instrumente am Endeffektor [95]

Koordinatensystem festgelegt. Die Pose des Referenzkörpers wird über eine Transformationsmatrix im Koordinatenursprung des Stereokamerasystems beschrieben. Am Roboter wurde ein Referenzkörper an der Halteplatte der Instrumentenführung angebracht, das mit dem Koordinatensystem $\{RRB\}$ (Robot Rigid Body) bezeichnet wird. Über eine Roboterkalibrierprozedur wird die Position des $\{RRB\}$ relativ zum Roboterflansch, der mit $\{TCP\}$ (Tool center point) bezeichnet wird, bestimmt und im System hinterlegt. Über eine bereits implementierte Instrumentenkalibrierung kann die Transformation von $\{RRB\}$ zur Spitze des chirurgischen Instruments, das als $\{TTP\}$ (Robot Tooltip) bezeichnet wird, bestimmt werden. Über die Robotersteuerung ist es hierdurch möglich, den Roboter zu Zielpositionen im Koordinatensystem von $\{TTP\}$ zu steuern.

Das Navigationssystem erfasst alle Referenzkörperpositionen relativ zu dem eigenen Koordinatensystem, welches mit $\{OTS\}$ (Optical Tracking System) bezeichnet wird. Die intraoperative Patientenposition wird durch das Anbringen eines Referenzkörpers am Patienten erfasst, das mit $\{ARB\}$ (Aim Rigid Body) bezeichnet wird. Als weiteres Instrument wird ein optisches Zeigeinstrument in dem System mit eingebunden. Dieses wird mit dem Koordinatensystem $\{PRB\}$ (Pointer Rigid

Body) benannt. Die Spitze des Zeigelinstrumentes kann über eine Pivotierfunktion bestimmt und direkt bei der Detektion berücksichtigt werden. Hierdurch liegen die Positionsinformationen im Koordinatensystem der Zeigerspitze, welches mit $\{PTP\}$ (Pointer Tool Tip) gekennzeichnet wird, vor. Die Abbildung 6.3 stellt die intraoperativ relevanten Koordinatensysteme dar.

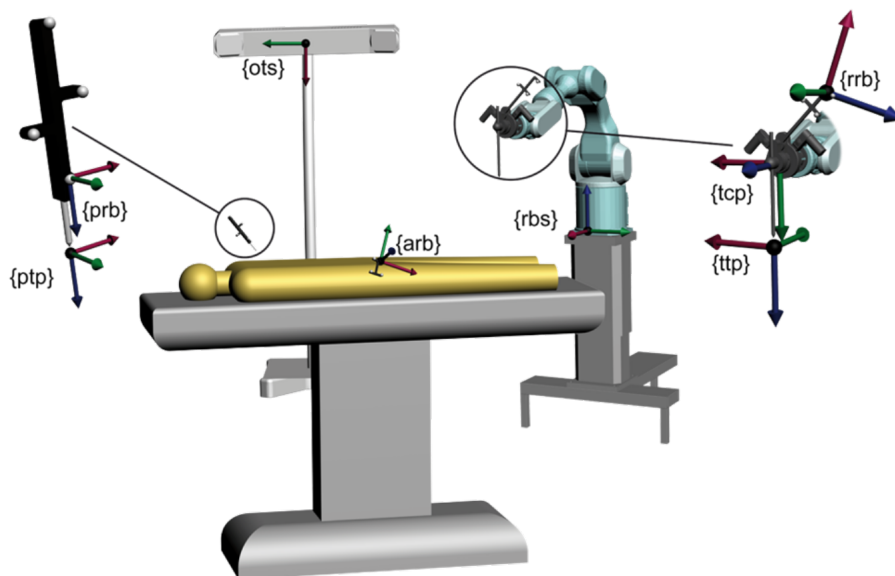


Abbildung 6.3: Skizzierung der intraoperativ verwendeten Koordinatensysteme [87]

Das eingesetzte Robotersystem wird von einem Echtzeitrechner gesteuert. An diesem war auch bisher die, zur intraoperativen Navigation eingesetzte, Stereokamera der Firma NDI (Polaris Spectra, NDI) angebunden. Das System wurde mit der Software LabView entwickelt und kommuniziert über eine Netzwerkverbindung (TCP Verbindung) mit dem zur Planung eingesetzten Computer. Die Kommunikation zum Steuerungsrechner des Roboters erfolgt über ein LabView Programm (modiCAS-Controller) auf dem zur Planung eingesetzten Computer. Die Architektur des Robotersystems ist in der Abbildung 6.4 skizziert.

Das vorhandene Host-Network-Interface (HNI) bildet die Schnittstelle zum modiCAS-Controller. Dieses steht als Netzwerkschnittstelle (TCP-Schnittstelle) mit einem auf die Anforderungen angepassten XML Übertragungsprotokoll zur Verfügung. Die roboterseitige Implementierung der Schnittstelle und die verfügbaren Befehle sind in einer vorhergehenden Arbeit beschrieben [26]. Diese Schnittstelle dient als Ausgangspunkt zur Anbindung des Robotersystems an die neu entwickelte Planungssoftware.

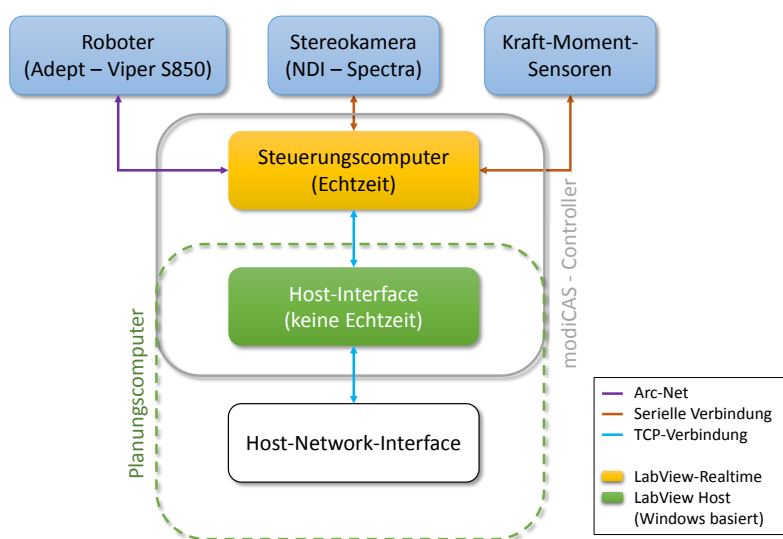


Abbildung 6.4: Systemarchitektur des MODICAS Assistenzrobotersystems

6.2 Entwicklung einer Software zur Operationsplanung mit dem Software Framework Mevislab

Die Umsetzung der vorgestellten Workflow Varianten erfolgt in der Implementierung einer neu zu entwickelnden Planungssoftware. Als Entwicklungswerkzeug wird hierfür das vom Fraunhofer Institut entwickelte Framework Mevislab eingesetzt.

Mevislab ist ein modulares Framework, das insbesondere für die Verarbeitung und Visualisierung von medizinischen Bilddaten entwickelt wurde. Hierfür stehen erweiterte Module zur Segmentierung, Registrierung und für die funktionale Analyse zur Verfügung. Mevislab bietet eine graphische Entwickleroberfläche in der verschiedene Module miteinander verbunden werden können. Durch graphische Verbindungen werden Daten zwischen den Modulen übertragen. Zu jedem Modul gehört auch ein Anzeigepanel, in dem das Modul parametrieren und auf Variablen zugegriffen werden kann. Module die zur Visualisierung eingesetzt werden enthalten auf dem Panel ein Anzeigefenster. Mevislab bietet auch die Möglichkeit eigene Module in C++ zu implementieren, mit denen eigene Methoden und Funktionen erstellt werden können. Zur Erstellung einer Benutzeroberfläche stellt Mevislab eine eigene Entwicklungsumgebung bereit, mit welcher eine Vielzahl von einsetzbaren Bedienelementen zur Benutzerinteraktion auf einer graphischen Benutzeroberfläche platziert werden können. Für graphische Visualisierungen sind in Mevislab Funktionen der open Inventor Bibliothek eingebunden [69]. Zur Umsetzung von eigenen Berechnungen und

aufwendigeren Benutzerinteraktionen können zusätzlich Python Skripte eingebunden werden, wodurch der Funktionsumfang für die Benutzeroberfläche vielfältig und variabel wird. Eine ausführlichere Beschreibung ist auf der Webseite von Mevislab zu finden [68].

Für die Implementierung der Planungssoftware müssen im ersten Schritt die medizinischen Bilddaten von der Bildgebungsmodalität empfangen und visualisiert werden. Daraufhin müssen die nötigen Funktionen für die Planung der Schraubenpositionen im aufgenommenen Bilddatensatz umgesetzt werden. Zur intraoperativen Navigation wird eine neue Stereokamera (Polaris VEGA, NDI) direkt an die Planungssoftware angebunden. Um die erstellte Planung robotergestützt auszuführen, wird das vorhandene Robotersystem angebunden. Die Abbildung 6.5 stellt die an die Planungssoftware angebotenen Hardware Komponenten dar.

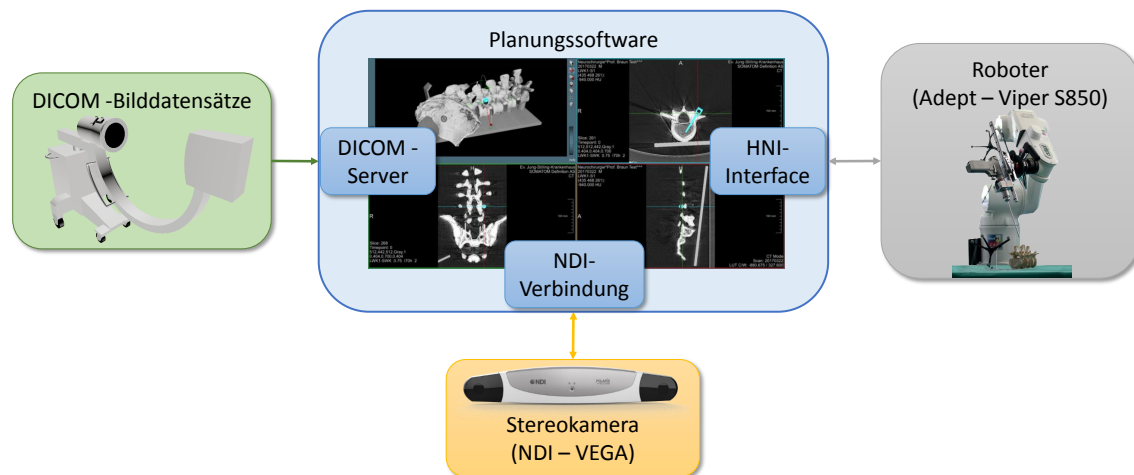


Abbildung 6.5: Anbindung der Hardware Komponenten an die Planungssoftware

Neben der technischen Implementierung wird eine graphische Benutzeroberfläche mit Mevislab erstellt, die eine schrittweise Durchführung des Workflows ermöglicht.

6.2.1 Visualisierung medizinischer Bilddaten

Medizinische Bildgebungsmodalitäten speichern die Bildaufnahmen im standardisierten DICOM Format. Solche Geräte senden die aufgenommenen Daten über ein Netzwerkprotokoll an das in Krankenhäusern eingesetzte PACS (picture archiving and communication system). Um die Daten von einem PACS oder auch direkt von der Bildgebungsmodalität empfangen zu können, bietet Mevislab ein DICOM Receiver Modul. Bei Auswahl der von der Bildgebungsmodalität verwendeten Netzwerk-

adresse und des zugehörigen Ports, wird eine TCP-Socket Verbindung zur Bildgebungsmodalität aufgebaut, die auf eingehende Bilddaten wartet. Sobald Bilddaten empfangen wurden, werden diese in einem ausgewählten Verzeichnis abgelegt.

Zum Laden von DICOM Bilddaten stellt Mevislab das Modul „*DirectDicomImport*“ bereit. Hiermit kann ein Verzeichnis ausgewählt und DICOM Bilddaten importiert werden. Anschließend stehen die Bilddaten am Ausgangsanschluss des Moduls im Mevislab Bildformat zur Weiterverarbeitung zur Verfügung. Neben den Bilddaten sind hier auch erweiterte Informationen aus dem DICOM Header, wie die Position und Lage des Patienten bei der Bildaufnahme und die Größe und Skalierung der Voxel enthalten.

Um 3-dimensionale DICOM Bilddaten maßstabsgetreu darstellen zu können, muss die Voxelgröße, die bei der Aufnahme festgelegt wurde (siehe 4.1) , berücksichtigt werden. Diese und weitere Aufnahmeparameter aus dem DICOM-Header des Datensatzes sind ebenfalls im Mevislab Bildformat geladen. Aus diesen Parametern kann eine Transformationsmatrix erstellt werden, mit der sich der Bilddatensatz in eine maßstabsgetreue Volumendarstellung transformieren lässt. Die Formel 6.1 zeigt die Transformationsmatrix für einen 3D Bilddatensatz, der mit dem am ZESS vorhanden 3D C-Bogen (Ziehm, Vario 3D) aufgenommen wurde. Das Gerät besitzt ein würfelförmiges Aufnahmevolumen mit der Kantenlänge von 128mm und einer Auflösung von 256 Bildpunkten in jeder Richtung, woraus sich eine quadratische Voxelgröße mit einer Kantenlänge von $0,5\text{mm}$ ergibt.

$${}^{world}T_{\text{voxel}} = \begin{bmatrix} -0.5 & 0 & 0 & 0.25 \\ 0 & 0.5 & 0 & -0.25 \\ 0 & 0 & -0.5 & 127.75 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

Neben der Skalierung des Bildes, wird hier die X- und die Z-Achse der Volumenaufnahme gespiegelt und der Koordinatenursprung durch eine Translation in das Voxelzentrum gelegt. Die Spiegelung und Verschiebung ergibt sich aus der bei der Aufnahme gewählten Lage des Patienten, was bereits im Abschnitt 4.1 beschrieben ist, sodass die X-Achse von der rechten Körperseite zur linken, die Y-Achse von der Bauchseite in Richtung der Rückseite und die Z-Achse von den Füßen hin zum Kopf ausgerichtet ist.

Die aufgenommenen Voxelintensitäten liegen bei röntgen basierten Bildgebungsgeräten als 12-bit oder 16-bit Werte vor. Um knöcherne Strukturen deutlich dar-

zustellen, müssen Voxelwerten mit knöchernen Strukturen erkennbare Grau- oder Farbwerte zugewiesen werden. Da 12-bit oder 16-bit Grauwerte vom menschlichen Auge nicht unterschieden werden können, werden Transferfunktionen verwendet, um den knöchernen Strukturen deutlich erkennbare Werte zu zuweisen.

In Mevislab werden hierfür Lookuptables verwendet, die den Voxelwerten der Aufnahme Farb- oder Grauwerte zuweisen. Da die Histogramm Verteilung verschiedener Datensätze unterschiedlich sein kann, muss die Visualisierung für jeden Datensatz anwenderfreundlich angepasst werden können. Durch Auswahl eines Histogramm Bereichs wird ein Übergang zwischen der minimalen Intensitätsfarbe Schwarz und der maximalen Intensitätsfarbe Weiß bestimmt. In Mevislab erstellt das „*RampLUT*“ Modul eine entsprechende Lookuptable. Mit dem „*MouseGrabber*“ Modul werden Mauseingaben des Visualisierungsfensters zur Auswahl des Histogramm Bereichs an das „*RampLUT*“ Modul übertragen. Dadurch können Helligkeit und Kontrast und damit auch die Darstellung des Datensatzes schnell und einfach angepasst werden. Die Abbildung 6.6 zeigt das implementierte Mevislab Netzwerk zum Laden und Rendern eines Bilddatensatzes.

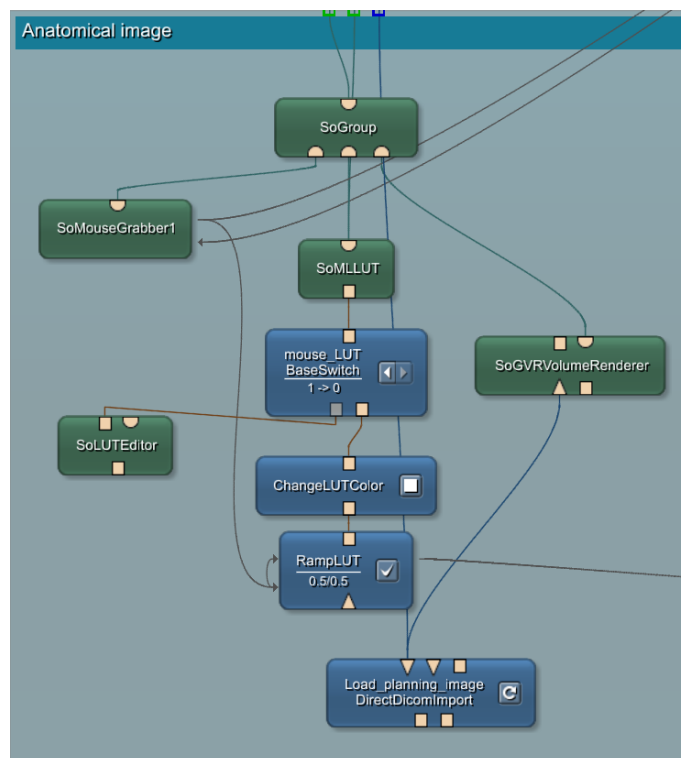


Abbildung 6.6: Mevislab Netzwerk zum Laden von DICOM Bilddaten

Alternativ hierzu bietet Mevislab mit dem „*LUTEditor*“ Modul eine interaktive Möglichkeit, Histogramm Werten beliebige Farbwerte zu zuweisen. Hierdurch

können aufwendigere und farblich hervorgehobene Darstellungen erzeugt werden. Mit dem BaseSwitch Modul kann zwischen beiden Methoden umgeschaltet werden.

Für die Berechnung einer auf dem Bildschirm darstellbaren Szene der 3D-Aufnahme bietet Mevislab das „SoGVRVolumeRenderer“ Modul. Dieses beinhaltet bereits Interpolationsmethoden zur Berechnung einer perspektivischen und interaktiv rotierbaren Volumendarstellung der 3D-Aufnahme. Zusätzlich kann die für die Aufnahme gültige Transformation, wie in der Formel 6.1 gezeigt, zur Umrechnung der Voxel­darstellung in eine maßstabsgetreue Darstellung mit dem Parameter „use world coordinates“ direkt auf den Datensatz angewendet werden. Die Abbildung 6.7 zeigt das Settings Panel des „SOGVRVolumeRenderer“ Moduls. Anschließend steht am Ausgang ein gerenderteter open Inventor Datensatz zur Verfügung, der von einem Anzeige Modul dargestellt werden kann.

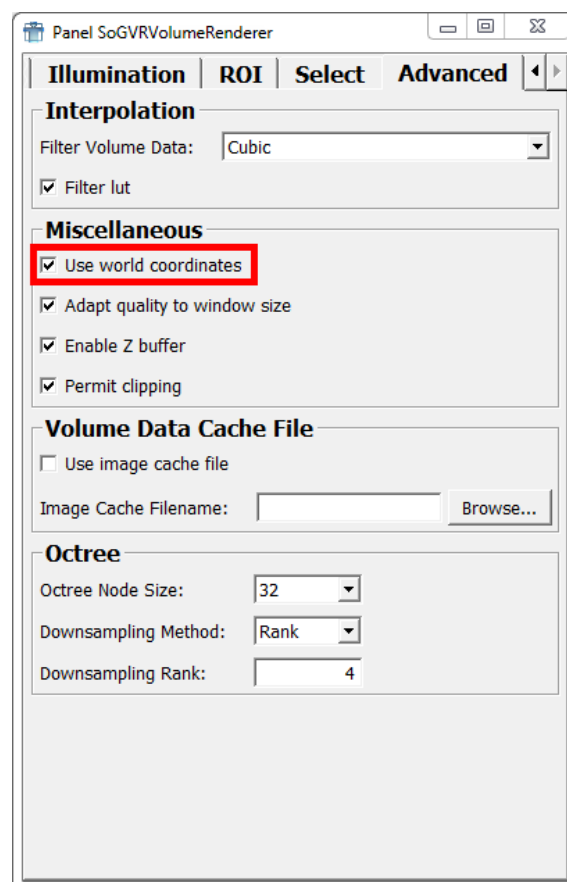


Abbildung 6.7: Settings Panel des SoGVRVolumeRenderer Moduls zur Berechnung einer maßstabsgetreuen Darstellung durch Multiplikation der Transformationsmatrix (Formel: 6.1) durch die Aktivierung des Parameters „use world coordinate“

In der Abbildung 6.8 ist als Ergebnis die Visualisierung im Anzeigefenster des „SoGroup“ Moduls dargestellt, in der die Lookuptable auf das gerenderte Bildvolumen angewendet wird.

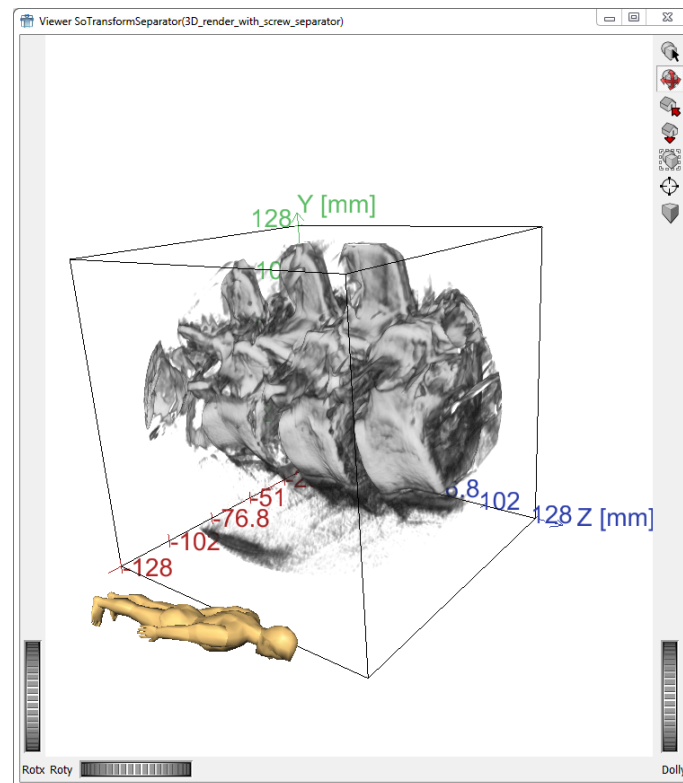


Abbildung 6.8: 3D-Visualisierung eines DICOM Datensatzes mit dem eingezeichneten Patientenkoordinatensystem und der Ausrichtung des Patienten

Die in der Abbildung 6.8 dargestellte Visualisierung ist hilfreich um einen Überblick über den gesamten Datensatz zu erhalten. Zur chirurgischen Planung werden in der Regel orthogonale Schichtansichten des 3D-Bilddatensatzes verwendet, da hier jedem Bildpunkt direkt eine eindeutige Position im Bilddatensatz zugeordnet werden kann.

Für die Entwicklung der Planungssoftware wird das Modul „SoOrthoView2D“ verwendet, um aus dem 3D-Bilddatensatz eine Schichtebene auszuwählen, die anschließend in dem „SoRenderAreaModul“ dargestellt wird. In der Abbildung 6.9 ist das erstellte Netzwerk zum Visualisieren einer Schichtansicht dargestellt. Als Eingang dient der in der Abbildung 6.6 geladene Bilddatensatz.

Die Abbildung 6.10 zeigt die Parametrierung des „SoOrthoView2D“ Moduls zur Darstellung der koronalen Schichtansicht eines Datensatzes. Hierbei bestimmt der Parameter „World Position“ einen zentralen Schnittpunkt für die koronale, sagittale

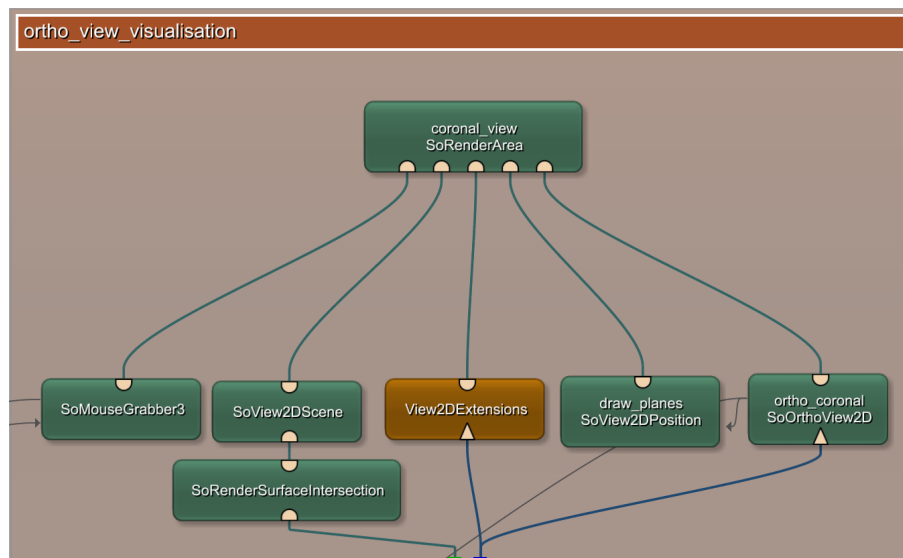


Abbildung 6.9: Mevislab Netzwerk zur Visualisierung der koronalen Schichtebene eines 3D-Bilddatensatzes

und die axiale Schichtansicht. Die Darstellung aller orthogonalen Schichtansichten erfolgt analog hierzu.

Der aktuelle „World Position“ Vektor wird zwischen den „*SoOrthoView2D*“ Modulen für alle 3 Schichtansichten synchronisiert und auch an das „*SoView2DPosition*“ Modul übergeben. Mit diesem kann die aktuelle Mausposition und Tastatureingaben erfasst werden. Auch die aktuelle Schichtebene wird hiermit farblich hervorgehoben dargestellt und mit dem gerenderten Datensatz überlagert. Hier werden die sagittale Ebene rot, die koronale Ebene grün und die axiale Ebene in blau hervorgehoben. Dies erleichtert die Navigation durch verschiedene Schichtansichten.

Um auch in der Schichtansicht die Darstellung der Visualisierung noch anpassen zu können wird, wie bei der 3D-Visualisierung, ein „*SoMouseGrabber*“ Modul eingesetzt und mit dem „*SoMouseGrabber*“ Modul für die 3D-Visualisierung, dass in der Abbildung 6.6 vorgestellt ist, synchronisiert. Weitere Interaktionsfunktionen bringt das „*View2D – Extensions*“ Modul mit, das neben der Einblendung von relevanten Patientendaten, auch Funktionen zur Navigation durch die Schichtebenen mit dem Mousrad, sowie Optionen zum Zoomen und Verschieben des sichtbaren Bildausschnittes zur Verfügung stellt.

Um auch weitere Objekte wie Pedikelschrauben, die in der 3D-Szene eingefügt werden, ebenfalls in der Schichtansicht darzustellen, wird mit dem Modul „*SoRenderSurfaceIntersection*“ die Überlagerung der 3D-Szene mit der Schichtansicht berechnet und gerendert. Unter Verwendung des *SoView2DScene* Moduls wird dieses

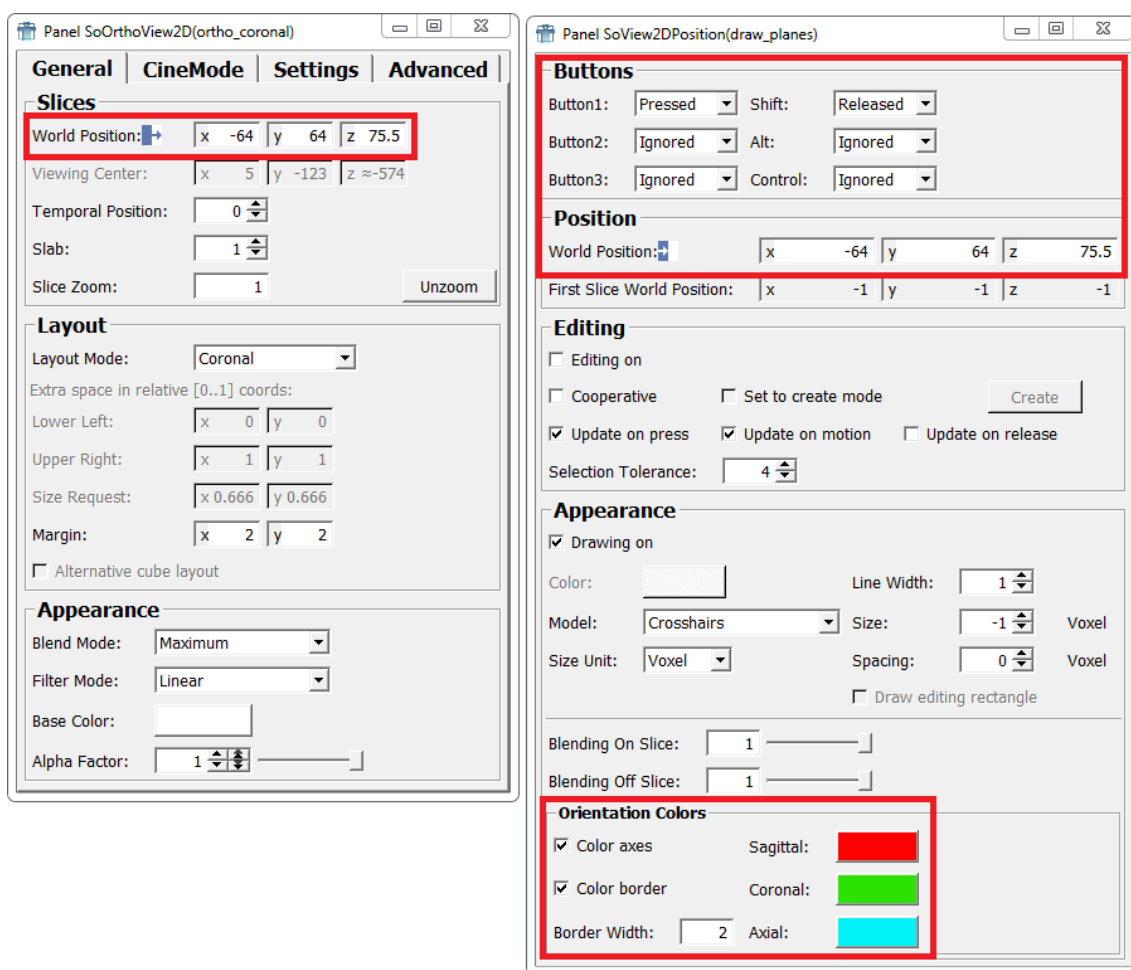


Abbildung 6.10: Parametrierung des „SoOrthoView2DModuls“ und orthogonale 2D-Schichtansicht der koronalen Schichtansicht eines DICOM Datensatzes (links) und die Parametrierung des „SoView2DPosition“ Moduls zum Erfassen der Mauszeigerposition und zum Hervorheben der aktuellen Schichtansichten (rechts)

Schnittbild mit der aktuellen Schichtansicht überlagert und eingefügte Objekte werden auch in der Schichtansicht eingeblendet. Die Abbildung 6.12 zeigt eine koronale Schichtansicht des in der Abbildung 6.8 dargestellten 3D-Bilddatensatzes. Dabei weißt die grüne Umrandung auf die koronale Ebene hin, während die sagittale und axiale Ebene hier in rot und blau dargestellt sind.

Als Orientierungsunterstützung sollen die angezeigten Schichtebenen ebenfalls in der 3D-Szene hervorgehoben werden. Hierfür wird das Mevislab Modul „MPR“ (Multi Planar Reconstruction) verwendet. Die Abbildung 6.11 zeigt den Einsatz sowie die Parametrierung des MPR Moduls. Als Eingangssignal benötigt das „MPR“ Modul den geladenen Bilddatensatz. Des Weiteren wird die aktuelle Position, des in der Abbildung 6.10 dargestellten „SoOrthoView2DModuls“, an den Translationspara-

meter des „MPR“ Moduls übergeben. Über die „World Views“ Schaltfläche kann die anzuzeigende Schichtebene ausgewählt werden, womit die Ausrichtung des Normalenvektors der Ebene, die dargestellt werden soll, festgelegt wird. Neben dem Normalenvektor wird ein Stützpunkt zum Festlegen der Ebene bestimmt. Dieser ergibt sich aus dem Abstand der Ebene vom Koordinatenursprung. Aufgrund der orthogonalen Ausrichtung der Schichtebenen, entspricht der Abstand, je nach ausgewählter Schichtebene, genau der X-, Y- oder der Z-Komponente des Translationsparameters. In der Abbildung 6.11 ist als Beispiel die Parametrierung der koronalen Schichtebene aufgezeigt. Hier ist der Normalenvektor in negativer Y-Richtung ausgerichtet und hat einen Abstand von -64 mm vom Ursprung.

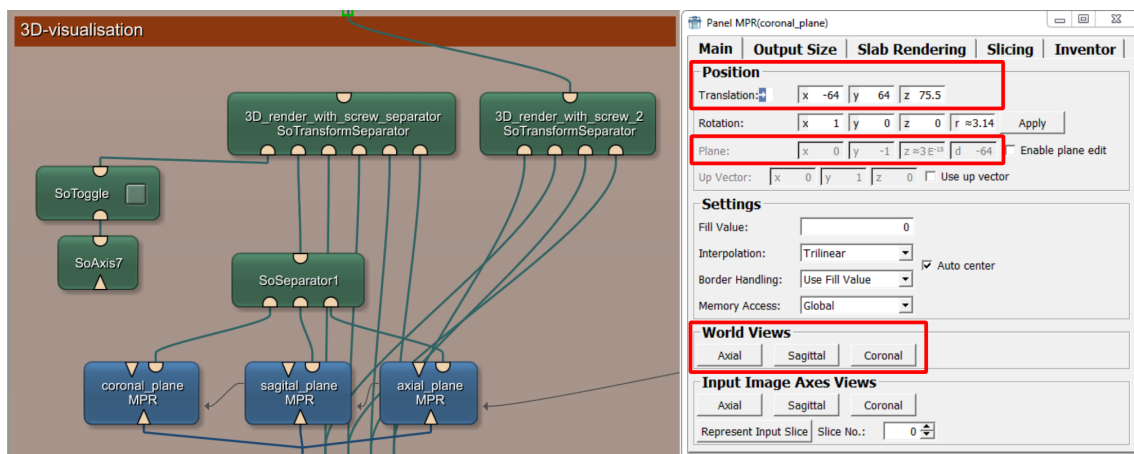


Abbildung 6.11: Mevislab Netzwerk zum Hervorheben der aktuell angezeigten Schichtebenen und Parametrierung des „MPR“ Moduls

Zum Hervorheben der sagitalen und axialen Ebenen werden zwei weitere MPR Module eingebunden, deren Parametrierung analog erfolgt. Am Ausgang des „MPR“ Moduls liegt nun die in der Szene hervorgehobene Ebene mit einer ausgewählten farblichen Umrandung vor. Die Ausgänge der MPR Module werden in dem „SoSeparator“ Modul gebündelt und zur Ansicht an den „SoTransformSeparator“ übergeben, der die 3D-Visualisierung der aktuellen Szene übernimmt. Als weiteres dargestelltes Signal wird hier die in der Abbildung 6.6 erstellte 3D-Visualisierung des Datensatzes eingefügt. Auch weitere Elemente, wie platzierte Schraubenmodelle oder weitere darzustellende Objekte können hier angezeigt werden. Die Abbildung 6.12 zeigt den gerenderten 3D-Bilddatensatz (a) und farblich umrandet die aktuell in den orthogonalen Schichtansichten dargestellten Ebenen. Die axiale Schichtansicht (b) stellt dabei die X-Y Ebene dar, die koronale Schichtansicht (c) zeigt die X-Z Ebene und in der sagitalen Schichtansicht (d) ist die Y-Z Ebene dargestellt.

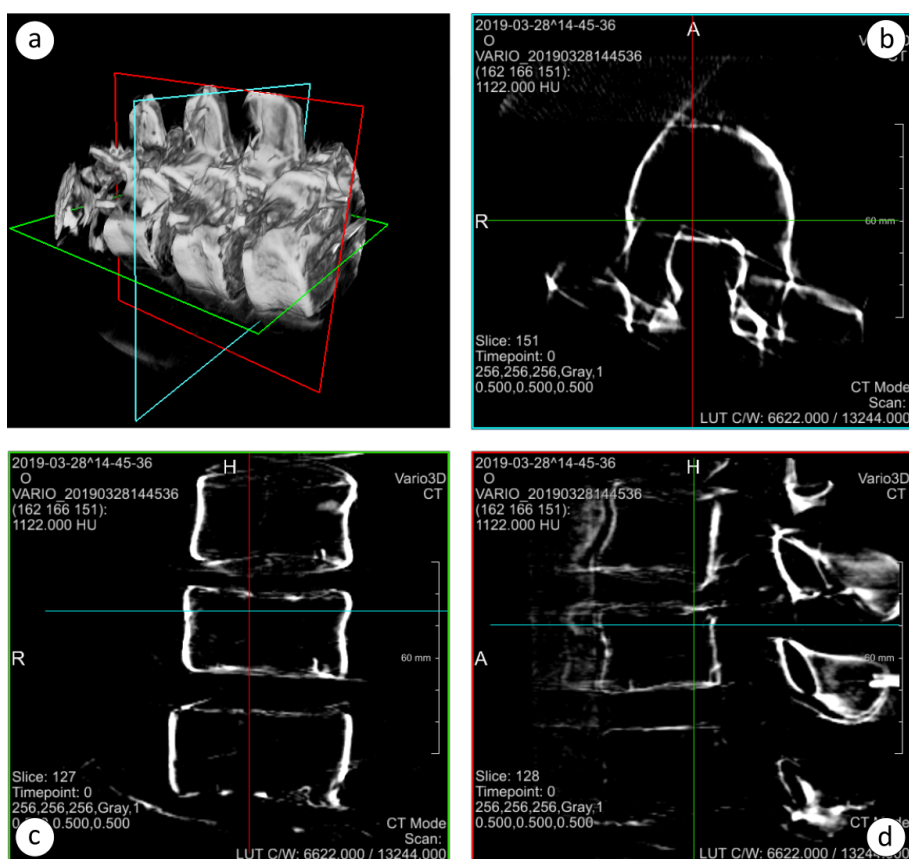


Abbildung 6.12: 3D-Visualisierung eines 3D-Datensatzes, in dem die aktuell angezeigten Schichtansichten farblich umrandet eingeblendet sind (a), axiale Schichtansicht (b) die blau umrandet dargestellt ist, koronale Schichtansicht (c) die grün umrandet dargestellt ist und die sagittale Schichtansicht (d) die rot umrandet dargestellt ist

6.2.2 Planung von Pedikelschrauben

Die Planung von Pedikelschrauben soll es dem Chirurgen ermöglichen, eine Schraubenposition im Bilddatensatz des Patienten festzulegen. Hierdurch wird eine Zielposition und Orientierung für eine zu platzierende Pedikelschraube relativ zum Koordinatenursprung des Bilddatensatzes festgelegt. Den Ausgangspunkt hierfür bilden die in dem Abschnitt 6.2.1 eingeführten Schichtansichten aus der Abbildung 6.12 und die 3D-Darstellung des Bilddatensatzes.

Im ersten Schritt muss ein Schraubenmodell geladen und angezeigt werden. Hierfür bietet Mevislab das „*WEMLoad*“ Modul, mit dem verschiedene 3D-Objekte importiert werden können. Mit dem „*SoWEMRenderer*“ Modul können diese Objekte zur Darstellung gerendert werden. Hier sind verschiedene Darstellungsoptionen wie Farbe, Transparenz und auch die Umgebungsbeleuchtung einstellbar. Die Abbil-

Abbildung 6.13 (a) zeigt das Mevislab Netzwerk zum Laden eines 3D-Objektes, dass im „SoSeparator“ Anzeigemodul dargestellt werden kann und an dessen Ausgang als open Inventor Objekt vorliegt. In der Arbeitsgruppe für Medizintechnik am ZESS wurden bereits 3D-Modelle von Pedikelschrauben mit einem Durchmesser von 5 mm und einer Länge von 35 mm bis 50 mm entworfen. Die Abbildung 6.13 (c) zeigt das gerenderte Schraubenmodell mit einer Länge von 50 mm und dem im 3D-Modell definierten Koordinatenursprung im Zentrum des Schraubenkopfes.

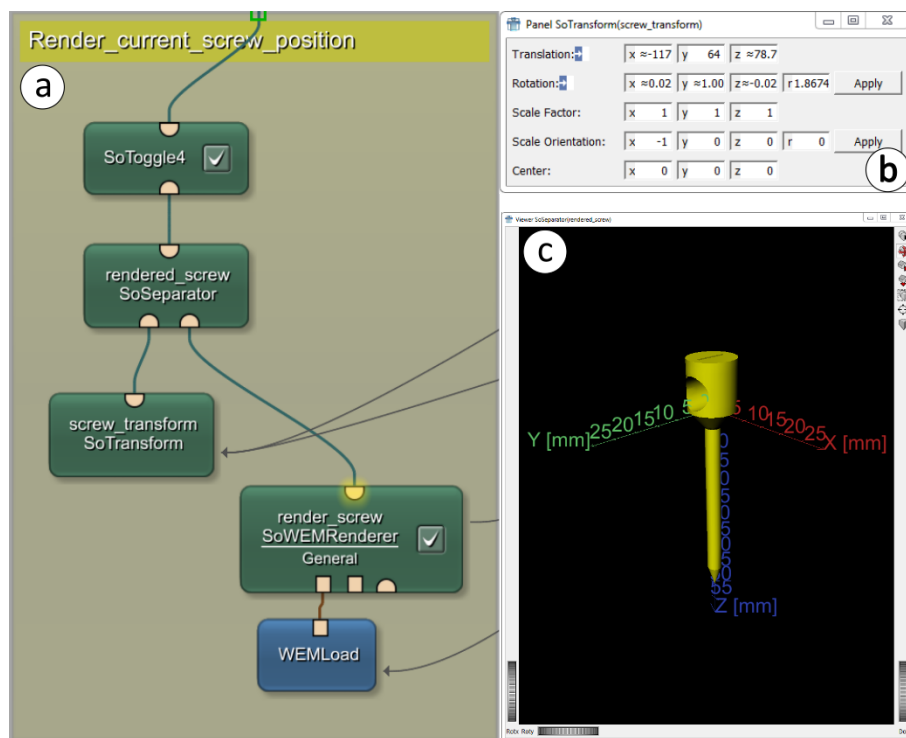


Abbildung 6.13: Laden und Anzeigen eines 3D-Objektes in Mevislab (a), das Panel des „SoTransform“ Moduls (b) und die gerenderte Darstellung des SoSeparator Moduls wo ein Schraubenmodell in Gelb dargestellt ist (c)

Um das Schraubenmodell an eine bestimmte Position zu verschieben, wird mit dem „SoTransform“ Modul eine Transformation, bestehend aus einem Translationsvektor und einem Rotationsvektor, durchgeführt. Die Parametrierung des „SoTransform“ Moduls ist in der Abbildung 6.13 (b) dargestellt. Dabei setzt sich der Rotationsvektor aus einem Richtungsvektor und einen Rotationswinkel zusammen. Die daraus generierte Transformation transformiert das gerenderte Schraubenmodell im „SoSeparator“ Modul. Am „SoSeparator“ Modul werden die angeschlossenen Elemente in der Anschlussreihenfolge verarbeitet. Hierdurch wird die Schraube an der transformierten Position dargestellt.

Diese Darstellung kann nun zur 3D-Visualisierungsszene aus der Abbildung 6.11 hinzugefügt werden. Da diese Szene auch über das „*SoRenderSurfaceIntersection*“ Modul in den Schichtansichten angezeigt wird (Abbildung 6.9), ist die gerenderte Schraube ebenfalls in den Schichtansichten dargestellt.

Die Planung der Schraubenposition erfolgt in den, in der Abbildung 6.12 dargestellten, Schichtansichten. Diese können durch die im Abschnitt 6.2.1 beschriebene Parametrierung bereits im Patientenkoordinatensystem dargestellt werden. Der Translationsvektor der Schraubenposition $scr\vec{e}w_{translation}$ kann hierdurch direkt aus der Bildposition der Schichtansicht extrahiert werden. Hierfür wird mit einem weiteren „*SoView2DPosition*“ Modul die Position des Mauszeigers beim Drücken der linken Maustaste in den Schichtansichten erfasst. Als Ergebnis erhält man einen Translationsvektor $\vec{t}_{leftclick}$, der die aktuelle X-, Y- und Z-Position relativ zum Patientenkoordinatensystem angibt. Dieser Vektor wird an das „*SoTransform*“ Modul zur Translation der Schraubenposition übergeben.

$$scr\vec{e}w_{translation} = \vec{t}_{leftclick} \quad (6.2)$$

Die Bestimmung der Orientierung erfolgt über die Eingabe einer zweiten Mauszeigerposition $\vec{t}_{rightclick}$. Hierfür wird ein weiteres „*SoView2DPositionModul*“ verwendet, das die Position des Mauszeigers beim Betätigen der rechten Maustaste erfasst. Aus der Differenz ergibt sich die Richtung für die zu platzierende Schraube.

$$scr\vec{e}w_{direction} = \vec{t}_{leftclick} - \vec{t}_{rightclick} \quad (6.3)$$

Um die Schraube mit dem „*SoTransform*“ Modul entsprechend der festgelegten Orientierung auszurichten, muss die Rotationsachse und der Rotationswinkel bestimmt werden. Wie in der Abbildung 6.13 (c) dargestellt, ist die Ausrichtung des geladenen Schraubenmodells in Z-Richtung festgelegt. Hieraus ergibt sich der Richtungsvektor $scr\vec{e}w_{origin}$ des geladenen Schraubenmodells.

$$scr\vec{e}w_{origin} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (6.4)$$

Die gesuchte Rotationsachse $scr\vec{e}w_{axes}$ ergibt sich aus dem Kreuzprodukt zwischen $scr\vec{e}w_{origin}$ und $scr\vec{e}w_{direction}$.

$$\vec{screw}_{rotaxes} = \vec{screw}_{origin} \times \vec{screw}_{direction} \quad (6.5)$$

Der Rotationswinkel kann gemäß der Formel 6.6 bestimmt werden.

$$screw_{angle} = \arccos \left(\frac{\vec{screw}_{origin} \cdot \vec{screw}_{direction}}{|\vec{screw}_{origin}| |\vec{screw}_{direction}|} \right) \quad (6.6)$$

Die Umsetzung in Mevislab erfolgt in der eingebundenen Python Funktion 6.1. Diese berechnet den Vektor $\vec{screw}_{direction}$ aus den erfassten Mauszeigerpositionen und verwendet die *SbRotationd* Funktion aus der *SbRotationd* Klassenbibliothek um den Rotationsvektor, bestehend aus dem Richtungsvektor $\vec{screw}_{rotaxes}$ und dem Rotationswinkel $screw_{angle}$, zu bestimmen. Abschließend wird das Ergebnis in das Mevislab Modul *calculate_rotation* übertragen.

```

1 from Inventor.base import SbVec3d, SbRotationd
2 from mevis import *
3 from PythonQt import QtCore, QtGui, QtWebKit
4
5 def updateOrientation():
6     screw_direction = ctx.field("SoView2DPosition_rightclick.worldPosition").inventorValue() - \
7         ctx.field("SoView2DPosition_leftclick.worldPosition").inventorValue()
8     orientation = SbRotationd(SbVec3d(0,0,1), screw_direction)
9     ctx.field("calculate_rotation.rotation2").setValue(orientation)

```

Quellcode 6.1: Python Funktion „updateOrientation“ zur Berechnung der Schraubenorientierung

Der Aufruf des Python Scripts erfolgt in dem Mevislab Script File, das für die Planungssoftware erstellt wird. Bei jeder Aktualisierung der Position durch einen Rechtsklick in einem der Schichtansichtfenster wird damit auch die aktuelle Schraubenorientierung neu berechnet.

```

1 FieldListener SoView2DPosition_rightclick.worldPosition {command = updateOrientation}

```

Quellcode 6.2: Aufruf der Python Funktion „updateOrientation“ im Mevislab Script File über den Fieldlistener Parameter

Diese Implementierung erlaubt bereits eine einfache Planung einer Schraubenposition. Die Abbildung 6.14 zeigt ein Schraubenmodell, das mit zwei Mausklicks in den Schichtansichten des Bilddatensatzes geplant wurde.

Zur präzisen Feinjustierung der Schraubenposition werden Schaltflächen in den Anzeigefenstern der Schichtansichten eingefügt, die bei Betätigung eine Translation

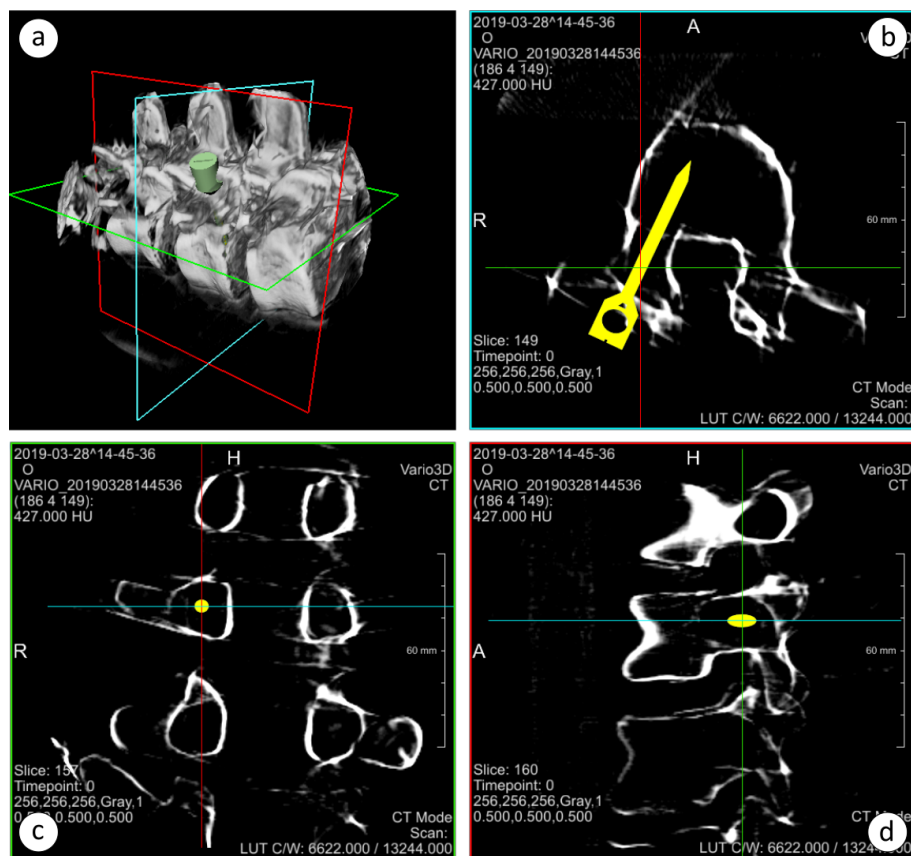


Abbildung 6.14: Positionierung eines Schraubenmodells (gelb) in der 3D-Ansicht des Bilddatensatzes (a) und in den Schichtansichten des Datensatzes (b), (c), (d)

oder Rotation der Schraubenposition ausführen. Um eine intuitive Bedienung zu ermöglichen, muss die Schraubenposition für jede Schichtansicht in der dargestellten Ebene justiert werden können. In der axialen Schichtansicht, die in der X-Y Ebene liegt, muss der Translationsvektor in X-Richtung angepasst werden, um die Schraubenposition in der dargestellten Schichtansicht nach links oder rechts zu verschieben. Durch eine Verschiebung in Y-Richtung ergibt sich eine Auf- oder Abwärtsverschiebung. Zur Feinjustierung der Rotation muss um die Z-Achse rotiert werden. Verschiebungen in der sagittalen Schichtansicht erfolgen entsprechend in der Y-Z Ebene des Bilddatensatzes und für Änderung der Rotation wird hier um die X-Achse rotiert. Änderungen in der koronalen Schichtansicht erfolgen in der X-Z Ebene, wobei hier Rotationsänderungen um die Y-Achse appliziert werden.

Zur Einbindung von Schaltflächen bietet Mevislab ein „SoView2DButton“ Modul, mit dem eine Schaltfläche auf einem Anzeigefenster platziert wird. Die Parametrierung des Moduls erlaubt die Auswahl eines eigenen Schaltflächensymbols und legt die Größe sowie Position der Schaltfläche fest. Da für jede Schaltfläche ein eigenes

„SoView2DButton“ Modul nötig ist, wurden diese in einem Makromodul zusammengefasst. Der Aufbau des Makromoduls ist in der Abbildung A.3 im Anhang dargestellt.

Die Berechnung der Translations- und Rotationsänderungen erfolgt über Python Funktionen. Translationsänderungen erfolgen durch Addition eines Verschiebungsschrittes von 1 mm. Auszugsweise werden die Funktionen zur Verschiebung in X-Richtung, sowie auch die Bestimmung der Rotationsänderung in der axialen Schichtansicht entlang der Z-Achse im Quellcode 6.3 dargestellt. Rotationsänderungen bei Betätigung der Schaltfläche erfolgen in Schritten von 0,05 rad.

```
1 from Inventor.base import SbVec3d, SbRotationd
2 from mevis import *
3 from PythonQt import QtCore, QtGui, QtWebKit
4
5 rot = 0.05
6 step = 1
7
8 def translationXPlus():
9     x = ctx.field("TranslationVector.x").value
10    y = ctx.field("TranslationVector.y").value
11    z = ctx.field("TranslationVector.z").value
12    ctx.field("XMarkerListContainer.vecXYZ").setValue(x+step,y,z)
13
14 def translationXMinus():
15    x = ctx.field("TranslationVector.x").value
16    y = ctx.field("TranslationVector.y").value
17    z = ctx.field("TranslationVector.z").value
18    ctx.field("XMarkerListContainer.vecXYZ").setValue(x-step,y,z)
19
20 def rotclockwiseAxial():
21    rot2 = SbRotationd(SbVec3d(0,0,-1), rot)
22    ctx.field("calculate_rotation.rotation1").setValue(rot2)
23
24 def rotcounterclockwiseAxial():
25    rot2 = SbRotationd(SbVec3d(0,0,1), rot)
26    ctx.field("calculate_rotation.rotation1").setValue(rot2)
```

Quellcode 6.3: Python Funktion zur Translation und Rotation der Schraubenposition in der Axialen Schichtansicht

Die Funktionen zur Verschiebung der Schraubenposition in Y- und Z-Richtung, sowie die Generierung der Rotationsvektoren für Rotationsänderungen in der sagitalen und koronalen Schichtansicht sind analog hierzu implementiert. Der Aufruf der Python Funktionen erfolgt über FieldListener in dem zur Planungssoftware gehörenden Script File. Dieses ist im Quellcode 6.4 veranschaulicht.

```

1 FieldListener Buttons.addAxial {command = rotclockwiseAxial }
2 FieldListener Buttons.subtractAxial {command = rotcounterclockwiseAxial }
3 FieldListener Buttons.TranslationLeftAxial {command = translationXMinus }
4 FieldListener Buttons.TranslationRightAxial {command = translationXPlus}

```

Quellcode 6.4: Aufruf der Python Funktionen zur Verschiebung und Rotation der Schraubenposition bei Betätigung der Schaltfläche in der Axialen Schichtansicht, aufgerufen aus dem Mevislab Script File

Während die Translationsverschiebung direkt im Python Skript ausgeführt wird, erfolgt die Berechnung der resultierenden Rotation im Mevislab Modul „*calculate_rotation*“. Dieses berechnet den resultierenden Rotationsvektor und Winkel aus dem ursprünglichen Rotationsvektor der Schraube und dem Änderungsvektor aus dem Pythonskript. Bei jeder Betätigung einer Rotationsschaltfläche wird so ein neuer Rotationsvektor generiert und die Berechnung erneut ausgeführt.

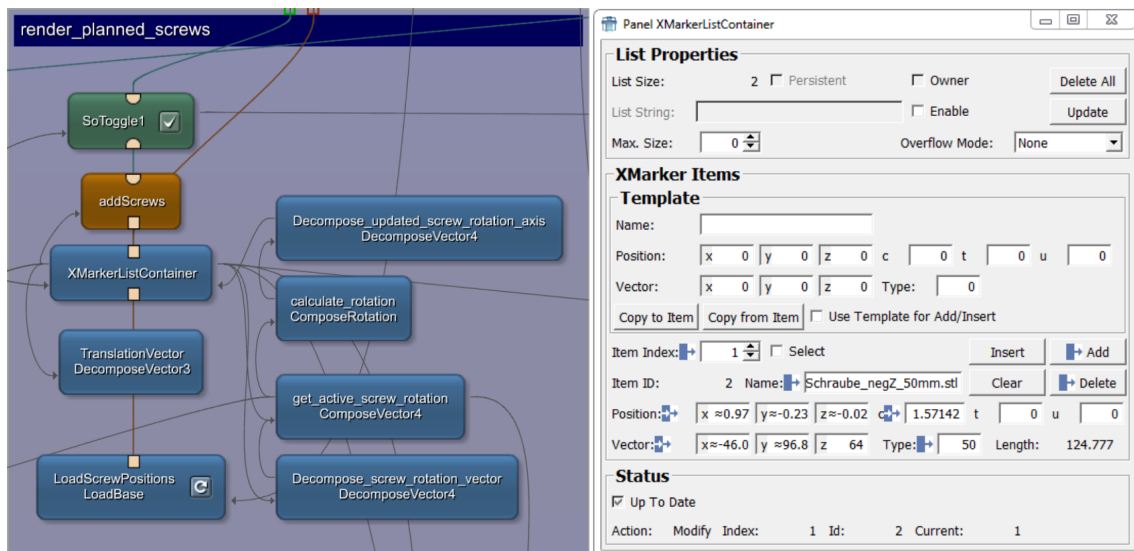


Abbildung 6.15: Mevislab Netzwerk zur Berechnung des aktuellen Rotationsvektors und zur Generierung der Liste an geplanten Schraubenpositionen im XMarkerListContainer Modul, sowie das Panel zur Parametrierung des XMarkerListContainer Moduls

Um mehrere Schraubenpositionen planen zu können, muss der Translations- und Rotationsvektor für jede Schraubenposition gespeichert werden. Hierfür wird der „*XMarkerListContainer*“ in Mevislab verwendet. Hier wird für jede geplante Schraube ein Listeneintrag generiert, in dem der Translationsvektor $screw_{translation}$ und der berechnete Rotationsvektor abgelegt werden. Zusätzlich wird hier auch das ausgewählte Schraubenmodell und die damit verbundene Schraubenlänge gespeichert. Der „*XMarkerListContainer*“ bietet ausserdem Möglichkeiten zur Verwaltung der Listeneinträge durch Auswahl des entsprechenden Listeneintrags (Index),

sowie Optionen zum Hinzufügen (Add) und Löschen (Delete) von Listeneinträgen, die in dem Parameterpanel des „*XMarkerListContainers*“, in der Abbildung 6.15, dargestellt sind. Die aktuelle Schraubenposition und der Rotationsvektor werden an das „*screw_transform*“ Modul, welches in der Abbildung 6.13 dargestellt ist, übergeben.

Mit diesem Ansatz kann lediglich die aktuell ausgewählte Schraube dargestellt werden. Um alle hinzugefügten Schrauben anzuzeigen, wird ein eigenes Mevislab Modul „addScrew“ erstellt. Dieses soll für jede geplante Schraube, das entsprechende Schraubenmodell laden und an der geplanten Zielposition darstellen. Da die Anzahl von Listeneinträgen sich mit jeder hinzugefügten Schraube ändert, muss das Modul dynamisch für jeden Listeneintrag eine Schraubensvisualisierung, wie in Abbildung 6.13, erstellen. Dies erfolgt über die Ausführung eines erstellten Python Scripts 6.5.

```

1  _addedModules = []
2
3  def createScrew():
4      resetModule()
5      screw_index = 0
6      listSize = ctx.field("XMarkerListContainer1.numItems").value
7
8      for screw_index in range(0, listSize, 1):
9          sceneVar = "scene" + str(screw_index)
10         markerVar = "Marker" + str(screw_index)
11         scene = ctx.network().createModule("ScrewSeparator")
12         scene.name = sceneVar
13         _addedModules.append(scene)
14         marker = ctx.network().createModule("XMarkerAtIndex")
15         marker.name = markerVar
16         _addedModules.append(marker)
17         ctx.connectField(sceneVar + ".input0" , markerVar + ".outputXMarkerList")
18         ctx.connectField(markerVar + ".inputXMarkerList" , "XMarkerListContainer1.outXMarkerList")
19         ctx.field(markerVar + ".index").value = screw_index
20         ctx.module("SoSeparator").addChild(ctx.module(sceneVar).field("output0"))
21     return
22
23 def resetModule():
24     global _addedModules
25     for module in _addedModules:
26         module.remove()
27     _addedModules = []

```

Quellcode 6.5: Python Funktionen zur dynamischen Erstellung eines Mevislab Netzwerks mit einer Visualisierung für jede geplante Schraube im „*XMarkerListContainer*“

Das Script 6.5 wird bei jeder Änderung der Liste von Schraubenpositionen neu ausgeführt. Bei der Ausführung wird innerhalb des „*add_screw*“ Moduls ein neu-

es Mevislab Netzwerk erstellt. Für jeden Listeneintrag wird eine Visualisierung im „*add_screw*“ Modul erzeugt. Hierfür wurde die in der Abbildung 6.13 dargestellte Visualisierung als eigenes Makro Modul „*ScrewSeparator*“ abgespeichert und hier eingebunden. Zur farblichen Unterscheidbarkeit werden die gespeicherten Schrauben in blau dargestellt. Als Eingang des Moduls wird der aktuelle Listeneintrag mit dem „*XMarkerAtIndex*“ Modul ausgelesen, womit der Translations- und Rotationsvektor für die jeweilige Schraube ausgelesen wird. Die Ergebnisse der Schraubenvisualisierung werden in einem „*SoSeparator*“ Anzeigemodul zusammengeführt. Als Ergebnis wird eine 3D-Visualisierung aller gespeicherten Schraubendarstellungen erzeugt, die in der 3D-Visualisierung der Planungssoftware, aus der Abbildung 6.8, eingebunden wird.

Das erstellte „*ScrewSeparator*“ Modul und ein Beispiel-Netzwerk mit 4 visualisierten Schrauben sind in den Abbildungen A.4 und A.5 im Anhang dargestellt.

Mit dieser Implementierung einer Schraubenplanung ist eine einfache und anwenderfreundliche Planung einer beliebigen Anzahl von Schraubenpositionen unter Verwendung von vorhandenen 3D-Schraubenmodellen möglich.

6.2.3 Anbindung der Stereokamera

In vorhergehenden Arbeiten wurde das Stereokamerasystem NDI Spectra am Echtzeitrechner des Robotersystems angebunden. Dieses lässt sich nur über den Echtzeitrechner ansteuern. Für diese Arbeit soll das Nachfolgemodell der Stereokamera (VEGA, NDI) zur intraoperativen Lokalisation der verwendeten optischen Referenzkörper eingesetzt werden. Hierfür soll die Stereokamera direkt an die Planungssoftware angebunden werden, sodass auch ein Betrieb ohne den Echtzeitrechner möglich wird. Die Abbildung 6.16 zeigt die Stereokamera und dessen Messvolumen mit der Lage des Koordinatenursprungs {OTS}.

Die Kamera wird über eine Netzwerkverbindung an dem Planungsrechner angeschlossen. Zur Einbindung der Stereokamera an das Mevislab Framework wurde im Rahmen der betreuten Masterarbeit [2] ein Mevislab Modul „*NDIConnection*“ entwickelt, mit dem die Kamera direkt angesteuert werden kann. Dafür wird im Hintergrund eine Socket Verbindung zur Kamera aufgebaut und die benötigten Befehle zum Betrieb der Kamera ausgeführt. Zur Parametrierung der Kamera und zur

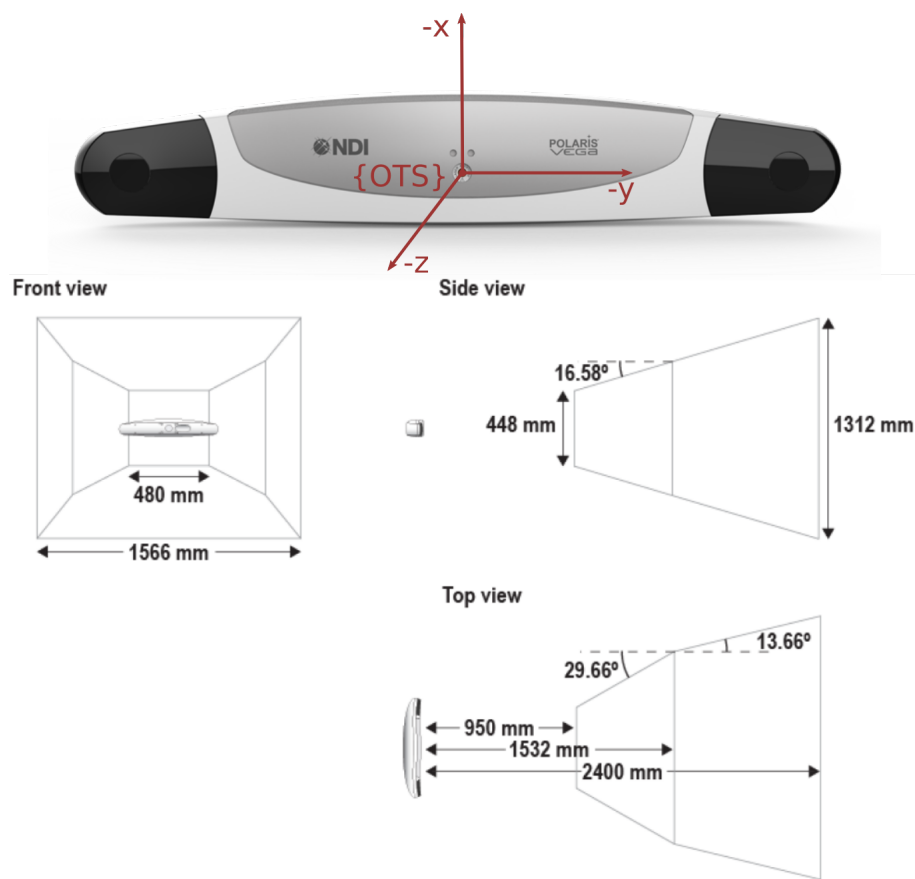


Abbildung 6.16: Stereokamera NDI Polaris VEGA und dessen Messvolumen mit der Lage des Koordinatenursprungs, Quelle: [76]

Auswahl der verwendeten Referenzkörper wurde eine Benutzeroberfläche für das entwickelte Mevislab Modul aufgebaut. Nach Auswahl der an der Kamera eingestellten Netzwerkadresse und des Ports, wird die Verbindung zur Kamera aufgebaut. Anschließend können bis zu 6 Referenzkörper Definitionsdateien geladen und der Tracking Modus aktiviert werden. Die Kamera beobachtet geladene Referenzkörper im Messvolumen und überträgt deren Position und Orientierung relativ zu ihrem Koordinatenursprung {OTS}. Die Abbildung 6.17 zeigt die Benutzeroberfläche zur Parametrierung der Kamera, sowie das Ausgabefenster mit den Positionsdaten der beobachteten Referenzkörper.

Hierbei werden jeweils die Translation vom Koordinatenursprung zum Referenzkörper, sowie die Rotation des Referenzkörpers als Quaternion ausgegeben. Zusätzlich stehen die Positionsdaten auch als Transformationsmatrix zur Verfügung, was die weitere Berechnung vereinfacht.

$${}^{OTS}T_{DRB1\dots6} \quad (6.7)$$

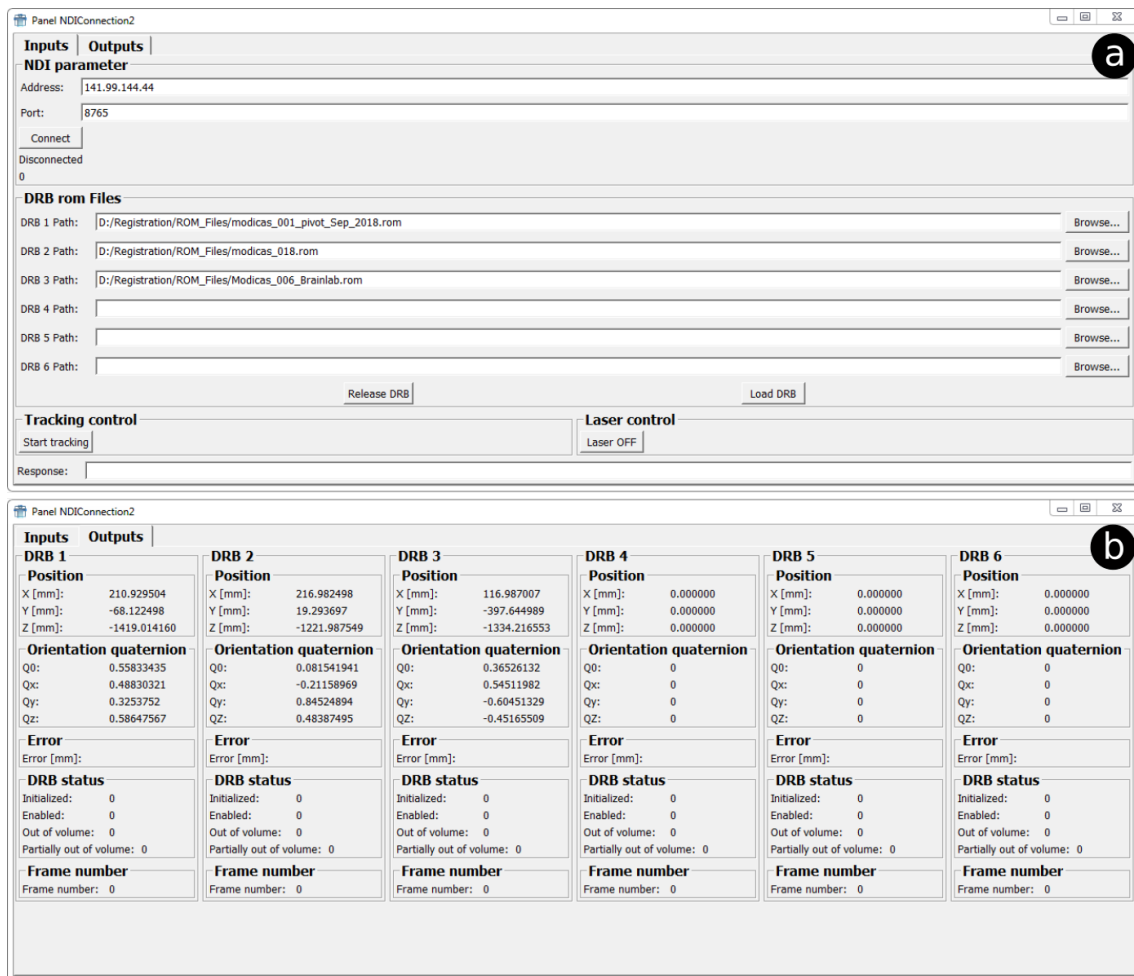


Abbildung 6.17: Benutzeroberfläche des Mevislab Moduls „*NDIConnection*“ zur Parametrierung der Stereokamera (a) und zur Ausgabe der empfangenen Positionsdaten der Referenzkörper (b)

Hierbei gibt die Transformationsmatrix aus 6.7 die Transformation der Referenzkörper (DRB 1-6) im Koordinatensystem der Kamera {OTS} an.

Die vollständige Implementierung und der Aufbau des Mevislab Moduls sind in der Arbeit [2, S.25-43] beschrieben.

6.2.4 Anbindung an das Robotersystem

Die robotergestützte Ausführung zur Implantierung von geplanten Pedikelschrauben soll mit dem am ZESS aufgebauten Robotersystem erfolgen. Zur Anbindung des Robotersystems an eine Planungssoftware wurde in vorhergehenden Arbeiten die HNI-Schnittstelle am Robotersystem konzipiert [88, S.98-104]. Die HNI-Schnittstelle ist mit einem LabVIEW Programm implementiert, welches die Kommunikation zum

Robotersystem steuert. Neben Steuerungsbefehlen für den Roboterarm können hier auch Positionsdaten einer am Echtzeitrechner des Robotersystems angebotenen Stereokamera (Polaris SPECTRA, NDI) übertragen werden. Hierbei handelt es sich um das Vorgängermodell der, in dem Abschnitt 6.2.3 beschriebenen, neu angebotenen Stereokamera.

Wichtig zur Steuerung des Roboters ist der Fahrbefehl „*APPRobMove*“, welcher den Roboter in einem ausgewählten Koordinatensystem bewegt. Zum Erreichen von geplanten Zielpositionen wird der Roboter im Koordinatensystem des Referenzkörpers am Patienten $\{ARB\}$ bewegt. Daher wird auch die Zielposition als Transformationsmatrix ${}^{ARB}\underline{T}_{TTP}$ im Koordinatensystem von $\{ARB\}$ angegeben, wobei $\{TTP\}$ das Koordinatensystem der Werkzeugspitze beschreibt. Da die verwendeten chirurgischen Instrumente zum Einbringen der Pedikelschrauben rotationsymmetrisch sind, wird die Rotationsachse angegeben, die bei dem Robotersystem in der Z-Achse von $\{TTP\}$ liegt. Die Abbildung 6.18 zeigt den Referenzkörper am Roboter sowie das Modell eines Werkzeugs zum Eindrehen der Schrauben mit den eingezeichneten Koordinatensystemen an dem Flansch des Roboters.

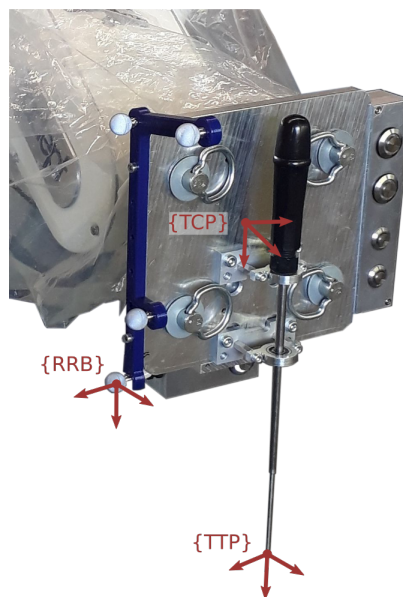


Abbildung 6.18: Referenzkörper am Roboter, sowie das Modell eines Werkzeugs zum Eindrehen der Schrauben mit den Koordinatensystemen an dem Flansch des Roboters

Hierbei beschreibt $\{RRB\}$ das Koordinatensystem des Referenzkörpers am Roboter, $\{TCP\}$ das Koordinatensystem des Mittelpunktes der Werkzeugaufnahme und $\{TTP\}$ das Koordinatensystem der Werkzeugspitze. Um den Roboter unter Verwendung des Navigationssystems bewegen zu können, muss die Position von $\{TCP\}$

relativ zu $\{RRB\}$ bekannt sein. Hierzu wurde in vorhergehenden Arbeiten eine Roboterkalibrierung zum Abgleich der Roboterbewegung mit dem angebrachten Referenzkörper implementiert [26] [54]. Um die Werkzeugspitze an eine Zielposition navigieren zu können, muss die Position von $\{TTP\}$ im Koordinatensystem von $\{RRB\}$ bestimmt werden. Durch den Einsatz eines Vermessungsreferenzkörpers wird die Position der Werkzeugspitze, relativ zu $\{RRB\}$ vermessen und die Transformation ${}^{RRB}\underline{T}_{TTP}$ bestimmt. Diese wird im Robotersystem abgespeichert und kann über die HNI-Schnittstelle ausgelesen werden.

Die Anbindung des Robotersystems an das Mevislab Software Framework erfolgt über ein Mevislab Modul, das im Rahmen einer betreuten Masterarbeit aufgebaut wurde [2]. In dem erstellten Modul wird eine Verbindung zur HNI-Schnittstelle aufgebaut, sodass die benötigten Befehle gesendet und die über die HNI-Schnittstelle zur Verfügung stehenden Daten ausgelesen werden können. Neben dem Auslesen der Transformation ${}^{RRB}\underline{T}_{TTP}$ wird auch das, über das Echtzeitbetriebssystem angebundene, Vorgängermodell der Stereokamera (Polaris SPECTRA, NDI) angesteuert. Damit können verschiedene Referenzkörper ausgewählt und die von der Kamera erfassten Positionsdaten übertragen werden. Analog zum direkt in Mevislab angebotenen NDI VEGA Navigationssystem, werden die erfassten Positionsdaten der Referenzkörper als Transformationsmatrizen im Koordinatenursprung der Kamera übertragen.

$$\begin{aligned} & {}^{OTS}\underline{T}_{ARB} \\ & {}^{OTS}\underline{T}_{PRB} \\ & {}^{OTS}\underline{T}_{RRB} \end{aligned} \tag{6.8}$$

Die in 6.8 dargestellten Transformationen stehen als Parameter im HNI-Modul zur Verfügung und können in Mevislab weiterverarbeitet werden.

Zur Übermittlung von Fahrbefehlen wird die Zielposition als Transformationsmatrix angegeben. Zusätzlich muss ausgewählt werden, in welchem Koordinatensystem diese Zielposition vorliegt. Ausserdem wird die rotationssymmetrische Achse der Zielposition ausgewählt. Über die Schaltfläche „MoveRobot“ wird der Fahrbefehl an den Roboter gesendet. Unter der kontrollierten Freigabe des Anwenders, durch den am Roboter angebrachten Freigabeknopf, wird die Zielposition angefahren. Zur Sicherheit wird die Bewegung unterbrochen, sobald der Freigabeknopf losgelassen wird.

Das Panel des implementierten Mevislab Moduls zur Robotersteuerung ist in der Abbildung 6.19 dargestellt.

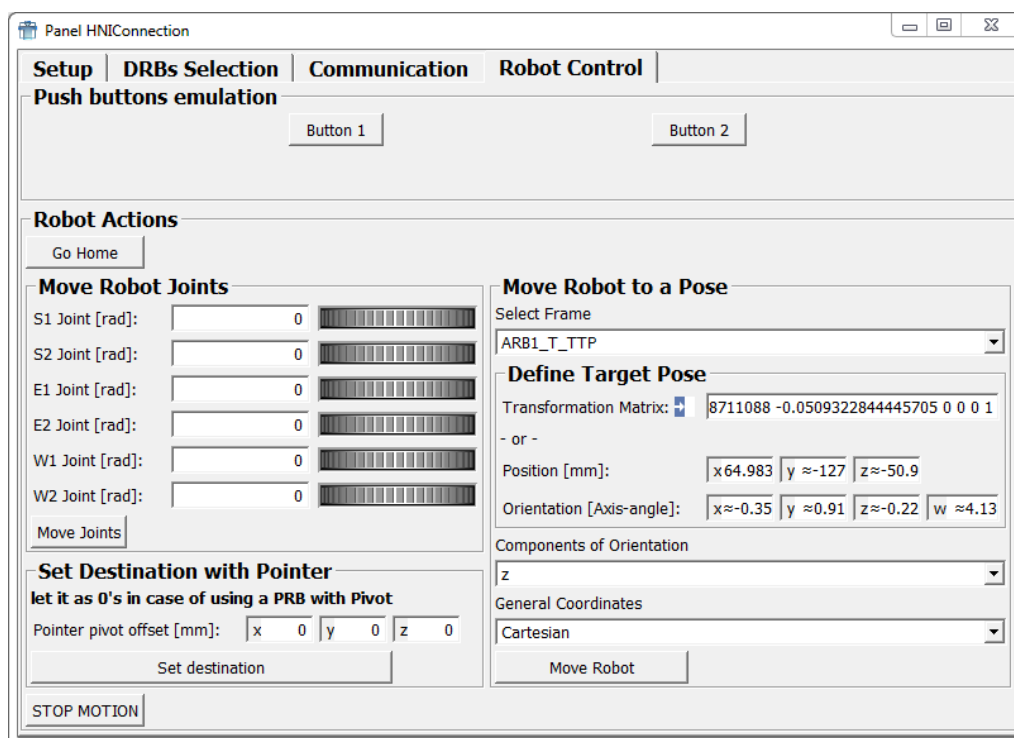


Abbildung 6.19: Mevislab Panel des aufgebauten HNI-Moduls zur Ansteuerung des Roboters [2]

6.2.5 Bestimmung der Zielposition

Die Planung von Pedikelschrauben erfolgt im Koordinatensystem $\{CBVol\}$ des 3D-Bilddatensatzes und gibt die Position einer geplanten Schraube im Koordinatensystem $\{CBVol\}$ an. Bei der Planung liegt der Koordinatenursprung der geplanten Schraube im Zentrum des Schraubenkopfes, wie in der Abbildung 6.13 dargestellt. Um das Instrument zum Einbringen der Schraube an die finale Schraubenposition zu navigieren, muss die Zielposition um die Länge der Schraube, entlang der Z-Achse der Schraube, verschoben werden. Dies erfolgt durch Multiplikation mit der Transformation in der Formel 6.9.

$${}^{screw}T_{screwtip} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & length \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.9)$$

Hierbei beschreibt *length* die Länge der geplanten Pedikelschraube, die vom Anwender gewählt wurde und ebenfalls als Parameter in dem „*XMarkerListContainer*“ der geplanten Schraubenpositionen abgespeichert ist.

Um die Schraubenposition im Koordinatensystem des Patienten {ARB} anzugeben, erfolgt die intraoperative Registrierung der Patientenposition, welche im Abschnitt 6.3 erklärt ist. Damit wird die Transformation von {ARB} im Koordinatensystem der 3D-C-Bogenaufnahme {CBVol} bestimmt. Anschließend lässt sich die Schraubenposition, gemäß der Formel 6.10, im Koordinatensystem {ARB} beschreiben.

$${}^{ARB}\underline{T}_{screw\ tip} = {}^{ARB}\underline{T}_{CBVol} \cdot {}^{CBVol}\underline{T}_{screw} \cdot {}^{screw}\underline{T}_{screw\ tip} \quad (6.10)$$

Die Transformation ${}^{ARB}\underline{T}_{screw\ tip}$ kann als Zielposition an das im Abschnitt 6.2.4 beschriebene HNI-Modul zur Ansteuerung des Roboters übergeben werden um die Zielposition mit dem Robotersystem anzufahren.

6.2.6 Visualisierung von Komponenten im Bilddatensatz

Zur Unterstützung des Chirurgen während eines navigierten Eingriffes und zur Verifikation der angefahrenen Schraubenpositionen kann die Position des Referenzkörpers am Patienten, das Robotersystem und ein chirurgisches Zeigeeinstrument in dem 3D-Bilddatensatz eingeblendet werden.

Hierfür kann, analog zur in der Abbildung 6.13 aufgezeigten Visualisierung einer Schraubenposition, das „*SoWEMRender*“ Modul verwendet werden, um ein 3D-Modell des verwendeten Patientenreferenzkörpers einzublenden. Zur Einblendung des Patientenreferenzkörpers im 3D-Bilddatensatz muss dessen Position im Koordinatensystem von {CBVol} vorliegen. Diese ergibt sich aus der Patientenregistrierung die im Abschnitt 6.3 erklärt wird und liefert die Transformation ${}^{ARB}\underline{T}_{CBVol}$ als Ergebnis.

Eine analoge Implementierung kann auch verwendet werden um ein chirurgisches Zeigeeinstrument in dem 3D-Bilddatensatz einzublenden. Dessen Position wird von dem Navigationssystem erfasst und muss zur Visualisierung in das Koordinatensystem des 3D-Bilddatensatzes transformiert werden.

$${}^{CBVol}\underline{T}_{PRB} = {}^{CBVol}\underline{T}_{ARB} \cdot {}^{OTS}\underline{T}_{ARB}^{-1} \cdot {}^{OTS}\underline{T}_{PRB} \quad (6.11)$$

Hierbei beschreibt ${}^{OTS}\underline{T}_{ARB}^{-1}$ die inverse Pose des Patientenreferenzkörpers im Koordinatensystem des Navigationssystems $\{OTS\}$ und ${}^{OTS}\underline{T}_{PRB}$ die Pose des Zeiginstrumentes, welche vom Navigationssystem erfasst wird.

Zusätzlich kann auch ein 3D-Modell des Robotersystems im 3D-Bilddatensatz dargestellt werden. Hierfür muss neben der vom Navigationssystem erfassten Position des Referenzkörpers am Robotersystem $\{RRB\}$ auch die Position der Werkzeugspitze, relativ zum Referenzkörper am Robotersystem, berücksichtigt werden. Diese wird bei der Werkzeugvermessung bestimmt und ist im Robotersystem hinterlegt ist.

$${}^{CBVol}\underline{T}_{TTP} = {}^{CBVol}\underline{T}_{ARB} \cdot {}^{OTS}\underline{T}_{ARB}^{-1} \cdot {}^{OTS}\underline{T}_{RRB} \cdot {}^{RRB}\underline{T}_{TTP} \quad (6.12)$$

Der Aufbau des Netzwerks zur Visualisierung des Patientenreferenzkörpers, des Zeiginstrumentes sowie der Robotersystemposition ist im Anhang A.2 dargestellt. Die Visualisierung der beschriebenen Komponenten ist in der Abbildung 6.20 dargestellt.

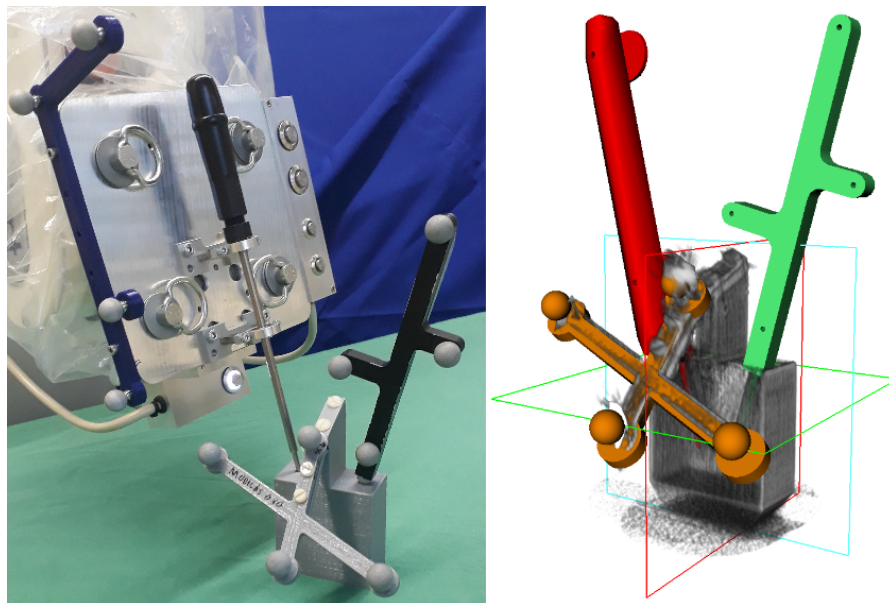


Abbildung 6.20: Die reale Szene (links) und die Visualisierung des am Patienten angebrachten Referenzkörpers $\{ARB\}$ (orange), sowie eines chirurgischen Zeiginstrumentes $\{PRB\}$ (grün) und der aktuellen Robotersystem Position $\{TTP\}$ (rot) im 3D-Bilddatensatz (rechts)

6.2.7 Entwurf einer Benutzeroberfläche

Für den Einsatz der erstellten Planungssoftware wurde eine Benutzeroberfläche erstellt. Mevislab bietet hierfür eine eigene Skript Sprache. Mit dieser können Bedienelemente wie Schaltflächen oder Anzeigeelemente aus den Modulen des Mevislab Netzwerks auf einer Benutzeroberfläche angeordnet werden. Das Ziel der Benutzeroberfläche ist es, den Anwender Schrittweise durch die einzelnen Programmschritte zu führen. Hierfür wurden die, für die einzelnen Schritte der Planung benötigten Bedienfelder, in aufeinander folgenden Registerkarten arrangiert.

In der Registerkarte „*LoadData*“ können die Bilddatensätze und die zugehörigen Dateien, wie das Ergebnis des Registrierungsschrittes, dass für die Navigation benötigt wird, geladen werden. Hierfür wurden die Pfadauswahl und die Ladeschaltfläche des im Abschnitt 6.2.1 vorgestellten „*DirectDicomImport*“ Moduls in der Benutzeroberfläche eingebunden. Die Benutzeroberfläche zum Laden der Bilddatensätze ist in der Abbildung 6.21 dargestellt.

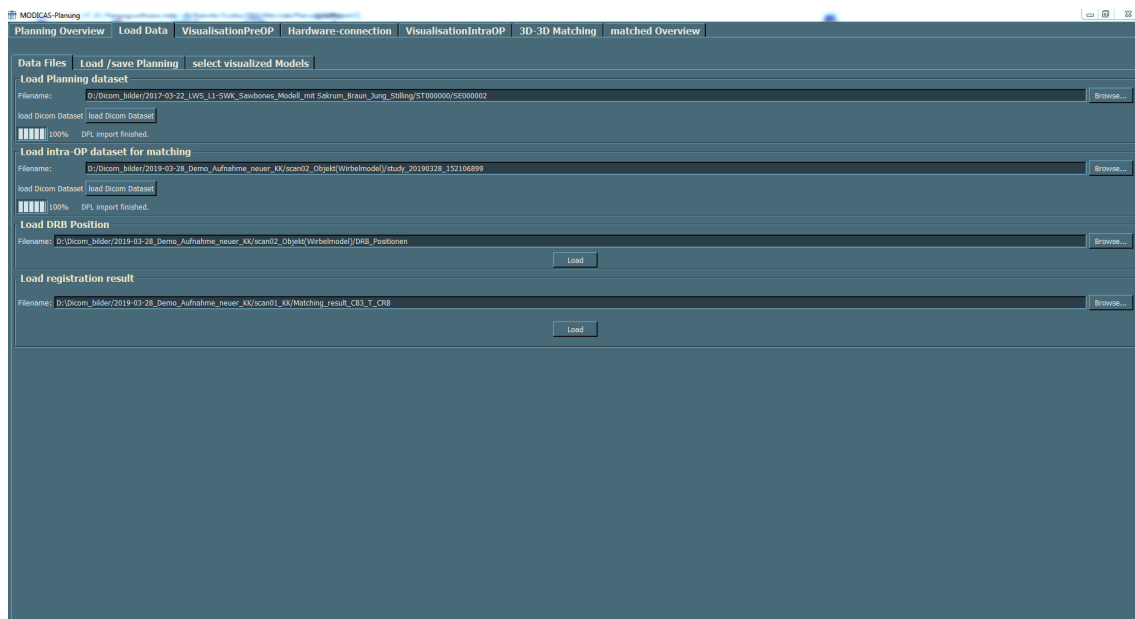


Abbildung 6.21: Benutzeroberfläche zum Laden von Bildaufnahmen

Des Weiteren wurde die Funktionalität um das Speichern und Laden einer erstellten Planung erweitert. Hierfür wird ein „*Settingsmanager*“ Modul verwendet um die bei der Planung erstellten Visualisierungseinstellungen zu speichern. Die geplanten Schraubenpositionen liegen als XMarkerList Objekt vor und können mit einem „*SaveBase*“ Modul in einer Datei abgelegt werden und mit einem „*LoadBase*“ Modul wieder geladen werden. Um eine komfortable Bedienung zu ermöglichen,

wurde die Speicher- und die Ladeprozedur in ein Python Skript ausgelagert, das über eine einzelne Schaltfläche aktiviert werden kann. Im Hintergrund werden dann alle notwendigen Visualisierungseinstellungen und Programmparameter durch den „*SettingsManager*“ in einer Datei gespeichert. Gleichzeitig wird eine weitere Datei mit den geplanten Schraubenpositionen angelegt. Der Quellcode 6.6 zeigt das Python Skript zum Speichern und Laden einer erstellten Planung.

```
1 # Speichert die aktuelle Planung in 3 Verschiedenen Dateien ab
2 def savedata():
3     ctx.field("SavePlanning.save").touch()
4     ctx.field("SaveScrewPositions.save").touch()
5     ctx.field("SaveMatchedScrewPositions.save").touch()
6
7 # Aktiviert die entsprechenden Buttons der Module, die geladen werden müssen (LoadBase für das Laden
8     der geplanten Schrauben, Settingsmanager für die Visualisierungseinstellungen und DicomImport
9     zum Laden der Bilddatensätze)
10 def loaddata():
11     ctx.field("SavePlanning.load").touch()
12     ctx.field("Load_planning_image.dplImport").touch()
13     ctx.field("Load_intraOP_image.dplImport").touch()
14     ctx.field("Load_registration.load").touch()
15     ctx.field("Load_OTS_recording.load").touch()
16     ctx.field("List_of_planned_screws.add").touch()
17     ctx.field("List_of_planned_screws.delete").touch()
18     ctx.field("LoadScrewPositions.load").touch()
19     ctx.field("LoadMatchedScrewPositions.load").touch()
20     ctx.field("SavePlanning.load").touch()
```

Quellcode 6.6: Python Funktionen zum Speichern und Laden einer erstellten Schraubenplanung

Nach dem Laden eines Bilddatensatzes kann im folgenden Reiter die Darstellung des Datensatzes angepasst werden. Hierfür werden die Panelemente, aus dem im Abschnitt 6.2.1 vorgestellten „*RampLUT*“ Modul, auf der Benutzeroberfläche angeordnet. Hier kann die farbliche Darstellung des Bilddatensatzes angepasst werden. Zur detaillierteren Darstellung kann auf der Benutzeroberfläche auch die Visualisierungseinstellung über das „*LUTEditor*“ Modul ausgewählt werden. Die Abbildung 6.22 zeigt die Benutzeroberfläche zur Anpassung der Visualisierung des Bilddatensatzes.

Im Reiter „*PlanningOverview*“ wird zur Übersicht das gerenderte 3D-Volumen des Bilddatensatzes angezeigt, dass mit dem im Abschnitt 6.2.1 beschriebenen „*SoGVR-VolumeRenderer*“ Modul erzeugt wurde. Daneben werden die bereits zur Planung erstellten Schnittbildansichten aus dem Abschnitt 6.2.2 auf der Oberfläche angeordnet. In diesen kann dann die Planung der Pedikelschrauben erfolgen. Die Abbildung 6.23 präsentiert die Benutzeroberfläche zur Pedikelschraubenplanung.

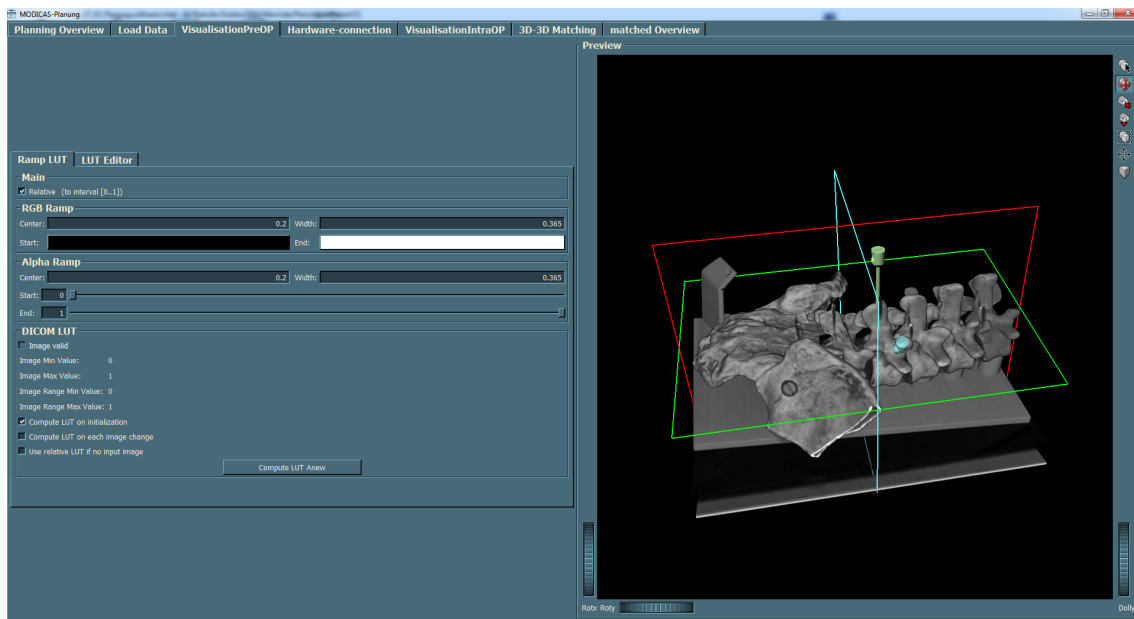


Abbildung 6.22: Benutzeroberfläche zur Einstellung der Visualisierung des 3D-Bild Datensatzes

In der linken Spalte der Benutzeroberfläche ist ein Bereich für Kontrollelemente angeordnet. Im oberen Bereich kann unter dem Titel „*Visualisation*“ die Einblendung der von der Kamera detektierten Komponenten, die im Abschnitt 6.2.6 beschrieben sind, aktiviert werden. Über Schaltflächen kann so die Roboterposition (TTP), ein ausgewähltes Zeigeelement (PRB), der Referenzkörpers am Patienten (ARB) sowie die Sichtbarkeit der geplanten Schrauben ausgewählt werden. Darunter sind Schaltflächen zum Hinzufügen und Entfernen von Pedikelschrauben platziert. Die Betätigung der Schaltfläche führt die im Abschnitt 6.2.2 beschriebene Funktion 6.5 aus, um eine neue Schraube in der Liste der geplanten Schrauben zu erzeugen. Hierüber können neue Pedikelschrauben hinzugefügt und über die im Abschnitt 6.2.2 beschriebenen Platzierfunktionen geplant werden. Hierzu wird nach dem Hinzufügen einer neuen Schraube, die Schraubenlänge ausgewählt und damit auch das entsprechende Schraubenmodell geladen. Durch einen rechten Mausklick in eine der Schnittbildansichten, wird die Schraube an dieser Stelle platziert. Über die Betätigung der linken Maustaste in eine der Schnittbildansichten wird die Schraubenorientierung in Richtung der aktuellen Mauszeigerposition festgelegt.

Zur Feinjustierung wurden zusätzlich noch Schaltflächen auf allen Schnittbildansichten eingeblendet, über deren Betätigung die im Abschnitt 6.2.2 beschriebenen Funktionen 6.3 und 6.4 ausgeführt werden, die die Position und Orientierung der geplanten Schraube justieren.

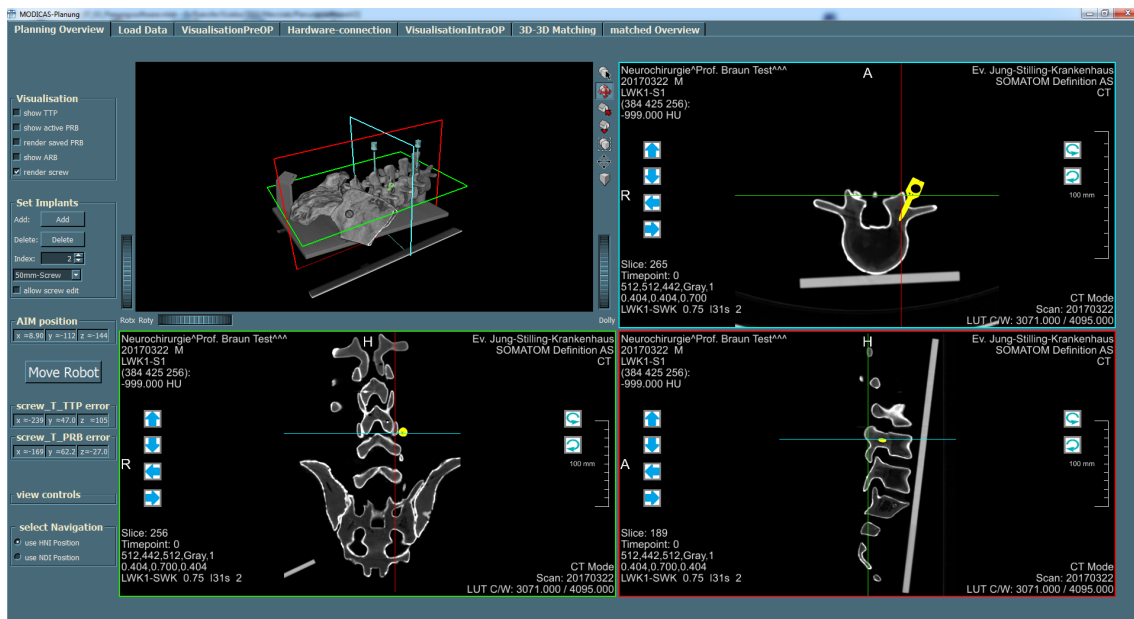


Abbildung 6.23: Benutzeroberfläche mit der Darstellung des Datensatzes zur Planung von Pedikelschrauben

Die Schaltfläche „Move Robot“ sendet einen Fahrbefehl zur aktuell ausgewählte Schraubenposition an das Robotersystem.

Im untersten Bereich der linken Spalte kann ausgewählt werden ob das Vorgängermodell des Navigationssystems (Polaris SPECTRA) verwendet werden soll oder das neue Navigationssystem (Polaris VEGA).

Damit der Fahrbefehl ausgeführt und die Kamera verwendet werden kann, muss im Reiter „Hardware – connection“ das Robotersystem und die Kamera verbunden werden. Hierfür wurde das Bedienpanel des „HNICconnection“ Moduls, das im Abschnitt 6.2.4 beschrieben ist, im oberen Bereich des Reiters angeordnet. Hier kann die Verbindung zum Robotersystem aufgebaut werden und nachfolgend die verwendeten Referenzkörper ausgewählt werden. Danach kann der „MoveRobot“ Befehl verwendet werden.

Die neu angebundene Stereokamera (NDI VEGA) kann im unteren Bereich des Reiters angesteuert werden. Hierzu wurden die relevanten Bedienelemente des im Abschnitt 6.2.3 beschriebenen Moduls „NDICconnection“ auf der Benutzeroberfläche angeordnet. Nachdem die Verbindung zum Kamerasystem aufgebaut ist, können hier die verwendeten Referenzkörper ausgewählt und das Tracking gestartet werden. Zusätzlich werden die aktuellen Positionsdaten und die Sichtbarkeit der Referenzkörper auf der Benutzeroberfläche ausgegeben. Die Abbildung 6.24 zeigt die

Benutzeroberfläche mit der Anbindung an das Robotersystem und die Ansteuerung des neu angebundenes Kamerasystems.

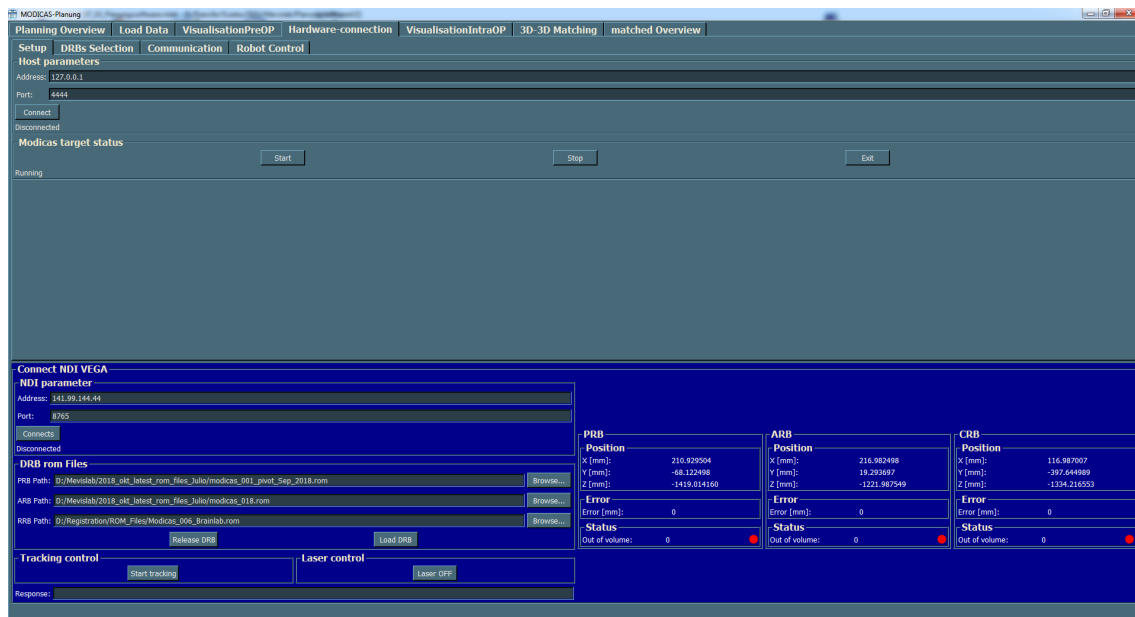


Abbildung 6.24: Benutzeroberfläche zur Anbindung an das Robotersystem und zur Ansteuerung des neu angebundenes Navigationssystems

6.3 Registrierung der intraoperativen Patientenposition

Intraoperativ werden zur Bildgebung 3D C-Bögen, O-Arme oder auch kleine CT Geräte eingesetzt. Um ein chirurgisches Assistenzsystem intraoperativ navigieren zu können und so eine geplante Implantatposition oder einen geplanten Zugangspfad zu erreichen, muss die Patientenposition mit dem Bildvolumen der Bildgebungsmodalität registriert werden. Ein 3D-Lokalisiersystem wird eingesetzt um die Lage der intraoperativen Bildgebungsmodalität, sowie der eingesetzten chirurgischen Instrumente zu überwachen. Die Lage des Patienten wird über einen am Patienten angebrachten Referenzkörper {ARB} vom 3D-Lokalisiersystem erfasst.

Konventionelle Methoden zur Registrierung, wie eine Oberflächenregistrierung oder ein Paired-Point Matching von bekannten Landmarken, sind für minimalinvasive Eingriffe, wie der vorgestellten minimalinvasiven Platzierung von Pedikelschraubenplatzierung, ungeeignet. Eine Oberflächenregistrierung bringt eigene Herausforderungen mit sich, da die Haut flexibel ist und sich leicht verschieben kann, wo

durch das Registrierungsergebnis ungültig wird. Für die Registrierung von bekannten Landmarken sind zusätzliche Inzisionen notwendig, um rigide Strukturen zu erreichen.

Zum Zeitpunkt der eigenen Entwicklung gab es Lösungen einzelner Hersteller, die eine Patientenregistrierung ermöglichen. Ziehm bietet hierfür die eigene Naviport Schnittstelle, die für aktuelle Ziehm 3D C-Bögen verfügbar ist [102]. Hierüber können Positionsdaten eines Navigationssystems empfangen werden. Hierfür werden am Bilddetektor angebrachte Mulden mit einem navigierten Zeigeelement angetastet und so die Position der Bildgebungsmodalität zur Registrierung erfasst. In der neuesten Gerätegeneration können auch spezielle Referenzkörper am C-Bogen detektiert werden, womit die Registrierung bestimmt wird [102].

Auch der Hersteller Brainlab hat für die eigenen Navigationslösungen Verfahren zur Registrierung von Bildgebungsmodalitäten entwickelt. Unter anderem ist der Einsatz eines entwickelten XSpot Markers patentiert, der zur Registrierung vor den Bilddetektor des C-Bogens gehalten werden muss und so ein Punkteraster in der Aufnahme erzeugt, über die die Registrierung bestimmt wird [22]. Nach dem aktuellen Stand wurden die Verfahren erheblich erweitert, sodass Registrierungsverfahren für unterschiedliche Anwendungen angeboten werden [22].

Diese Verfahren haben den Nachteil, dass Sie nur mit den Komponenten der Hersteller betrieben werden können. So ist die Naviport Schnittstelle nur für aktuelle Ziehm Bildgebungsmodalitäten verfügbar und das Brainlab Navigationssystem kann nur mit den Brainlab Navigationskomponenten betrieben werden.

Als eigener Beitrag soll daher eine Methode implementiert werden, die eine herstellerunabhängige Registrierung von beliebigen intraoperativen 3D-Bildgebungsmodalitäten ermöglicht [40].

Zur Bestimmung der Registrierung wird ein zweischrittiges Verfahren vorgestellt. Diese Schritte entsprechen den ersten beiden Schritten des intraoperativen Workflows, der in der Abbildung 6.1 vorgestellt wurde. Der erste Schritt beinhaltet die Registrierungsprozedur für die Bildgebungsmodalität. Dieser Schritt kann im Operationsraum vor dem chirurgischen Eingriff ausgeführt werden und bleibt gültig, solange das Messvolumen der Bildgebungsmodalität relativ zu dem angebrachten Referenzkörper unverändert bleibt.

Der zweite Schritt erfolgt während des Eingriffes und beinhaltet die Akquise eines registrierten Bilddatensatzes des Patienten. Mit dem Ergebnis des ersten Schrittes

wird die Verknüpfung zwischen dem Bildvolumen $\{CBvol\}$ und der Lage des Patienten, über den am Patienten angebrachten Referenzkörper $\{ARB\}$, bestimmt.

Ziel der Patientenregistrierung ist die Bestimmung der Transformation des Patientenreferenzkörpers im Bildvolumen der Bildgebungsmodalität.

$${}^{CBVol}T_{ARB} \quad (6.13)$$

Implementiert und getestet wurde die Methode mit dem Navigationssystem NDI VEGA und dem Bildgebungsgerät Ziehm Vario 3D.

6.3.1 Einsatz eines Kalibrierkörpers zur Registrierung

Im ersten Schritt wird die Position des Bildvolumens $\{CBVol\}$ relativ zu dem an der Bildgebungsmodalität angebrachten Referenzkörper $\{CRB\}$ bestimmt. Hierfür wird ein Kalibrierkörper eingesetzt, der sowohl röntgenintransparente Marker, als auch einen Referenzkörper für das Navigationssystem enthält $\{KRB\}$. Dieser besteht aus einer Plexiglasplatte mit Haltepunkten für röntgenintransparente Markerkugeln mit einem Durchmesser von 8 mm . Die Abbildung 6.25 zeigt den verwendeten Kalibrierkörper.

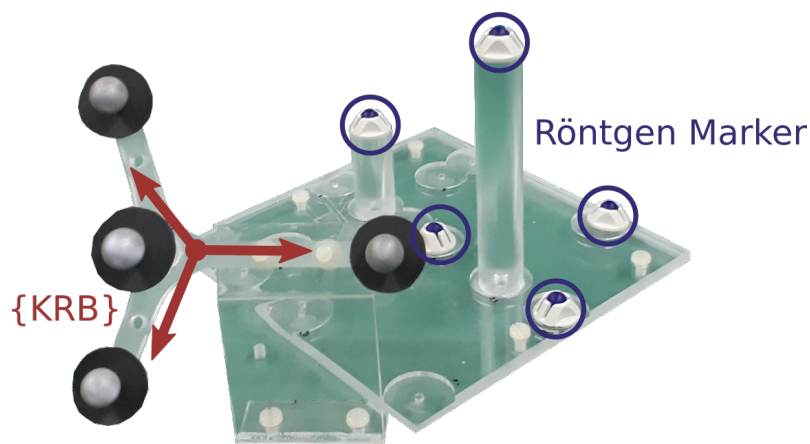


Abbildung 6.25: Zur Registrierung verwendeter Kalibrierkörper mit röntgenintransparenten Markerkugeln (blau) sowie einem Referenzkörper für das optische 3D-Lokalisiersystem $\{KRB\}$

Die Position der röntgenintransparenten Markerkugeln, relativ zu dem Referenzkörper $\{KRB\}$ ist durch die Geometrie festgelegt und kann aus dem Modell des Kalibrierkörpers bestimmt werden. Alternativ dazu können die verwendeten Markerku-

geln über die vorhandene Steckverbindung durch antastbare Mulden ersetzt werden, die eine optische Vermessung des Kalbrierkörpers ermöglichen. Dazu werden die Mulden mit einem Zeigeeinstrument {PRB} angetastet und die Position der Markerkugeln relativ zu {KRB} gespeichert. Die Position der angetasteten Mulde entspricht dem Mittelpunkt einer angebrachten Markerkugel.

$${}^{KRB}\underline{Marker}_{PRB}(i) = {}^{OTS}\underline{T}_{KRB}^{-1} \cdot {}^{OTS}\underline{T}_{PRB}(i) \quad (6.14)$$

Hierbei liefert das optische Lokalisiersystem die Transformationen ${}^{OTS}\underline{T}_{KRB}$ und ${}^{OTS}\underline{T}_{PRB}$. Der Index (i) bezeichnet die verschiedenen Markerkugelpositionen. Da lediglich die Position der Marker von Bedeutung ist, werden die Positionsvektoren der Marker im Koordinatensystem von {KRB} in der Punktliste 6.15 hinterlegt.

$${}^{KRB}\vec{Markerpos}(i) \quad (6.15)$$

Mit dem in der Abbildung 6.26 skizzierten Setup wird eine 3D Aufnahme des Kalibrierkörpers mit der Bildgebungsmodalität aufgenommen.

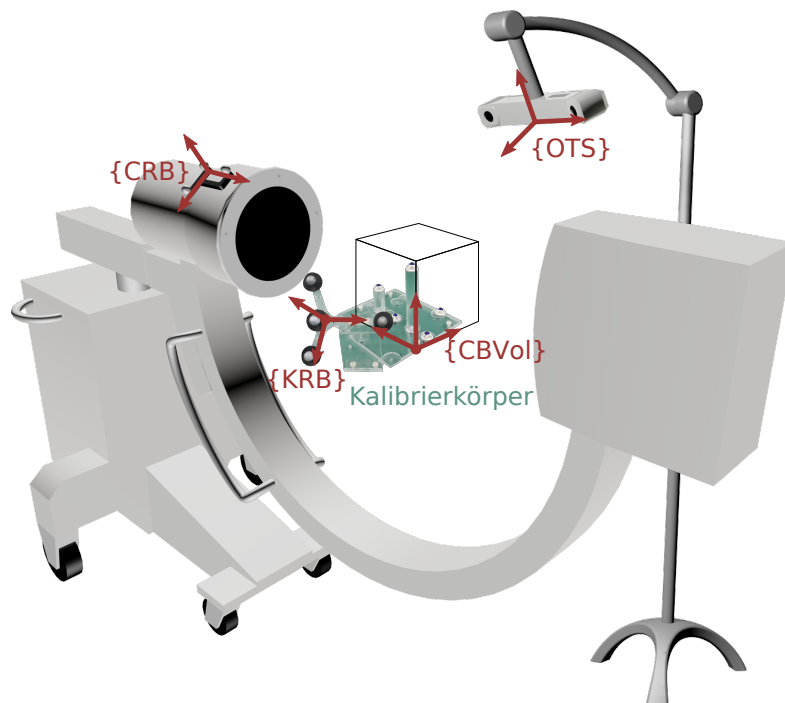


Abbildung 6.26: Setup zur Erfassung des verwendeten Kalibrierkörpers im Bildvolumen der Bildgebungsmodalität

Zum Zeitpunkt der Bildaufnahme wird die Position des Kalibrierkörpers relativ zu dem Referenzkörper der Bildgebungsmodalität vom Lokalisiersystem erfasst.

$${}^{CRB}\underline{T}_{KRB} = {}^{OTS}\underline{T}_{CRB}^{-1} \cdot {}^{OTS}\underline{T}_{KRB} \quad (6.16)$$

Hierdurch können die Positionen der Markerpunkte aus der Formel 6.15 im Koordinatensystem vom CRB beschrieben werden.

$${}^{CRB}\vec{Markerpos}(i) = {}^{CRB}\underline{T}_{KRB} \cdot {}^{KRB}\vec{Markerpos}(i) \quad (6.17)$$

6.3.2 Detektion des Kalibrierkörpers im Volumen der Bildgebungsmodalität

Zur Detektion des Kalibrierkörpers im Volumen der Bildgebungsmodalität wird eine Aufnahme mit dem in der Abbildung 6.26 skizzierten Setup erfasst. Anschließend werden die Kugelzentren der röntgenintransparenten Marker im Bilddatensatz bestimmt. Die Abbildung 6.27 skizziert den implementierten Programmablauf.

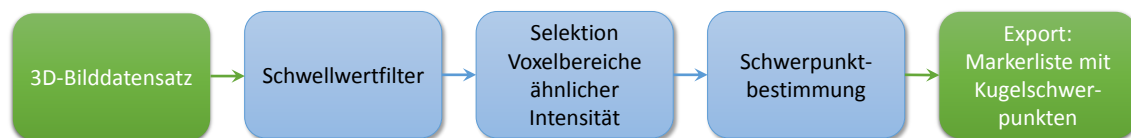


Abbildung 6.27: Programmablauf zur Detektion der röntgenintransparenten Kugelmarkerzentren

Auf Grund der Röntgenintransparenz weisen die Markerkugeln im 3D Bilddatensatz hohe Intensitätswerte auf. Im ersten Schritt wird ein Schwellwertfilter angewendet um die Marker von dem Hintergrund zu separieren. Hierdurch werden auch die Artefakte um die Kugeln, die aus der Aufnahme stammen, erfolgreich separiert. Die Abbildung 6.28 visualisiert die 3D Aufnahme des Kalibrierkörpers sowie die Anwendung des Schwellwertfilters.

Die Abbildung 6.29 zeigt die Implementierung in Mevislab. Im zweiten Schritt werden die Bildvoxel analysiert und benachbarte Voxel mit ähnlichen Intensitätswerten als zusammenhängender Bereich selektiert. Hierfür wird in dem Modul „*Compute-ConnectedComponents*“ eine 3D-Nachbarschaftsoperation durchgeführt, in der die Intensitätswerte eines jeden Voxels mit den Intensitätswerten der 6 benachbarten Voxel verglichen werden. Um eine benutzergesteuerte Auswahl der Markerpunkte zu

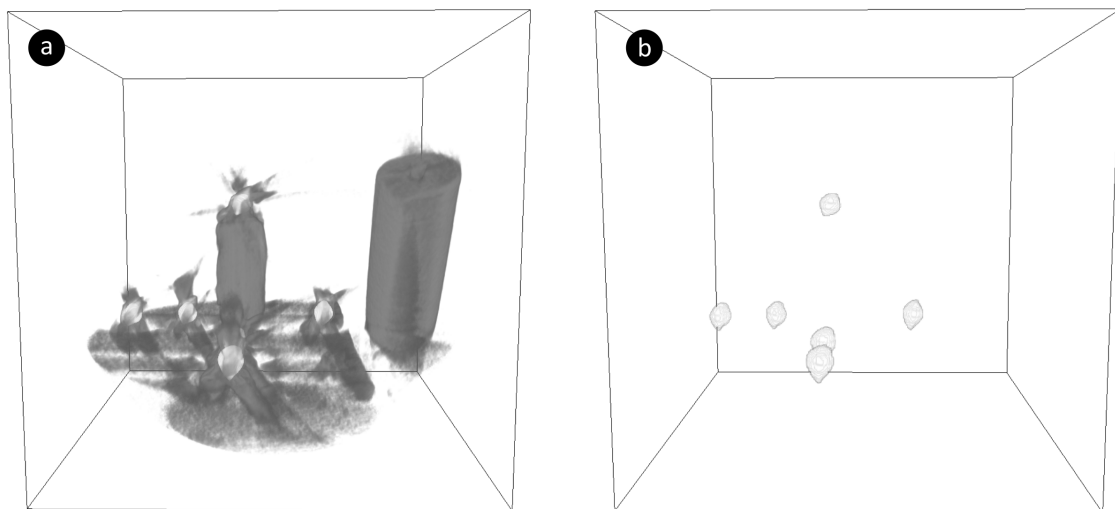


Abbildung 6.28: Visualisierung der 3D Aufnahme des Kalibrierkörpers mit heller Darstellung hoher Intensitätswerte(a), sowie das Ergebnis des ersten Schrittes: Anwendung des Schwellwertfilters (b)

ermöglichen, wurde eine Schnittebenenansicht des Datensatzes angelegt, in der vom Anwender die verwendeten Marker ausgewählt werden können. Diese werden im „So-View2DMarkerEditor“ Modul angelegt. Über das „*FilterConnectedComponents*“ Modul werden die vom Benutzer ausgewählten Marker gefiltert und im Datensatz selektiert. Die selektierten Marker werden dann im „*ConnectedComponentsToImage*“ Modul in einen Bilddatensatz geschrieben, der angezeigt und weiterverarbeitet werden kann. Hierin enthalten sind gefilterte, zusammenhängende Bereiche der Markerpunkte.

Im dritten Schritt wird durch das Modul „*calculatemarker_center*“ der Mittelpunkt eines jeden zusammenhängenden Bereiches bestimmt. Die Sortierung der Markerpunkte, in derselben Reihenfolge wie diese vom Benutzer ausgewählt wurden, erfolgt in dem eigens dafür erstellten „*sort_markers*“ Modul. Dieses führt das Python Skript 6.7 aus, und sortiert die detektierten Markerzentren.

```
1 from mevis import *
2 import numpy
3
4 def sortmarkers():
5     #get input markerlists
6     selectedmarkerList = ctx.field("inselectedmarkers").object()
7     detectedmarkerList = ctx.field("indetectedmarkers").object()
8     outmarker = ctx.field("sortedmarkers.outputXMarkerList").object()
9     outmarker.clear()
10    selectedpos = selectedmarkerList.getMarkerPositions3DAsNumPyArray()
11    detectedpos = detectedmarkerList.getMarkerPositions3DAsNumPyArray()
12    sortedpos = numpy.zeros((selectedpos.shape))
13    #find min deviation between markerlists
```

```

14 for i in range(selectedpos.shape[0]):
15     error = []
16     for j in range(detectedpos.shape[0]):
17         error.append((detectedpos[j][0]-selectedpos[i][0])*(detectedpos[j][0]-selectedpos[i][0])+
18                     detectedpos[j][1]-selectedpos[i][1])*(detectedpos[j][1]-selectedpos[i][1])+
19                     (detectedpos[j][2]-selectedpos[i][2])*(detectedpos[j][2]-selectedpos[i][2]))
20     min_error_index=error.index(min(error))
21     sortedpos[i,:]=detectedpos[min_error_index]
22 #write closest element to output
23 outmarker_=outmarker.addMarkersFromNumPyArray(sortedpos)
24 MLAB.log("points sorted")
25 ctx.field("sortedmarkers.outputXMarkerList").touch()

```

Quellcode 6.7: Python Funktionen zur Sortierung der detektierten Markerzentren

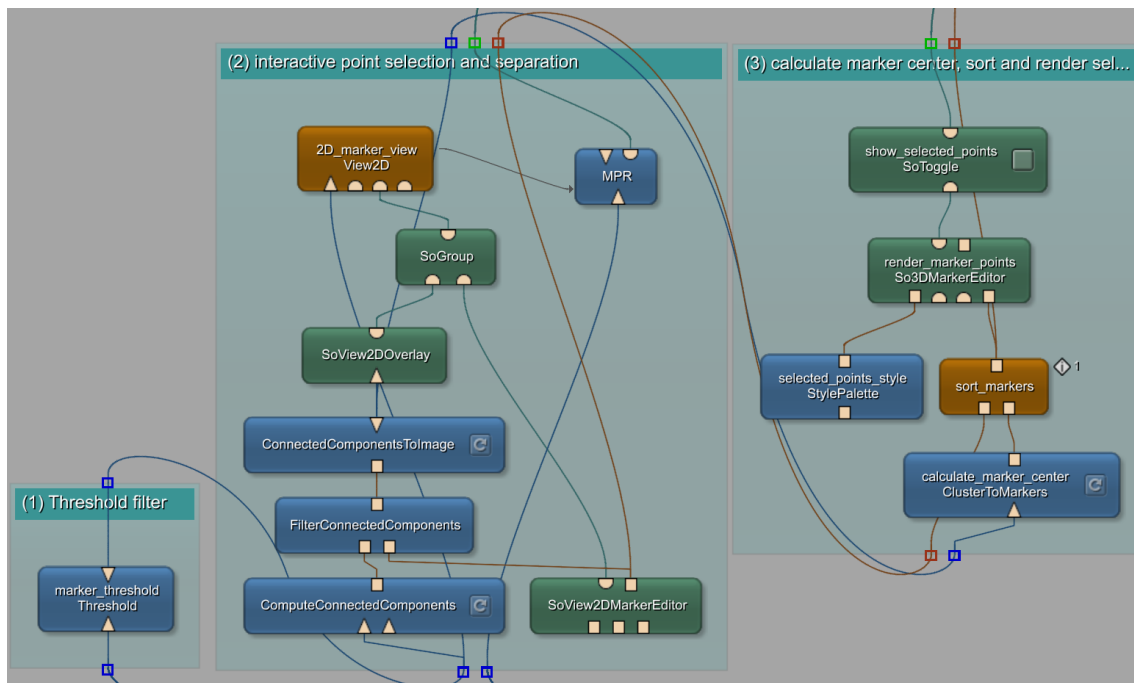


Abbildung 6.29: Mevislab Programm zur Detektion der röntgenintransparenten Markerpunkte in der 3D-Aufnahme des Kalibrierkörpers

Die Abbildung 6.30 (a) zeigt die detektierten zusammenhängenden Bereiche der Markerpunkte im Bilddatensatz des Kalibrierkörpers, die im Schritt 2 selektiert wurden. In der Abbildung 6.30 (b) sind die Markerzentren den detektierten Bereichen überlagert dargestellt.

Als Ergebnis stehen die detektierten Markerpositionen als sortierte Markerliste am Ausgang des „*sort_marker*“ Moduls zur Weiterverarbeitung zur Verfügung. Hierbei sind die Markerpositionen aus der Formel 6.18 im Koordinatensystem der Bildgebungsmodalität {CBVol} bestimmt.

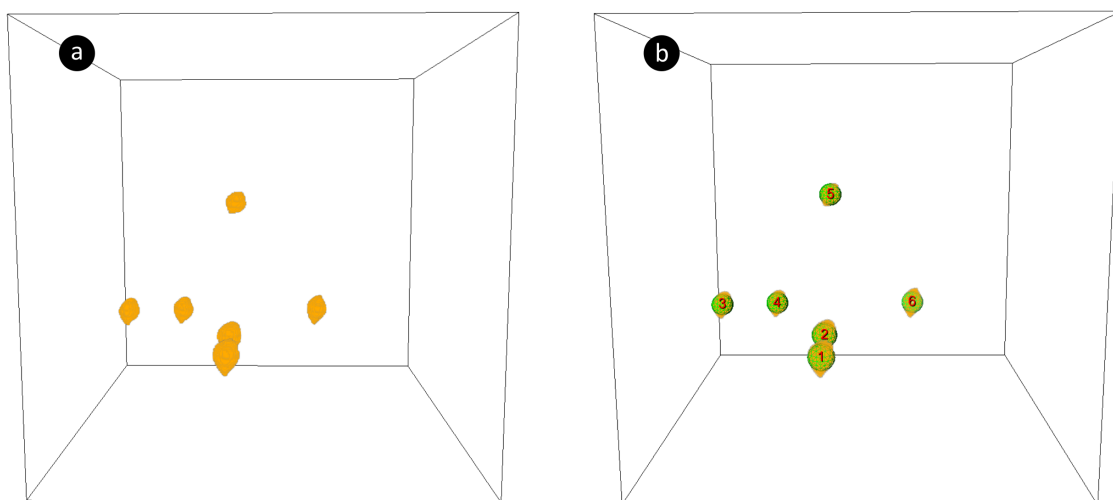


Abbildung 6.30: Die im 2. Schritt detektierten Markerbereiche im Bilddatensatz der Kalibrierkörperaufnahme (a), sowie die im 3. Schritt bestimmten Markerzentren die als grüne Kugeln dargestellt sind (b)

$${}^{CBVol} \text{Markerpos}(i) \quad (6.18)$$

6.3.3 Registrierung der detektierten Markerpositionen

Nach der erfolgreichen Detektion der Markerkugeln im Volumen der Bildgebungsmodalität werden die detektierten Markerpositionen ${}^{CBVol} \text{Markerpos}(i)$ mit den aus der Geometrie bekannten Markerkugelposition ${}^{CRB} \text{Markerpos}(i)$ durch eine Registrierungsprozedur überlagert. Hierfür wird diejenige Transformation gesucht, die die detektierten Markerpunkte auf die aus der Geometrie bekannten Markerpunkte so abbildet, dass die Abweichung minimal ist. Aufgrund von Messungenauigkeiten und einer limitierten Genauigkeit der Detektion der Markerkugeln im Bildvolumen, wird die Abbildung immer einen Restfehler $\vec{\epsilon}(i)$ behalten.

$${}^{CBVol} \text{Markerpos}(i) = {}^{CBVol} \underline{T}_{CRB} \cdot {}^{CRB} \text{Markerpos}(i) + \vec{\epsilon}(i) \quad (6.19)$$

Um den Fehler zu minimieren, ergibt sich das in der Formel 6.20 dargestellte Minimierungsproblem. Hierbei gibt N die Anzahl von Markerpunkten an.

$$\min \sum_{i=1}^N \left\| {}^{CBVol} \text{Markerpos}(i) - {}^{CBVol} \underline{T}_{CRB} \cdot {}^{CRB} \text{Markerpos}(i) \right\|^2 \quad (6.20)$$

Das Ergebnis des Minimierungsproblems liefert die Transformation zwischen dem Bildvolumen und dem Referenzkörper an der Bildgebungsmodalität ${}^{CBV}{}^{ol}\underline{T}_{CRB}$. Nachfolgend werden Verfahren zur punktbasierten Registrierung vorgestellt, mit denen die gesuchte Transformation bestimmt werden kann.

Verfahren zur punktbasierten Registrierung

In einer vorhergehenden Arbeit wurden, basierend auf den Ergebnissen von [31], bereits verschiedene Verfahren zur punktbasierten Registrierung beschrieben, implementiert und verglichen [55]. Ziel der Ansätze ist, die Transformation zwischen zwei Punktwolken \vec{p}_i und \vec{q}_i zu finden, die die kleinsten Fehlerquadrate liefert. Hierzu wird die Transformation in dem Minimierungsproblem als Skalierungsfaktor s , Translation \vec{t} und Rotation \underline{R} ausgedrückt.

$$\min \sum_{i=1}^N \|\vec{p}_i - sR\vec{q}_i - \vec{t}\|^2 \quad (6.21)$$

Für die in dieser Arbeit vorliegenden Punktdatensätze kann die Skalierung vernachlässigt werden, da es sich hier physikalisch um dieselben Punkte handelt, die lediglich rotiert und verschoben, aber nicht skaliert zueinander vorliegen.

Der erste Ansatz beruht auf dem Einsatz der Singulärwertzerlegung [4]. Hierfür werden beide Punktdatensätze in je eine Matrix geschrieben. Zu diesen beiden Matrizen wird eine orthogonale Matrix gefunden, die beide möglichst optimal aufeinander abbildet.

Ein weiterer Ansatz besteht in dem Einsatz von orthonormalen Matrizen [50]. Die Translation und die Skalierung kann einfach bestimmt werden, wenn die Rotation bekannt ist. Zur Bestimmung der Rotation wird je ein Koordinatensystem in jedem Punktdatensatz aufgespannt. Die Lösung für die Rotation ist dann die Drehmatrix, die beide Koordinatensysteme optimal aufeinander abbildet.

Einen ähnlichen Ansatz beschreibt die Methode zur Bestimmung der Drehmatrix durch den Einsatz von Quaternionen [49]. Durch den Einsatz von Quaternionen zur Beschreibung der Rotation lässt sich das Problem umformen. Als Lösung ergibt sich eine Einheitsquaternion, die dem Eigenvektor des größten Eigenwertes einer symmetrischen Matrix entspricht.

Für die vollständige Beschreibung der Methoden wird auf die originalen Publikationen [4], [50], [49] verwiesen.

Bei dem Vergleich der Methoden wurde die Genauigkeit und die Zuverlässigkeit analysiert. Dabei stellte sich der Einsatz der Methode der Einheitsquaternionen als besonders empfehlenswert heraus, da hier die höchste Stabilität bei einer geringen Anzahl an Punktpaaren erreicht wurde. Des Weiteren verringerte sich der Fehler bei der Verwendung zusätzlicher Punktpaare [55].

Als Ergebnis der vorhergehenden Arbeit [55] liegt die Implementierung des Einheitsquaternionen Algorithmus als C++ Klasse vor, die als Eingabe zwei Punktdatensätze erwartet und als Ausgabe eine Rotationsmatrix und den Translationsvektor liefert.

Um den Algorithmus in die erstellte Software einzubinden, wurde ein eigenes Mevislab Modul für die Registrierung von Punktdatensätzen erstellt. Der vollständige Quellcode des Moduls ist im Anhang B.1 dargestellt. Die Abbildung 6.31 zeigt den Einsatz des erstellten „*PCL_matching*“ Moduls in dem Mevislab Netzwerk.

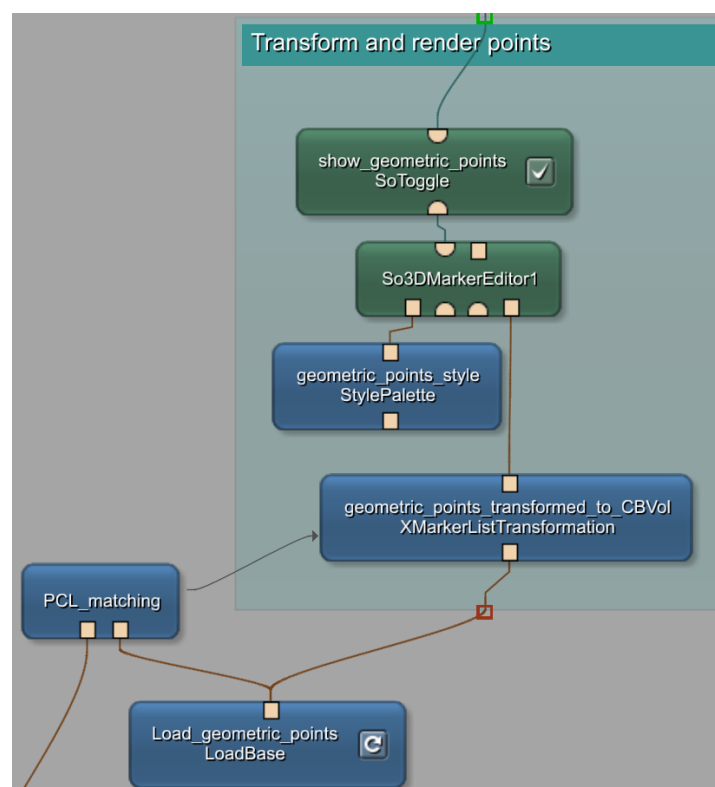


Abbildung 6.31: Einsatz des erstellten „*PCL_matching*“ Moduls zur Registrierung der erkannten Markerpunkte mit den aus der Geometrie bekannten Markerpunkten

Am Eingang des Moduls werden die im Abschnitt 6.3.2 detektierten Markerpunkte ${}^{CBV}ol\vec{Markerpos}(i)$ übergeben. Die Zielpunkte bilden die aus der Geometrie bekannten Markerpunkte ${}^{CRB}\vec{Markerpos}(i)$, die gemäß der Formel 6.17 im Koordinatensys-

tem von $\{CRB\}$ abgespeichert sind und ebenfalls an das „PCL_matching“ Modul übergeben werden. Durch den im Quellcode B.1(Zeile35) verwendeten „Notification handler“ wird die gesuchte Ergebnis Transformation ${}^{CBVol}T_{CRB}$ berechnet, sobald sich die Eingangsparameter des Moduls ändern, also die Punktdatensätze geladen werden und am Eingang anliegen. Die Tabelle 6.1 zeigt die im Bildvolumen detektierten Markerpunkte ${}^{CBVol}Markerpos(i)$ und die aus der Geometrie bekannten Markerposition ${}^{CRB}Markerpos(i)$, die für die Registrierung verwendet wurden.

Markernr	${}^{CBVol}Markerpos(i)[mm]$	${}^{CRB}Markerpos(i)[mm]$
1	(-44.82 / 36.28 / 48.24)	(43.80 / -181.83 / 29.26)
2	(-75.05 / 34.66 / 108.22)	(55.21 / -198.04 / 93.14)
3	(-75.33 / 35.35 / 88.13)	(47.53 / -186.92 / 79.12)
4	(-75.11 / 76.78 / 48.72)	(63.91 / -140.7 / 50.21)
5	(-95.37 / 36.10 / 48.43)	(23.18 / -152.32 / 63.80)
6	(-115.95 / 35.56 / 48.79)	(14.69 / -140.58 / 77.50)

Tabelle 6.1: Die zur Registrierung verwendeten Markerpunkte ${}^{CBVol}Markerpos(i)$ und ${}^{CRB}Markerpos(i)$

Nach der Ausführung der Registrierungsberechnung steht die in der Formel 6.22 angegebene Ergebnis Transformation als Parameter in dem „PCL_matching“ Modul und wird von dort an das „XMarkerListTransformation“ Modul übergeben.

$${}^{CBVol}T_{CRB} = \begin{bmatrix} 0.41759 & -0.59312 & -0.68834 & -151.36657 \\ 0.80914 & 0.58740 & -0.01526 & 107.91283 \\ 0.41339 & -0.55059 & 0.72522 & -91.33285 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.22)$$

Hiermit werden die Markerpunkte ${}^{CRB}Markerpos(i)$ mit der Ergebnis Transformation aus der Formel 6.22, wie in der Formel 6.23 angegeben, in das Koordinatensystem des Bildvolumens $\{CBVol\}$ transformiert.

$${}^{CBVol}transformed_Markerpos(i) = {}^{CBVol}T_{CRB} \cdot {}^{CRB}Markerpos(i) \quad (6.23)$$

Zur Visualisierung werden an den Markerpunkten Kugelobjekte gerendert, die im Bilddatensatz eingeblendet werden können. Die Abbildung 6.32 zeigt die, gemäß

der Formel 6.23 transformierten Markerpunkte, in dem Bilddatensatz des Kalibrierkörpers (rot). Hier ist eine gute Überlagerung zu den detektierten Markerpunkten (grün) erkennbar.

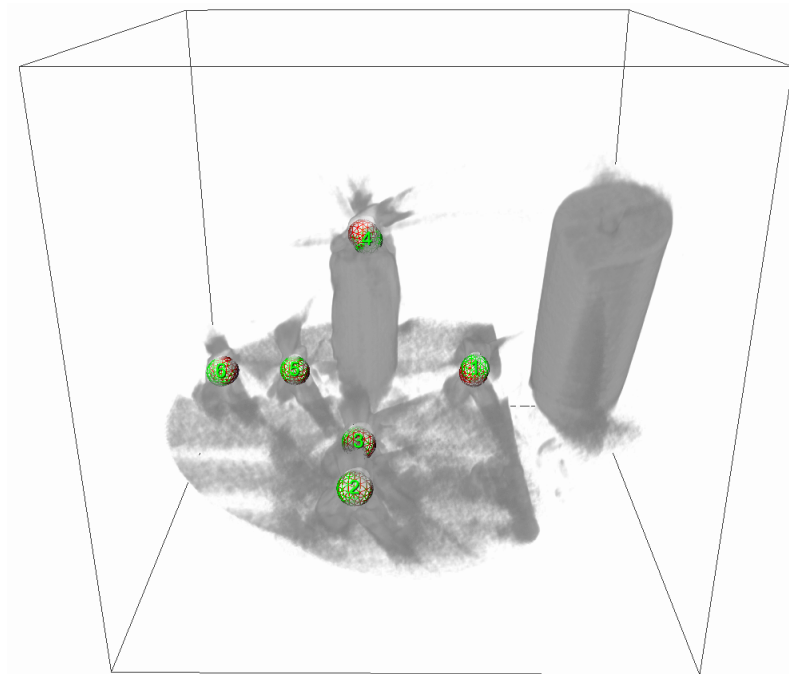


Abbildung 6.32: Kalibrierkörperaufnahme mit den detektierten Markerpunkten (grün) und die aus der Geometrie bekannten Markerpunkten (rot), die nach der erfolgreichen Registrierung überlagert sind

Zur Untersuchung des Ergebnisses kann die Abweichung zwischen den detektierten Markerpunkten (grün) und den gemäß der Formel 6.23 transformierten Markerpunkten bestimmt werden.

$$\vec{\epsilon}(i) = {}^{CBVol} \text{Markerpos}(i) - {}^{CBVol} \text{transformed_Markerpos}(i) \quad (6.24)$$

Die Tabelle 6.2 zeigt die in der Formel 6.24 bestimmten Abweichungen der einzelnen Markerpunkte.

Die Interpretation der Abweichungen lässt keine unmittelbaren Schlüsse auf mögliche Abweichungen bei der Verwendung des Registrierungsergebnisses zu, da die Zieltrajektorien der geplanten Schraubenpositionen nicht entlang der Koordinatenachsen von $\{CBVol\}$ liegen. Als Maß für die durchgeführte Registrierung wird eine gemittelte Abweichung $\bar{\epsilon}$ bestimmt und in der Formel 6.25 angegeben.

Markernr (i)	$\vec{\epsilon}(i)[mm]$
1	(0.54 / 0.18 / 0.13)
2	(-0.09 / -0.16 / 0.13)
3	(-0.21 / -0.01 / -0.48)
4	(0.65 / 0.58 / -0.26)
5	(-0.11 / -0.11 / 0.03)
6	(-0.75 / -0.47 / 0.44)

Tabelle 6.2: Resultierende Abweichung $\vec{\epsilon}(i)$ der einzelnen Markerpunktepaare nach der erfolgten Registrierung

$$\begin{aligned}\bar{\epsilon} &= \frac{1}{N} \sum_{i=1}^N \left| {}^{CBV}ol\vec{Marker}(i) - transformed_Marker(i) \right| \\ &= 0.57mm\end{aligned}\quad (6.25)$$

6.3.4 Benutzeroberfläche zur Registrierung des Kalibrierkörpers

Zur Durchführung der Registrierung wurde eine Benutzeroberfläche mit Mevislab erstellt. Die Einzelschritte sind in aufeinander folgenden Registern angeordnet. Für den ersten Schritt muss die Bildgebungsmodalität und der Kalibrierkörper vom Navigationssystem erfasst werden.

Hierfür bietet die in der Abbildung 6.33 dargestellte Registeransicht „*recorddata*“ auf der linken Bildschirmhälfte die relevanten Bedienelemente. Auf der rechten Bildschirmseite befindet sich ein Vorschauenfenster zur Anzeige der erfassten Bilddaten. Zum Empfang von DICOM-Bilddaten wurde der Mevislab eigene DICOM-Receiver eingesetzt. Über die „*StartDicomServer*“ Schaltfläche wird auf eingehende Bilddaten gewartet, die in dem ausgewählten Ziel Pfad abgelegt werden. Darunter befindet sich die Ansteuerung der Stereokamera, die bereits im Abschnitt 6.2.3 beschrieben wurde. Das Feld „*save DRB Position*“ exportiert die aktuell gemessenen Referenzkörper Positionen und wird verwendet um bei Beginn der Kalibrierkörpervermessung die Position der Bildgebungsmodalität {CRB} und des Kalibrierkörpers {KRB} zu erfassen.

Anschließend kann auf der in der Abbildung 6.34 dargestellten „*LoadData*“ Registeransicht der empfangene Bilddatensatz importiert werden. Die rechte Bildschirmhälfte visualisiert den Bilddatensatz. Im linken unteren Teil der Bildansicht

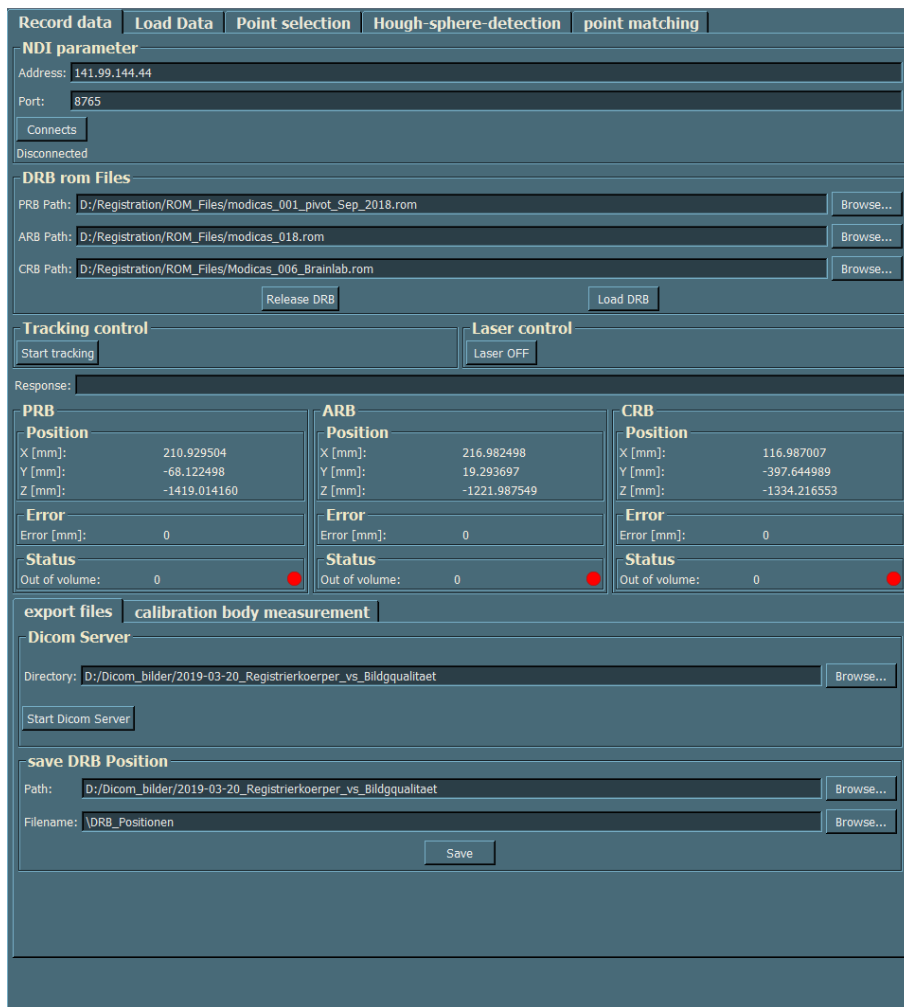


Abbildung 6.33: Benutzeroberfläche zur Ansteuerung des Navigationssystems und zum Erfassen der Referenzkörperpositionen des Kalibrierkörpers {KRB} und der Bildgebungsmodalität {CRB} sowie zur Aktivierung des DICOM Receiver Moduls zum Empfangen von DICOM Bilddatensätzen von der Bildgebungsmodalität

kann die Visualisierung des Bilddatensatzes angepasst werden. Ausserdem können die zuvor gespeicherten Referenzkörperpositionen, aus denen die Transformation ${}^{CRB}T_{KRB}$ bestimmt wird, geladen werden. Im Feld „Loadcalibrationpoints“ können die aus der Geometrie bekannten Markerpunkte geladen werden.

In der darauffolgenden Registeransicht „Pointselection“ werden die zum Matching benötigten Röntgenmarker im Bilddatensatz detektiert. Hierzu bietet die linke Bildhälfte eine Schichtansicht des Bildvolumens. Die aktuelle Schichtansicht kann über den Schieber am linken Bildschirmrand durch den gesamten Datensatz verschoben werden. Zur Orientierung wird die aktuelle Schichtansicht in der 3D-Visualisierung des Datensatzes als weißer Rahmen mit blauer Füllung überlagert. In dem Schichtan-

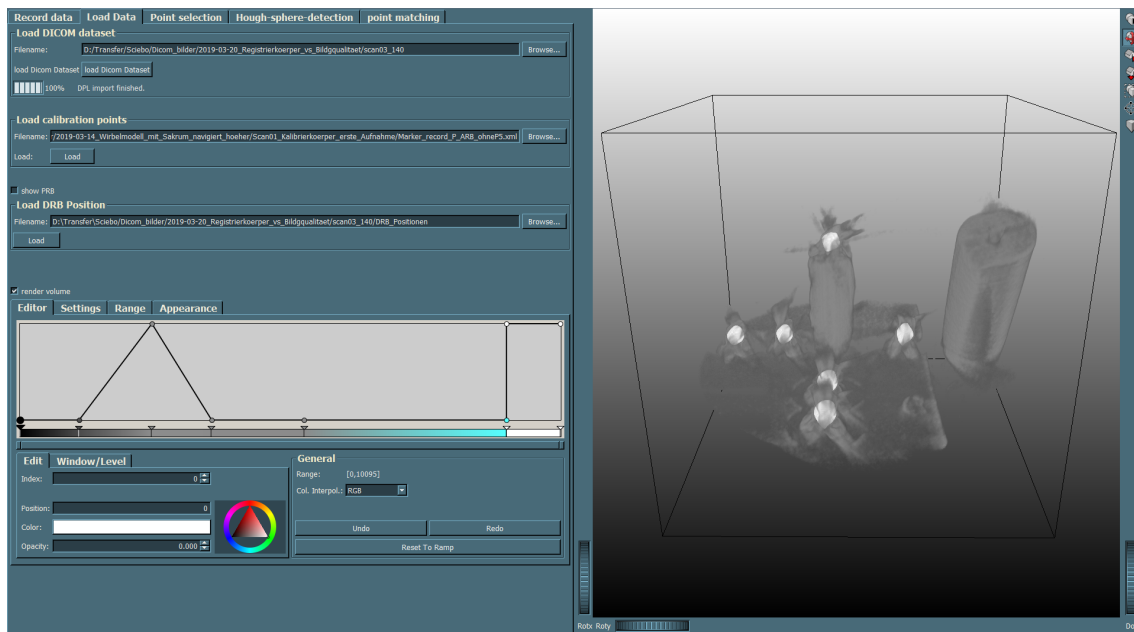


Abbildung 6.34: Benutzeroberfläche zum Laden und Visualisieren des DICOM Bild-datensatzes der Kalibrierkörperaufnahme, sowie zum Laden der aus der Geometrie bekannten Röntgenmarkerpositionen

sichtsfenster werden die Röntgenmarker über einen einfachen Mausklick ausgewählt und werden dann von der im Abschnitt 6.3.2 beschriebenen Methode detektiert. Anschließend sind die detektierten Kugelmarkerzentren berechnet und werden in der Schichtansicht rot eingefärbt. In der 3D-Visualisierung auf der rechten Bildschirmseite sind die detektierten Regionen in rot dargestellt und die Zentren der Kugelmarker als grüne Kugelobjekte abgebildet. Die Registeransicht zur Punktdetektion ist in der Abbildung 6.35 dargestellt.

Anschließend können in der Registeransicht „point matching“ über die Schaltfläche „match points“ die detektierten Punkte, mit den aus der Geometrie bekannten Punkten, registriert werden. Hierzu wird im Programmablauf das im Abschnitt 6.3.3 erstellte „PCL_matching“ Modul ausgeführt. Die Visualisierung auf der rechten Bildschirmhälfte zeigt anschließend die überlagerten Kugelpositionen an, wobei die aus der Geometrie bekannten Kugelpositionen (lila), mit den detektierten Markerpositionen (grün) überlagert sind. Als Ergebnis ergibt sich die Transformation ${}^{CBV}{}^{ol}T_{CRB}$ aus der Formel 6.22. Diese wird über die Schaltfläche „save“ gespeichert und kann für den anschließend aufgenommenen Patienten Bilddatensatz, bei der Planung verwendet werden.

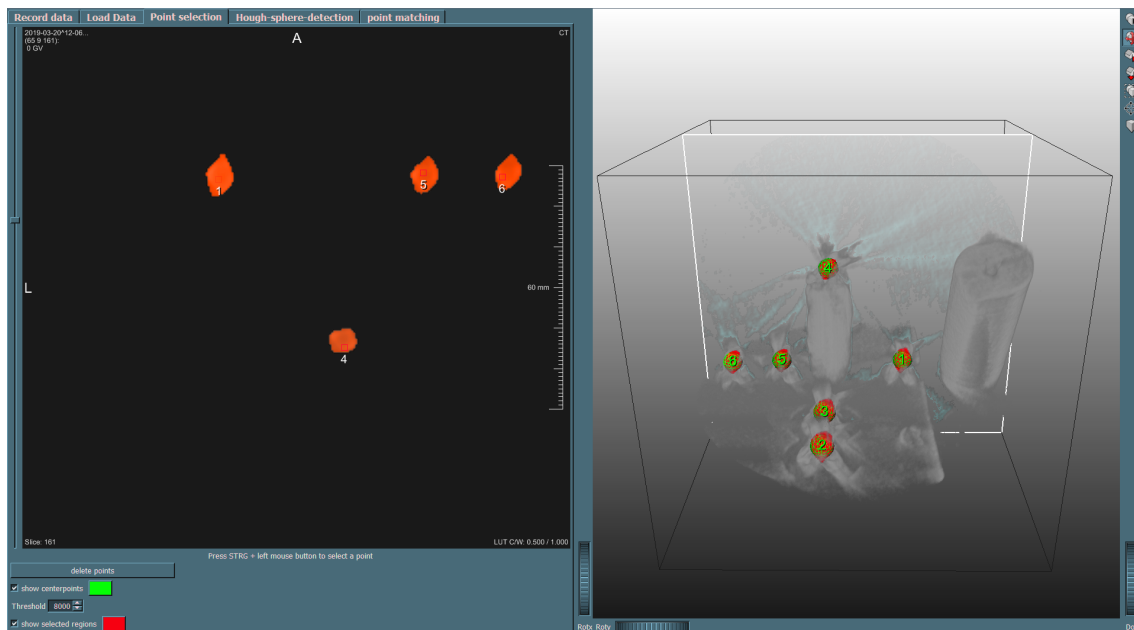


Abbildung 6.35: Benutzeroberfläche zur Detektion der Kugelmarkerzentren in der Kalibrierkörperaufnahme. Die detektierten Kugelbereiche sind in rot visualisiert und die Kugelmarkerzentren als grüne Kugelobjekte.

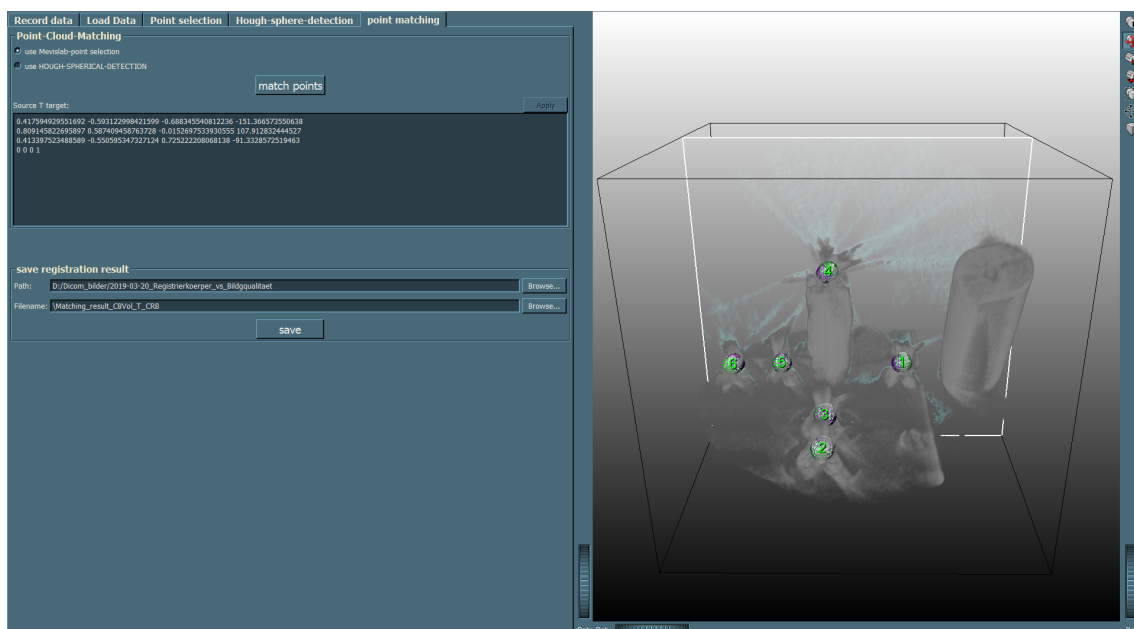


Abbildung 6.36: Benutzeroberfläche zur Ausführung der Punktregistrierung und zum Abspeichern der Ergebnis Transformation. Die aus der Geometrie bekannten Markerpositionen (lila) werden mit den detektierten Markerpositionen (grün) überlagert.

6.3.5 Akquise eines registrierten 3D Bilddatensatzes

Zur Akquise eines registrierten Bilddatensatzes wird der Patient in demselben Messvolumen, in dem die Registrierung des Kalibrierkörpers durchgeführt wurde, positioniert. Die Abbildung 6.37 skizziert die Positionierung des Patienten mit dem angebrachten Referenzkörper {ARB} im Messvolumen der Bildgebungsmodalität.

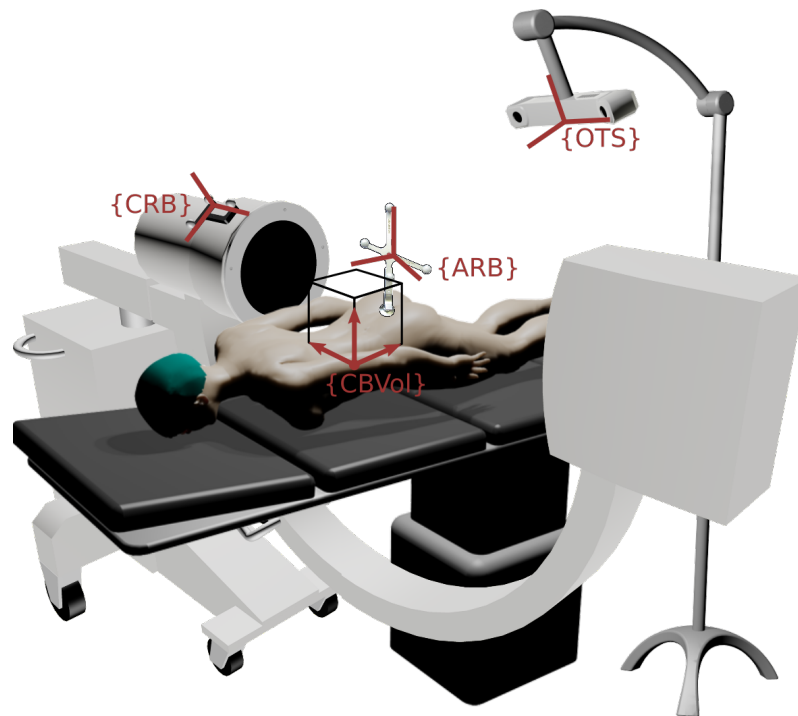


Abbildung 6.37: Intraoperative Bildgebungsmodalität mit dem angebrachten Referenzkörper {CRB} und der Patient mit dem Patientenreferenzkörper {ARB} im Bildvolumen der Bildgebungsmodalität {CBVol} sowie das optische Navigationssystem

Ebenso wie bei der Aufnahme des Kalibrierkörpers, wird der C-Bogen in der Ausgangsposition lateral positioniert (0°). Zu diesem Zeitpunkt wird die Position der Referenzkörper {CRB} und {ARB} von dem Navigationssystem erfasst. Aus den von dem Navigationssystem erfassten Positionen der Referenzkörper {ARB} und {CRB} ergibt sich die Transformation ${}^{ARB}\underline{T}_{CRB}$ gemäß der Formel 6.26. Anschließend wird der Datensatz des Patienten aufgenommen und über den beschriebenen DICOM Receiver auf dem Planungsrechner abgelegt.

$${}^{ARB}\underline{T}_{CRB} = ({}^{OTS}\underline{T}_{ARB})^{-1} \cdot {}^{OTS}\underline{T}_{CRB} \quad (6.26)$$

Da die Aufnahme in demselben Messvolumen des C-Bogens erfolgt, bleibt das Registrierungsergebnis des vorigen Schrittes gültig. Mit dem Registrierungsergebnis aus der Formel 6.22 kann nun die Lage des Bildvolumens {CBVol} im Koordinatensystem des Patientenreferenzkörpers {ARB} bestimmt werden.

$${}^{ARB}\underline{T}_{CBVol} = {}^{ARB}\underline{T}_{CRB} \cdot ({}^{CBVol}\underline{T}_{CRB})^{-1} \quad (6.27)$$

Mit der Transformation ${}^{ARB}\underline{T}_{CBVol}$ ist der Bilddatensatz mit dem Referenzkörper registriert. Die Bilddaten können zusammen mit dem Registrierungsergebnis von der im Abschnitt 6.2 beschriebenen Planungssoftware geladen und zur Ansteuerung des Roboters im Koordinatensystem {ARB} verwendet werden.

6.4 Präoperative Operationsplanung

Zur Diagnostik werden in der Regel vor dem chirurgischen Eingriff hochauflösende CT-Aufnahmen angefertigt. Diese Aufnahmen können verwendet werden um die Operation und die Positionierung der Pedikelschrauben vor dem Eingriff zu planen, was hier als präoperative Operationsplanung bezeichnet wird. So kann die Planung zu einem beliebigen Zeitpunkt vor dem Eingriff ohne zeitlichen Druck erfolgen, da während der Planung kein Patient anästhesiert ist, noch der chirurgische Eingriff aufgehalten wird. Das Ziel der präoperativen Planung ist die Festlegung der Zielpositionen der Pedikelschrauben im Bilddatensatz der präoperativen Bildgebungsmodalität.

Um diese präoperative Planung für den chirurgischen Eingriff verwenden zu können, muss diese mit dem Patientenkoordinatensystem verknüpft werden. Dies geschieht über den am Patienten angebrachten Referenzkörper mit dem Koordinatensystem {ARB}. Den Ansatz zur Lösung dieses Schrittes bildet die im Abschnitt 6.3.5 vorgestellte intraoperativ angefertigte, registrierte 3D-Aufnahme. Durch ein „Matching“ des präoperativen Datensatzes, mit dem das Koordinatensystem {CTVol} assoziiert wird, mit dem intraoperativen Datensatz im Koordinatensystem {CBVol}, ergibt sich die Transformation in der Formel 6.28 zur Überlagerung beider Datensätze.

$${}^{CTVol}\underline{T}_{CBVol} \quad (6.28)$$

Hiermit können die im präoperativen Bilddatensatz geplanten Schraubenposition über die registrierte intraoperative Aufnahme, unter Verwendung des Ergebnisses der Registrierung aus der Formel 6.27, in das Patientenkoordinatensystem {ARB} überführt werden. Die Transformation ergibt sich gemäß der Formel 6.29.

$${}^{ARB}\underline{T}_{CTVol} = {}^{ARB}\underline{T}_{CBVol} \cdot ({}^{CTVol}\underline{T}_{CBVol})^{-1} \quad (6.29)$$

6.4.1 Matchingstrategien für multimodale Wirbelaufnahmen

Für den vorliegenden Anwendungsfall wird ein Matching von einer präoperativen CT-Aufnahme mit einer intraoperativen C-Bogenaufnahme betrachtet, da dies den Gegebenheiten in den meisten Operationsräumen entspricht. Ein Matching derselben Aufnahme-region in Bilddatensätzen von unterschiedlichen Bildgebungsmodalitäten, entspricht einem multimodalen Matching. Die Abbildung 6.38 zeigt das für das Matching verwendete Wirbelmodell und eine 3D-Visualisierung der zugehörigen CT und C-Bogen Aufnahmen.

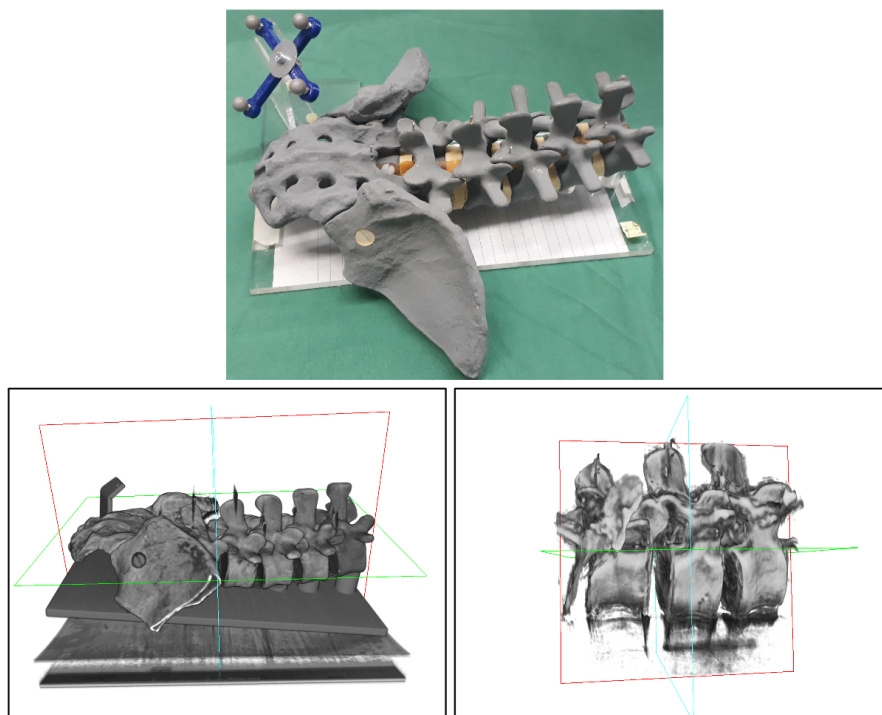


Abbildung 6.38: Das verwendete Wirbelmodell (oben), visualisierte 3D-Darstellung einer CT-Aufnahme (unten links) und die visualisierte 3D-Darstellung einer C-Bogen Aufnahme (unten rechts)

Da mit C-Bögen ein geringeres Aufnahmevolumen zur Verfügung steht, kann abhängig von der Anzahl der zu operierenden Wirbel, nicht das gesamte Operationsvolumen in einer Aufnahme erfasst werden. Es können Verschiebungen zwischen den Wirbeln auftreten, da diese nicht rigide miteinander verbunden sind. Um diesen Einfluss untersuchen zu können, sollen einzelne Wirbel überlagert werden. Dazu müssen die einzelnen Wirbel separiert betrachtet werden. Hierfür werden die Wirbel mit geeigneten Methoden segmentiert und anschließend gematcht.

Im Rahmen der betreuten studentischen Arbeit [93] wurden verschiedene Methoden zur Segmentierung und zum Matching einzelner Wirbelkörper umgesetzt und verglichen. Die größte Herausforderung stellen dabei die qualitätsärmeren C-Bogen Aufnahmen dar. Aufgrund der verfahrensbedingten Aufnahmemethode wird nicht das vollständige Aufnahmevolumen durchleuchtet. Daher fehlt ein geringer Bildausschnitt und es treten Bildartefakte auf.

Da die einzelnen Wirbel zusammenhängende Bereiche darstellen, die sich deutlich von der Umgebung abgrenzen, wurde zur Segmentierung ein regionenbasiertes Verfahren angewendet [87, S.68-71]. Dabei wird ein Punkt oder mehrere Punkte, innerhalb des zu segmentierenden Wirbelkörpers, ausgewählt. Anschließend werden die angrenzenden Voxel auf Übereinstimmungen überprüft und ein zusammenhängender Bereich detektiert. Die Abbildung 6.39 zeigt die segmentierten Wirbelkörper in beiden Bilddatensätzen.

Die Ergebnisse der Segmentierung wurden verwendet um eine Oberflächenregistrierung durchzuführen. Hierbei wird die Oberfläche des segmentierten Wirbelkörpers extrahiert und überlagert. Aufgrund der Artefakt behafteten Ergebnisse der Segmentierung des Wirbelkörpers in der C-Bogen Aufnahme, erscheint die extrahierte Oberfläche ebenfalls Artefakt behaftet und verkompliziert ein gutes Ergebnis des Matchings.

Mit dem Ziel, die Benutzerinteraktion möglichst einfach zu gestalten, wurde als weiterer Ansatz eine Segmentierung durch Festlegung der Volumenbereiche der einzelnen Wirbelkörper umgesetzt. Hierbei wird für jeden Wirbel die erste und letzte Schicht, in dem der Wirbel im Bilddatensatz erscheint, ausgewählt. Die Abbildung 6.40 zeigt die ausgewählten Wirbelkörper in dem C-Bogen und dem CT Datensatz.

Für das anschließende Matching der einzelnen Wirbelkörper wurden verschiedene Matchingverfahren implementiert und verglichen. Für das vorliegende Anwendungsszenario stammen beide Datensätze von derselben Knochenregion. Ausserdem sind Wirbelkörper rigide, weshalb nicht mit einer Deformierung zu rechnen ist. Daher

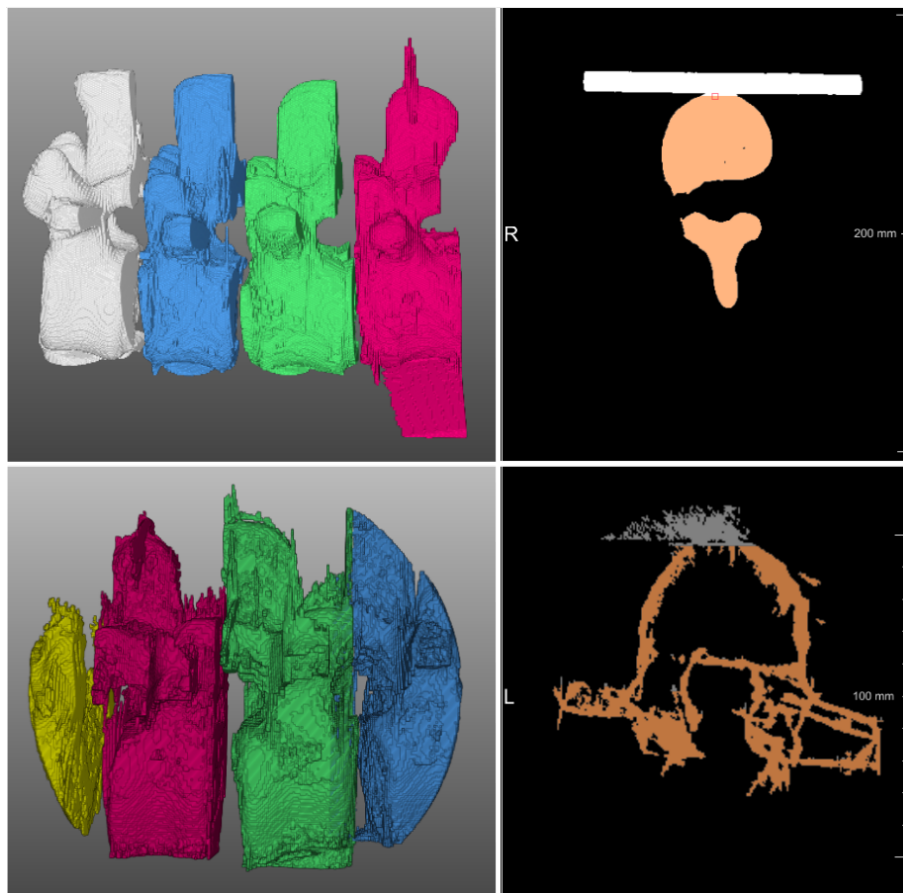


Abbildung 6.39: 3D-Darstellung der segmentierten Wirbelkörper der CT Aufnahme (oben links), sowie einer Querschnittsansicht (oben rechts) und die 3D-Darstellung der segmentierten Wirbelkörper einer C-Bogen Aufnahme (unten links) sowie einer Querschnittsansicht (unten rechts) [93]

eignen sich für das vorliegende Szenario Matchingverfahren, die eine Verschiebung und eine Rotation zwischen den Datensätzen ermitteln. In der Literatur wird dies als eine rigide Transformation bezeichnet. Eine Übersicht verschiedener Verfahren ist in der Literatur zu finden [87, S.85-115].

Als erste Matching Methode wurde die landmarkenbasierte Registrierung untersucht. Dafür werden korrespondierende Bildpunkte in beiden Datensätzen ausgewählt, um daraus die Registrierung zu bestimmen. Hierbei hängt das Ergebnis von der Auswahl und Platzierung der Landmarken ab. Insbesondere die exakte Festlegung der korrespondierenden Punkte an anatomisch wiedererkennbaren Strukturen ist fehleranfällig. Zum einen erfordert dies einen erhöhten Aufwand für den Anwender, zum anderen sind die Ergebnisse nur schwierig zu reproduzieren.

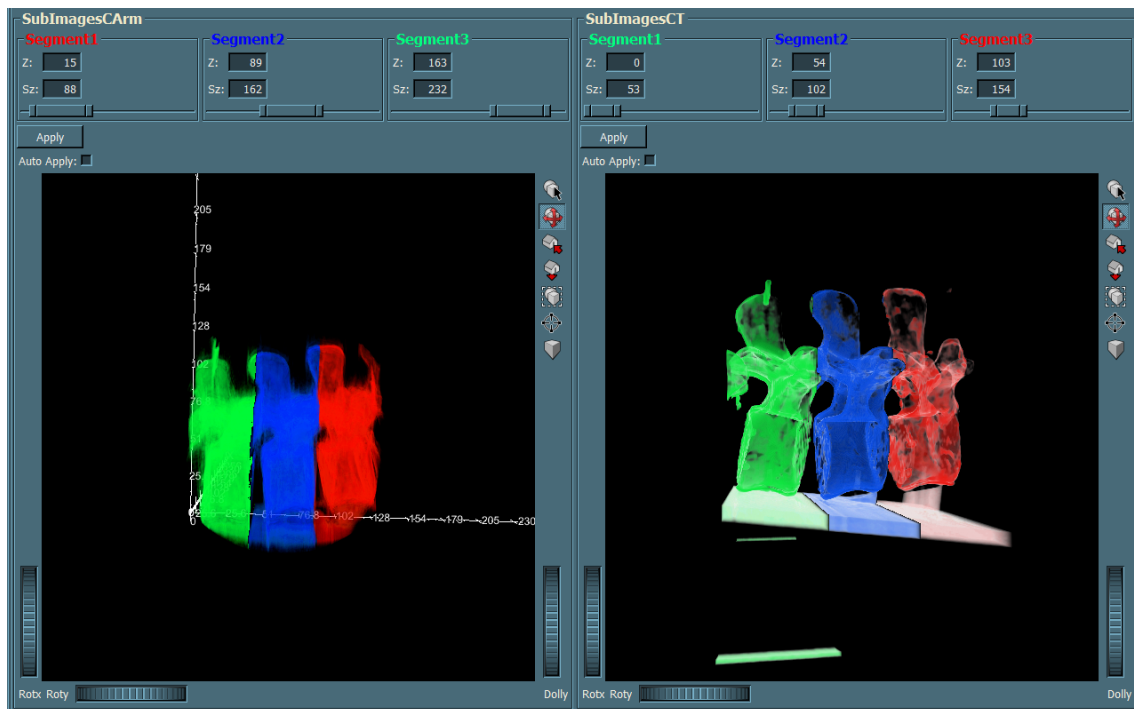


Abbildung 6.40: Die ausgewählten Wirbelkörperbereiche in der C-Bogen Aufnahme (links) und der CT Aufnahme (rechts). Die einzelnen Wirbelkörper sind farblich hervorgehoben. [93]

Als weitere Methode wurde eine Oberflächenregistrierung mit dem Softwareframework 3D-Slicer umgesetzt. Dieses wendet den „Iterative-Closest-Point“ [12] Algorithmus an, um die Bildpunkte zu vergleichen und daraus ein Maß für die Abweichung zu bestimmen. Den Ausgangspunkt hierfür bilden zwei segmentierte Wirbelkörper der Datensätze. Das Ergebnis wird schrittweise optimiert, bis die Abweichung minimal ist. Auf Grund der Artefakt behafteten Oberfläche der C-Bogenaufnahme, lieferte diese Methode keine geeigneten Ergebnisse [93].

Des Weiteren wurde das im Softwareframework Mevislab vorhandene „MERIT“ Modul (MEVIS Image Registration Toolkit) [15] [16] angebunden und zur Registrierung eingesetzt. Dieses beinhaltet Methoden für vielfältige Anwendungsfälle und kann über die Parametrierung an den Anwendungsfall angepasst werden. Die Abbildung 6.41 zeigt die Parametrierung des „MERIT“ Moduls, das eine Registrierung der Datensätze ermöglicht.

Als Registrierungstyp wurde hier „Rigid“ ausgewählt, wodurch die Translation und die Rotation transformiert werden, während die Skalierung unverändert bleibt. Da die Datensätze als 3D-Aufnahmen vorliegen, wird diese Dimension gewählt. Der „similarity“ Parameter wählt die Methode, nach der die Bilder verglichen werden. Die

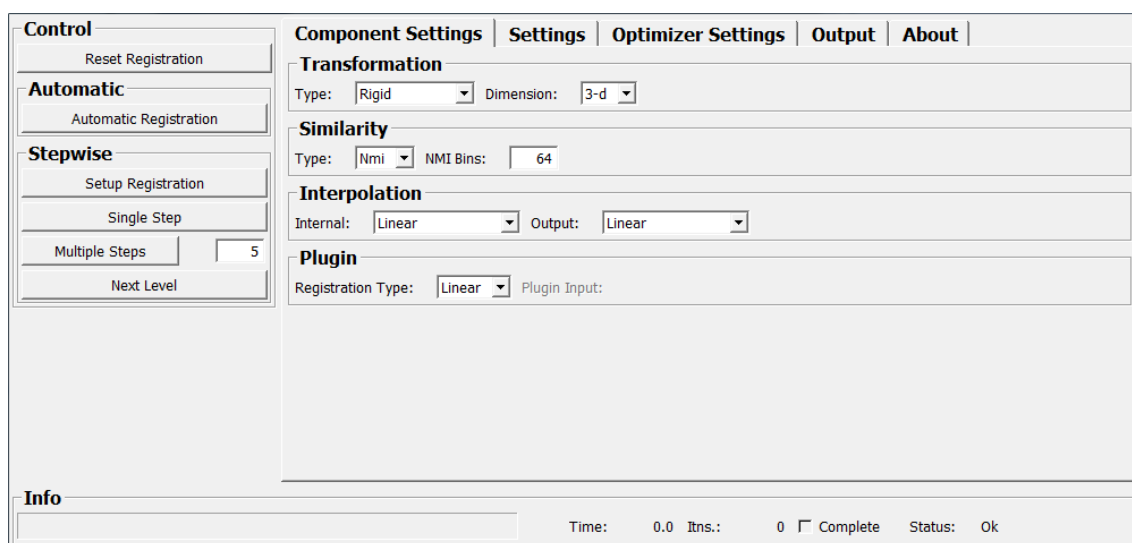


Abbildung 6.41: Parametrierung des „MERIT“ Moduls zur Registrierung von zwei 3D-Bilddatensätzen [93]

„NMI“ (Normalized Mutual Information) Methode betrachtet die Entropie beider Bilder, basierend auf deren Histogramm Analyse und eignet sich besonders für multimodale, intensitätsbasierte Registrierungen. Die „NMI Bins“ geben die Auflösung des Histogramms an. Somit werden die Intensitätswerte in einem Histogramm mit 64 gleichverteilten Stufen bestimmt.

Die Entropie eines Bildes kann aus dem Histogramm bestimmt werden und gibt die minimale Anzahl von Bits an, die benötigt werden um den Informationsgehalt eines Bildpixels zu speichern. Die Entropie kann gemäß der Formel 6.30 bestimmt werden.

$$H(I) = - \sum_{g \in G} P_I(g) \log(P_I(g)) \quad (6.30)$$

Hierbei beschreibt $P_I(g)$ die Häufigkeit eines Grauwertes g im Bild I , wobei die Summe über alle Grauwerte G gebildet wird. Ebenso lässt sich auch die Entropie für ein Referenzbild R und ein zu vergleichendes Bild Q bestimmen.

$$H(R, Q) = - \sum_{g_R \in G_R, g_Q \in G_Q} P_{R,Q}(g_R, g_Q) \log(P_{R,Q}(g_R, g_Q)) \quad (6.31)$$

Dabei gibt $P_{R,Q}(g_R, g_Q)$ die relative Häufigkeit für das gleichzeitige auftreten der Grauwerte g_R in R und g_Q in Q an. Ist die Entropie beider Bilder $H(R, Q)$ mi-

nimal, so ergibt sich die bestmögliche Übereinstimmung beider Bilder. Da die gemeinsame Entropie auch sinkt, wenn die Entropie eines der Bilder sich verringert, wird dies in der „Mutual Information“ (MI), die in der Formel 6.32 angegeben ist, berücksichtigt.

$$MI(R, Q) = H(R) + H(Q) - H(R, Q) \quad (6.32)$$

Um ein Distanzmaß D zu erhalten, wird das zu vergleichende Bild Q mit der Transformation \underline{T} verschoben und rotiert und anschließend mit dem Referenzbild verglichen. Die Formel 6.33 definiert dieses Distanzmaß.

$$D(R, Q, \underline{T}) = -MI(R, Q(\underline{T})) \quad (6.33)$$

Hieraus ergibt sich ein minimales Distanzmaß für die Bilder R und Q bei der gesuchten Transformation \underline{T} . Die Transformation beschreibt die durch den Algorithmus bestimmte Überlagerung der Bilder. Da lediglich der Zusammenhang der Grauwerte in den Bildern verglichen wird, ist diese Methode ein mächtiges Werkzeug zum Vergleich von Bilddaten von unterschiedlichen Modalitäten. Allerdings liegt hierin auch die Herausforderung, da vorab nicht abgeschätzt werden kann wie das Ergebnis ausfällt und ungewollte Registrierungen möglich sind. Daher muss das Ergebnis abschließend überprüft werden. Eine vollständige Beschreibung sowie die hier angegebenen Formeln sind in der Quelle [81, S.97-98] beschrieben.

Als Ergebnis der Registrierung ergibt sich die in der Abbildung 6.42 dargestellte Überlagerung der CT und C-Bogen Aufnahmen des Wirbelmodells aus der Abbildung 6.38.

Aufgrund der erzielten Ergebnisse, wurde für die weitere Arbeit das „MERIT“ Modul des Mevislab Frameworks für das Matching der 3D-Datensätze eingesetzt [93].

6.4.2 Erweiterung der Software zur präoperativen Operationsplanung

Für die Umsetzung einer präoperativen Planung wird die im Abschnitt 6.2 beschriebene Planungssoftware erweitert. Im ersten Schritt wird die Planung um die Visualisierung eines präoperativen Bilddatensatzes ergänzt. Da beide Datensätze als

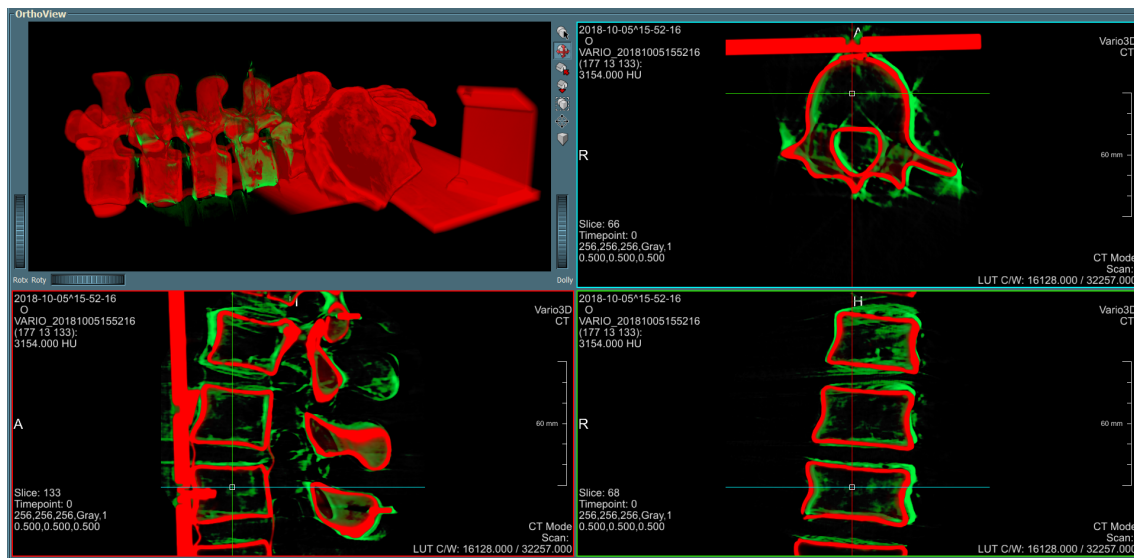


Abbildung 6.42: Ergebnis der Überlagerung einer CT Aufnahme eines Wirbelmodells (rot) mit einer C-Bogen Aufnahme (grün), erstellt mit dem „MERIT“ Modul [93]

DICOM Bilddaten zur Verfügung stehen, kann das Mevislab Netzwerk zum Laden und Visualisieren der Datensätze, das im Abschnitt 6.2.1 beschrieben ist, dupliziert werden. Dadurch ergeben sich dieselben Anzeigefenster mit einer 3D-Visualisierung und den Ebenen Ansichten des Datensatzes, die bereits in der Abbildung 6.12 dargestellt sind.

Die Planung von Pedikelschrauben kann in einem präoperativen CT Datensatz genauso erfolgen wie in einem Datensatz aus einer intraoperativen Bildgebungsmodalität, was bereits im Abschnitt 6.2.2 beschrieben ist. Als Ergebnis der Planung liegen die Schraubenpositionen bei der präoperativen Planung im Koordinatensystem von $\{CTVol\}$ vor und beschreiben die Transformation der geplanten Schraubenposition durch die Transformation 6.34.

$${}^{CTVol}T_{screw} \quad (6.34)$$

Die Anwendung einer präoperativen Planung entspricht der zweiten Workflow Variante aus der Abbildung 6.1. Diese Workflow Variante sieht vor, dass nach der präoperativen Planung, zu Beginn des chirurgischen Eingriffes, eine registrierte intraoperative 3D-Aufnahme, wie im Abschnitt 6.3.5 beschrieben, angefertigt wird. Anstelle der intraoperativen Planung, erfolgt nun ein Matching des präoperativen Datensatzes mit dem intraoperativen Datensatz. Hierfür wurde das im Rahmen der

betreuten studentischen Arbeit [93] entwickelte Matching Programm in einem eigenen Mevislab Makro Modul zusammengefasst. Am Eingang des Moduls werden beide 3D-Bilddatensätze an das Matching Modul übergeben. Zur Anpassung der Visualisierung werden ausserdem die Visualisierungseinstellungen der Planungssoftware als Parameter übergeben. Die Abbildung 6.43 zeigt das erstellte „3D3DMatching“ Modul mit den zwei Eingängen für die Bilddatensätze und das Parameterfenster mit den übergebenen Parametern.

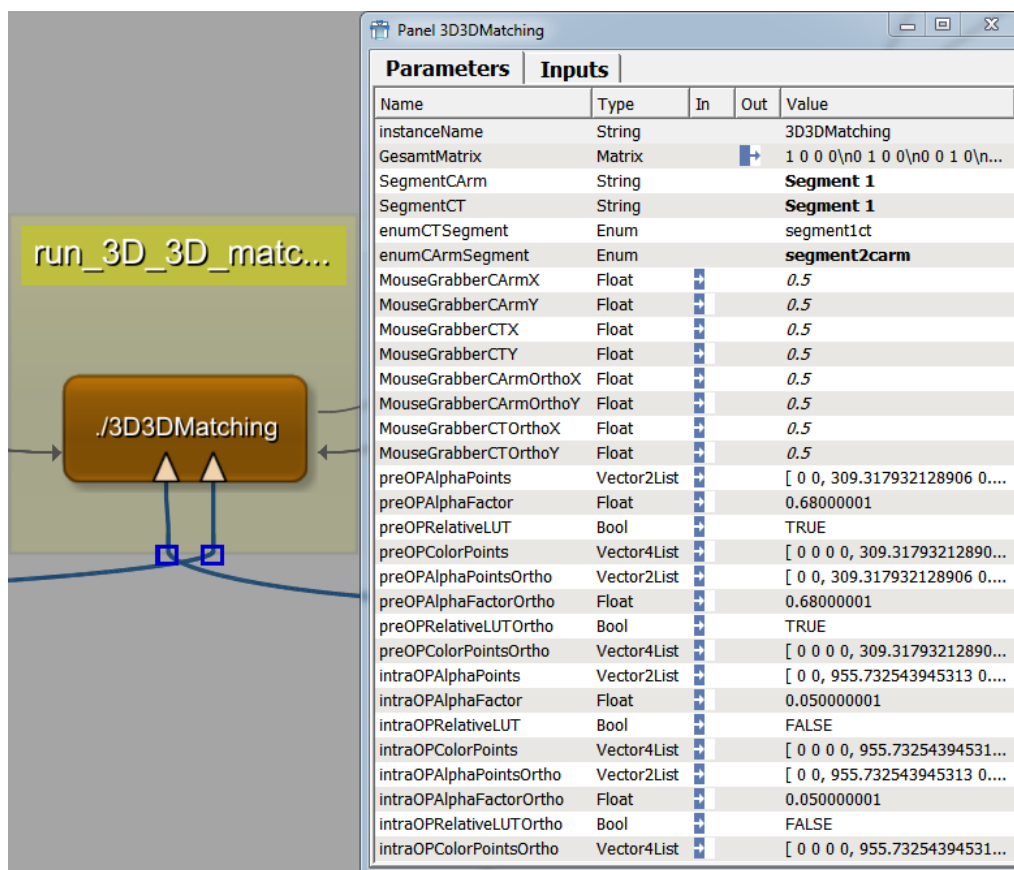


Abbildung 6.43: „3D3DMatching“ Makro Modul und das Parameterpanel mit den übergebenen Visualisierungseinstellungen

In dem erstellten Modul wurde eine schrittweise Benutzerführung über einzelne Registeransichten umgesetzt. Im ersten Schritt können Visualisierungsparameter angepasst werden und die Datensätze werden zur besseren Erkennbarkeit eingefärbt. Bei der Evaluierung zeigte sich, dass für die automatische Registrierung beide Datensätze dieselbe Ausrichtung haben sollten. Dies ist in der Regel der Fall, denn die Lage des Patienten Koordinatensystems wird bei der Aufnahme gemäß dem DICOM Standard festgelegt. Da einige Testdatensätze mit einer falschen Ausrichtungseinstellung vorlagen, wird für den nächsten Schritt eine Vorpositionierung implementiert, die eine Korrektur der Ausrichtung des Datensatzes ermöglicht. Hier wird durch

die Festlegung einer Rotation um die X-, Y-, und Z-Achse und eine Translation eine Transformationsmatrix erzeugt und der Datensatz damit vorpositioniert. Bei weiteren Tests zeigte sich, dass Aufnahmeinformationen vom Rand des Datensatzes das Matching Ergebnis negativ beeinflussen können. Daher wurde in dem darauffolgenden Reiter eine Zuschnitt Funktion implementiert, mit der sich der Datensatz auf die relevanten Wirbelregionen zuschneiden lässt. Im nächsten Schritt werden die Einzelwirbel über definierbare Bereiche festgelegt und ausgewählt, welcher Auswahlbereich für das Matching verwendet werden soll. Dieser Schritt ist in der Abbildung 6.40 dargestellt. Daraufhin wird das Matching durchgeführt und die Ergebnis Transformation aus der Formel 6.34 generiert. Nach der Transformation des präoperativen Datensatzes wird das Ergebnis durch eine Überlagerung beider Datensätze visualisiert. Hier wurden, wie schon bei der Planungssoftware im Abschnitt 6.2.1 eingeführt, die einzelnen Ebenen Ansichten und eine 3D-Ansicht zur Visualisierung verwendet. Die erstellte Benutzeroberfläche, mit einer Darstellung der einzelnen Schritte für das Matching der in der Abbildung 6.38 vorgestellten Wirbelmodell Aufnahmen, ist im Anhang C dargestellt.

Wie die Abbildung 6.42 zeigt, liefert die automatische Registrierung nicht immer das visuell optimale Ergebnis. Ein Grund hierfür liegt in den Artefakten der C-Bogen Bilddaten, die das Ergebnis beeinflussen. Um das Ergebnis zu überprüfen und zu optimieren wurde die Oberfläche um eine manuelle Feinjustierung ergänzt. Hierfür wurden die bereits bei der Planung von Pedikelschrauben implementierten Schaltflächen zur Justierung der Position und Orientierung der geplanten Schraubenposition eingebunden, um den präoperativen Datensatz in feinen Schritten verschieben und rotieren zu können. Die Abbildung 6.44 zeigt die Benutzeroberfläche mit einer Darstellung der optimierten Überlagerung der 3D-Datensätze aus der Abbildung 6.42 und den Schaltflächen für die manuelle Justierung.

Nach dem durchgeführten Matching beschreibt die Transformation ${}^{CTVol}T_{CBVol}$ das Ergebnis und steht als Parameter des „3D3DMatching“ Makro Moduls für die weiteren Berechnungen zur Verfügung.

Um die Schraubenpositionen zusätzlich in dem intraoperativen Datensatz anzeigen zu können, müssen diese in das Koordinatensystem $\{CBVol\}$ transformiert werden. Hierfür wurde ein zweiter XMarkerList-Container angelegt, in welchem die nach dem Matching transformierten, geplanten Schraubenpositionen im Koordinatensystem $\{CBVol\}$ gespeichert werden. Die Transformation erfolgt mit dem Ergebnis des Matchings gemäß der Formel 6.35.

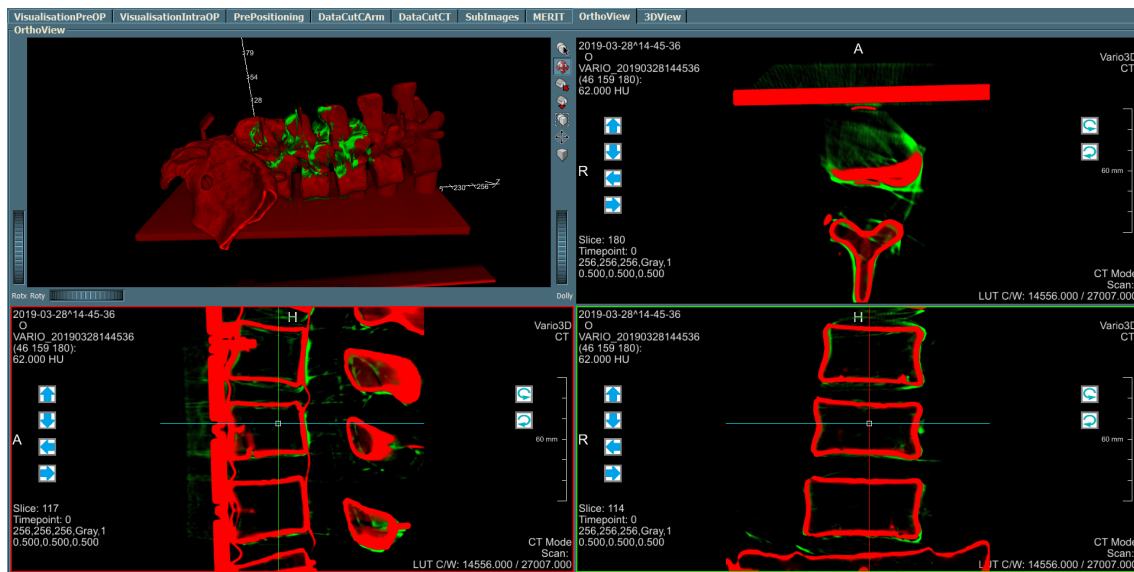


Abbildung 6.44: 3D-Ansicht und Ebenen Darstellung des Matchingergebnisses nach einer Feinjustierung über die ergänzten Schaltflächen

$$CBVol \underline{T}_{screw} = CBVol \underline{T}_{CTVol} \cdot CTVol \underline{T}_{screw} \quad (6.35)$$

Das erstellte Mevislab Netzwerk mit beiden XMarkerList Containern und der erfolgten Umrechnung ist in der Abbildung A.1 im Anhang dargestellt. Diese transformierten Schraubenpositionen können nun in der intraoperativen Aufnahme dargestellt werden und ermöglichen so eine Überprüfung des Matchingergebnisses. Die Abbildung 6.45 zeigt die Benutzeroberfläche der erweiterten Planungssoftware in der Registeransicht des „3D-3D Matching“ Fensters. Von hier kann das „3D – 3D Matching“ Makromodul über die Schaltfläche „Start 3D-3D Matching“ gestartet werden. Nach dem Matching werden in den 3D-Visualisierungen der Datensätze die geplanten Schraubenpositionen in beiden Datensätzen angezeigt.

Für Testzwecke wurden an dem Wirbelmodell Metallstifte als Landmarken angebracht, die auch in den Bildaufnahmen sichtbar sind. Die Abbildung 6.45 zeigt drei geplante Schrauben, die auf den Metallstiften des Wirbelmodells im präoperativen Datensatz geplant wurden und nach dem Matching auf die intraoperative C-Bogen Aufnahme übertragen sind. In dieser Ansicht kann überprüft werden, ob das Matching erfolgreich war und die Position der geplanten Schrauben korrekt übertragen wurden.

Zur Bestimmung der vom Roboter anzufahrenden Zielposition, muss die geplante Schraubenposition in das Koordinatensystem des Referenzkörpers am Patienten

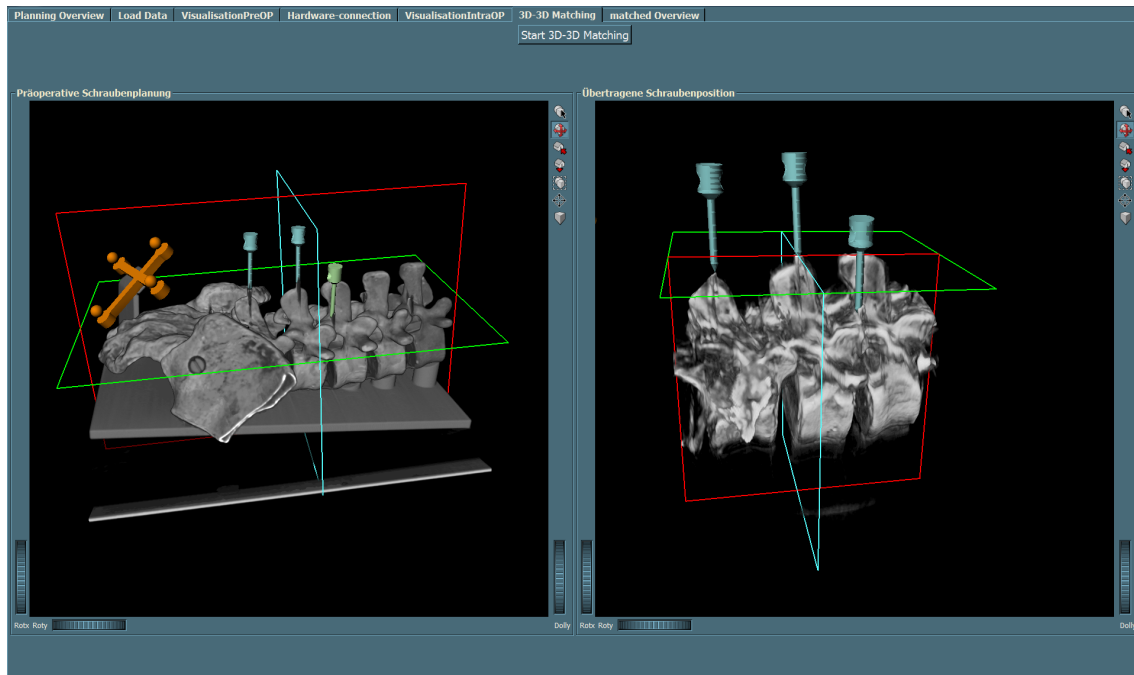


Abbildung 6.45: Benutzeroberfläche zum Start des 3D-3D Matchings und zur Visualisierung des Ergebnisses durch die Übertragung der im präoperativen Datensatz geplanten Schraubenpositionen (links) in die intraoperative C-Bogen Aufnahme (rechts)

{ARB} transformiert werden. Für die präoperative Planung muss daher die Zielposition aus der Formel 6.10 um das Matching Ergebnis erweitert werden. Dies geschieht durch Einsetzen und einer Multiplikation mit dem Matching Ergebnis aus der Formel 6.35. Damit ergibt sich die vom Roboter anzufahrende Zielpositionen gemäß der Formel 6.36.

$${}^{ARB}\underline{T}_{screw\ tip} = {}^{ARB}\underline{T}_{CBVol} \cdot {}^{CBVol}\underline{T}_{CTVol} \cdot {}^{CTVol}\underline{T}_{screw} \cdot {}^{screw}\underline{T}_{screw\ tip} \quad (6.36)$$

Um die Kompatibilität für eine intraoperative Planung beizubehalten, wird die Transformation ${}^{CBVol}\underline{T}_{CTVol}$ mit der Einheitsmatrix initialisiert. Hierdurch kann das erstellte Mevislab Netzwerk zur präoperativen Planung auch für eine intraoperative Planung verwendet werden. Für eine präoperative Planung wird nach dem durchgeführten Matching die Einheitsmatrix der Transformation ${}^{CBVol}\underline{T}_{CTVol}$ mit dem Matching Ergebnis überschrieben. Hierdurch lässt sich die erstellte Software für beide Workflow Varianten einsetzen, ohne dass eine zusätzliche Parametrierung erforderlich ist.

6.5 Ergebnis Betrachtung des Gesamtsystems

Zur Überprüfung der implementierten Verfahren werden Genauigkeitsuntersuchungen am Gesamtsystem durchgeführt. Die Gesamtgenauigkeit setzt sich aus der Messunsicherheit aller eingesetzten Komponenten zusammen. Hier muss berücksichtigt werden, dass diese sich aus der Genauigkeit bei der Planung einer Schraubenposition, der Registriergenauigkeit, der Genauigkeit der optischen Referenzkörpererfassung, der Werkzeugvermessung und der Positioniergenauigkeit des Robotersystems zusammensetzt. Die Aussagen über die erzielbare Genauigkeit sind für den jeweils untersuchten Datensatz gültig. Mit jeder neu durchgeführten Registrierung der Bildgebungsmodalität kann die Messunsicherheit variieren, weshalb folgende Bilddaten eigenständig untersucht werden müssen. Die Planungsgenauigkeit, als auch die Registriergenauigkeit können nicht alleinstehend untersucht werden, da zur Überprüfung eines erreichten Ergebnisses das Navigationssystem eingesetzt werden muss. Daher werden verschiedene Szenarien betrachtet, die einen Überblick über die erzielten Ergebnisse ermöglichen sollen. Für den klinischen Einsatz werden Abweichungen von $< 1 \text{ mm}$ bei der Platzierung der Schrauben angestrebt.

6.5.1 Messung der Planungsgenauigkeit am Modell

Im ersten Szenario soll das erzielte Registrierungsergebnis zusammen mit der Planungsgenauigkeit untersucht werden. Hierfür wurde ein Kunststoffmodell angefertigt, das mit zwei Bohrungen versehen wurde, die die Lage von Pedikelschrauben in der Wirbelsäule nachbilden. Die Bohrungen wurden mit einem Durchmesser von 5 mm präzise gefertigt. Zur Lokalisation des Modells wurde an diesem ein Referenzkörper angebracht, der die Anbringung eines Patientenreferenzkörpers nachbildet und mit dem Koordinatensystem $\{ARB\}$ beschrieben wird. Zur Überprüfung der Positioniergenauigkeit wird ein Zeigeinstrument verwendet, das ebenfalls mit einem Referenzkörper ausgestattet ist und dessen Zeigerspitze mit dem Koordinatensystem $\{PTP\}$ beschrieben wird. Mit einem Zeigerspitzendurchmesser von 5 mm kann dieses passgenau in der Bohrung positioniert werden und die Position über das Lokalisiersystem bestimmt werden. Mit den detektierten Referenzkörpern kann die Pose der Zeigerspitze relativ zum Koordinatensystem $\{ARB\}$ angegeben werden.

$${}^{ARB}\underline{T}_{PTP} = ({}^{OTS}\underline{T}_{ARB})^{-1} \cdot {}^{OTS}\underline{T}_{PTP} \quad (6.37)$$

Die Abbildung 6.46 zeigt das erstellte Modell und das Zeigeeinstrument, das in einer Bohrung platziert wurde.

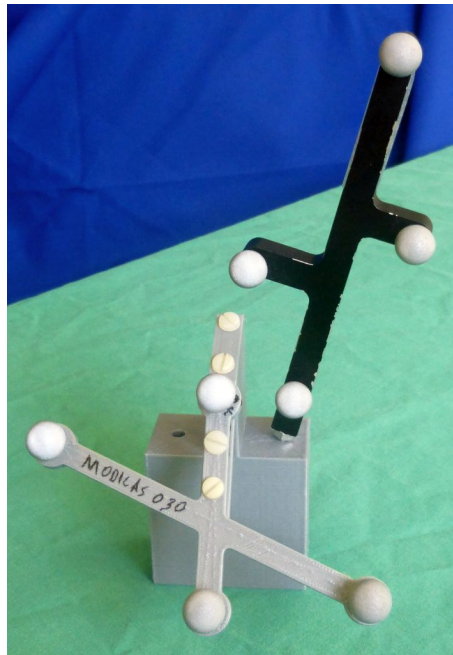


Abbildung 6.46: Erstelltes Modell mit zwei Bohrungen und das darin platzierte Zeigeeinstrument, das vom Lokalisiersystem lokalisiert wird

Von dem Modell wurde eine registrierte 3D-Aufnahme mit einem intraoperativen C-Bogen (Ziehm Vario 3D), wie im Abschnitt 6.3.5 beschrieben, angefertigt. In dieser Aufnahme wird, durch die Planung einer Schraubenposition in der Bohrung des Modells, die Zielposition festgelegt. Unter Verwendung des Registrierungsergebnisses wird so die Zielposition im Koordinatensystem $\{ARB\}$ angegeben und durch die Transformationsmatrix ${}^{ARB}T_{screw}$ beschrieben.

Die Abbildung 6.47 zeigt die 3D-Aufnahme des Modells, in der eine Schraubenposition geplant wurde. Überlagert wird die erfasste Position des Zeigeeinstruments im Datensatz visualisiert.

Da sowohl die geplante Zielposition als auch die Position des Zeigeeinstruments im Koordinatensystem $\{ARB\}$ vorliegen, kann die Abweichung als Transformation zwischen beiden Positionen bestimmt werden.

$${}^{screw}T_{PTP} = {}^{ARB}T_{screw}^{-1} \cdot {}^{ARB}T_{PTP} \quad (6.38)$$

Hierbei gibt der Translationsvektor der Transformation aus dem Ergebnis 6.38 die Positionsabweichung an. Aufgrund der Rotationssymmetrie der Schraube entlang

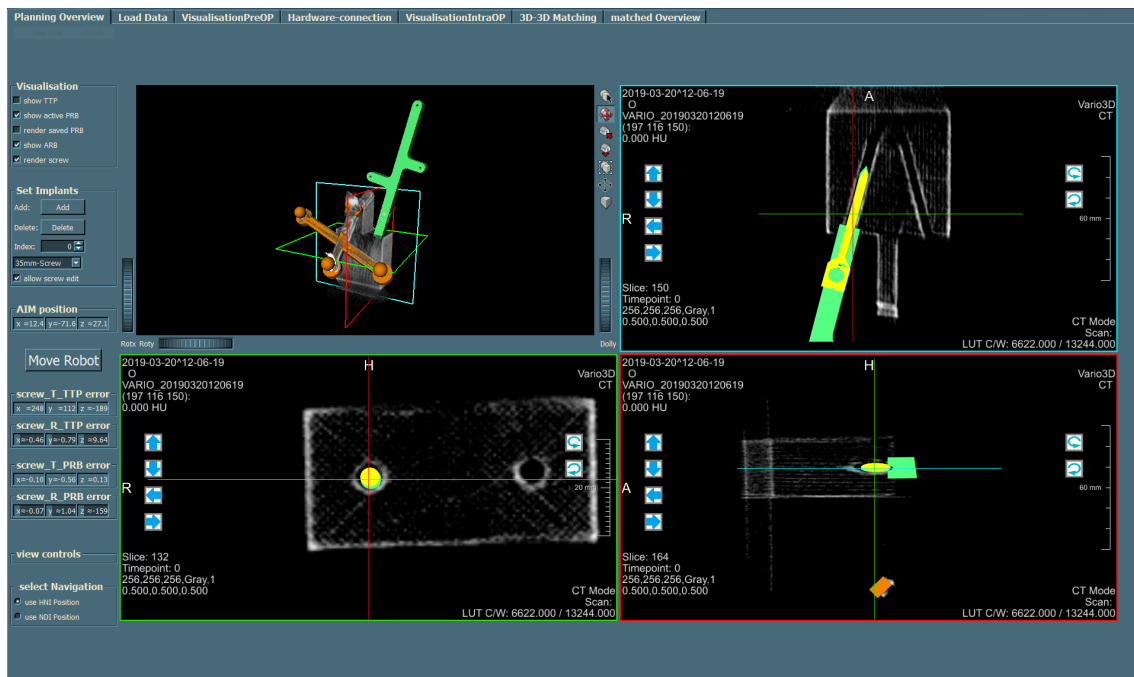


Abbildung 6.47: 3D-Datensatz des Modells mit einer geplanten Schraubenposition als Zielposition (gelb) und dem visualisierten Zeigeeinstrument (grün)

ihrer Z-Achse, geben die X- und die Y-Komponenten des Translationsvektors die radiale Abweichung, relativ zur Schraubenachse an. Die Z-Komponente gibt die Abweichung der Schraubenposition entlang der Schraubenachse an.

Zur Beschreibung der Rotationsabweichung zwischen der geplanten Schraubenposition und der Position des Zeigeeinstruments, wird der Rotationsanteil der Ergebnis Transformation durch 3 Winkel (Roll(γ), Pitch (β) und Yaw (α)) ausgedrückt. Nach der Formel 6.39 kann die Orientierung eines Koordinatensystem $\{B\}$ im Referenzkoordinatensystem $\{A\}$ mit der Rotationsmatrix ${}^A R_B$ durch eine Rotation mit dem Winkel γ um die X-Achse, eine Rotation um den Winkel β um die Y-Achse und eine Rotation um den Winkel α um die Z-Achse beschrieben werden [96].

$$\begin{aligned}
 {}^A\underline{R}_B &= \text{Rot}\{\vec{Z}_A, \alpha\} \cdot \text{Rot}\{\vec{Y}_A, \beta\} \cdot \text{Rot}\{\vec{X}_A, \gamma\} \\
 &\text{mit } c\alpha = \cos(\alpha); s\alpha = \sin(\alpha); \text{ usw.} \\
 &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \\
 &= \begin{bmatrix} cac\beta & cas\beta s\gamma - sac\gamma & cas\beta c\gamma + sas\gamma \\ sac\beta & sas\beta s\gamma + cac\gamma & sas\beta c\gamma - cas\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \tag{6.39} \\
 &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
 \end{aligned}$$

Hieraus können die Winkel α , β und γ gemäß den Formeln 6.40, 6.41 und 6.42 bestimmt werden [96].

$$\beta = \text{atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \tag{6.40}$$

Für $\cos(\beta) \neq 0$

$$\alpha = \text{atan2}(r_{21}/c\beta, r_{11}/c\beta) \tag{6.41}$$

$$\gamma = \text{atan2}(r_{32}/c\beta, r_{33}/c\beta) \tag{6.42}$$

Die Tabelle 6.3 gibt die Ergebnisse für die Rotationsabweichung, beschrieben durch 3 Rotationswinkel und die Translationsabweichung zwischen der geplanten Schraubenposition und der Position des Zeigeinstrumentes an.

Achse	Translationsabweichung [mm]	Rotationsabweichung [Grad]
X	-0.10	$\gamma = -0.07$
Y	-0.56	$\beta = 1.04$
Z	0.13	$\alpha = -159$

Tabelle 6.3: Translations- und Rotationsabweichung zwischen der geplanten Schraubenposition und der gemessenen Zeigeinstrumentposition

Aufgrund der Rotationssymmetrie ist die Rotationsabweichung um die Z-Achse (Winkel α) ohne Bedeutung. Insgesamt zeigen Translationsabweichungen mit einem Fehler von unter 1 mm und die maximalen Rotationsabweichungen von $1,04^\circ$, dass die implementierten Verfahren zur Planung und Registrierung mit einer für den klinischen Einsatz benötigten Genauigkeit ausgeführt werden können.

Bei diesem Szenario setzt sich die ermittelte Abweichung aus mehreren Komponenten zusammen. Bedingt durch die begrenzte Aufnahmequalität und die Auflösung von $0,5\text{ mm}$ Kantenlänge eines Bildvoxels des 3D-Datensatzes, ergibt sich bereits in der Planung eine begrenzte Genauigkeit. Bei der registrierten 3D-Aufnahme entsteht ein Registrierungsfehler, bedingt durch die angefertigte Aufnahme des Kalibrierkörpers und der Registrierung der detektierten Kugelmarker in der Kalibrierkörperaufnahme. Hinzu kommt der Fehler des Lokalisierensystems, der vom Hersteller mit einer Genauigkeit von $0,12\text{ mm}[RMS]$ für die Detektion von Einzelmarkerkugeln angegeben ist [73]. Auch die Geometrie der Referenzkörper beeinflusst die Genauigkeit mit der diese von dem Lokalisierensystem detektiert werden, was bereits in einer vorhergehenden Arbeit untersucht wurde [53]. Durch die Berücksichtigung der Designrichtlinien [75] bei der Erstellung der Referenzkörper wird dieser Fehler minimal gehalten.

Dieses Szenario präsentiert die Ergebnisse des chirurgischen Ablaufs von der Akquise des registrierten Bilddatensatzes über die Planung, bis hin zur Überprüfung der tatsächlichen Position durch das Zeigeinstrument.

6.5.2 Positioniergenauigkeit des Systems am Beispiel einer intraoperativen Planung

Das zweite Szenario bildet die erste chirurgische Workflow Variante nach. Dabei wird eine registrierte intraoperative 3D-Aufnahme eines Wirbelmodells zur Planung verwendet. Hierfür wurde an dem Wirbelmodell ein Referenzkörper, der mit dem Patientenkoordinatensystem $\{ARB\}$ bezeichnet wird, angebracht. Die Bilddaten wurden mit einem intraoperativen C-Bogen (Vario 3D, Ziehm) aufgenommen. Zusätzlich wurden an dem Wirbelmodell Metallstifte als Landmarken angebracht, die in dem Bilddatensatz erkennbar sind. Zur Überprüfung der Positioniergenauigkeit wird eine Schraube auf die Spitze des Metallstiftes geplant. Unter Verwendung des Registrierungsergebnisses, kann die Planung einer Schraubenposition im

Patientenkoordinatensystem $\{ARB\}$ erfolgen. Hierbei wird die geplante Schraubenposition, wie im ersten Szenario, durch die Transformation ${}^{ARB}\underline{T}_{screw}$ angegeben. Anschließend wird die geplante Schraubenposition mit dem Roboter angefahren. Dabei tritt eine Abweichung zwischen der erreichten Position und der tatsächlichen Zielposition auf den Metallstiften des Wirbelmodells auf. Um diese Abweichung zu bestimmen wird das Wirbelmodell manuell an die tatsächliche Zielposition verschoben. Diese Verschiebung entspricht der Abweichung für diese Zielposition und kann von dem Stereokamerasystem gemessen werden. Die Abbildung 6.48 zeigt das verwendete Wirbelmodell mit dem Patienten Referenzkörper und den angebrachten Metallstiften. Der Roboterarm ist mit einem Metallstab als chirurgisches Werkzeug ausgestattet, der als Platzhalter für eine chirurgische Schraube mit Schraubendreher dient. Die Spitze dieses Metallstabes entspricht der Spitze einer Schraube. Diese Spitze wird an der tatsächlichen Zielposition des Wirbelmodells platziert.

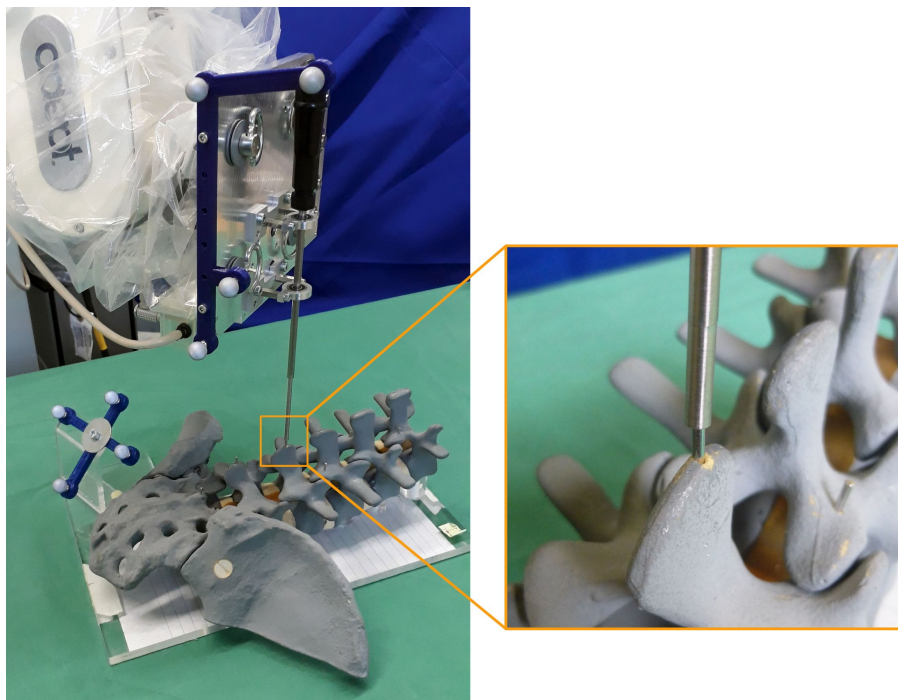


Abbildung 6.48: Verwendetes Wirbelmodell und das Robotersystem, platziert über dem Metallstift, der als Zielposition geplant wurde

Zur Überprüfung des Ergebnisses wird die Werkzeugspitze im Bilddatensatz der Planung eingeblendet. Die Abbildung 6.49 zeigt die geplante Schraubenposition auf der Spitze des Metallstiftes sowie die eingeblendete Werkzeugspitze des Roboters.

Mit dem Lokalisiersystem kann die Position des Roboterreferenzkörpers $\{RRB\}$ und die Position des Patientenreferenzkörpers $\{ARB\}$ bestimmt werden. Aus der im Abschnitt 6.2.4 beschriebenen Werkzeugvermessung ergibt sich die Transformation der

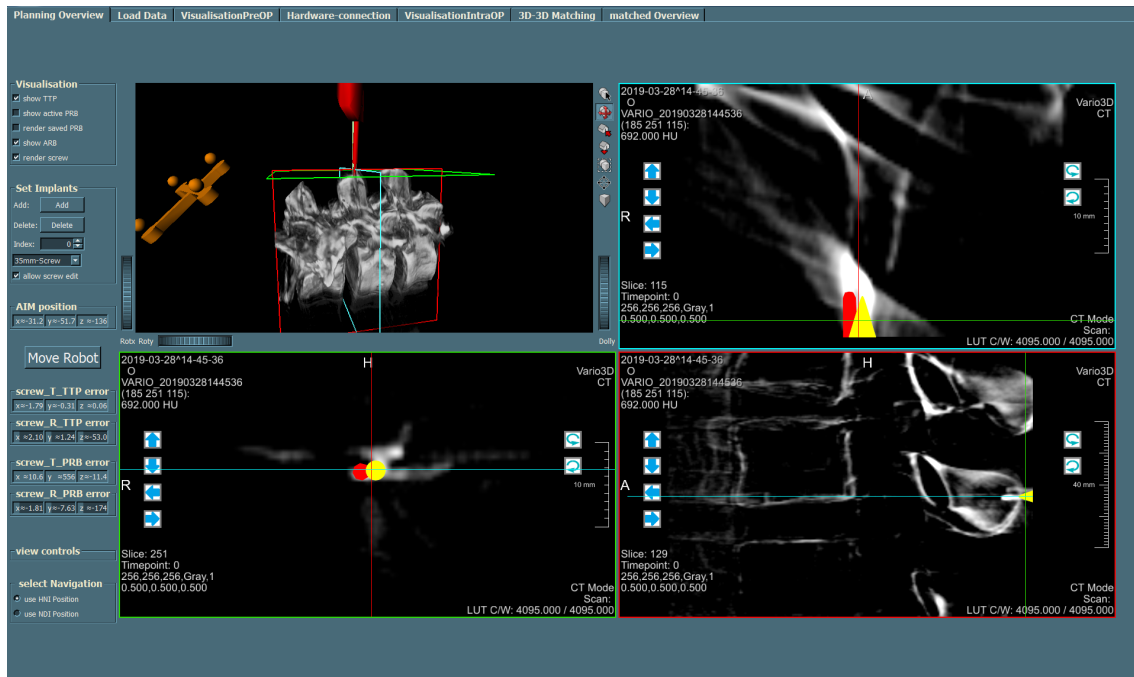


Abbildung 6.49: Planung einer Schraubenposition (gelb) im intraoperativen 3D-Bilddatensatz des Wirbelmodells mit der eingblendeten Spitze des Werkzeugs am Roboter (rot) der an der Zielposition positioniert wurde

Werkzeugs Spitze im Koordinatensystem von $\{RRB\}$ welche durch die Transformation ${}^{RRB}\underline{T}_{TTP}$ beschrieben wird und über die HNI-Schnittstelle vom Robotersystem übertragen wird. Die Formel 6.43 gibt die Transformation der Werkzeugs Spitze des Roboters $\{TTP\}$ im Koordinatensystem von $\{ARB\}$ an.

$${}^{ARB}\underline{T}_{TTP} = {}^{OTS}\underline{T}_{ARB}^{-1} \cdot {}^{OTS}\underline{T}_{RRB} \cdot {}^{RRB}\underline{T}_{TTP} \quad (6.43)$$

Nach der Positionierung des Roboters an der geplanten Zielposition kann aus der erfassten Position der Werkzeugs Spitze des Roboters, im Koordinatensystem von $\{ARB\}$, beschrieben durch die Transformation ${}^{ARB}\underline{T}_{TTP}$ und der geplanten Zielposition ${}^{ARB}\underline{T}_{screw}$ die Abweichung bestimmt werden.

$${}^{screw}\underline{T}_{TTP} = {}^{ARB}\underline{T}_{screw}^{-1} \cdot {}^{ARB}\underline{T}_{TTP} \quad (6.44)$$

Dabei beschreibt der Translationsvektor der Transformation ${}^{screw}\underline{T}_{TTP}$ die ermittelte Positionsabweichung zwischen der tatsächlichen Werkzeugs Spitze und

der geplanten Schraubenposition. Die Tabelle 6.4 zeigt die ermittelten Positionsabweichungen.

Achse	Positionsabweichung [mm]
X	-1.79
Y	-0.31
Z	0.06

Tabelle 6.4: Positionsabweichung zwischen der in der C-Bogen Aufnahme geplanten Schraubenposition und der gemessenen Werkzeugspitzenposition

Diese Ergebnisse zeigen größere Abweichungen als die Ergebnisse des ersten Szenarios. Dies liegt zum einen daran, dass hier der Planungsfehler größer ist, da die zur Planung verwendeten Metallstifte und insbesondere die genaue Lage des Zentrums der Metallstifte, nur ungenau in dem Datensatz lokalisiert werden können. Dies verdeutlicht insbesondere die grün umrahmte koronale Schichtansicht der Abbildung 6.49. Zum anderen kommt zu dem Registrierungsfehler des ersten Szenarios noch der Fehler der Werkzeugvermessung am Roboterflansch hinzu. Da die Position des Werkzeugs über den angebrachten Referenzkörper am Roboterflansch erfasst wird, geht die Positioniergenauigkeit des Robotersystems nicht mit in die Fehlerbetrachtung ein. Da in dem Datensatz die Spitze des Metallstiftes nicht abgebildet ist, kann der Positionsfehler entlang der Schraubenachse, die der Z-Richtung entspricht, nicht verifiziert werden und ist daher nicht zur weiteren Betrachtung heranzuziehen. Dieser gibt die Schraubentiefe an und wird für den klinischen Workflow nur für die Visualisierung herangezogen. Bei dem tatsächlichen Einbringen einer Schraube im klinischen Umfeld verlassen sich die Chirurgen bei dem Eindrehen einer Schraube auf ihr Handgefühl, das durch ein erhöhtes Drehmoment an der Zielposition der Schraube entsteht, wenn diese vollständig eingedreht wurde sodass der Schraubenkopf auf dem Knochen aufliegt.

Für den klinischen Workflow relevant sind die Abweichungen in der X- und Y-Achse relativ zur Schraubenachse, die in diesem Szenario dargestellt werden.

6.5.3 Positioniergenauigkeit des Systems am Beispiel einer präoperativen Planung

Im dritten Szenario wird die zweite Workflow Variante mit einer präoperativen Planung untersucht. Hierzu wird eine CT-Aufnahme des in der Abbildung 6.38 darge-

stellten Wirbelmodells für die Operationsplanung eingesetzt. Als Zielposition werden, wie im Abschnitt 6.5.2, die am Wirbelmodell angebrachten Metallstifte verwendet. Hierfür wird eine Schraube exakt über der Spitze des Metallstiftes geplant. Die Abbildung 6.50 zeigt die in der hochauflösenden CT-Aufnahme geplante Schraubenposition. Besonders im Vergleich zur intraoperativen C-Bogen Aufnahme aus der Abbildung 6.49 wird hier deutlich, dass die höhere Bildqualität eine deutlich präzisere Planung ermöglicht.

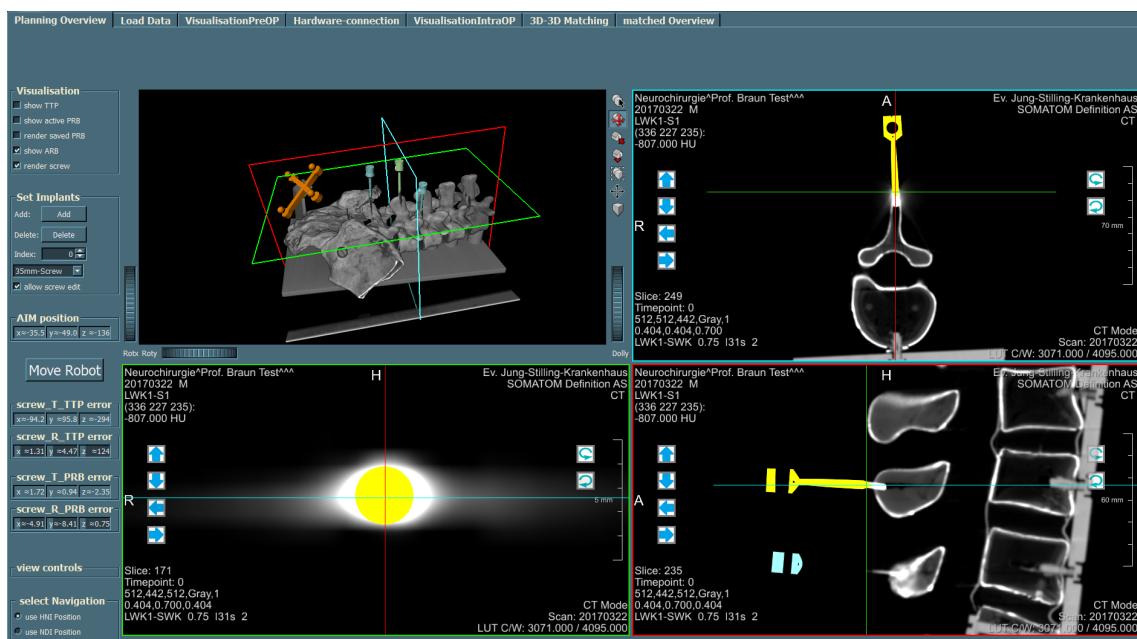


Abbildung 6.50: Planung einer Schraubenposition (gelb) im präoperativen CT Bild-datensatz des Wirbelmodells

Anschließend wird eine registrierte 3D-Aufnahme des Wirbelmodells mit einem intraoperativen C-Bogen (Ziehm Vario 3D) aufgenommen. Zur besseren Vergleichbarkeit mit dem Ergebnis der präoperativen Planung wird hierfür dieselbe Aufnahme wie im Abschnitt 6.5.2 verwendet. Mit dem im Abschnitt 6.4.2 beschriebenen Verfahren wird die präoperative CT-Aufnahme mit dem intraoperativen C-Bogen Bild-datensatz überlagert. Hierdurch können die im CT-Bildvolumen geplanten Schraubenpositionen in das Bildvolumen der C-Bogen Aufnahme übertragen werden und im Koordinatensystem des Referenzkörpers an dem Wirbelmodell {ARB} beschrieben werden. Die Abbildung 6.45 zeigt das Ergebnis nach dem 3D Matching, bei der drei in der CT-Aufnahme geplante Schraubenpositionen auf den intraoperativen C-Bogendatensatz übertragen wurden.

Anschließend wird eine Schraubenposition mit dem Robotersystem angefahren. Nachdem die Zielposition erreicht wurde, wird das Wirbelmodell zur Bestimmung der

aufgetretenen Abweichung, wie bereits im Abschnitt 6.5.2 in der Abbildung 6.48 dargestellt, an der geplanten Zielposition positioniert. Die Abbildung 6.51 zeigt die abschließend in der präoperativen Aufnahme eingeblendete Werkzeugspitze des Roboters.

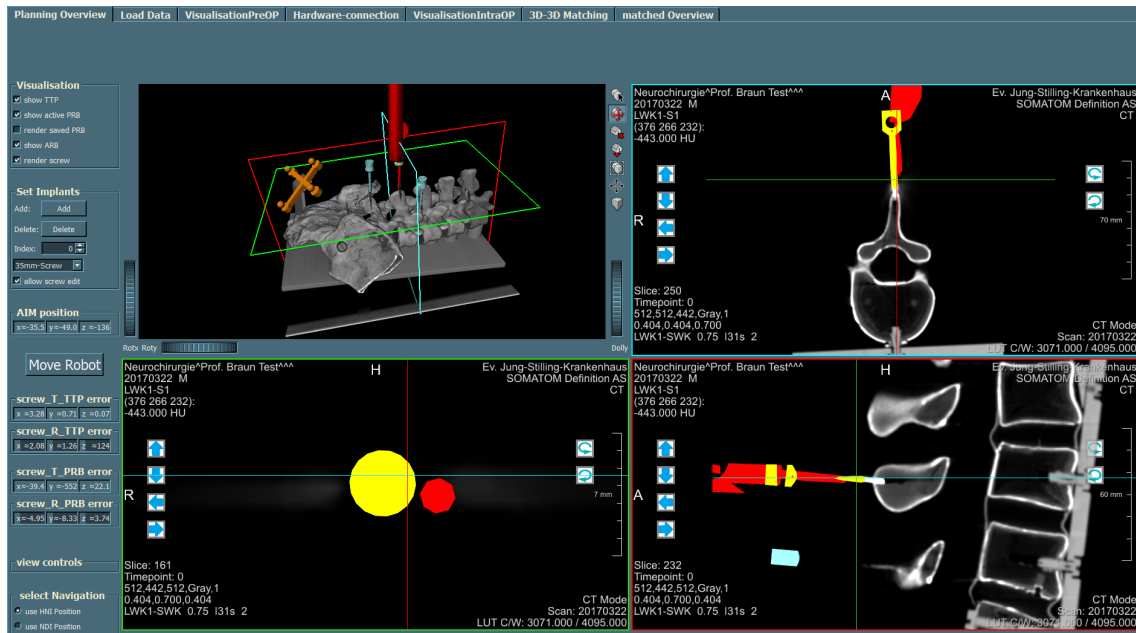


Abbildung 6.51: Planung einer Schraubenposition (gelb) im präoperativen 3D-Bild-datensatz des Wirbelmodells mit der eingeblendeten Spitze des Werkzeugs am Roboter (rot)

Zum Vergleich wird die Werkzeugspitze zusätzlich in der intraoperativen C-Bogen Aufnahme eingeblendete. Dies ist in der Abbildung 6.52 dargestellt.

Insbesondere die koronale Schichtansicht (grün umrandet) der Abbildungen 6.51 und 6.52 visualisieren die aufgetretene Abweichung. Da ein Datensatz mit einer abweichenden Patientenausrichtung aufgenommen wurde, sind die Visualisierungen der Schichtansichten aus unterschiedlichen Perspektiven dargestellt. Während der präoperative Datensatz die koronale Schichtansicht (grün umrandet) vom Rücken in Richtung Bauchdecke darstellt, zeigt die intraoperative Visualisierung diese Schichtansicht aus der umgekehrten Richtung. Dies hat keinen Einfluss auf das Ergebnis, erklärt aber warum im intraoperativen Datensatz die Werkzeugspitze linksseitig von der geplanten Schraubenposition dargestellt ist, wogegen diese im präoperativen Datensatz rechtsseitig von der geplanten Schraubenposition angezeigt wird.

Da die geplante Schraubenposition nach dem Matching mit dem intraoperativen Datensatz im Koordinatensystem von $\{ARB\}$ vorliegt, kann diese mit der gemessenen

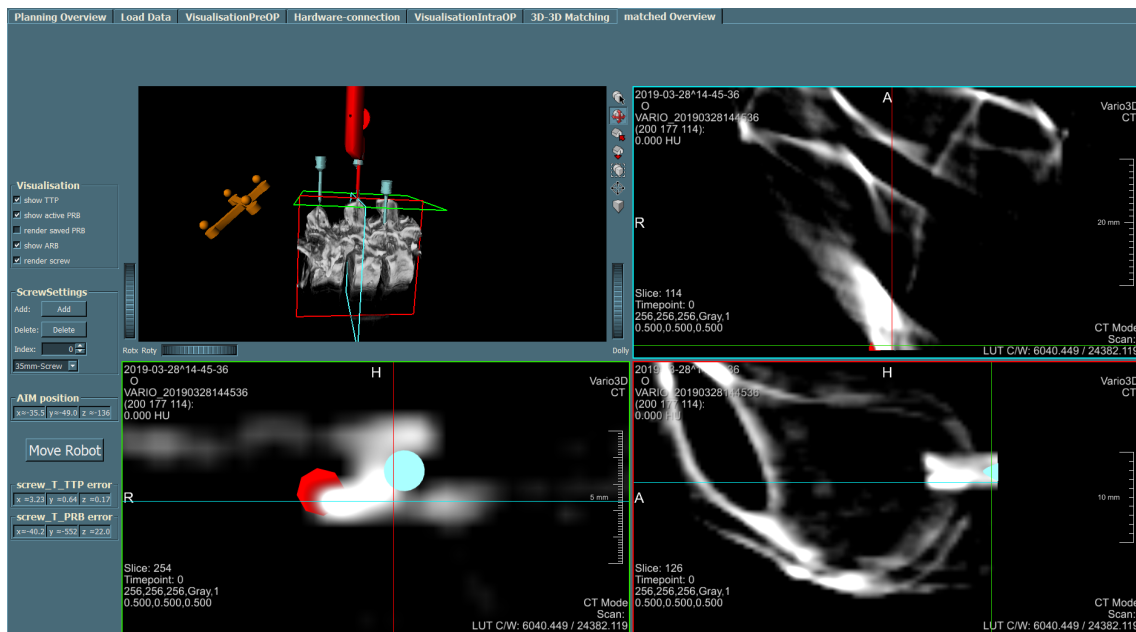


Abbildung 6.52: Nach dem Matching mit der präoperativen Planung übertragene Schraubenposition (blau) im intraoperativen C-Bogen Bilddatensatz des Wirbelmodells mit der eingblendeten Spitze des Werkzeugs am Roboter (rot)

Position der Werkzeugspitze des Roboters {TTP} an der Zielposition verglichen werden. Nach dem durchgeführten Matching gelten daher dieselben Formeln 6.43 und 6.44 zur Berechnung der Abweichung zwischen der geplanten Schraubenposition und der tatsächlichen Zielposition. Die Tabelle 6.5 zeigt die ermittelten Abweichungen der Zielposition in der X-, Y- und Z-Richtung.

Achse	Translationsabweichung [mm]
X	3.28
Y	0.71
Z	0.07

Tabelle 6.5: Translationsabweichung zwischen der in der präoperativen Aufnahme geplanten Schraubenposition und der gemessenen Werkzeugspitzenposition

Im Vergleich zu den Ergebnissen der intraoperativen Planung aus dem Abschnitt 6.5.3 fällt die Translationsabweichung größer aus. Einen Grund dafür bildet das Matching des CT-Datensatzes mit dem intraoperativen C-Bogen Datensatz. Da dieselben intraoperativen Datensätze verwendet wurden, verdeutlicht die erhöhte Translationsabweichung aus der Tabelle 6.5 gegenüber den Ergebnissen aus der Tabelle 6.4 die Abweichung, die durch das Matching hinzugekommen ist.

7 Operationsplanung und -durchführung in der Weichteilchirurgie

In der Weichteilchirurgie liegt ein besonderes Augenmerk auf sensiblen Bereichen wie Blutgefäßen und Organen. Um sensible Bereiche nicht zu beschädigen und im Körper liegende Operationsgebiete zu erreichen, werden flexible Operationsinstrumente eingesetzt. Auf Grund der weichen und beweglichen Organe, kann die Operationsplanung nicht analog zur knöchernen Chirurgie erfolgen. Insbesondere komplizierte Operationen oder auch Eingriffe an schwer zugänglichen Operationsgebieten werden daher meist nicht minimalinvasiv durchgeführt. Dies hat auch eine erhöhte Belastung und längere Erholungszeit für den Patienten zur Folge. Ein Beispiel hierfür ist die totale mesorektale Exzision, bei der zur Behandlung einer Krebserkrankung, Teile des Enddarms entfernt werden [3].

Einen neuen Ansatz hierfür bietet der im Rahmen des STIFF-FLOP Projektes entwickelte flexible Roboterarm mit der Möglichkeit sich zu versteifen, der im Abschnitt 7.1 vorgestellt wird. Um den flexiblen Roboterarm sicher im Operationsgebiet zu bewegen, sind in dem Arm Sensoren integriert. Zur Überwachung der Position im Operationsgebiet soll ein optisches Lageüberwachungssystem aufgebaut werden. Zur intraoperativen Navigation setzen Chirurgen endoskopische Kameras ein. Diese Bild-daten sollen verwendet werden, um den flexiblen Roboterarm zu detektieren und Rückschlüsse auf die Position zu ermöglichen. Hierbei wird die Position des Endoskops mit einer Stereokamera überwacht.

Das aufgebaute Lageüberwachungssystem wird im Abschnitt 7.2 beschrieben. Den Schwerpunkt der eigenen Beiträge bildet die Erkennung des flexiblen Roboterarmes im Bild einer Endoskopkamera. Hierfür wurde ein Verfahren des maschinellen Lernens angewendet, welches im Abschnitt 7.3 beschrieben wird.

Teile des entwickelten Lageüberwachungssystems sind bereits in einer Dissertation veröffentlicht [64], auf die an entsprechenden Stellen verwiesen wird.

7.1 Beschreibung des verwendeten flexiblen Roboterarmes

Im Rahmen des EU-Projektes STIFF-FLOP wurde in Kooperation mit 8 europäischen Partnern der flexible STIFF-FLOP Roboterarm entwickelt [44]. Die Inspiration hierfür lieferte die Bewegung des Oktopusses, der sich durch kleine Öffnungen bewegen kann und seine Arme ausstrecken und versteifen kann, um Nahrung zu ergreifen. Ziel des Projektes war die Entwicklung eines flexiblen Roboterarmes, der durch einen minimalinvasiven Zugang das Operationsgebiet erreicht und sich versteifen kann. In der technischen Umsetzung wurden flexible Armsegmente entwickelt, die sich verbinden lassen. Die Aktivierung des Armes erfolgt pneumatisch, was einen sicheren Einsatz bei minimalinvasiven Eingriffen ermöglicht. Eine erste Version des entwickelten Armes bestand aus Silikon und verfügte über drei interne Luftkammern. Über den Druck in den Luftkammern konnte der Arm gebogen werden. Um eine Expansion des Armes zu unterbinden, wurde der Arm von einer gewobenen Hülle umschlossen. Die Abbildung 7.1 zeigt eine erste Version des entwickelten Armes, bestehend aus zwei fest verbundenen Modulen.



Abbildung 7.1: Erste Version des STIFF-FLOP Armes, bestehend aus 2 Arm Segmenten mit einer gewobenen äusseren Hülle. Quelle: Projekt Newsletter

Im Projektverlauf wurde dieser Arm weiterentwickelt. Um die Expansion des Armes zu unterdrücken und gleichzeitig die Bewegungsfähigkeit zu verbessern, wurden drei unabhängig kontrollierbare doppelte Luftkammern integriert. Durch eine Umman- telung der Luftkammern wird eine Expansion verhindert und nur eine Elongation zugelassen. Wenn kein Druck an den Luftkammern anliegt, verkürzt sich der Arm auf seine Ursprüngliche Größe. Der Druck in den Kammern wird von Druckreg- lern, die an ein RoNeX Modul (Shadow Robot company, London) angeschlossen sind, kontrolliert. Über den Druck in den Kammern lässt sich der Arm in beliebige

Richtungen biegen. In zusätzlichen Kammern befindet sich ein Granulat, das durch ein Vakuum versteift werden kann. Hierdurch kann der Arm in beliebigen Positionen versteift werden. Die Abbildung 7.2 zeigt den schematischen Aufbau und die Abbildung eines flexiblen STIFF-FLOP Arm Moduls.

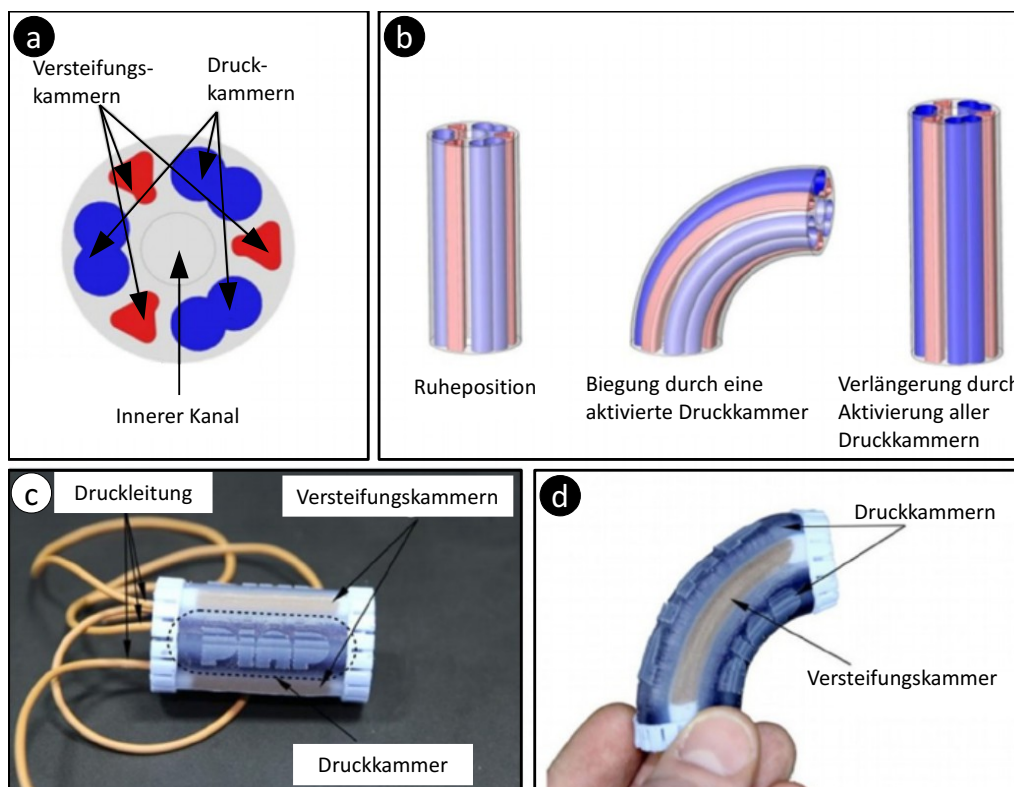


Abbildung 7.2: Aufbau eines STIFF-FLOP Manipulator Moduls. Die Abbildung (a) visualisiert den schematischen Querschnitt des Moduls. Die Abbildung (b) zeigt die schematische Darstellung des Moduls in der Ruheposition, bei Druckerhöhung in einer Luftkammer und bei einer Druckerhöhung in allen Luftkammern. Die Abbildung (c) zeigt ein STIFF-FLOP Modul und Abbildung (d) zeigt die Biegung eines Moduls. Quelle:[3]

In den Verbindungsstücken zwischen den Modulen sind neu entwickelte optische Kraft-Moment Sensoren integriert, die die Krafteinwirkung auf den Arm erfassen und bei der Steuerung verwendet werden [71]. Zusätzlich wird die ermittelte Krafteinwirkung als haptische Rückmeldung über das zur Steuerung eingesetzte Eingabemodul an den Chirurgen weitergeleitet. Gefertigt wird der Arm aus flexiblem Silikon [84] [37]. Für den Einsatz des Armes können bis zu 3 Elemente verbunden werden und bieten an der Spitze der Modulkette eine Aufnahme für verwendbare Werkzeuge. Im Rahmen des Projektes wurde der Einsatz eines pneumatisch aktivierten Greifers, eines Ablationsinstruments und eines Kameramoduls getestet.

7.2 Aufbau des optischen Positionsbestimmungssystems

Um den STIFF-FLOP Arm gezielt zu dem Operationsgebiet zu bringen, wurde dieser an dem Flansch eines Roboters angebracht (Leichtbauarm LWA, Schunk). Als Testszenario wurde für den flexiblen STIFF-FLOP Arm ein Phantommodell des Abdomens aufgebaut [62]. Hierbei erfolgt der Zugang durch einen Trokar. Die Steuerung des Armes wurde so ausgelegt, dass dieser in Kombination mit dem Schunk Roboterarm durch einen fixierten Zugangspunkt gesteuert werden kann. Ein weiterer Trokar Zugang ermöglicht den Einsatz einer endoskopischen Kamera mit einer starren Optik. Innerhalb des Phantommodells wird dem Chirurgen hierdurch ein Kamerabild zur Verfügung gestellt, das zur Orientierung und Navigation dient. Dieses Kamerabild soll verarbeitet werden, um darin den STIFF-FLOP Arm zu detektieren und die Position des Armes zu überwachen.

Als Kamerasystem wird eine laparoskopische Endoskopkamera (HD Endocam 5509, Richard Wolf) mit einer zugehörigen Lichtquelle eingesetzt. Zusätzlich wird die Position der Endoskopkamera und der Basis des STIFF-FLOP Armes mit einer Stereokamera (Cambar B2, Axios) überwacht. Hierdurch kann die Position der Endoskopkamera im Koordinatensystem der Basis des STIFF-FLOP Armes bestimmt werden. Dadurch ist es möglich, die Position des Armes im Endoskopkamerabild relativ zu der Basis des Armes zu bestimmen, sodass die Positionsdaten für eine Weiterverarbeitung zur Verfügung stehen [41].

Die Abbildung 7.3 zeigt den STIFF-FLOP Arm, bestehend aus 2 Modulen, über dem Phantommodell. Hier ist jeweils ein Zugangspunkt für den STIFF-FLOP Arm und die verwendete Endoskopkamera integriert. In den rechten Abbildungen ist die verwendete Stereokamera und das Endoskop Kamera System dargestellt.

Die im Rahmen des STIFF-FLOP Projektes entwickelte Software zur Steuerung des Manipulators wurde mit dem Robot Operating System (ROS) aufgebaut. ROS ist ein open source Meta-Betriebssystem, das bereits vielfältige Funktionen zur Ansteuerung, Simulation und Visualisierung von Robotersystemen enthält. Die Kommunikation innerhalb von ROS läuft über den „*ROS – Master*“ der auf einem Rechner laufen muss. Eigene C++ oder Python Funktionen werden in sogenannten „*ROS – Nodes*“ implementiert, die sich mit dem Master verbinden und auf unterschiedlichen Rechnern laufen können, während der Master die gesamte Kommunikation verwaltet. Der Datenaustausch erfolgt asynchron über „*ROS – Topics*“, mit denen Nach-

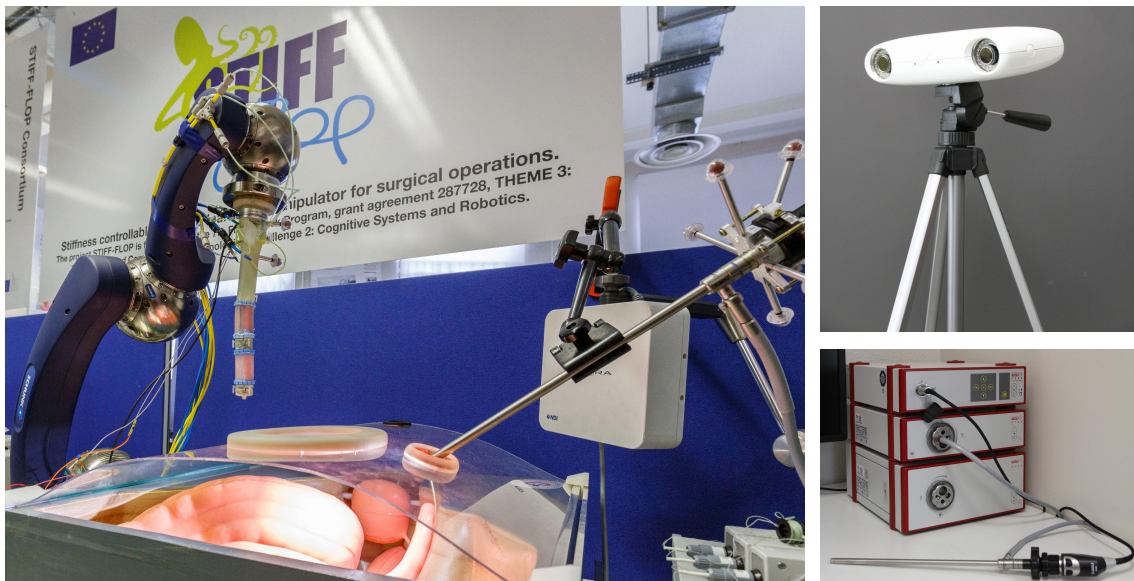


Abbildung 7.3: STIFF-FLOP Arm bestehend aus 2 Modulen, montiert an einem Schunk Roboterarm über einem Phantommodell (links). Optische Referenzkörper an der STIFF-FLOP Arm Basis und an der eingesetzten Endoskopkamera (unten rechts), werden von der Stereokamera (oben rechts) überwacht.

richten mit vordefinierten oder selbst erstellten Datenstrukturen von einer „*ROS – Node*“ publiziert werden. Andere „*ROS – Nodes*“ können diese Nachricht über einen Subscriber-Befehl abonnieren und erhalten automatisiert alle neu gesendeten Nachrichten. Hierüber können einzelne „*ROS – Nodes*“ unabhängig entwickelt werden. Über „*ROS – Services*“ lassen sich auch direkte Anfrage-Antwort-Interaktionen zwischen „*ROS – Nodes*“ implementieren. Über den „*ROS – Parameterserver*“ können Parameter zentral gespeichert werden und von beliebigen „*ROS – Nodes*“ ausgelesen und angepasst werden. Zusätzlich bietet ROS die Möglichkeit versendete „*ROS – Topics*“ in „*Bag – files*“ aufzuzeichnen. Hiermit können Sensordaten aufgezeichnet werden und anschließend durch Abspielen simuliert werden, sodass die Entwicklung auch ohne einen Betrieb der Sensorik erfolgen kann.

Zur Verarbeitung von Bilddaten wird die open source Bibliothek openCV (open Computer Vision) eingesetzt. Diese enthält Funktionen zur Segmentierung, Objekt-, Bewegungs- und Gestenerkennung, sowie Algorithmen des maschinellen Lernens. Die Algorithmen sind für eine effiziente Ausführung optimiert und verwenden dafür parallele Verarbeitungsmöglichkeiten. Weitere Beschreibungen der verwendeten Software und Bibliotheken sind in den Quellen [34], [19], [35], [98] zu finden.

Um die Beiträge des optischen Lageüberwachungssystems in die Software des STIFF-FLOP Projektes zu integrieren, wurden diese, unter Verwendung von ROS, entwickelt. Ziel ist es, dass die Positionsdaten zur weiteren Sensordatenfusion zur Verfügung stehen. Für die Umsetzung des optischen Lageüberwachungssystems wurden verschiedene „*ROS – Nodes*“ entwickelt, die nachfolgend vorgestellt werden. Die Abbildung 7.4 zeigt die Softwarearchitektur der implementierten Module.

In der ROS Node „*Intensity*“ werden die Bilddaten der Endoskopkamera erfasst und vorverarbeitet. Als Ausgang wird hier das Kamerabild der Endoskopkamera als ROS-Topic zur Verfügung gestellt. Die ROS-Node „*cambar*“ baut die Verbindung zur verwendeten Stereokamera auf und sendet ein ROS-Topic mit den detektierten Referenzkörperpositionen. Die Detektion des STIFF-FLOP Armes im Bild der Endoskopkamera erfolgt in der „*Tubedetect*“ Node, die im Abschnitt 7.3.6 beschrieben wird. In der ROS-Node „*DetectTubeConnectors*“ wurde eine Methode zur Detektion der Verbindungspunkte des STIFF-FLOP Armes umgesetzt, die in der Dissertation [64] vorgestellt ist. Über den „*DynamicReconfigure*“ Parameterserver können Parameter für einzelne Module im laufenden Betrieb angepasst werden. Durch den von ROS zur Verfügung gestellten „*TransformationTree*“, können die Positionsdaten verschiedener Frames transformiert werden und stehen im Anschluss für die Datenfusion bereit.

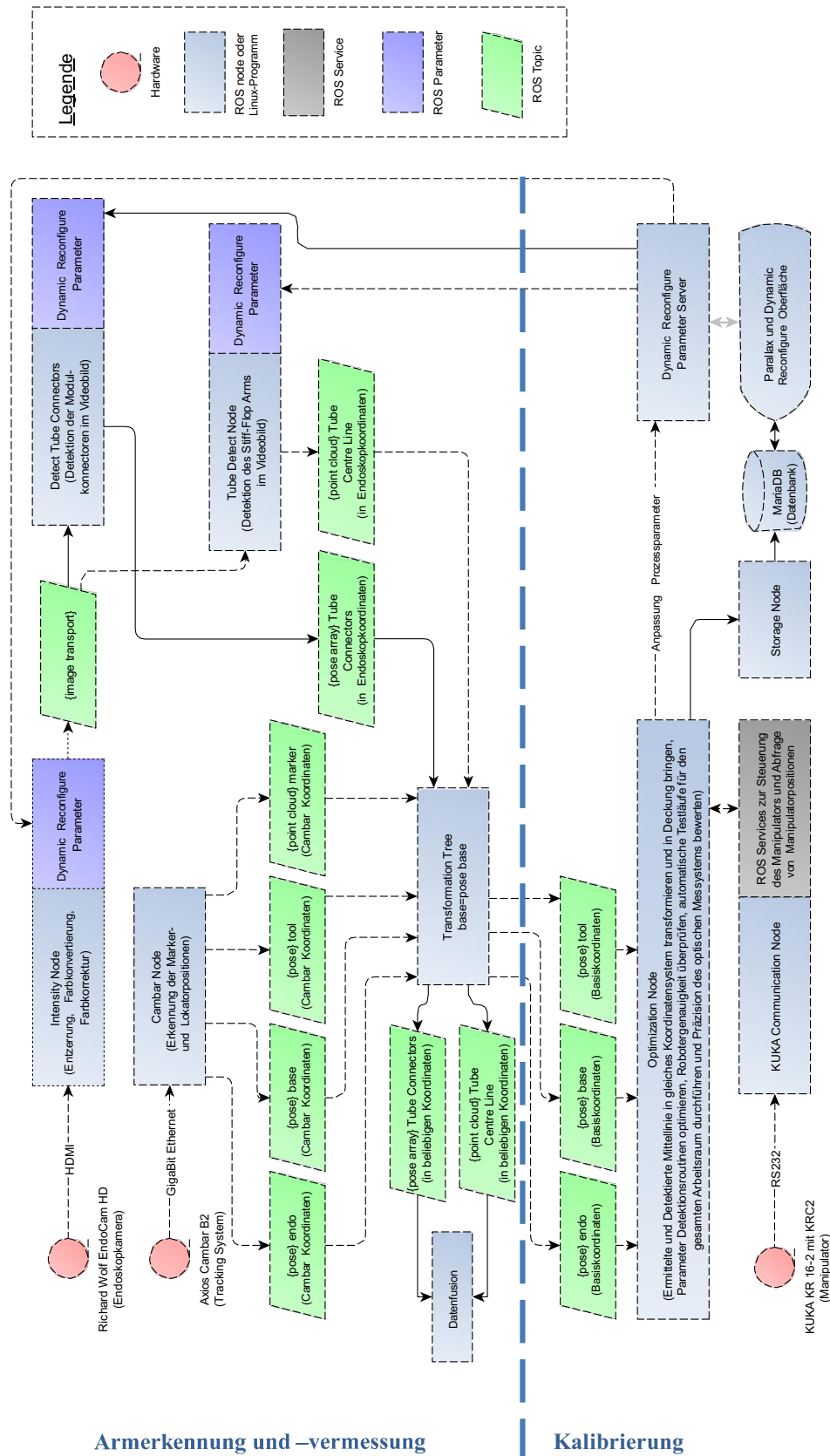


Abbildung 7.4: Softwarestruktur der implementierten ROS Module zur Positionsbestimmung des STIFF-FLOP Armes [64]

7.2.1 Lokalisation der intraoperativ verwendeten Komponenten

Zur intraoperativen Lokalisation wird die Stereokamera (Cambar B2, Axios) an das System angebunden. Die Kamera ist über einen Gigabit Ethernet Anschluss mit dem Rechner verbunden. Die Positioniergenauigkeit eines Punktfeldes wird vom Hersteller mit 0.25 mm in einem Messabstand von $80 - 240\text{ cm}$ angegeben [7]. Die Ansteuerung erfolgt durch die Einbindung der vom Hersteller zur Verfügung gestellten Kameratreiber in der „Cambar“ ROS-Node. Darüber hinaus stellt der Hersteller eine Software zum Einlernen neuer Referenzkörper zur Verfügung. Hiermit können erstellte Referenzkörper eingemessen und als XML Dokument abgespeichert werden. Dabei wird der Referenzkörper statisch vermessen und ein Koordinatensystem durch die Markerkugelanordnung festgelegt. Für den Betrieb der Kamera können diese Dateien geladen werden. Anschließend können die im Messbereich der Kamera liegenden Referenzkörper erkannt werden. Als Ausgabe liefert die „Cambar“ Node jeweils ein ROS-Topic für jeden erfassten Referenzkörper. Darin wird die aktuelle Position und Orientierung des Referenzkörpers im Koordinatensystem der Kamera angegeben. Die Implementierung und der Ablauf der „Cambar“ Node sind in der Dissertation [64, S.60] beschrieben.

Um die Position der endoskopischen Kamera und die Position der STIFF-FLOP Arm Basis mit dem Stereokamerasystem zu erfassen, werden diese mit optischen Referenzkörpern ausgestattet. Bedingt durch den Schnellverschlussmechanismus, der den endoskopischen Kamerakopf mit der Optik verbindet, traten Abweichungen in der Bildebene auf, da diese nicht fest miteinander verbunden waren. Um diese Verbindung zu versteifen und den optischen Referenzkörper an der endoskopischen Kamera zu befestigen, wurde ein Verbindungsstück aus Aluminium entworfen, gefertigt und an der endoskopischen Kamera angebracht. Bei einer Operation wird die Endoskopkamera bewegt, wodurch Markerkugeln des Referenzkörpers verdeckt werden können. Aus diesem Grund wurden 7 Markerkugeln geschickt an dem Referenzkörper angeordnet, sodass dieser aus verschiedenen Perspektiven zuverlässig erkannt werden kann, auch wenn einige Markerkugeln verdeckt sind. Der Entwurf des Verbindungselementes ist bereits in einer vorhergehenden Arbeit beschrieben [64, S.58]. Bei dem Einlernen des Referenzkörpers wird das Koordinatensystem in das Zentrum der endoskopischen Kameraebene gelegt. Die Abbildung 7.5 zeigt das entworfene Versteifungselement und dessen Anbringung an der endoskopischen Kamera.

Die Basis des STIFF-FLOP Armes bildet ein im 3D-Druck Verfahren hergestellter Kunststoffadapter, der den STIFF-FLOP Arm mit dem Schunk Roboterarm ver-



Abbildung 7.5: Modell des Versteifungselements für die Endoskopkamera mit Haltern für Markerkugeln (links) und das gefertigte Element an der Endoskopkamera (rechts) mit dem festgelegten Koordinatensystem, Quelle: [64]

bindet. Da diese Basis nicht in das Operationsgebiet eindringt und als rigide Basis für den STIFF-FLOP Arm dient, werden hier Markerkugeln für die Stereokamera angebracht. Um mögliche Verdeckungen durch Kabel und pneumatische Leitungen zu vermeiden und um eine einzigartige Markeranordnung, in Anlehnung an die herstellerspezifischen Designrichtlinien umzusetzen, werden die Referenzmarker über Aluminium Distanzstücke an der Basis angebracht und anschließend durch die vom Hersteller zur Verfügung gestellte Software angelernt. Nach einmaligem Anlernen der Referenzkörperbeschreibung, kann diese für die spätere Überwachung der Position geladen und eingesetzt werden. Das Koordinatensystem wird hierbei in dem Zentrum der Basis festgelegt. Die Abbildung 7.6 zeigt die STIFF-FLOP Basis mit zwei flexiblen STIFF-FLOP Modulen und den angebrachten Markerkugeln.

Zur Verknüpfung verschiedener Transformationen und Koordinatensysteme wird das ROS Package „TF“ verwendet [35]. Die von der „Cambar“ Node publizierten Positionsdaten der Referenzkörper sind bereits in dem für das TF Paket benötigten Format angelegt und können daher direkt verarbeitet werden. Da die Koordinatensysteme der Referenzkörper der endoskopischen Kamera und der STIFF-FLOP Basis im Koordinatensystem der Stereokamera vorliegen, können diese direkt miteinander verknüpft werden. Nach erfolgreicher Parametrierung erfolgt dies automatisiert mit dem „TF“ Package. Hier werden auch jegliche Änderungen der Transformationen, durch mögliche Bewegungen der Referenzkörper, berücksichtigt und immer die aktuell gültige Transformation verwendet. Damit wird ein sogenannter Transformationsbaum aufgebaut, der die verwendeten Koordinatensysteme verknüpft und die zu dem aktuellen Zeitpunkt gültigen Transformationen beinhaltet. Dieser Transformationsbaum kann um beliebig viele Transformationen erweitert werden und ermöglicht eine effiziente Umrechnung vorliegender Koordinatensysteme in ein gemeinsames

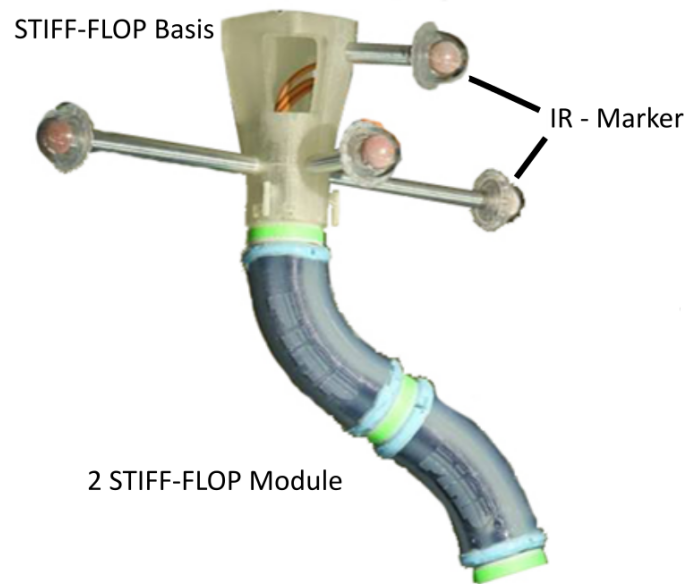


Abbildung 7.6: STIFF-FLOP Arm an dem Basiselement, das mit dem Schunk Roboter verbunden wird. An der Basis sind Markerkugeln für eine Detektion mit der Stereokamera angebracht. [43]

Weltkoordinatensystem. Die Anwendung des „TF“ Package ist in der Quelle [36] beschrieben.

Weitere Informationen zur Anbindung der Stereokamera und dem Entwurf der Referenzkörper sowie der auftretenden Fehler sind in der Dissertation [64] zu finden.

7.2.2 Bilderfassung und Bildvorverarbeitung

Die Bilddaten der endoskopischen Kamera werden über eine HDMI Verbindung mit einem Framegrabber (Intensity Pro, Blackmagic Design) aufgenommen. Das Eingangssignal wird mit einer Auflösung von 1980 x 1080 und einer Bildwiederholrate von 50 Bildern pro Sekunde erfasst. Um die Kameradaten weiter mit ROS verarbeiten zu können, müssen diese zunächst aufbereitet werden. Die Abbildung 7.7 skizziert den Ablauf der Bildvorverarbeitung innerhalb der ROS-Node „Intensity“.

Die Verarbeitung des vollständigen Videostroms im Full HD 50p Modus erfordert eine hohe Rechenleistung. Zur Reduktion der Rechenlast wurde die Option eingefügt, einzelne Frames zu überspringen oder auch die Auflösung zu reduzieren. Die Parameter zur Anzahl an übersprungenen Frames und zur Reduktion der Auflösung

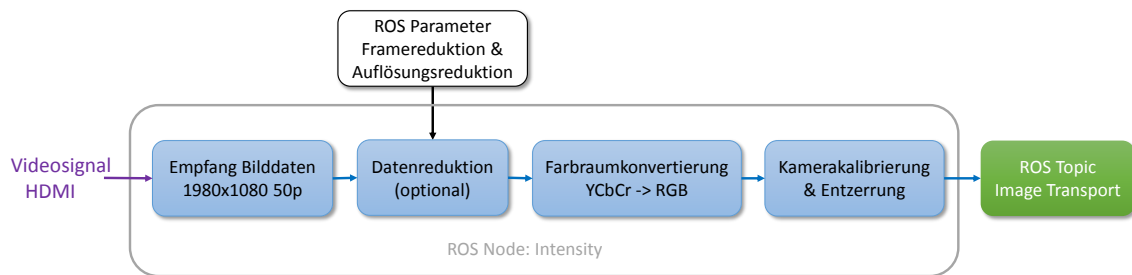


Abbildung 7.7: Schritte der Bildvorverarbeitung im Programmablauf der *Intensity* ROS-Node

können dynamisch während der Laufzeit des Programms über den ROS Parameter Server angepasst werden.

Die von der Kamera empfangenen Bilddaten liegen in einer Farbtiefe von 8-Bit im YCbCr 4:2:2 Farbraum vor. Hierbei beschreibt Y die Helligkeit, Cb die blau-gelb Chrominanz und Cr die rot-grün Chrominanz. Im 4:2:2 Format wird die Helligkeit aller Pixel abgetastet, jedoch erfolgt zur Videostromkomprimierung eine Unterabtastung in der horizontalen Bildebene, wobei sich jeweils 2 Pixel einen Chrominanzwert teilen.

Um die Bilddaten mit der openCV Bibliothek verarbeiten zu können und anschließend über ein ROS-Topic zu versenden, wird das CV:MAT Format verwendet. Da hierfür die Bilddaten in einer Farbtiefe von 8-Bit im RGB-Farbraum vorliegen müssen, werden die Bilddaten konvertiert. Die Formel 7.1 gibt die vom Hersteller angegebene Umrechnungsformel an [14, S.310].

$$\begin{aligned}
 R &= 1.164(Y - 16) + 1.793(Cr - 128) \\
 G &= 1.164(Y - 16) - 0.534(Cr - 128) - 0.213(Cb - 128) \\
 B &= 1.164(Y - 16) + 2.115(Cb - 128)
 \end{aligned}
 \tag{7.1}$$

Um Vermessungen in einem aufgenommenen Bild vorzunehmen, muss dies eine möglichst genaue Projektion der aufgenommenen Szene vorweisen. Ein reales Kamerabild unterliegt den Abbildungsparametern der verwendeten Komponenten. Dies beinhaltet die Brennweite der Kamera, eine mögliche Verschiebung oder Schrägheit der optischen Achse, sowie Skalierungen der Größe eines Bildpunktes auf dem Kame-rachip. Des Weiteren entstehen bei dem Einsatz von Linsen radiale und tangentiale Verzerrungen.

Zur Bestimmung dieser Parameter sind in der Literatur bereits verschiedene Verfahren dokumentiert. Zur Charakterisierung der endoskopischen Kamera wurde das „camera_calibration“ Paket von ROS verwendet. Dieses beruht auf der Kamerakalibrierung von openCV [77], die auf die Kamerakalibrierung nach den Methoden von Zhang [100] und anderen zurückgreift. Für die Kalibrierung wird ein bekanntes Schachbrettmuster aus verschiedenen Perspektiven aufgenommen. Die Abbildung 7.8 zeigt die Kamerakalibrierung mit ROS.

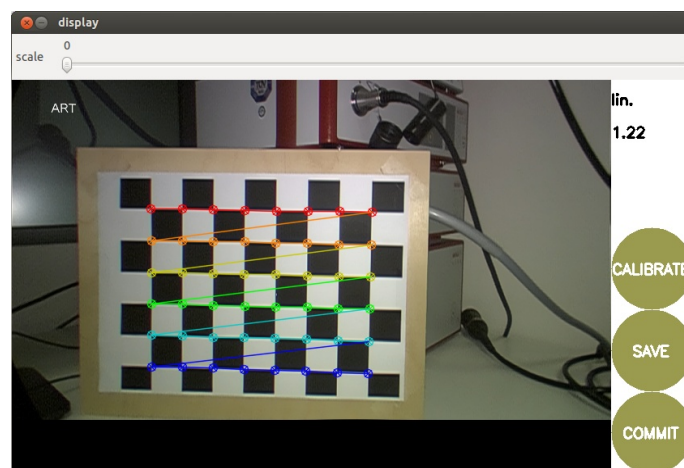


Abbildung 7.8: Kamerakalibrierung mit einem Schachbrettmuster in ROS [64]

Aus mehreren Aufnahmen werden die Eckpunkte des Schachbrettmusters detektiert. Daraus werden die Kameraparameter und die Verzerrungsparameter bestimmt. Anschließend wird das aufgenommene Bild entzerrt und steht für die Weiterverarbeitung zur Verfügung. Der Quellcode 7.1 zeigt die zur Entzerrung verwendeten Parameter der Kamerakalibrierung.

```
# Camera intrinsics
[image]

width: 1920
height: 1080

[EndoCam 5509]
camera matrix
1349.27603 0.00000 1028.25149 =K
0.00000 1323.66811 483.48426
0.00000 0.00000 1.00000

distortion
-0.15633 0.12363 -0.00093 -0.00011 0.00000 =D

rectification
1.00000 0.00000 0.00000 =R
0.00000 1.00000 0.00000
0.00000 0.00000 1.00000
```

projection				
1287.66870	0.00000	1026.13709	0.00000	=P'
0.00000	1289.85388	480.03962	0.00000	
0.00000	0.00000	1.00000	0.00000	

Quellcode 7.1: Ergebnisse der endoskopischen Kamerakalibrierung mit ROS [64]

Eine detaillierte Beschreibung der Kalibrierung der verwendeten endoskopischen Kamera ist in der Dissertation [64, S.36-46] zu finden.

7.3 Erkennung des flexiblen Roboterarmes in der Endoskop Aufnahme

In dem erfassten Videobild soll der STIFF-FLOP Arm detektiert werden. Den Ausgangspunkt dafür bildet die erste Version des STIFF-FLOP Armes aus der Abbildung 7.1. Hierfür sollte ein möglichst effizientes Verfahren implementiert werden, das eine echtzeitnahe Detektion des STIFF-FLOP Armes ermöglicht. Die Herausforderung liegt in der Detektion des flexiblen Armes, dessen Position sich im Bild verändert, weil der Arm oder die endoskopische Kamera bewegt werden. Bedingt durch den flexiblen Aufbau des Arms ist auch die Form und Orientierung des Arms variabel. Bei der Auswertung des Bildes kann zuvor auch noch nicht abgeschätzt werden, ob der vollständige Arm oder nur Teile hiervon von der Kamera erfasst werden. Da der Arm während der Projektlaufzeit weiterentwickelt wurde, soll die Detektion auch für weiterentwickelte Versionen des STIFF-FLOP Arms anwendbar sein.

Dies stellt besondere Herausforderungen an den Detektionsalgorithmus, wodurch klassische Ansätze zur Detektion von bekannten Formen oder Farben für dieses Anwendungsszenario nicht anwendbar sind. Für die Detektion des Arms ist insbesondere die äussere Beschaffenheit ausschlaggebend. Die erste Version des Arms erhielt durch die gewobene Ummantelung eine strukturierte Oberfläche, die als Ausgangspunkt für die Auswahl von Detektionsalgorithmen dient. Daher wurde hierfür die generalisierte Hough Transformation [8] sowie eine Merkmal basierte Detektion [59] zur Evaluation ausgewählt.

Während der Weiterentwicklung des STIFF-FLOP Arms entfiel die gewobene Hülle des Armes durch die Integration der Doppelkammerstruktur. Hierdurch änderten sich die äusseren Merkmale signifikant. Die Abbildung 7.9 zeigt ein Videobild des

finalen STIFF-FLOP Armes in dem Phantommodell. Dieses zeigt das Szenario zum Projektende, in dem der STIFF-FLOP Arm erfolgreich erkannt werden soll.

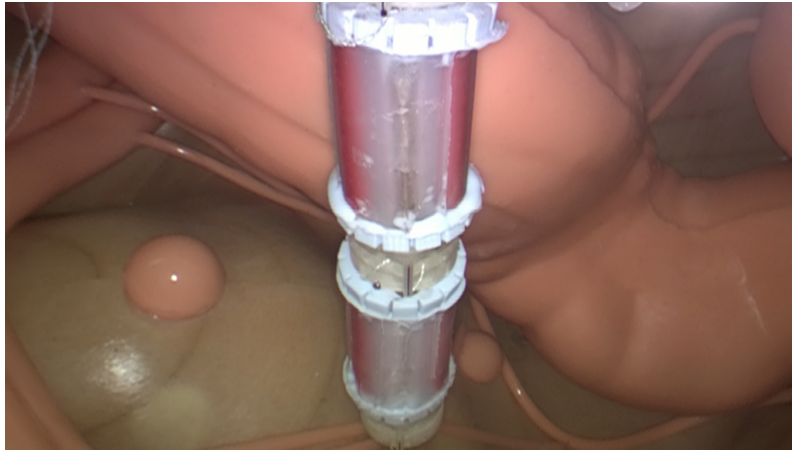


Abbildung 7.9: Endoskopische Aufnahme des weiterentwickelten STIFF-FLOP Armes im Phantommodell

Um eine anpassbare Detektion des STIFF-FLOP Armes zu ermöglichen, wurde ein Verfahren des maschinellen Lernens, basierend auf dem Einsatz von Support Vector Machines (SVM), implementiert. Durch den trainierbaren Algorithmus konnte dieser auch für weiterentwickelte Varianten des STIFF-FLOP Armes angewendet werden.

7.3.1 Anwendung der generalisierten Hough Transformation zur Detektion

Die Hough Transformation wurde ausgewählt, da diese in einer generalisierten Form, als Methode zur Detektion von beliebigen Formen vorgestellt wurde [8]. Daher wurde der Einsatz zur Detektion des STIFF-FLOP Armes im Rahmen einer betreuten studentischen Arbeit evaluiert [29]. Die Methode wird hier in zusammengefasster Form vorgestellt. Die umfassende Implementierung ist in der Arbeit [29] zu finden. Die Abbildung 7.10 skizziert den Programmablauf zur Erstellung der R-Tabelle und den Programmablauf der anschließenden Online Detektion des gesuchten Objektes im Videobild.

Im ersten Schritt wird ein Modell des gesuchten Objektes in einer Datenbank, der sogenannten R-Tabelle angelegt. Hierfür wird die Kontur des STIFF-FLOP Armes aus der Abbildung 7.1 bestimmt und dessen Schwerpunkt berechnet. Die Abbildung 7.11 zeigt die Kontur des STIFF-FLOP Armes aus der Abbildung 7.1.

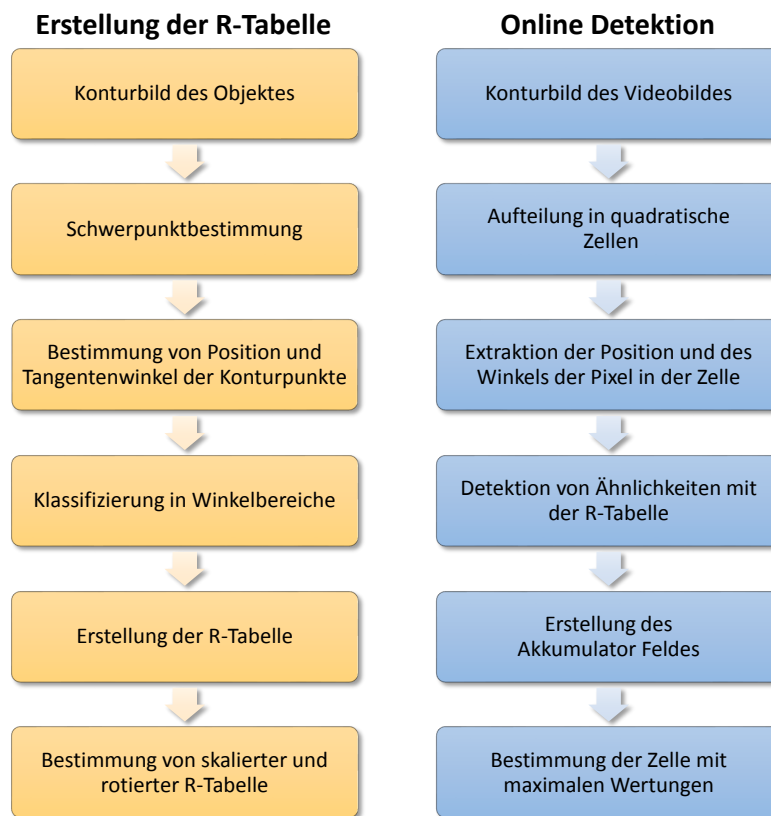


Abbildung 7.10: Programmablauf zur Erstellung der R-Tabelle und der anschließenden Online Detektion

Anschließend wird für jeden Punkt der Kontur, die relative Position \vec{r}_i zum Schwerpunkt, sowie der Winkel der Tangente des Konturpunktes Φ_i bestimmt. Diese Werte werden in n Winkelbereiche Θ_n klassifiziert und in der sogenannten Radius-Tabelle (R-Tabelle) abgelegt. Diese Werte dienen zur Beschreibung eines Modells des gesuchten Armes und werden gemäß der Tabelle 7.1 abgespeichert.

n	Θ_n	\vec{r}_i
0	$0, \Delta\Phi$	$r_1, r_2, r_3, \dots \{r_i \Phi \in \Theta_0\}$
1	$\Delta\Phi, 2\Delta\Phi$	$r_{14}, r_{21}, r_{23}, \dots r_i \Phi \in \Theta_1\}$
2	$2\Delta\Phi, 3\Delta\Phi$	$r_{10}, r_{30}, r_{34}, \dots r_i \Phi \in \Theta_2\}$
...

Tabelle 7.1: Format der R-Tabelle der generalisierten Hough Transformation zur Beschreibung eines Modells des gesuchten Objektes

Die erstellte R-Tabelle kann nun verwendet werden um das Objekt in dem Videobild zu detektieren. Hierzu wird zunächst eine Konturerkennung auf dem Videobild durchgeführt. Dafür wird ein Canny-Filter angewendet [23], für den es bereits eine Implementierung in openCV gibt. Anschließend wird das Bild in kleine Quadrate

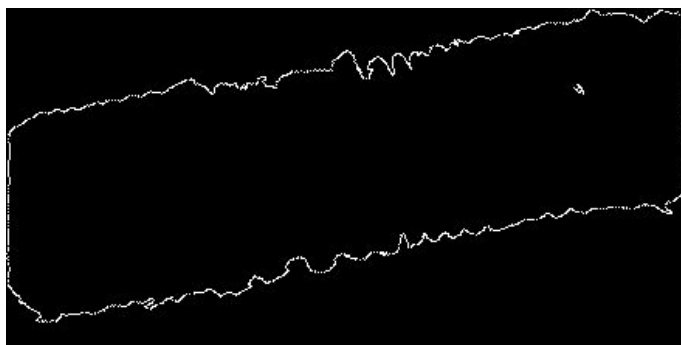


Abbildung 7.11: Kontur des STIFF-FLOP Armes aus der Abbildung 7.1 [29]

mit einer Kantenlänge von 5-10 Pixeln aufgeteilt. Für jeden Konturpixel wird der Winkel der Tangente bestimmt und, genauso wie bei der Erstellung der R-Tabelle, klassifiziert. Dieser Winkel wird mit der R-Tabelle verglichen, um den Eintrag mit dem passenden Index zu bestimmen. Unter Kenntnis der Koordinaten des Konturpunktes wird der zugehörige Schwerpunkt aus der R-Tabelle bestimmt. Anschließend wird das entsprechende Schwerpunktpixel in einem neuen Bild inkrementiert. Die Abbildung 7.12 illustriert diesen Vorgang. Dieses neue Bild wird als Akkumulator Feld bezeichnet. Die Zelle mit dem höchsten Inkrement entspricht dem Schwerpunkt des gesuchten Objektes in dem analysierten Bild.

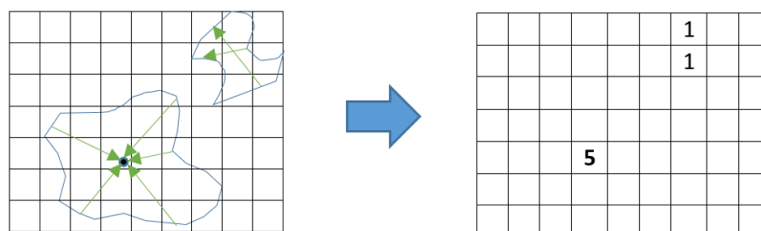


Abbildung 7.12: Beispielhafte Darstellung der Erstellung des Akkumulator Feldes (rechts) zur Bestimmung des Objektschwerpunktes in dem untersuchten Videobild (links) [29]

Um das erstellte Modell auch skaliert oder rotiert zu detektieren, kann die R-Tabelle transformiert werden. Hierzu wird das Modell der R-Tabelle mit $R(\Phi)$ und die Skalierung mit s bezeichnet. Die Transformation um die Skalierung s wird mit T_s ausgedrückt. Die Formel 7.2 beschreibt die Skalierung aller Vektoren der R-Tabelle.

$$T_S [R(\Phi)] = sR(\Phi) \quad (7.2)$$

Ist das gesuchte Objekt um den Winkel Θ rotiert, so wird die Transformation T_Θ durch die Formel 7.3 beschrieben. Hierdurch werden alle Indizes der R-Tabelle

um $-\Theta \text{Modulo } 2\pi$ inkrementiert und die zugehörigen Schwerpunkts Vektoren \vec{r}_i rotiert.

$$T_{\Theta} [R(\Phi)] = \text{Rot}\{R[(\Phi - \Theta) \text{mod } 2\pi], \Theta\} \quad (7.3)$$

Um auch rotierte und skalierte Objekte zu detektieren, wird das beschriebene Akkumulator Feld um die Parameter der Skalierung und der Rotation erweitert und stellt nun ein 4-Dimensionales Feld mit den Parametern (X, Y, Skalierung, Rotation) dar.

Auch wenn die hier vorgestellte Methode sich zur Detektion von beliebigen Strukturen eignet, zeigten sich doch erhebliche Hürden in der Erkennung eines nicht rigiden Objektes wie dem hier gesuchten flexiblen STIFF-FLOP Arm. Die vorgestellte Methode beschreibt die Detektion einer STIFF-FLOP Arm Konfiguration. Da der STIFF-FLOP Arm aber beliebig gebogen werden kann, müsste jede mögliche Biegung des Armes als eigenständige Form modelliert und anschließend das Bild auf alle möglichen Modelle untersucht werden. Ein weiteres Problem entsteht, wenn nur ein Teil des STIFF-FLOP Armes sichtbar ist und der Schwerpunkt des Modells ausserhalb des sichtbaren Bildbereiches liegt. Auch dann schlägt die Detektion des Armes fehl und der Algorithmus liefert kein Ergebnis.

7.3.2 Anwendung der merkmalsbasierten Detektion

Eine weitere Möglichkeit zur Detektion des STIFF-FLOP Armes bietet die Auswertung von charakteristischen Merkmalen. Hierfür wurde die von David Lowe vorgestellte Methode zur Merkmaldetektion von charakteristischen skalierungsinvarianten Punkten (SIFT - Scale Invariant Feature Detection) ausgewählt [59]. Die Evaluierung der Methode erfolgte im Rahmen einer betreuten studentischen Arbeit [29]. Daher wird die Methode hier nur zusammengefasst vorgestellt. Eine umfassende Beschreibung der Implementierung ist in der Masterarbeit [29] zu finden.

Im ersten Schritt werden hierfür zunächst die charakteristischen Merkmale des Bildes untersucht. Charakteristisch sind insbesondere Ecken und Kanten in Bildern. Zur Extraktion wird ein Laplacefilter zusammen mit einem Gaußfilter angewendet, der sogenannte Laplacian of Gaussian (LoG) [52]. Durch verschiedene Werte für die Standardabweichung σ wird der Filter gesteuert und kann dadurch unterschiedlich große Kanten detektieren. Durch die Subtraktion der Ergebnisbilder verschiedener σ

Werte ergibt sich der Filtertyp Difference of Gaussian (DoG) [52]. Hierdurch lassen sich potentielle Extremstellen des Bildes identifizieren. Aus diesen Werten werden die kontrastschwachen Punkte und Eckpunkte herausgefiltert, sodass charakterisierende Merkmalspunkte übrigbleiben. Zusätzlich wird für jeden Merkmalspunkt die Orientierung bestimmt und mit abgespeichert.

Anschließend wird für diese Merkmalspunkte eine Merkmalbeschreibung aus der 16×16 Pixelnachbarschaft erstellt. Dieser Nachbarschaftsbereich wird in 4×4 Blöcke unterteilt, für den jeweils ein Orientierungshistogramm mit 8 Einträgen erstellt wird. Hieraus ergeben sich für jeden Merkmalspunkt eine Merkmalbeschreibung mit 128 Einträgen.

Anhand dieser Merkmalbeschreibungen kann das Objekt in neuen Bilddaten gesucht werden. Hierzu wird das neue Bild auf charakteristische Merkmale untersucht, die dann mit den gespeicherten Merkmalen verglichen werden.

Bei der Anwendung des SIFT Algorithmus zur Detektion des STIFF-FLOP Armes wurden 3898 Merkmalspunkte in einem Bild gefunden. Diese sind als rote Kreise in der Abbildung 7.13 eingezeichnet. Die Ergebnisse wurden bereits mit der optimierten SIFT Implementierung aus openCV erzeugt [29].

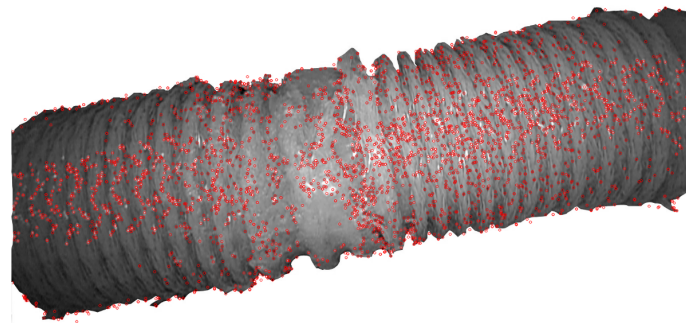


Abbildung 7.13: Mit dem SIFT Algorithmus detektierte Merkmalspunkte des STIFF-FLOP Armes aus der Abbildung 7.1, hervorgehoben durch rote Kreise [29]

Zur Erzielung einer robusten und zuverlässigen Detektion des Armes sollten Merkmalspunkte aus verschiedenen Bilddaten erzeugt und zusammengefasst werden. Bei der Evaluierung des Algorithmus war die Berechnung der Merkmalspunkte, trotz der optimierten Algorithmen, sehr zeitintensiv. Für die Bestimmung der dargestellten Merkmalspunkte wurden 620 Millisekunden benötigt. Dies berücksichtigt noch nicht den Vergleich der detektierten Merkmalspunkte mit den Daten einer Datenbank mit

vielen STIFF-FLOP Arm SIFT-Merkmalpunkten. Damit ist diese Methode zu langsam um eine echtzeitnahe Detektion des STIFF-FLOP Armes in dem Videobild zu ermöglichen [29].

7.3.3 Detektion des flexiblen Roboterarmes unter Verwendung von maschinellem Lernen

Mit dem Ziel auch weiterentwickelte Versionen des STIFF-FLOP Armes detektieren zu können, wurde ein Verfahren des maschinellen Lernens ausgewählt und evaluiert. Ausgewählt wurde die Verwendung von Support Vector Machines (SVM) [10]. Die Methode wird verbreitet zur Klassifikation von Daten eingesetzt. Ein erster Ansatz hierfür ist im Rahmen der betreuten Masterarbeit [29] beschrieben. Die hier beschriebene Methode wurde auch im STIFF-FLOP Buch veröffentlicht [44].

Die grundlegende Idee dahinter ist, das Videobild in kleine Teilstücke zu unterteilen und dann zu untersuchen, ob diese Teile zu dem STIFF-FLOP Arm oder zu dem Hintergrund gehören. Dieser Ansatz zur Klassifikation wird auf die Bilddaten angewendet. Durch eine Analyse werden die Bilddaten in zwei Kategorien klassifiziert. Hierfür werden Trainingsdaten benötigt, in denen der STIFF-FLOP Arm und der Hintergrund entsprechend bekannt sind. Die SVM Methode stellt diese Daten als Punkte in einem Raum dar, sodass diese durch eine Ebene separiert sind. Bei einer anschließenden Untersuchung von neuen Bilddaten werden diese, wie die Trainingsdaten, in Teilstücke aufgeteilt und überprüft, auf welcher Seite der Ebene sich das Teilstück befindet. Hierdurch werden neue Bilddaten klassifiziert und bestimmt, welche Teilstücke zum gesuchten STIFF-FLOP Arm gehören. Entscheidend hierfür ist die Bestimmung der Ebene zur Separation der Bilddaten. Für die Ebene, die den größten Abstand zwischen den Daten liefert, werden bei dem Einsatz der Methode auf unbekannte Bilddaten, die zuverlässigsten Ergebnisse erzielt.

Zur Detektion des STIFF-FLOP Armes wird das Videobild in kleine Quadrate mit einer Kantenlänge von 25×25 Pixel aufgeteilt. Diese werden analysiert und mit der Variable $y_i \in \{-1, 1\}$ als Hintergrund oder Objekt klassifiziert.

$$D_i = \{(\vec{x}_i, y_i) | \vec{x}_i \in \mathbb{R}^P, y_i \in \{-1, 1\}\}_{i=1}^n \quad (7.4)$$

Hierbei beschreibt D_i die Anzahl n an quadratischen Teilstücken \vec{x}_i des Videobildes mit der zugehörigen Klassifikation y_i . Die Abbildung 7.14 visualisiert eine verein-

fachte Darstellung der Formel 7.4. Die roten Quadrate repräsentieren dabei die erste Klasse und die blauen Kreise die zweite.

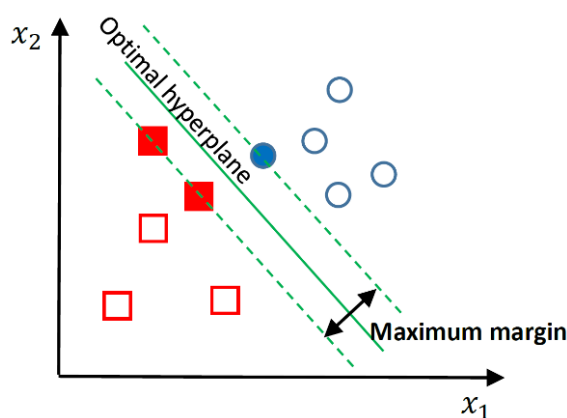


Abbildung 7.14: Visualisierung verschiedener Musterpunkte entsprechend ihrer Charakterisierung, wobei rote Quadrate die erste Klasse und blaue Kreise die zweite Klasse repräsentieren [44]

Ausgehend von einem Datensatz aus der Abbildung 7.14 kann eine Ebene gefunden werden, die beide Klassen voneinander trennt. Die grüne Linie in der Abbildung 7.14 zeigt eine optimale Ebene, die beide Klassen mit einem maximalen Abstand trennt. Diese Ebene dient als Klassifikator für die spätere Detektion des STIFF-FLOP Armes im Videobild. Basierend auf den beschriebenen Methoden aus den Quellen [10] und [24] kann eine Ebene aus den Punkten \vec{x}_i bestimmt werden. Für die 2-Dimensionale Darstellung der Punkte aus der Abbildung 7.14 genügt eine Gerade zur Separation. Für höher dimensionale Darstellungen spricht man von einer Hyperebene, die die Punkte trennt.

$$\vec{w}' \cdot \vec{x}_i + b = 0 \quad (7.5)$$

Hierbei beschreibt \vec{w} den Normalenvektor der Hyperebene und b eine Konstante der Gleichung. Die Punkte der ersten Klasse werden unterhalb der Hyperebene und die Punkte der zweiten Klasse oberhalb der Hyperebene gefunden.

$$\begin{aligned} \vec{w}' \cdot \vec{x}_i + b &\geq +1 && \text{für } y_i = +1 \\ \vec{w}' \cdot \vec{x}_i + b &\leq -1 && \text{für } y_i = -1 \end{aligned} \quad (7.6)$$

Die Formel 7.6 kann mit der Klassifikation y_i zusammengefasst werden in der Formel 7.7.

$$y_i(\vec{w}' \cdot \vec{x}_i + b) - 1 \geq 0 \quad (7.7)$$

Die Punkte, die auf der oberen und unteren Grenze liegen, was als gestrichelte Linie in der Abbildung 7.14 dargestellt ist, werden als Support Vektoren bezeichnet. Für genau diese Support Vektoren ist die Formel 7.7 gleich 0.

Um den maximalen Abstand zwischen den Support Vektoren der ersten und der zweiten Klasse zu finden, wird der Abstand zwischen den Support Vektoren auf der oberen und unteren Grenze berechnet. Die Abbildung 7.15 visualisiert die Abstandsbestimmung.

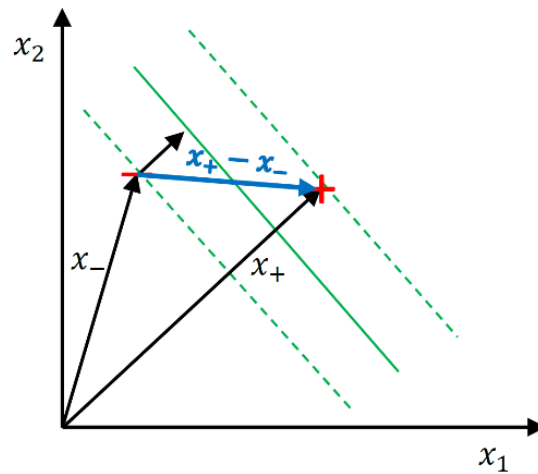


Abbildung 7.15: Visualisierung der Abstandsberechnung zwischen den Supportvektoren auf der unteren Grenze \vec{x}_- und der oberen Grenze \vec{x}_+ [44]

Die Punkte der Supportvektoren sind durch die Gleichung 7.7 definiert.

$$\begin{aligned} \vec{w}' \cdot \vec{x}_+ &= 1 - b & \text{für } y_i &= +1 \\ \vec{w}' \cdot \vec{x}_- &= -1 - b & \text{für } y_i &= -1 \end{aligned} \quad (7.8)$$

Wie in der Abbildung 7.15 dargestellt, kann aus der Geometrie der Abstand zwischen den Support Vektoren aus zwei Klassen bestimmt werden.

$$width = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|} \quad (7.9)$$

Für eine möglichst zuverlässige Klassifikation soll die Hyperebene so festgelegt werden, dass der Abstand maximal wird. Um den maximalen Abstand zu erhalten, muss

das Minimum von $\|\vec{w}\|$ bestimmt werden. Hierfür kann $\|\vec{w}\|$ gemäß der Formel 7.10 substituiert werden.

$$\text{substituiere: } \min \|\vec{w}\| \quad \text{durch} \quad \min \frac{1}{2} \|\vec{w}\|^2 \quad (7.10)$$

Damit ergibt sich ein Optimierungsproblem mit der Nebenbedingung aus der Formel 7.7. Dieses kann durch Einführen eines Lagrange-Multiplikators α gemäß der Formel 7.11 formuliert werden.

$$L = \arg \min_{w,b} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\vec{w}' \cdot \vec{x}_i + b) - 1] \right\} \quad (7.11)$$

Das Minimum des Optimierungsproblems kann mit Hilfe der ersten Ableitung, die gleich null gesetzt wird, bestimmt werden. Damit ergeben sich die Terme aus der Formel 7.12.

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 & \Rightarrow \vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \\ \frac{\partial L}{\partial b} = 0 & \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned} \quad (7.12)$$

Aus der Formel 7.12 wird deutlich, dass \vec{w} sich als Linearkombination von Vektoren deren $\alpha > 0$ ist, zusammensetzt. Dies sind die bereits vorgestellten Supportvektoren. Mit dieser Funktion kann die optimale Hyperebene bestimmt werden. Durch Einsetzen der Ableitung in das Optimierungsproblem aus der Formel 7.11 kann das Optimierungsproblem vereinfacht werden. Durch das $\|\vec{w}\|^2$ wird eine zweite Summationsvariable j eingeführt, wodurch die Terme zusammengefasst werden können. Das Ergebnis zeigt die Formel 7.13. Die vollständige Herleitung kann dem Anhang der Publikation [25] entnommen werden.

$$L = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \vec{x}_i' \cdot \vec{x}_j \quad (7.13)$$

Hierbei liegt der Lagrangemultiplikator α_i und α_j im Intervall $0 \leq \alpha \leq C$. Durch eine Begrenzung von α durch den Parameter C wird zugelassen, dass sich auch einzelne

Supportvektoren in dem Grenzbereich, also zwischen den gestrichelten Linien der Abbildung 7.14, befinden dürfen. Dies wurde von Cortes und Vapnik [10] als weicher Grenzbereich (soft-margin) eingeführt. Das ist besonders bei großen Datenmengen hilfreich. So kann der Einfluss von einzelnen Ausreisser Punkten, auf die Lage der Hyperebene, begrenzt werden.

Damit hängt das Optimierungsproblem aus der Formel 7.13 nur von dem Skalarprodukt von $\vec{x}_i \cdot \vec{x}_j$ ab. Dieses bleibt auch gültig wenn die Vektoren, durch eine Transformation $\phi(\vec{x}_i)$, transformiert werden. Hierdurch können die Vektoren \vec{x}_i und \vec{x}_j in einen Raum überführt werden, in dem sich diese passender voneinander separieren lassen. Dies übernehmen sogenannte Kernelfunktionen $K(\vec{x}'_i \cdot \vec{x}_j)$.

$$K(\vec{x}_i \cdot \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \quad (7.14)$$

In der Literatur wurden bereits einige Kerntypen vorgestellt, die sich für unterschiedlich verteilte Daten eignen [18] [25]:

- Lineare Basis Funktion: $K(\vec{x}_i \cdot \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)$
- Polynomiale Basis Funktion: $K(\vec{x}_i \cdot \vec{x}_j) = (\gamma \vec{x}_i \cdot \vec{x}_j + 1)^d, \gamma > 0$
- Radiale Basis Funktion: $K(\vec{x}_i \cdot \vec{x}_j) = e^{-\gamma \|\vec{x}_i - \vec{x}_j\|}, \gamma > 0$
(in einigen Quellen auch als $\gamma = 1/2\sigma^2$ bezeichnet)
- Sigmoidale Basis Funktion: $K(\vec{x}_i \cdot \vec{x}_j) = \tanh(\gamma \vec{x}_i \cdot \vec{x}_j + r)$

Diese, zur Justierung der Kernelfunktionen verwendeten Parameter, werden als Hyperparameter bezeichnet. Der lineare Kernel eignet sich für Daten, die durch eine einfache Ebene separiert werden können. Mit dem Hyperparameter d wird der Grad des polynomialen Kerns festgelegt. Wobei $d = 1$ einem linearen Kernel entspricht und höhere Grade eine dem Polynom entsprechende Separation der Daten erlauben. Der Hyperparameter γ der radialen Basisfunktion, die auch als Gauss'sche Kernelfunktion bezeichnet wird, erlaubt eine kreisförmige Separierung der Daten. Ein höherer Wert $\gamma > 1$ erhöht die Flexibilität der Hyperebene, sodass diese sich gebogen um die Datenpunkte legt, kann aber bei zu hohen Werten zu einer Überanpassung führen. Weitere Erläuterungen und Beispielbilder für die Funktionsweise der Kernelfunktionen sind in den Quellen [25] und [10] zu finden. Im Rahmen dieser Arbeit wurden mit der Polynomiale und der Radialen Basis Funktion die besten Ergebnisse erzielt, die im Abschnitt 7.3.5 vorgestellt sind.

7.3.4 Training des Support Vector Machine Algorithmus

Der Einsatz des Support Vector Machine Algorithmus zur Detektion des STIFF-FLOP Armes gliedert sich in eine Trainingsprozedur und eine anschließende Online Erkennung. In der Trainingsprozedur wird der Algorithmus parametrisiert und mit Trainingsdaten, in welchen der Bildausschnitt mit dem STIFF-FLOP Arm bekannt ist, trainiert. Hierbei wird die Hyperebene zur Trennung von Vordergrunddaten, die den STIFF-FLOP Arm enthalten (positive Teilstücke) und Hintergrunddaten (negative Teilstücke) bestimmt. Diese Hyperebene wird verwendet um den STIFF-FLOP Arm in dem Online Videobild zu erkennen.

In der Implementierung wird auf die vorhandene LIBSVM Bibliothek [24], die bereits in openCV enthalten ist [19], zurückgegriffen. Diese beinhaltet Klassen zur Lösung des Optimierungsproblems mit den vorgestellten Kerneltypen. Eine Erklärung zum Einsatz der SVM Klassen kann in der openCV Dokumentation des SVM Algorithmus gefunden werden [79]. Die Abbildung 7.16 zeigt die Schritte zum Training des Algorithmus.

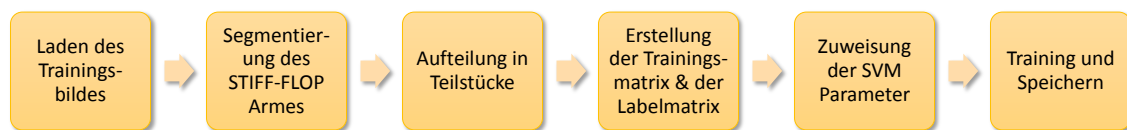


Abbildung 7.16: Programmablauf zum Training der SVM-Methode

In den ersten Ansätzen zur Implementierung des SVM Algorithmus im Rahmen der betreuten studentischen Arbeit [29] zeigte sich, dass das Training entscheidend für die erzielten Ergebnisse ist. Um das Training des SVM Algorithmus zu verfeinern, wurde dieses in einem eigenständigen Programm umgesetzt. Die Online Erkennung des STIFF-FLOP Arms wurde in das vorhandene ROS Framework integriert, um eine Anbindung zu den anderen Komponenten des STIFF-FLOP Projektes zu ermöglichen.

Für die Trainingsprozedur werden Videobilddaten mit dem STIFF-FLOP Arm in unterschiedlichen Biegungen verwendet, um möglichst viele Positionsmöglichkeiten des Armes zu trainieren. Hierbei muss bekannt sein, welche Bildausschnitte den STIFF-FLOP Arm enthalten und welche zum Hintergrund gehören.

Um die Teilstücke klassifizieren zu können, wird der Bildausschnitt mit dem STIFF-FLOP Arm von dem Hintergrund separiert. Da dieser Schritt für verschiedene Szenarien und Bilddaten wiederholt werden muss, wurde hierfür im Rahmen einer weiteren

studentischen Arbeit, ein Verfahren zur manuellen Segmentierung des STIFF-FLOP Armes implementiert [61]. Um die Segmentierung möglichst einfach und effizient zu gestalten wurde die „Grabcut“ Methode [86], für die es bereits eine Implementierung in der openCV Bibliothek gibt [78], verwendet. Diese verbindet verschiedene Methoden, die die Texturinformation und die Kontrastinformation des Bildes auswerten.

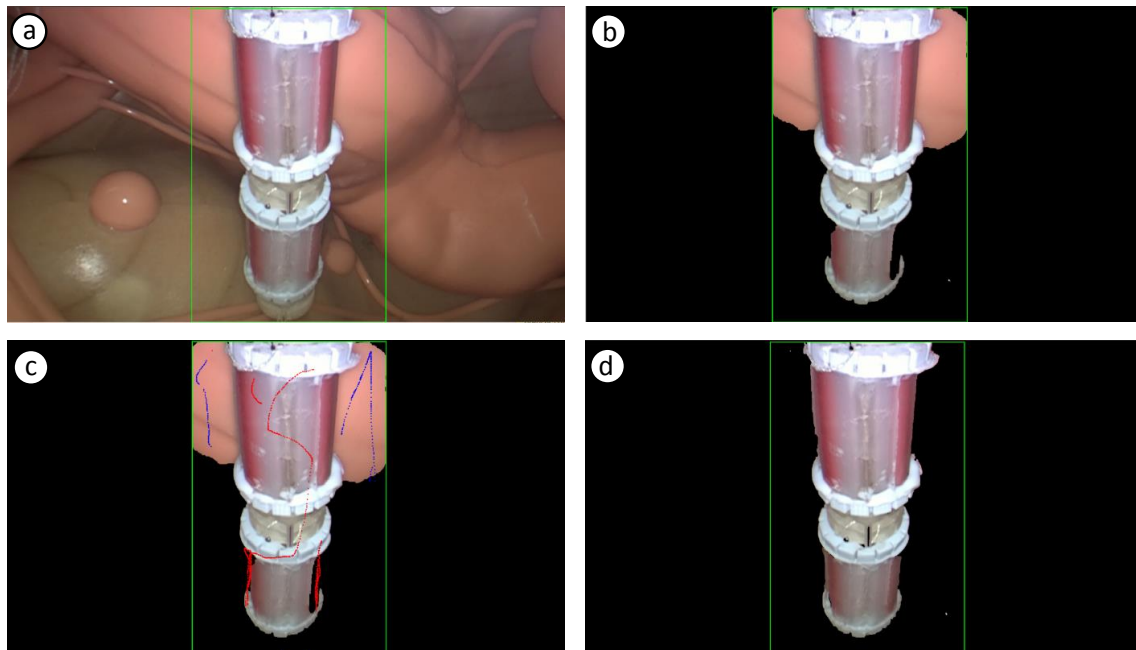


Abbildung 7.17: Segmentierung des STIFF-FLOP Armes mit der Grabcut Methode: (a) originales Videobild, in dem der Bereich mit dem STIFF-FLOP Arm ausgewählt wird (grünes Quadrat), (b) automatisch segmentierte Teile des Originalbildes, (c) Benutzereingaben zur Markierung von Bereichen die zum STIFF-FLOP Arm gehören (rot), sowie Bereiche die zum Hintergrund gehören (blau), (d) Ergebnis der Segmentierung [61]

Für die Segmentierung mit der „Grabcut“ Methode wird zunächst ein Rechteck ausgewählt, das den STIFF-FLOP Arm enthält. Die Bereiche außerhalb des Rechtecks werden automatisch als Hintergrund klassifiziert. Innerhalb des Rechtecks wird bereits eine erste Analyse der Pixel durchgeführt, wobei abgeschätzt wird, welche Bereiche zum Vordergrund gehören. Anschließend können die Bereiche, die noch falsch klassifiziert sind, durch einfache Mauseingaben als Vordergrund- und Hintergrundbereiche markiert werden. In der darauffolgenden Iteration des Grabcut Algorithmus wird eine verbesserte Segmentierung unter Berücksichtigung der Eingaben durchgeführt. Die Abbildung 7.17 zeigt die implementierte Segmentierung des STIFF-FLOP Armes mit der verwendeten Grabcut Methode. Als Abschluss der

Segmentierung wird ein Vordergrundbild mit dem segmentierten STIFF-FLOP Arm, wie in der Abbildung 7.17 dargestellt und ein Bild mit dem Hintergrund, ohne den STIFF-FLOP Arm, erzeugt.

Zum Training der SVM-Methode wird ein Trainingsbild geladen und in Teilstücke mit einer Größe von 25×25 Pixel aufgeteilt. Zur Verwendung der SVM-Klassen muss aus diesen Teilstücke eine Trainingsmatrix erstellt werden, die die Merkmale der Teilstücke enthält. Die Merkmale setzen sich aus den Farbinformationen aller Pixel des Teilstückes zusammen. Diese werden als Zeilenvektor in die Trainingsmatrix eingetragen. Jede Zeile der Trainingsmatrix repräsentiert damit ein Teilstück des Trainingsdatensatzes.

Anschließend wird eine Labelmatrix angelegt, die für jedes Teilstück das entsprechende Label enthält. Dieses ist Positiv (+1) für alle Teilstücke die den STIFF-FLOP Arm enthalten und negativ (-1) für alle Teilstücke aus dem Hintergrund. Die Abbildung 7.18 illustriert die Erstellung der Trainingsmatrix und der Labelmatrix zum Training der SVM-Methode.

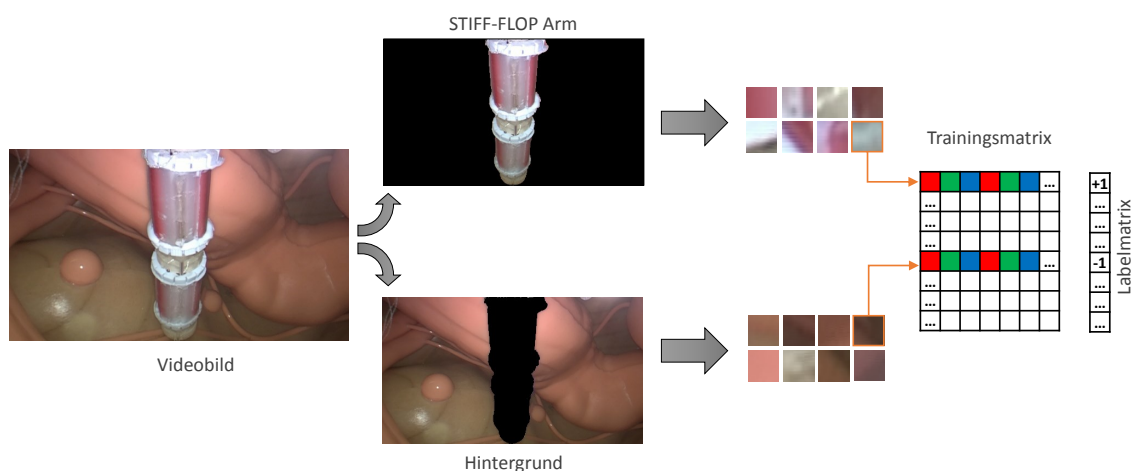


Abbildung 7.18: Aufteilung eines Videobildes in Teilstücke zur Erstellung der Trainingsmatrix und der Labelmatrix

Insbesondere für die Teilstücke, die nur teilweise den STIFF-FLOP Arm enthalten, muss entschieden werden, ob diese dem STIFF-FLOP Arm oder dem Hintergrund zugeordnet werden. In den Teilstücken des segmentierten STIFF-FLOP Armes aus der Abbildung 7.17 wird die Summe von nicht schwarzen Pixeln bestimmt. Dies entspricht der Summe von Vordergrundpixeln die zum STIFF-FLOP Arm gehören. Dieser Anteil wird in der Formel 7.15 als Vordergrundanteil bezeichnet.

$$\text{Vordergrundanteil} = \frac{\sum \text{Vordergrundpixel}}{25 \times 25} \quad (7.15)$$

Die Implementierung der Trainingsprozedur erfolgte in Zusammenarbeit mit den betreuten studentischen Arbeiten [29] [61]. Hierfür wurde ein eigenständiges Programm erstellt und openCV in der Version 2.4.9 verwendet. Zur Umsetzung der SVM Methode wurde die Klasse „SupVecMac“ erstellt, in der die benötigten Funktionen implementiert sind. Der Quellcode 7.2 zeigt die „process_image“ Funktion, wo die Funktionen für das Training der SVM Methode aufgerufen werden.

```

1 void SupVecMac::process_image(std::string training_image, std::string bg_image, std::string fg_image
  ,std::string result_masked_unfiltered,std::string result_masked_filtered, int offset, int
  initial_no_fg, int initial_no_bg, int bg_percentage, int fg_percentage)
2 {
3 int bg_files=0; //file counter for foreground pieces
4 int fg_files=0; //file counter for background pieces
5 int bg=1; //this is to extract the background images
6 int fg=2; //this is to extract the foreground images
7
8 //extract images
9 clear_all_folders(offset);
10 breakdown_img(training_image,"Small_Images",offset,bg_percentage, bg, &bg_files);
11 breakdown_img(bg_image,"Train_samples_Background",offset,bg_percentage, bg, &bg_files);
12 breakdown_img(fg_image,"Train_samples_Tube",offset,fg_percentage, fg, &fg_files);
13
14 //training
15 bg_files=bg_files+initial_no_bg;
16 fg_files=fg_files+initial_no_fg;
17 train_and_save_svm(&fg_files,&bg_files,"Train_samples_Tube","Train_samples_Background");
18
19 //prediction
20 predict_svm(training_image);
21 predict_svm_masked(training_image);
22 predict_svm_masked_filtered(training_image);
23
24 //result comparison
25 result_comparison(fg_image,result_masked_unfiltered,result_masked_filtered);
26 }

```

Quellcode 7.2: Quellcode zum Training des SVM Algorithmus

An die „process_image“ Funktion werden die Pfade zu dem originalen Trainingsbild, dem unter Verwendung des Grabcut Algorithmus erzeugten Vordergrundbild und der Pfad zum Hintergrundbild, übergeben. Ausserdem werden noch Pfade für den Export eines gefilterten und ungefilterten Ergebnisses übermittelt. Der „offset“, sowie die Variablen mit der initialen Anzahl an Vorder- und Hintergrundbildnummern, ermöglichen die Verwendung von mehreren Trainingsbildern zum Training des Algorithmus. Über den Vordergrundanteil wird, gemäß der Formel 7.15, der Vorder-

grundanteil der segmentierten Teilstücke festgelegt, die für das Training verwendet werden. Analog dazu wird ein Anteil für die Hintergrundteilstücke festgelegt.

Im Quellcode 7.2 (Zeile 10 - 12) erfolgt zunächst die Aufteilung des Trainingsbildes in Teilstücke mit der „breakdown_Img“ Funktion. Hierfür wird zunächst das originale Trainingsbild in Teilstücke geteilt, die in einem temporären Ordner abgelegt werden. Anschließend erfolgt der Schritt für das Hintergrundbild ohne den segmentierten STIFF-FLOP Arm. Diese Teilstücke werden in einen Ordner für die Hintergrundteilstücke extrahiert. Abschließend wird das Vordergrundbild mit dem segmentierten STIFF-FLOP Arm in Teilstücke geteilt und abgespeichert. Durch die Segmentierung sind alle Hintergrundkomponenten aus dem Bild entfernt. Die Teilstücke, die den Rand des STIFF-FLOP Armes abbilden, enthalten auch einen kleinen Hintergrundanteil, der durch die Segmentierung herausgeschnitten wurde. Diese Teilstücke werden von dem originalen Trainingsbild übernommen und ebenfalls in dem Ordner für die Vordergrund Teilstücke abgelegt. Der Quellcode der „breakdown_Img“ Funktion ist im Quellcode D.1 im Anhang D angefügt.

Im zweiten Schritt (Zeile 17) werden die Trainingsbilder geladen und wie in der Abbildung 7.18 vorgestellt, zeilenweise in einer Trainingsmatrix und einer Labelmatrix abgebildet. Nach der Auswahl des Kerneltypen und der Parametrierung der SVM Methode, die im Quellcode D.2 dargestellt sind, erfolgt das Training der SVM Methode. Als Ergebnis werden die ermittelten Stützvektoren, die die Hyperebene festlegen, in eine Datei exportiert. Diese wird für die Online Erkennung eingesetzt.

Zur Überprüfung des Trainingsergebnisses erfolgt eine Erkennung des STIFF-FLOP Armes mit der SVM-Methode. Dies berechnet die Funktion „predictsvm“. Hierin werden alle detektierten Teilstücke mit einem Rahmen in dem untersuchten Bild gekennzeichnet. Die Abbildung 7.19 zeigt die detektierten Teilstücke.

Um anschließend die Ergebnisse zu überprüfen, wird in der Funktion „predict_svm_masked“ ein neues binäres Ergebnisbild erzeugt, das alle erkannten Teilstücke enthält. Bei dem Ergebnisbild in der Abbildung 7.19 zeigt sich, dass einzelne Hintergrundteile als STIFF-FLOP Arm erkannt wurden und einzelne Teilstücke innerhalb des STIFF-FLOP Arms nicht erkannt wurden. Um diese einzelnen Fehldetektionen zu unterdrücken wurde ein Konturfilter implementiert. Den Ausgangspunkt hierfür bildet die Bildmaske des binären Ergebnisbildes. In dieser Bildmaske werden alle Konturen gesucht. Die Kontur des detektierten STIFF-FLOP Armes entspricht der größten zusammenhängenden Kontur. Alle anderen Konturen werden verworfen,

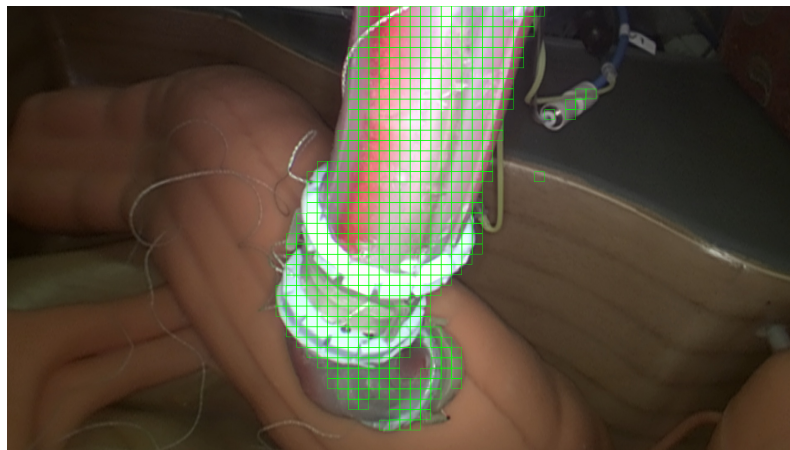


Abbildung 7.19: Erkante Teilstücke des STIFF-FLOP Armes, gekennzeichnet mit grünem Rahmen, nach Ausführung der Detektionsfunktion

wodurch lediglich der detektierte STIFF-FLOP Arm erhalten bleibt. Die Implementierung erfolgt in der „predict_svm_masked_filtered“ Funktion, die im Quellcode D.3 dargestellt ist.

Abschließend wird das Ergebnis der Detektion mit dem segmentierten Trainingsbild verglichen um festzustellen, wie viele Anteile des STIFF-FLOP Armes erkannt wurden. Die Implementierung der „result_comparison“ Funktionen ist im Quellcode D.4 dargestellt. Die Ergebnisse werden im Abschnitt 7.3.5 erläutert.

7.3.5 Ergebnisse der Detektion des flexiblen Roboterarmes

Zur Überprüfung der Ergebnisse der Detektion wurden verschiedene Bilder im Anschluss an das Training aus dem Abschnitt 7.3.4 analysiert. Im ersten Schritt wird untersucht, welcher Vordergrundanteil, gemäß der Formel 7.15 zu der höchsten Detektionsrate führt. Durch eine Überlagerung des manuell segmentierten STIFF-FLOP Armes, der in der Abbildung 7.17 dargestellt ist, mit der gefilterten Bildmaske des Detektionsergebnisses, kann die Anzahl der detektierten Bildpunkte und die Anzahl der nicht detektierten Bildpunkte, die zum STIFF-FLOP Arm gehören, bestimmt werden. Zusätzlich wird überprüft, wie viele Pixel, die den Hintergrund abbilden, als STIFF-FLOP Arm detektiert wurden (false positive).

Die Tabelle 7.2 zeigt eine Versuchsreihe, in der die falsch detektierten Bildpunkte (false positive), die nicht detektierten Bildpunkte, sowie die korrekt detektierten Bildpunkte bei variiertem Vordergrundanteil dargestellt sind. Als Bilddatensatz

für die Erkennung des STIFF-FLOP Armes wurde das Videobild aus der Abbildung 7.23(b) verwendet. Dieses wurde unter Verwendung der radialen Basisfunktion detektiert und anschließend gefiltert.

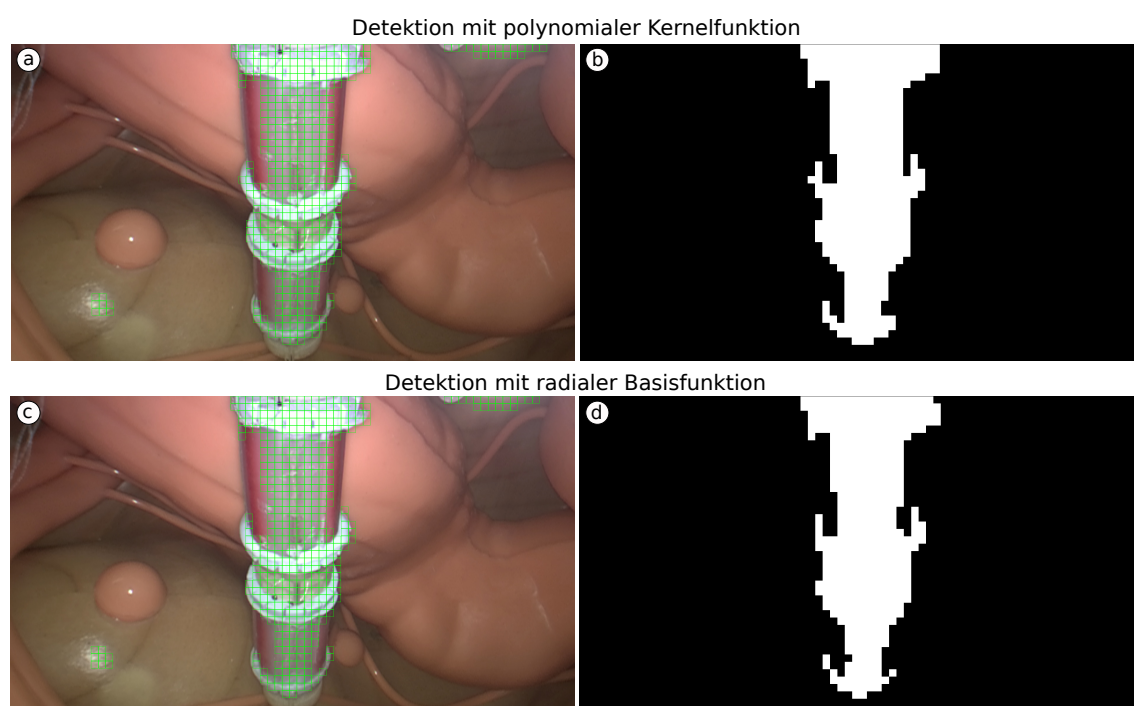
Vordergrundanteil [%]	Falsch detektierte Bildpunkte	Nicht detektierte Bildpunkte	Korrekt detektierte Bildpunkte	Prozentsatz der erkannten Bildpunkte	Prozentsatz falsch detektierter Bildpunkte	Anzahl Bildpunkte die den STIFF-FLOP Arm enthalten
100	8924	22545	349376	93,94%	2,40%	371921
95	8924	21920	350001	94,11%	2,40%	371921
90	8956	20702	351219	94,43%	2,41%	371921
85	8956	20702	351219	94,43%	2,41%	371921
80	8956	20077	351844	94,60%	2,41%	371921
75	8956	19452	352469	94,77%	2,41%	371921
70	9234	19105	352816	94,86%	2,48%	371921
65	9234	19105	352816	94,86%	2,48%	371921
60	9427	18673	353248	94,98%	2,53%	371921
55	9427	18673	353248	94,98%	2,53%	371921
50	9427	18673	353248	94,98%	2,53%	371921
45	9427	18673	353248	94,98%	2,53%	371921
40	9427	18673	353248	94,98%	2,53%	371921
35	9427	18673	353248	94,98%	2,53%	371921
30	9427	18673	353248	94,98%	2,53%	371921
25	9427	18673	353248	94,98%	2,53%	371921
20	9427	18673	353248	94,98%	2,53%	371921
15	9868	18489	353432	95,03%	2,65%	371921
10	10358	17754	354167	95,23%	2,79%	371921
5	10358	17754	354167	95,23%	2,79%	371921

Tabelle 7.2: Versuchsreihe zur Analyse des berücksichtigten Vordergrundanteils für das Training der SVM-Methode am Beispiel der Abbildung 7.23(b)

In der Tabelle 7.2 gibt der Prozentsatz der erkannten Bildpunkte den prozentualen Anteil der korrekt detektierten Bildpunkte (Spalte 4), von der Gesamtanzahl der Bildpunkte die den STIFF-FLOP Arm abbilden (Spalte 7), an. Der Prozentsatz der falsch detektierten Bildpunkte bezeichnet den Anteil der falsch detektierten Bildpunkte (false positive) (Spalte 2) von der Gesamtanzahl der Bildpunkte, die den STIFF-FLOP Arm abbilden.

Bei geringerem Vordergrundanteil werden auch Teilstücke für das Training der SVM-Methode verwendet, die nur einen geringeren Anteil des STIFF-FLOP Armes enthalten. Erwartungsgemäß zeigt die Tabelle, dass bei sinkendem Vordergrundanteil der Prozentsatz der korrekt detektierten Bildpunkte steigt. Da nun auch mehr Teilstücke erkannt werden, die einen höheren Hintergrundanteil aufweisen, steigt hierdurch gleichzeitig die Anzahl der Bildpunkte die den Hintergrund darstellen, aber als STIFF-FLOP Arm erkannt wurden. Mit dem Ziel, die Anzahl der korrekt detektierten Bildpunkte zu maximieren, wurde für das verwendete Trainingsbild ein Vordergrundanteil von 10% angesetzt. Auch die Überprüfung weiterer Trainingsbilder zeigte ein vergleichbares Verhalten.

Nachfolgend werden die Ergebnisse verschiedener Videobilder aufgezeigt, die den STIFF-FLOP Arm in unterschiedlichen Orientierungen zeigen. Die Abbildungen 7.20 - 7.23 zeigen die Ergebnisse der Detektion bei der Verwendung der polynomialen Kernelfunktion und der radialen Basisfunktion. Die Abbildung (a) zeigt dabei jeweils das Ergebnis der Detektion unter Verwendung der polynomialen Kernelfunktion, wobei die detektierten Teilstücke hervorgehoben sind. In der Abbildung (b) ist die gefilterte Bildmaske der Detektion aus der Abbildung (a) dargestellt. Das Ergebnis der Detektion unter dem Einsatz der radialen Basisfunktion ist in der Abbildung (c) dargestellt. Die Abbildung (d) zeigt die gefilterte Bildmaske der Abbildung (c).

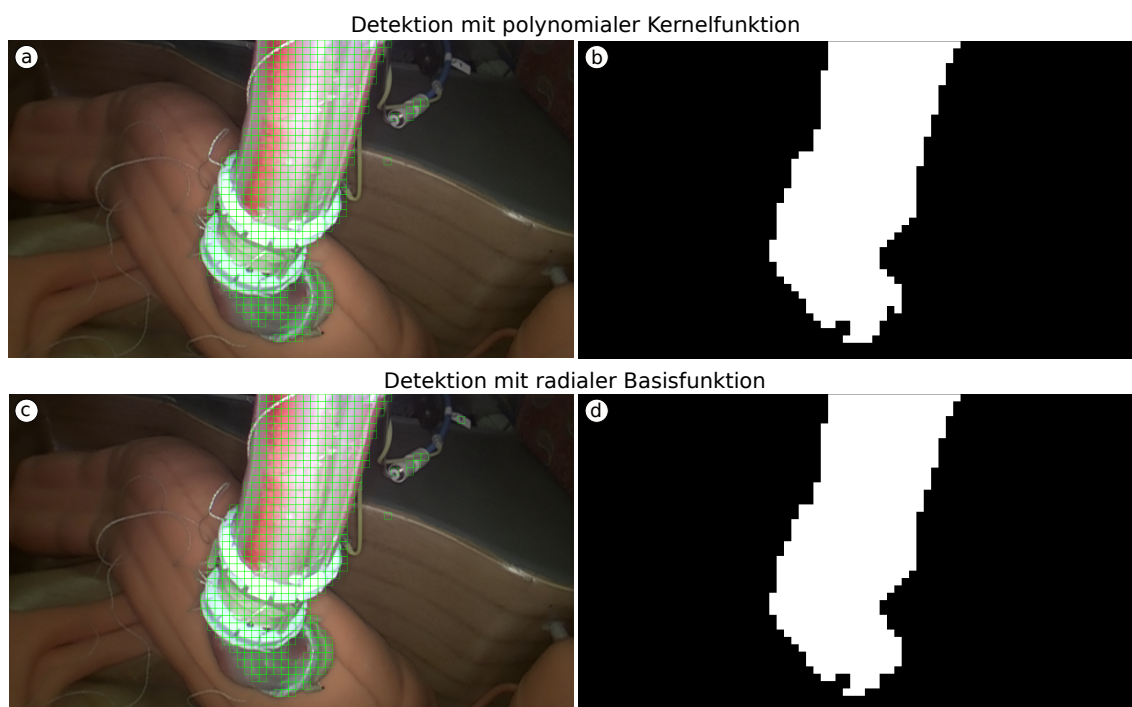


Parameter	Polynomialer Kernel		Radiale Basisfunktion	
	Ungefilterte Detektion	Gefilterte Bildmaske	Ungefilterte Detektion	Gefilterte Bildmaske
Korrekt detektierte Pixel [%]	81.03	81.07	78.68	78.44
Falsch detektierte Pixel [%]	17.74	2.73	16.81	2.19
Nicht detektiert [%]	18.69	18.93	21.32	21.55

Abbildung 7.20: Detektionsszenario 1 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]

In den Tabellen unter den Bildern 7.20 - 7.23 ist der prozentuale Anteil der detektierten Pixel, der falsch detektierten Pixel sowie der nicht detektierten Pixel dargestellt.

Hierbei wurde das ungefilterte Detektionsergebnis und die gefilterte Bildmaske für beide Kerneltypen untersucht.



Parameter	Polynomialer Kernel		Radiale Basisfunktion	
	Ungefilterte Detektion	Gefilterte Bildmaske	Ungefilterte Detektion	Gefilterte Bildmaske
Korrekt detektierte Pixel [%]	91.99	93.07	92.14	93.38
Falsch detektierte Pixel [%]	12.02	3.11	11.86	2.79
Nicht detektiert [%]	8.01	6.93	7.8	6.6

Abbildung 7.21: Detektionsszenario 2 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]

Aus den Ergebnissen der Tabellen unter den Bildern 7.20 - 7.23 wird deutlich, dass die Anwendung des Konturenfilters insbesondere die Anzahl der falsch detektierten Pixel deutlich reduziert, da hierdurch die falsch detektierten Teilstücke ausserhalb des STIFF-FLOP Armes erfolgreich herausgefiltert werden. Auch einzelne Löcher durch nicht detektierte Teilstücke innerhalb des STIFF-FLOP Armes können geschlossen werden, was in der Abbildung 7.21 deutlich wird. Hierdurch erhöht sich die Anzahl der korrekt detektierten Teilstücke.

Der korrekt detektierte Anteil variiert in den Beispielbildern zwischen 78,44% und 96,15%. Das Ergebnis für das erste Testbild aus der Abbildung 7.20 zeigt dabei

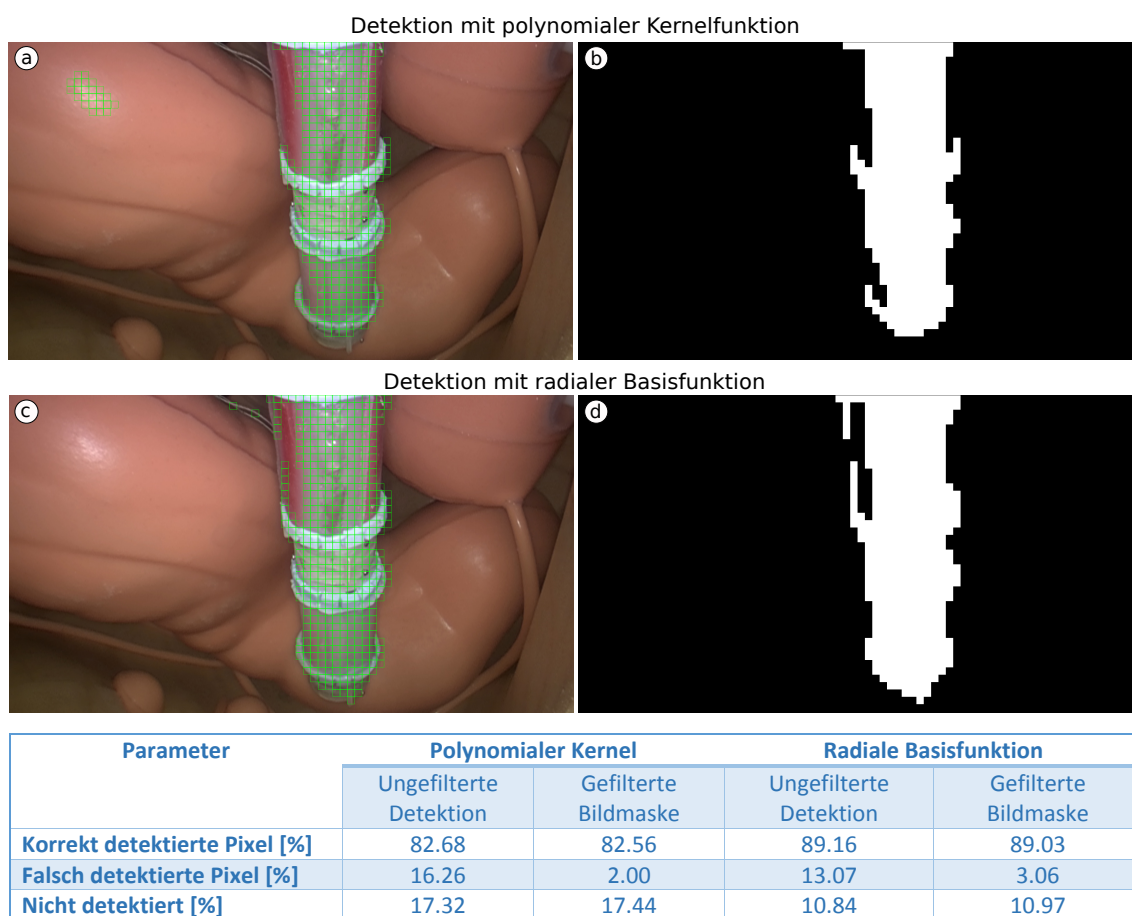
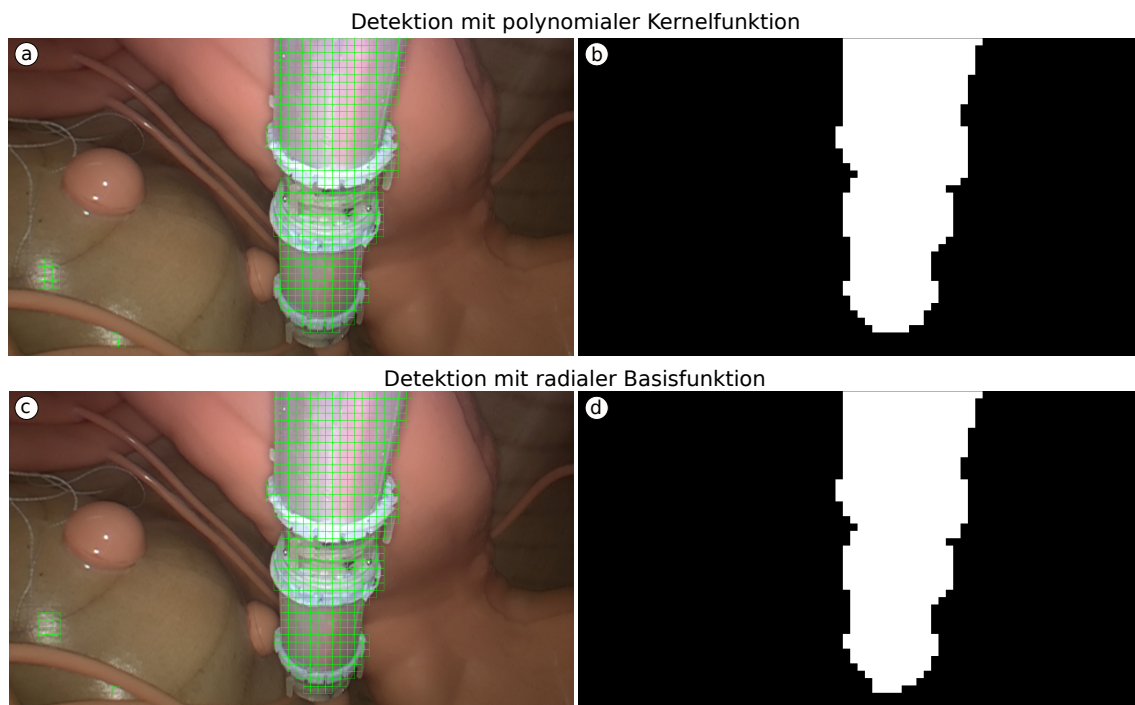


Abbildung 7.22: Detektionsszenario 3 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]

einen geringeren detektierten Anteil als die folgenden Beispielbilder. Es zeigt sich, dass wenn rote Teilstücke des Armes am Rand liegen, die Detektion nicht immer optimal gelingt. Dies ist vermutlich auf die hohe Ähnlichkeit der Teilstücke zu dem Hintergrund zurückzuführen. In den Testbildern 7.21 - 7.23 ist der Anteil der korrekt detektierten Pixel unter Verwendung der radialen Basisfunktion höher als der Anteil der mit der polynomialen Kernelfunktion detektiert wurde. Dies lässt darauf schließen, dass der Einsatz der radialen Basisfunktion zur Detektion des STIFF-FLOP Armes zu dem größten Anteil an detektierten Teilstücken führt, was sich beim Einsatz der Online Detektion bestätigte.



Parameter	Polynomialer Kernel		Radiale Basisfunktion	
	Ungefilterte Detektion	Gefilterte Bildmaske	Ungefilterte Detektion	Gefilterte Bildmaske
Korrekt detektierte Pixel [%]	95.23	95.23	96.15	96.15
Falsch detektierte Pixel [%]	13.63	2.79	13.38	2.70
Nicht detektiert [%]	4.77	4.77	3.85	3.85

Abbildung 7.23: Detektionsszenario 4 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]

Die aufgezeigten Ergebnisse entstanden in Zusammenarbeit mit der betreuten studentischen Arbeit [61].

7.3.6 Implementierung der Online Detektion

Nach dem Training der SVM-Methode kann die Online Detektion ausgeführt werden. Hierfür wird im ersten Schritt das Ergebnis des Trainings, das die Stützvektoren zur Bestimmung der Hyperebene enthält, geladen. Nach dem Erfassen der aktuellen Videoszene, wird diese in Teilstücke zerlegt, die dieselbe Größe wie die Teilstücke der Trainingsprozedur haben. Die Farbinformationen der Pixel werden in eine Matrix geladen und anschließend durch die SVM-Methode klassifiziert. Abschließend wird das Ergebnis, wie in Abschnitt 7.3.4 beschrieben, gefiltert. Die Abbildung 7.24 skizziert die Schritte für die Online Detektion.

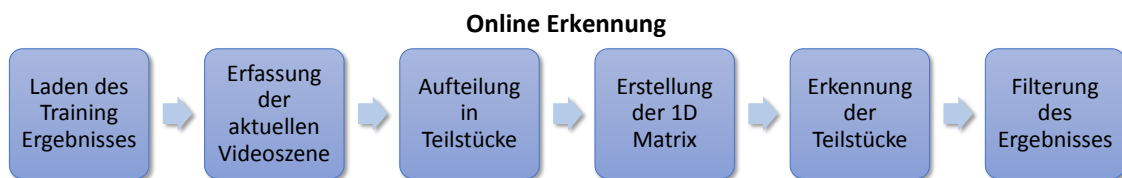


Abbildung 7.24: Programmablauf der „predict_svm“ Funktion zur Online Erkennung des STIFF-FLOP Armes mit der SVM Methode

Die Implementierung der Online Detektion erfolgt in der „Tubedetect“ ROS-Node, die in der Abbildung 7.4 eingeführt ist. Hierdurch erfolgt die Integration in das projektweit verwendete ROS-Framework. Die Node verbindet sich mit einem laufenden ROS-Core und empfängt die von der „Intensity“ Node publizierte Videobilder der Endoskopkamera. Über einen Parameterserver können zur Laufzeit Programmparameter angepasst werden und die Bildverarbeitungsmethoden aktiviert oder pausiert werden. Bei aktivierter Detektion, wird für jedes eingehende Bild eine Callback Funktion aufgerufen, die die Bildverarbeitung ausführt. Die Abbildung 7.25 skizziert den Ablauf der „Tubedetect“ Node.

Im ersten Schritt der Bildverarbeitung wird die bereits im Abschnitt 7.3.4 vorgestellte „predict_svm“ Funktion aufgerufen. Diese wurde insoweit modifiziert, dass die aktuelle Videoszene übergeben wird und als Ergebnis die gefilterte Bildmaske an das Programm zurückgegeben wird. Die Implementierung der Funktion ist im Anhang D.3 angefügt. Das Ergebnis der Detektion bildet den Ausgangspunkt für die Positionsbestimmung des STIFF-FLOP Armes. Die folgenden Schritte zur Positionsbestimmung wurden bereits in der Dissertation [64, S.95-102] vorgestellt und werden hier zusammengefasst.

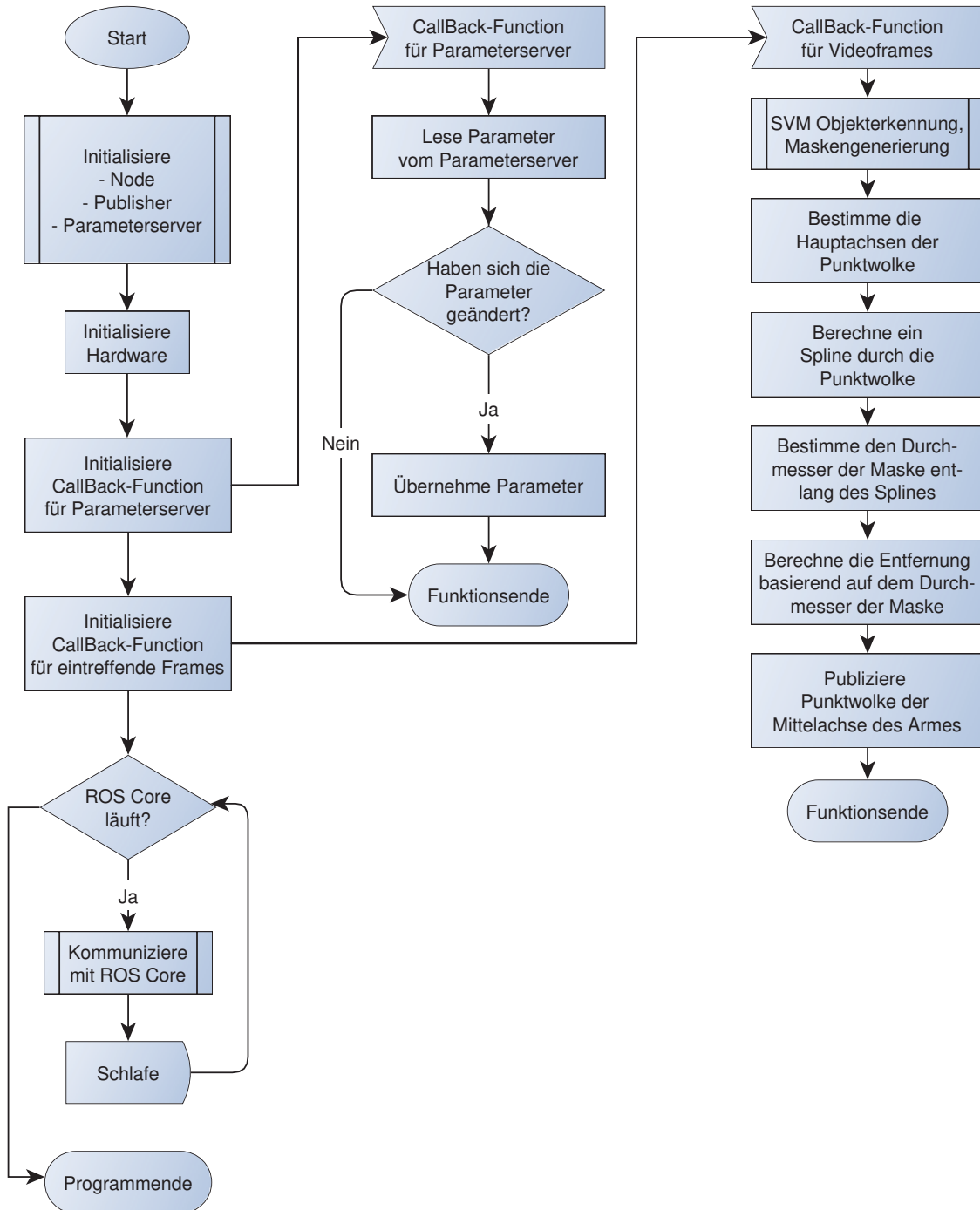


Abbildung 7.25: Programmablauf der „Tubedetect“ Ros-Node zur Detektion des STIFF-FLOP Armes im Videobild und zur Bestimmung der Position im Videobild [64]

In der Bildmaske wird zunächst die Richtung der Hauptachse bestimmt, um danach die Mittellinie durch eine Spline Interpolation dritten Grades zu ermitteln. Hierdurch können zu erwartende Biegungen approximiert werden. Durch die Spline Interpolation ergibt sich eine geglättete Position der Mittellinie aus den erkannten Teilstücken des STIFF-FLOP Armes in der Bildmaske. Anschließend wird die Höhe des STIFF-FLOP Armes an Punkten entlang der Mittellinie bestimmt. Hierfür wird in die Punkte der Mittellinie eine orthogonale Gerade gelegt und die Schnittpunkte mit der Bildmaske bestimmt. Aus den Schnittpunkten ergibt sich die Höhe des STIFF-FLOP Armes in Bildpunkten, an dem jeweiligen Punkt der Mittellinie. Die Abbildung 7.26 zeigt die detektierte Bildmaske sowie das Videobild, in dem die Mittellinie (rot) und die Höhe (cyan) dargestellt sind.

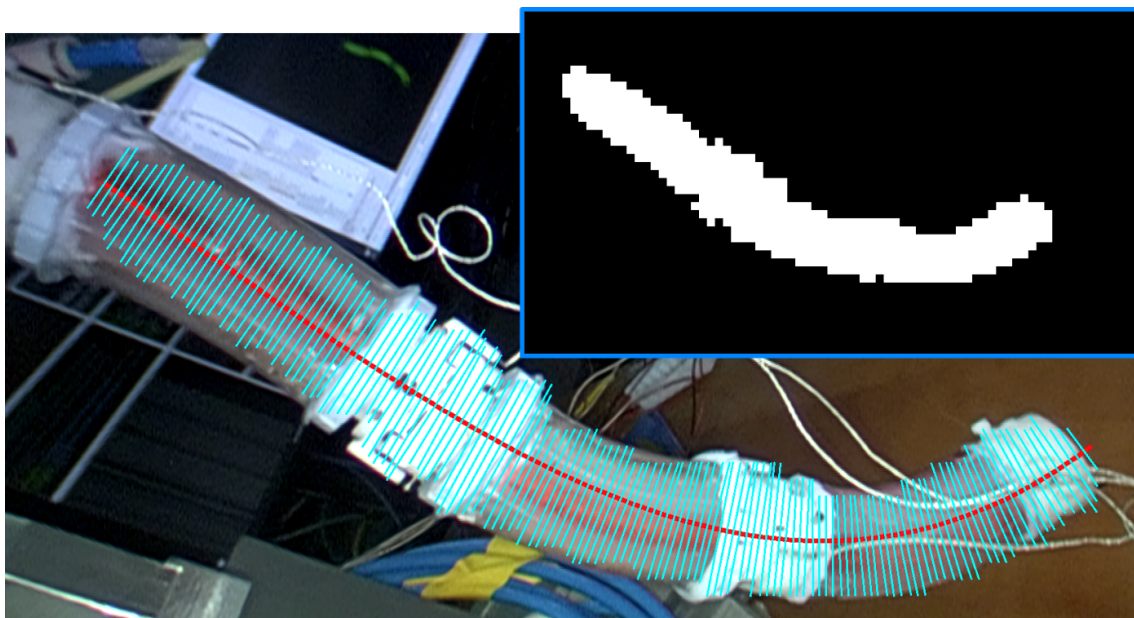


Abbildung 7.26: Mit der SVM Methode detektierte Bildmaske (oben rechts) der Videoszene (unten), in welcher die Mittellinie (rot) und die Höhe des STIFF-FLOP Armes (cyan) dargestellt sind

Die Position des STIFF-FLOP Armes wird im Koordinatensystem des Videobildes bestimmt. Die X- und Y-Komponente ergibt sich aus den Punkten der Mittellinie. Die Entfernung des STIFF-FLOP Armes von der Bildebene wird durch die Z-Komponente beschrieben. Um von der in Bildpunkten bestimmten Höhe auf die Entfernung des STIFF-FLOP Armes schließen zu können, wird zunächst der Abstand des STIFF-FLOP Armes ermittelt, bei dem dieser das Videobild vollständig ausfüllt. Befindet sich der STIFF-FLOP Arm in der Aufnahme weiter entfernt, was dem zu erwartenden Anwendungsszenario entspricht, so ergibt die in Bildpunkten ermittelte Höhe des Armes, einen relativen Anteil der zuvor ermittelten vollständigen

Ausfüllung des Videobildes. Unter Anwendung des Strahlensatzes kann aus dem relativen Anteil der Bildschirmfüllung auf die Entfernung des Objektes geschlossen werden.

Die ermittelten Positionspunkte des STIFF-FLOP Armes werden unter Verwendung der *“point – cloud“* Bibliothek [98] als Punktwolke erstellt und anschließend als eigenes ROS-Topic versendet. Hierdurch stehen diese Daten für anschließende Sensordatenfusionen und Auswertungen zur Verfügung. Durch die Lokalisation der Endoskopkamera und der Basis des STIFF-FLOP Armes kann der im Abschnitt 7.2.1 beschriebene Transformationsbaum erstellt werden. Hiermit kann das publizierte ROS-Topic mit den ermittelten Positionsdaten in das Basiskoordinatensystem des STIFF-FLOP Armes transformiert werden.

Das Ergebnis der Detektion in der Abbildung 7.26 zeigt, dass die Mittellinie des STIFF-FLOP Armes erfolgreich bestimmt wird. Bei der Auswertung der Positionsdaten bestätigte sich, dass die Positionen der X- und Y-Komponenten in der Bildebene eine Bestimmung der Lage in der Bildebene erlauben. Die Auswertung der Z-Komponente, die die Entfernung des STIFF-FLOP Armes von der Bildebene beschreibt, wies jedoch erhebliche Abweichungen zu der tatsächlichen Position auf. Eine genauere Charakterisierung der Ergebnisse erfolgte mit einem Kalibrationsaufbau, der in einer weiterführenden Arbeit durchgeführt wurde [32].

7.4 Weiterführende Arbeiten zur Erkennung der Verbindungspunkte des STIFF-FLOP Armes

Für den Regelalgorithmus war insbesondere die Position der Verbindungspunkte der STIFF-FLOP Arm Module relevant. Hierfür wurde eine Methode zur Detektion der Verbindungspunkte vorgestellt [42] [43]. Die Idee war es, die Rigidität der Verbindungspunkte auszunutzen, um dort farbige, ringförmige Markierungen anzubringen, was in der Abbildung 7.6 dargestellt wird. Hierfür wurden grüne Markierungen ausgewählt, da diese sich besonders gut von den zu erwartenden Farben im Operationsgebiet unterscheiden. Im Kamerabild erscheint der sichtbare Teil der Ringmarkierung als Teilstück einer Ellipse. Die Schritte zur Bestimmung der Mittelpunktposition der Ellipse sind in der Abbildung 7.27 dargestellt.

Im ersten Schritt wird der Marker im Videobild segmentiert und anschließend dessen Kontur bestimmt. Die Kontur wird in einen linken und rechten Teilbereich unterteilt.

Anschließend wird für beide Teilbereiche eine passende Ellipse bestimmt und in dem Bild eingezeichnet. Zwischen den Mittelpunkten der linken und der rechten Ellipse befindet sich der Mittelpunkt des Verbindungselementes. Die Implementierung der Ringmarkererkennung ist in der Dissertation [64] beschrieben.

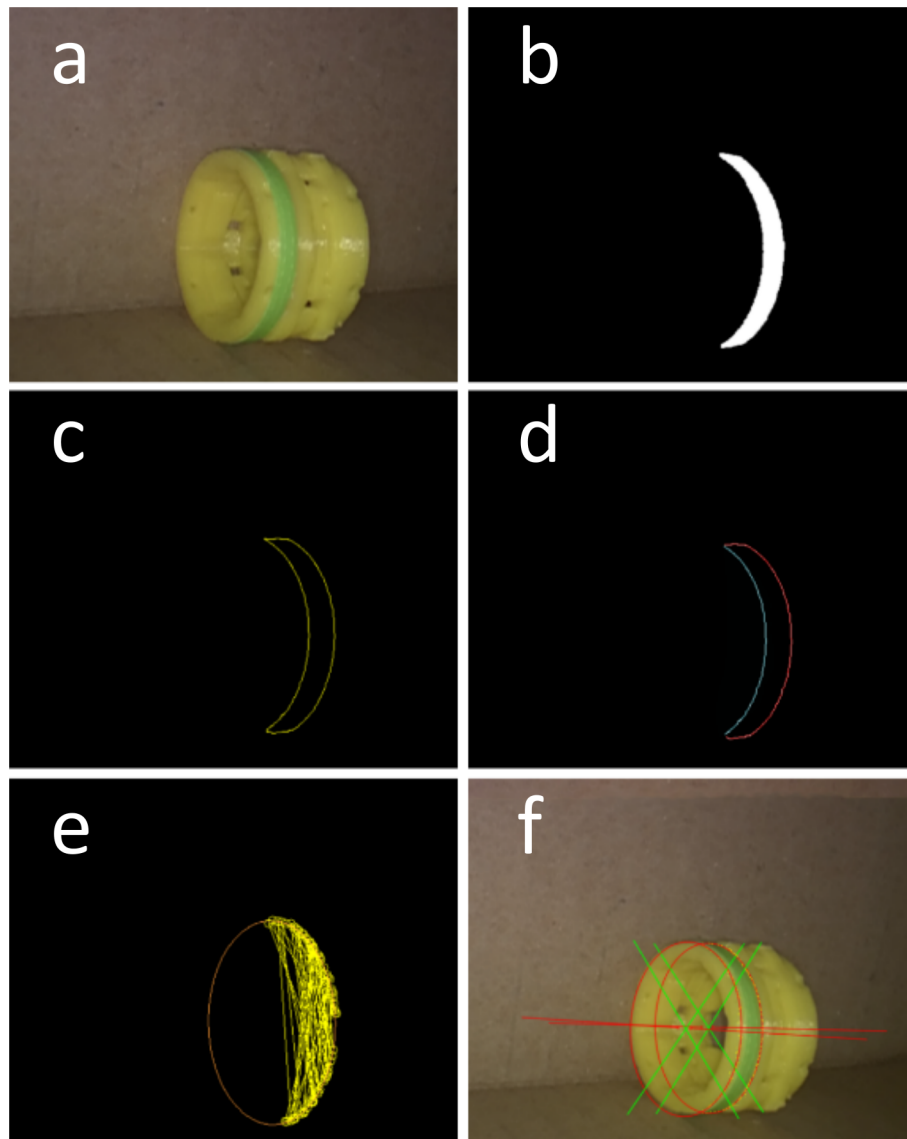


Abbildung 7.27: Schritte zur Detektion des Ringmarkers am Verbindungselement eines STIFF-FLOP Moduls (a), segmentierter Ringmarker(b), Kontur des segmentierten Ringmarkers mit einer Aufteilung in einen linken (blau) und einen rechten (rot) Randbereich (d), Bestimmung der passenden Ellipse (e), Originalbild mit den überlagerten Ellipsen (f) [43]

8 Diskussion der Ergebnisse für die knöchernen und die Weichteilchirurgie

Für das Anwendungsgebiet der knöchernen Chirurgie, mit der Zielsetzung der zielgenauen Platzierung von Pedikelschrauben, wurden im Abschnitt 6.5 Genauigkeitsuntersuchungen des vorgestellten Gesamtsystems durchgeführt. Für eine zielgenaue Platzierung muss die Schraubenposition im Datensatz geplant werden. Anschließend muss der Datensatz mit der intraoperativen Lage des Patienten registriert werden. Dann kann die Schraube mit dem eingesetzten Robotersystem zielgenau platziert werden.

Da die Schritte voneinander abhängen, wurde zur Genauigkeitsbetrachtung die gesamte Kette in mehreren Szenarien betrachtet. Die Ergebnisse des ersten Szenarios, das im Abschnitt 6.5.1 aufgezeigt wird, zeigen, dass die Schritte der Planung und Registrierung erfolgreich an einem Modell durchgeführt werden können und die Zielposition mit einer Abweichung kleiner 1 mm mit einem Zeigeinstrument im Bilddatensatz visualisiert wurde.

Bei der im Abschnitt 6.5.2 beschriebenen Planung an einem Wirbelknochen wurde der Datensatz zusammen mit dem Datensatz aus dem ersten Szenario aufgenommen. Hierdurch wird derselbe Registrierungsschritt aus dem ersten Szenario verwendet, sodass der Einfluss der Registrierung auf das Ergebnis bei beiden Szenarien identisch ist. In dem zweiten Szenario wird die Zielposition, anstelle des im ersten Szenario eingesetzten Zeigeinstrumentes, mit dem Robotersystem angefahren. Hierbei wurde eine Abweichung zwischen der geplanten und der vom Roboter angefahrenen Position von $1,79\text{ mm}$ ermittelt.

Die am Wirbelknochen angebrachten Metallpins, die als Landmarken für die Zielposition verwendet wurden, führten zu Bildartefakten in dem Bilddatensatz. Die Artefakt behaftete Darstellung der Metallpins in der intraoperativen Aufnahme beeinflusst die Präzision der Planung. Daher ist zu erwarten, dass es bereits bei der Planung zu Abweichungen kommt. Weiter gilt es zu berücksichtigen, dass durch den Einsatz des Roboters und dessen Werkzeugkalibrierung Abweichungen auftreten, die im ersten Szenario nicht betrachtet werden. Um das klinische Ziel einer Positioniergenauigkeit mit einer Abweichung von kleiner als 1 mm zu erreichen,

muss die Positioniergenauigkeit des Roboters verfeinert werden. Ein ausschlaggebender Parameter stellt dabei die Kalibrierung des verwendeten Werkzeugs dar. Bei dem dritten Szenario aus dem Abschnitt 6.5.3 wurde das zweite Szenario um eine präoperative Operationsplanung in einem CT-Datensatz ergänzt. Hierdurch ist eine präzise Planung in einem hochauflösenden Datensatz möglich. Der Workflow mit einer präoperativen Planung konnte mit dem Szenario demonstriert werden. Erwartungsgemäß ist die auftretende Abweichung bei der Positionierung in diesem Szenario am größten, da hier zu der Abweichung aus dem zweiten Szenario noch ein Fehler durch das multimodale Matching der 3D-Bilddatensätze hinzukommt. In dem untersuchten Szenario wurden Abweichung von $3,3\text{ mm}$ ermittelt. Hieraus ergibt sich, dass die Abweichung in dem aufgezeigten Szenario um etwa $1,5\text{ mm}$ höher ausfällt als in dem zweiten Szenario. Da derselbe Datensatz aus dem zweiten Szenario für das Matching verwendet wurde, kann dieses Ergebnis mit dem Ergebnis des zweiten Szenarios verglichen werden. Damit ist die Zunahme der Abweichung auf das multimodale Matching zurückzuführen. Für sich betrachtet, ist eine Abweichung von etwa $1,5\text{ mm}$ bei einem multimodalen Matching ein gutes Ergebnis. In der gesamten Anwendungskette zur Umsetzung der präoperativen Planung fällt der resultierende Fehler allerdings noch zu hoch für einen klinischen Einsatz aus.

Eine Verbesserung der Ergebnisse versprechen modernere Flatpanel C-Bögen oder O-Arme zur Bildgebung, die eine höhere Bildqualität liefern und dadurch sowohl die Präzision der Planung verbessern, als auch den Registrierungsfehler durch die höhere Bildqualität mindern. Zum anderen sollten die Tests mit Wirbelknochen durchgeführt werden, bei denen kein Material angebracht ist, dass Artefakte erzeugt.

Im Anwendungsgebiet der Weichteilchirurgie stellt die Detektion des entwickelten STIFF-FLOP Armes eine große Herausforderung an bekannte und neuartige Algorithmen dar. Da der STIFF-FLOP Arm zur Bearbeitungszeit stetig weiterentwickelt wurde und sich hierdurch auch die für die Detektion relevante äussere Form änderte, ist eine trainierbare und lernfähige Methode geeignet, um den STIFF-FLOP Arm zu detektieren. Mit der im Abschnitt 7.3.3 beschriebenen SVM-Methode konnten auch weiterentwickelte Versionen des STIFF-FLOP Armes im Bild der Endoskopkamera erkannt werden. Hierfür muss die SVM-Methode in dem jeweiligen Operationsszenario trainiert werden. Auch für die Verwendung von unterschiedlich vielen STIFF-FLOP Modulen hat sich gezeigt, dass die erfolgreichste Detektion mit Trainingsdaten aus dem aktuellen Szenario erzielt wird.

Die Ergebnisse des untersuchten Testszenarios aus dem Abschnitt 7.3.5 haben gezeigt, dass mit der SVM-Methode eine zuverlässige Detektion des STIFF-FLOP Armes im Videobild der Endoskopkamera erzielt wurde. Die Detektionsraten von 78,44% – 96,15% zeigen eine hohe, jedoch nicht immer vollständige Detektion. Bei Testmessungen im Online Betrieb zeigte die radiale Basisfunktion der SVM-Methode die besten Ergebnisse, im Gegensatz zur polynomialen Basisfunktion die mehrere Teilstücke nicht erkannt hat. Der erhöhte Rechenaufwand der radialen Basisfunktion führte dazu, dass Frameraten von 3-4 Frames pro Sekunde berechnet werden konnten. Bei der Verwendung der polynomialen Basisfunktion war die Berechnung von 7-8 Frames pro Sekunde möglich. Trotz der zuverlässigen Detektion muss beachtet werden, dass durch die Aufteilung des Videobildes in Teilstücke eine Quantisierung stattfindet, die insbesondere die Detektion der Randgebiete beeinflusst. Mit der Implementierung des anpassbaren Vordergrundanteils bei dem Training der Methode, konnten die Parameter für die höchste Detektionsrate bestimmt werden.

In der im Abschnitt 7.3.6 beschriebenen Online Detektion wird die Mittellinie des STIFF-FLOP Armes zuverlässig ermittelt. Hierdurch kann eine Lokalisation in der Bildebene erfolgen. Bei der Bestimmung der Entfernung des STIFF-FLOP Armes kam es hingegen zu Abweichungen, die durch Messungen in folgenden Arbeiten bestätigt wurden.

Mit der im Abschnitt 7.4 vorgestellten Ringmarkerdetektion können die Verbindungsstücke der STIFF-FLOP Module detektiert werden. Der Einsatz der Ringmarkerdetektion wurde durch eine Integration in das STIFF-FLOP Gesamtsystem demonstriert. Dadurch werden auftretende Abweichungen erkannt und als Warnung an den Anwender ausgegeben. Allerdings zeigte sich auch hier, wie schon bei der Erkennung des STIFF-FLOP Armes mit der SVM-Methode, dass bei der Berechnung der Entfernung größere Abweichungen auftreten.

Hieraus lässt sich schließen, dass mit dem aufgebauten optischen Lageüberwachungssystem die Position des STIFF-FLOP Armes sowie die Positionen der Verbindungspunkte bestimmt werden können. Die Ergebnisse liefern zuverlässige Aussagen über die Position in der Bildebene der Endoskopkamera. Bei der Bestimmung der Entfernung des STIFF-FLOP Armes in dem 2D-Bild der Endoskopkamera, führen bereits geringe Abweichungen des Durchmessers des STIFF-FLOP Armes zu großen Abweichungen bei der Berechnung der Entfernung.

9 Zusammenfassung und Ausblick

In dieser Arbeit werden medizinische Bilddaten verarbeitet die von verschiedenen prä- und intraoperativen Bildgebungsmodalitäten stammen können. Der Schwerpunkt liegt insbesondere darin, die Durchführung minimalinvasiver chirurgischer Operationsverfahren mit robotischen Assistenzsystemen zu unterstützen.

Den medizinischen Hintergrund dieser Arbeit bilden zwei verschiedene Anwendungsgebiete. Das erste Anwendungsszenario beschreibt den medizinischen Eingriff zur Wirbelsäulenversteifung. Hierfür müssen Pedikelschrauben zielgenau in die Wirbelknochen eingebracht werden. Als eigener Beitrag wurde im Rahmen dieser Arbeit eine neue Planungssoftware mit dem Software Framework Mevislab entwickelt. Mit dieser Software werden Schraubenposition in medizinischen 3D-Bilddaten geplant und anschließend roboterassistiert an die Zielposition geführt. Im ersten Schritt wird die Bildgebungseinheit mit der Lage des Patienten bei der Aufnahme registriert. Hierfür wurde in dieser Arbeit ein Verfahren basierend auf dem Einsatz von Markern, die sowohl von optischen Stereokameras als auch von intraoperativ eingesetzten C-Bögen erkannt werden, entwickelt. Da das entwickelte Verfahren herstellunabhängig ist, können Bilddatensätze verschiedener Bildgebungsgeräte verwendet werden.

Um dem Chirurgen eine Planung mit bereits vor dem Eingriff erstellten, hochauflösenden Bilddatensätzen zu ermöglichen, wurde die Planungssoftware um eine präoperative Planung erweitert. Hierfür werden die Bilddaten mit einem multimodalen 3D-3D Matching mit den intraoperativen Bilddaten überlagert.

Jeder Schritt von der Planung über die Registrierung, bis hin zur roboterassistierten Ausführung, beeinflusst die erzielbare Genauigkeit. Zur Untersuchung der Ergebnisse wurden verschiedene Szenarien, mit dem chirurgischen Workflow zur Planung und Platzierung der Pedikelschrauben, untersucht. Hierbei konnten die geplanten Schraubenpositionen mit einer Abweichung kleiner als 1 mm im Bilddatensatz visualisiert werden. Bei der anschließenden roboterassistierten Platzierung von Pedikelschrauben konnte die Zielposition mit einer Abweichung von $1,78\text{ mm}$ erreicht werden. Der Einsatz einer präoperativen Planung konnte in einem weiteren Szenario dargestellt werden.

Mit der neuen Planungssoftware und der implementierten Registrierungsmethode wurde die Basis geschaffen, um den mit einem Kooperationspartner neu entwickelten Operationsroboter an die Software anzubinden. Die Steuerung des neuen Roboterarmes erfolgt in einem eigenen Mevislab Modul, welches direkt in die Planungssoftware eingebunden werden kann, um die geplanten Zielposen anzufahren. Sobald der neue Roboterarm fertiggestellt ist, können die im Rahmen dieser Arbeit untersuchten Operationsszenarien direkt mit dem neuen Roboterarm evaluiert werden.

Im zweiten Anwendungsgebiet für die Weichteilchirurgie, wurde ein Lageüberwachungssystem für den im Rahmen des STIFF-FLOP Projektes neu entwickelten flexiblen Operationsroboter aufgebaut. Hierfür wird der Operationsroboter von einer Endoskopkamera beobachtet. Die Position der Endoskopkamera wird von einer Stereokamera überwacht. Die größte Herausforderung besteht in der Detektion des flexiblen Operationsroboters in dem Videobild der Endoskopkamera. Im Rahmen dieser Arbeit wurden hierfür verschiedene Methoden ausgewählt und deren Einsatz evaluiert. Eine besondere Herausforderung waren die speziellen Eigenschaften des flexiblen Manipulators, die sich durch die Weiterentwicklung des Armes während der Projektlaufzeit änderten. Insbesondere die Weiterentwicklung der ursprünglichen Variante mit einer gewobenen Aussenhülle, hin zu einer integrierten Lösung die eine Aufblähung des Manipulators verhinderte, veränderte die für die Detektion relevante äussere Form signifikant.

Als eigener Beitrag wurde im Rahmen dieser Arbeit durch den Einsatz von Support-Vector-Machines eine lernfähige Methode für die Erkennung des STIFF-FLOP Armes im Videobild der Endoskop Kamera umgesetzt. Diese Methode kann für das gewählte Einsatzgebiet und auch für weiterentwickelte Versionen des Manipulators trainiert werden. Mit dieser Implementierung wurden Detektionsraten von 78,44% - 96,15% erzielt. Anschließend wurde die Mittellinie und der Durchmesser des Manipulators bestimmt. Hierdurch kann die Position in der Kameraebene angegeben werden. Die Ergebnisse haben gezeigt, dass die Bestimmung der Position in der Kameraebene gelingt. Aus dem ermittelten Durchmesser erfolgte die Abschätzung der Entfernung des Manipulators von der Bildebene. Diese führte jedoch zu größeren Abweichungen. Prinzip bedingt führen hier bereits geringste Abweichungen bei der Bestimmung des Durchmessers zu großen Fehlern bei der Entfernungsberechnung. Über die eingesetzte Stereokamera können die ermittelten Positionsdaten in andere Bezugskoordinatensysteme transformiert werden. Dadurch können alle Positionsdaten relativ zur Basis des Manipulators angegeben werden.

Die vorgestellte Detektionsmethode ist sehr flexibel einsetzbar und liefert eine hohe Detektionsrate. Daher kann die Methode als Vorlage zur Detektion von Szenarien dienen, die mit konventionellen Bildverarbeitungsmethoden nicht erfasst werden können.

Die Methode kann erweitert werden, um die einzelnen Segmente des STIFF-FLOP Armes separiert voneinander zu detektieren. Hierfür können diese unterschiedlich eingefärbt werden und anschließend mit der implementierten Methode detektiert werden. Für die Regelung des Armes relevant war die Erfassung der Position der Verbindungsstücke der Armsegmente. Hierfür wurde eine weitere Methode implementiert die bereits in einer vorhergehenden Arbeit beschrieben ist [64].

Um die Entfernung des STIFF-FLOP Armes von der Kameraebene genauer erfassen zu können, bietet der Einsatz von aktuell verfügbaren Stereoendoskopkameras einen vielversprechenden Ansatz. Einen alternativen Ansatz zur Erfassung der Position des STIFF-FLOP Armes bietet die Integration von elektromagnetischen Markern eines elektromagnetischen Tracking Systems (NDI Aurora).

Literaturverzeichnis

- [1] Accuray Inc. Technologie des cyberknife systems. <https://cyberknife.com/cyberknife-technology/>. Aufgerufen am 17.09.2020.
- [2] J. C. Alvarez Gomez. Implementation of basic functions for a navigated surgical robot system based on optical 3d digitizing systems and the mevislab software framework system based on optical 3d digitizing systems and the mevislab software framework. Master's thesis, Universität Siegen, 2018.
- [3] A. Arezzo, Y. Mintz, M. Allaix, S. Arolfo, M. Bonino, G. Gerboni, M. Brancadoro, M. Cianchetti, A. Menciassi, H. Wurdemann, Y. Noh, K. Althoefer, F. Jan, J. Glowka, Z. Nawrat, G. Cassidy, R. Walker, and M. Morino. Total mesorectal excision using a soft and flexible robotic arm: a feasibility study in cadaver models. *Surgical Endoscopy*, 31:264–273, 01 2017.
- [4] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:698–700, 1987.
- [5] Atracsys. Produktdatenblatt: fusiontrack 500. <https://www.atracsys-measurement.com/products/fusiontrack-500/>. Aufgerufen am: 01.08.2020.
- [6] AVRA Medical Robotics Inc. (AVMR). The official medical robotic news center sponsored by avra medical robotics. <https://allaboutroboticsurgery.com/surgical-robots.html>. Aufgerufen am 27.08.2020.
- [7] AXIOS 3D Services GmbH. Datenblatt optisches messsystem cambar b2 c8. https://axios3d.de/wp-content/uploads/2018/07/Datenblatt_CamBarB2C8.pdf, Mai 2014. Aufgerufen am 08.02.2020.
- [8] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.
- [9] A. Bauer. *Computer and Robotic Assisted Knee and Hip Surgery*, chapter Total hip replacement – robotic assisted technique, pages 83–96. Oxford University Press, 2004.

- [10] A. Ben-Hur and J. Weston. A user's guide to support vector machines. *Methods in molecular biology (Clifton, N.J.)*, 609:223–39, 01 2010.
- [11] P. Berthet-Rayne, G. Gras, K. Leibrandt, P. Wisanuvej, A. Schmitz, C. Seneci, and G.-Z. Yang. The i2snake robotic platform for endoscopic surgery. *Annals of Biomedical Engineering*, 46:1663–1675, 06 2018.
- [12] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.
- [13] J. Beyerer and F. León. Die radontransformation in der digitalen bildverarbeitung (the radon transform in digital image processing). *Automatisierungstechnik - AT-AUTOM*, 50, 01 2002.
- [14] Blackmagic Design. *SDK-Decklink Software Developers Kit*, 2012.
- [15] T. Boehler, D. van Straaten, J. Strehlow, and S. Wirtz. Merit module reference. <https://mevislabdownloads.mevis.de/docs/2.4/FMEstable/ReleaseMeVis/Documentation/Publish/ModuleReference/MERIT.html>. Aufgerufen am 24.09.2020.
- [16] T. Boehler, D. van Straaten, S. Wirtz, and H.-O. Peitgen. A robust and extendible framework for medical image registration focused on rapid clinical application deployment. *Computers in Biology and Medicine*, 41(6):340 – 349, 2011.
- [17] M. Börner and W. Ditzen. *Navigation und Robotik in der Gelenk- und Wirbelsäulen Chirurgie*, chapter Aktueller Stand der Roboterchirurgie — Ausblick auf zukünftige Optionen, pages 435–448. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [18] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.
- [19] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 25, 2000.
- [20] BRAINLAB AG. Produktbeschreibung: Cirq - robotische assistenz. <https://www.brainlab.com/de/chirurgie-produkte/>

- uebersicht-ueber-plattformprodukte/cirq-robotics/. Aufgerufen am 16.08.2020.
- [21] BRAINLAB AG. Produktbeschreibung: Curve navigationssystem. <https://www.brainlab.com/de/chirurgie-produkte/uebersicht-ueber-plattformprodukte/curve-bildgestuetzte-chirurgie/>. Aufgerufen am 21.09.2020.
- [22] BRAINLAB AG. Spine navigation: Image guided spine surgery. <https://www.brainlab.com/wp-content/uploads/2019/07/Spine-Image-Guided-Surgery.pdf>, 2018. Aufgerufen am 21.08.2019.
- [23] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.
- [24] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [25] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [26] R. A. C. Cruces. *Concept and Design of a Cooperative Robotic Assistant Surgery System*. Dr.-Ing. Dissertation, Universität Siegen, 2008.
- [27] C. C. da Rosa. *Operationsroboter in Aktion. Kontroverse Innovationen in der Medizintechnik*. PhD thesis, RWTH Aachen University, Bielefeld, Mai 2013.
- [28] B. L. Davies, F. M. R. y Baena, A. R. W. Barrett, M. P. S. F. Gomes, S. J. Harris, M. Jakopc, and J. P. Cobb. Robotic control in knee joint replacement surgery. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 221(1):71–80, 2007. PMID: 17315770.
- [29] C. Dominguez. Recognition of a flexible tube for surgical operations using an endoscopic camera based on support vector machines. Master’s thesis, Universität Siegen, 2014.
- [30] P. E. Dupont and M. Mahvash. Snakes, worms and catheters: Continuum and serpentine robots for minimally invasive surgery. *IEEE ICRA 2010 Full Day Workshop*, 2010.

- [31] D. Eggert, A. Lorusso, and R. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Application*, 9:272–290, 1997.
- [32] O. G. A. Elghany Abd Elrazeq. Calibration of a vision sensor using an industrial manipulator. Master’s thesis, Universität Siegen, 2015.
- [33] J. Fitzpatrick, J. West, and C. Jr. Predicting error in rigid-body point-based registration. *Medical Imaging, IEEE Transactions on*, 17:694 – 702, 11 1998.
- [34] T. Foote. Ros Übersicht: Einführung. <http://wiki.ros.org/de/ROS/Introduction>, 12.06.2013. Aufgerufen am 28.10.2019.
- [35] T. Foote. Tf: The transform library. *IEEE Conference on Technologies for Practical Robot Applications, TePRA*, pages 1–6, 04 2013.
- [36] T. Foote, E. Marder-Eppstein, and W. Meeussen. Ros tf:package wiki. <http://wiki.ros.org/tf>. Aufgeufen am 12.02.2020.
- [37] J. Fras, J. Czarnowski, M. Macias, J. Glówka, M. Cianchetti, and A. Menciassi. New stiff-flop module construction idea for improved actuation and sensing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2901–2906, May 2015.
- [38] E. Gerz, D. Geraskin, J. Franke, P. Platen, A. Steimers, and M. Kohl-Bareis. Tissue oxygenation during exercise measured with nirs: Reproducibility and influence of wavelengths. In S. Van Huffel, G. Naulaers, A. Caicedo, D. F. Bruley, and D. K. Harrison, editors, *Oxygen Transport to Tissue XXXV*, pages 171–177, New York, NY, 2013. Springer New York.
- [39] E. Gerz, D. Geraskin, P. Neary, J. Franke, P. Platen, and M. Kohl-Bareis. Tissue oxygenation during exercise measured with nirs: a quality control study. In *Proc. SPIE 7368, Clinical and Biomedical Spectroscopy*, volume 7368, 2009.
- [40] E. Gerz, O. Gieseler, J. Wahrburg, and H. Roth. Intra-operative registration of any 3d imaging modality with navigation systems in computer assisted spine surgery. In *NSpine Main Conference*, London, UK, 1-3 Juli 2019.
- [41] E. Gerz, M. Mende, and H. Roth. Optische detektion und positionsbestimmung eines neuartigen flexiblen minimal-invasiven chirurgischen robotersystems. In *IEEE Workshop 2015 Medizinische Messtechnik, Vortragsreihe der Hochschule Ruhr West*, Mühlheim an der Ruhr, 17 April 2015.

- [42] E. Gerz, M. Mende, and H. Roth. Development of an optical tracking system for a novel flexible and soft manipulator with controllable stiffness for minimal invasive surgery (mis). In *IEEE Workshop Industrial and Medical Measurement and Sensor Technology, Vortragsreihe an der Hochschule Ruhr West*, Mülheim an der Ruhr, 16-17 Juni 2016.
- [43] E. Gerz, M. Mende, and H. Roth. Development of an optical tracking system for a novel flexible and soft manipulator with controllable stiffness for minimal invasive surgery (mis). *tm - Technisches Messen*, 84(1):47–52, 2017.
- [44] E. Gerz, M. Mende, and H. Roth. The stiff-flop vision system. In J. Konstantinova, H. Wurdemann, A. Shafti, A. Shiva, and K. Althoefer, editors, *Soft and Stiffness-controllable Robotics Solutions for Minimally Invasive Surgery: The STIFF-FLOP Approach*, chapter 8, pages 129–148. River Publishers Series in Automation, Control and Robotics, June 2018.
- [45] S. J. Harris, F. Arambula-Cosio, Q. Mei, R. D. Hibberd, B. L. Davies, J. E. Wickham, M. S. Nathan, and B. Kundu. The probot—an active robot for prostate resection. *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, 211:317–325, 1997.
- [46] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, chapter Epipolar Geometry and the Fundamental Matrix, page 239–261. Cambridge University Press, 2 edition, 2004.
- [47] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.
- [48] P. Heeckt, M. Rühl, G. Buchhorn, H. G. Willert, C. O. R. Grüneis, F. F. Hennig, J. Petermann, P. Heinze, L. Gotzen, R. Kober, H. Gerhardt, M. Romanowski, J. Repicci, U. Mall, and P. Habermeyer. *Rechnergestützte Verfahren in Orthopädie und Unfallchirurgie*, chapter Computer Assisted Surgical Planning and Robotics mit dem CASPAR-System, pages 414–433. Steinkopff, Heidelberg, 1999.
- [49] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629, 1987.
- [50] B. K. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5:1127, 1988.

- [51] Intuitive Surgical Inc. The davinci surgical system. <https://www.intuitive.com/en-us/products-and-services/da-vinci/systems>. Aufgerufen am 19.09.2020.
- [52] B. Jähne. *Digitale Bildverarbeitung*. Springer Vieweg, Berlin, 7 edition, 2012.
- [53] T. Kerstein. *Entwicklung von Verfahren und Elementen zur optischen 3D-Lokalisierung in der computerassistierten Chirurgie*. Dr.-Ing. Dissertation, Universität Siegen, 2014.
- [54] P. Knappe. *Konzeption und Realisierung eines navigierten interaktiven Chirurgierobotersystems*. Dr.-Ing. Dissertation, Universität Siegen, 2008.
- [55] M. Künkler. Matching sortierter punktwolken. Seminararbeit, Universität Siegen, April 2014.
- [56] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2):153–160, 1988.
- [57] O. Lechky. Worlds first surgical robot in b.c. *The Medical Post*, 21:92–93, 1985.
- [58] F. Leitner, F. Picard, R. Minfelde, H. J. Schulz, P. Cinquin, and D. Saragaglia. Computer-assisted knee surgical total replacement. In J. Troccaz, E. Grimson, and R. Mösges, editors, *CVRMed-MRCAS'97*, pages 629–638, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- [59] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 11 2004.
- [60] U. Ludwig. Roboter außer kontrolle. <https://www.spiegel.de/spiegel/spiegelwissen/d-65886423.html>, 30.06.2009. Aufgerufen am 18.09.2020.
- [61] P. K. Maddur. Automated training of support vector machines for the detection of a flexible surgical manipulator. Studienarbeit, Universität Siegen, Januar 2016.
- [62] Z. Malota, Z. Nawrat, and W. Sadowski. Benchmarking for surgery simulators. In J. Konstantinova, H. Wurdemann, A. Shafti, A. Shiva, and K. Althoefer, editors, *Soft and Stiffness-controllable Robotics Solutions for Minimally Invasive Surgery: The STIFF-FLOP Approach*, chapter 17, pages 309–323. River Publishers Series in Automation, Control and Robotics, June 2018.

- [63] Mazor Robotics. Renaissance - how it works step 4 - operate. <https://www.youtube.com/watch?v=4Yopv2PVJeg&feature=youtu.be>. Aufgerufen am 24.09.2020.
- [64] M. Mende. *Entwicklung eines optischen Lageüberwachungssystems für einen neuartigen flexiblen Roboter mit steuerbarer Steifigkeit zur Durchführung minimalinvasiver chirurgischer Eingriffe*. Dr.-Ing. Dissertation, Universität Siegen, 2018.
- [65] M. Mende, E. Gerz, and H. Roth. Evaluation of a magnetic 3d measurement system for application in computer assisted surgery compared to established optical tracking systems. In *9th International conference on Communications, Electromagnetics and Medical Application (CEMA 2014)*, Sofia, Bulgaria, 16-18 October 2014.
- [66] M. Mende, E. Gerz, and H. Roth. Evaluierung eines magnetischen tracking systems für den einsatz in einem neuartigen minimal-invasiven chirurgischen robotersystem. In *IEEE Workshop 2015: Medizinische Messtechnik*, pages 14–15, Mülheim an der Ruhr, 17. April 2015.
- [67] A. Mendivil, R. Holloway, and J. Boggess. Emergence of robotic assisted surgery in gynecologic oncology: American perspective. *Gynecologic oncology*, 114:S24–31, 08 2009.
- [68] MeVis Medical Solutions. Mevislab reference manual. <https://mevislabdownloads.mevis.de/docs/current/MeVisLab/Resources/Documentation/Publish/SDK/MeVisLabManual/index.html>. Aufgerufen am: 26.03.2019.
- [69] MeVis Medical Solutions. Open inventor integration in mevislab. <https://www.mevislab.de/mevislab/features/open-inventor>. Aufgerufen am 11.10.2020.
- [70] NEMA PS3 / ISO 12052 National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom) standard. <http://medical.nema.org/>. Aufgerufen am 28.07.2020.
- [71] Y. Noh, S. Sareh, J. Back, H. A. Würdemann, T. Ranzani, E. L. Secco, A. Faragasso, H. Liu, and K. Althoefer. A three-axial body force sensor for flexible manipulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6388–6393, Hong Kong, China, 2014.

- [72] Northern Digital Inc. Ndi aurora product specifications. <https://www.ndigital.com/medical/products/aurora/>. Aufgerufen am 15.08.2020.
- [73] Northern Digital Inc. Ndi polaris vega website. <https://www.ndigital.com/medical/products/polaris-vega/>. Aufgerufen am 10.10.2019.
- [74] Northern Digital Inc. Ndi 6d architect user guide. https://tsgdoc.socsci.ru.nl/images/c/cf/NDI_6D_Architect_User_Guide%28IL-1070059%29.pdf, 2004. Aufgerufen am 14.08.2020.
- [75] Northern Digital Inc. Polaris tool design guide, 2007.
- [76] Northern Digital Inc. Passive polaris vega user guide, November 2017.
- [77] OpenCV Documentation. Camera calibration and 3d reconstruction. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, 2011-2014. Aufgerufen am: 15.02.2020.
- [78] OpenCV Documentation. Interactive foreground extraction using grabcut algorithm. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_grabcut/py_grabcut.html, 11 2014. Aufgerufen am 18.03.2020.
- [79] OpenCV Documentation. Introduction to support vector machines. https://docs.opencv.org/3.4.0/d1/d73/tutorial_introduction_to_svm.html, 12 2017. Aufgerufen am: 11.03.2020.
- [80] T. Ota, A. Degani, D. Schwartzman, B. Zubiato, J. McGarvey, H. Choset, and M. Zenati. A highly articulated robotic surgical system for minimally invasive surgery. *The Annals of thoracic surgery*, 87:1253–6, 05 2009.
- [81] N. Papenberg, T. Lange, S. Heldmann, and B. Fischer. Bildregistrierung. In *Computerassistierte Chirurgie*, chapter 4, pages 86–118. Elsevier, Urban & Fischer, 2011.
- [82] S. Pieck. *LEISTUNGSANALYSE EINES OPTISCHEN 3D/6D LOKALISIERUNGSSYSTEMS UND DESSEN INTEGRATION IN EIN CHIRURGISCHES ASSISTENZSYSTEM*. Dr.-Ing. Dissertation, Universität Siegen, 2006.
- [83] A. C. Ramos, E. L. de Souza Bastos, and K. C. Kim. *Obesity and Diabetes: New Surgical and Nonsurgical Approaches*, chapter Perspectives of Robotic Bariatric Surgery, pages 69–87. Springer International Publishing, Cham, 2015.

- [84] T. Ranzani, M. Cianchetti, G. Gerboni, I. D. Falco, and A. Menciassi. A soft modular manipulator for minimally invasive surgery: Design and characterization of a single module. *IEEE Transactions on Robotics*, 32(1):187–200, Feb 2016.
- [85] F. Rodriguez y Baena and B. Davies. Robotic surgery: From autonomous systems to intelligent tools. *Robotica*, 28:163–170, 03 2010.
- [86] C. Rother, V. Kolmogorov, and A. Blake. “grabcut”: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04, page 309–314, New York, NY, USA, 2004. Association for Computing Machinery.
- [87] P. M. Schlag, S. H. Eulenstein, T. H. Lange, and J. Adermann. *Computerassistierte Chirurgie*. Elsevier, Urban & Fischer, München, 1. aufl. edition, 2011.
- [88] M. Schlimbach. *Konzeptionierung und Realisierung eines modularen Softwareframeworks zur interaktiven Planung und Durchführung computerassistierter chirurgischer Eingriffe mit einem navigierten Chirurgieroboter*. Dr.-Ing. Dissertation, Universität Siegen, 2012.
- [89] H.-C. Schneider. *Modellgestützte Analyse und Implementierung der roboterbasierten Echtzeitnachführung chirurgischer Instrumente*. Dr.-Ing. Dissertation, Universität Siegen, 2010.
- [90] H.-C. Schneider and J. Wahrburg. Experimentelle bestimmung geeigneter referenzgebergeometrien für eine navigierte mechatronische instrumentennachführung in echtzeit. *Fortschritt-Berichte VDI*, 274(274):29–30, 2009.
- [91] J. Shang, D. P. Noonan, C. Payne, J. Clark, M. H. Sodergren, A. Darzi, and G. . Yang. An articulated universal joint based flexible access robot for minimally invasive surgery. In *2011 IEEE International Conference on Robotics and Automation*, pages 1147–1152, 2011.
- [92] N. Smilowitz, J. Moses, F. Sosa, B. Lerman, Y. Qureshi, K. Dalton, L. Privitera, D. Canone-Weber, V. Singh, M. Leon, and G. Weisz. Robotic-enhanced pci compared to the traditional manual approach. *The Journal of invasive cardiology*, 26:318–21, 07 2014.
- [93] A. Spieß. Evaluierung von registrierungs- und segmentierungsverfahren für knöcherne strukturen, mithilfe der softwarepakete 3d-slicer und mevislab. Bachelorarbeit, Universität Siegen, Dezember 2018.

- [94] Statistisches Bundesamt. Die 20 häufigsten operationen insgesamt. <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Krankenhaeuser/Tabellen/drg-operationen-insgesamt.html>. Aufgerufen am 21.09.2020.
- [95] J. Wahrburg. modicas - modular interactive computer assisted surgery website. http://www.zess.uni-siegen.de/mecha_medi_home/mechatronik-medizintechnik/medizintechnik-modicas/. Aufgerufen am 04.05.2019.
- [96] J. Wahrburg. Robotik 1: Skript zur vorlesung. Lehrstuhl für Regelungs- und Steuerungstechnik, Universität Siegen, Oktober 2010.
- [97] R. J. Webster, A. M. Okamura, and N. J. Cowan. Toward active cannulas: Miniature snake-like surgical robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2857–2863, 2006.
- [98] W. Woodall and J. Kammerl. Ros point cloud library. <http://wiki.ros.org/pcl>. Aufgerufen am 12.02.2020.
- [99] B. Wowra and A. Muacvic. *Computerassistierte Chirurgie - State of the Art*, chapter Radiochirurgie mit Cyberknife, pages 587–590. Elsevier, Urban & Fischer, 2011.
- [100] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, December 2000. MSR-TR-98-71, Updated March 25, 1999.
- [101] G. Zheng and L. P. Nolte. Computer-assisted orthopedic surgery: Current state and future perspective. *Frontiers in surgery*, 2:66, 2015.
- [102] Ziehm imaging. Ziehm naviport 3d interface for image-guided navigation. http://aadee.com/00pdf/medicina/EN_NaviPort_3D_Brainlab_Broschure_global_2016_09_08.pdf. Aufgerufen am 21.08.2019.

Abbildungsverzeichnis

2.1	Das von Integrated Surgical Systems entwickelte ROBODOC System zum Ausfräsen des Femurschaftes für Hüftprothesen [67]	4
2.2	Das DaVinci Surgical System, (c) [2022] Intuitive Surgical Operations, Inc. — reproduced by courtesy of the manufacturer	6
2.3	Das Cyberknife System, bestehend aus dem Linearbeschleuniger am Roboterarm, zwei Röntgenröhren zur Positionsbestimmung, einer Stereokamera für die Bewegungsnachverfolgung, sowie der robotischen Behandlungsliege. Quelle: [1], Image courtesy of Accuray Incorporated - (c) [2022] Accuray Incorporated. All Rights Reserved.	7
3.1	Mit einem Referenzkörper ausgestattetes Zeigeinstrument und ein am Knochen befestigter Referenzkörper für eine intraoperative Navigation [101]	9
4.1	3D C-Bogen der Firma Ziehm - Vario 3D	13
4.2	Endoskopkamera Endocam HD 5509 (Richard Wolf) mit zugehöriger Xenon-Lichtquelle	14
4.3	Patientenkoordinatensystem mit der axialen Ebene (blau), der koronalen Ebene (grün) und der sagitalen Ebene (rot)	15
4.4	Funktionsprinzip zur Detektion eines Referenzkörpers mit 4 retroreflektierenden Markerkugeln mit einer Stereokamera [53]	16
4.5	Das elektromagnetische Tracking System NDI Aurora bei Messungen zur Untersuchung von Störeinflüssen	18
4.6	Das „Curve Navigation“ System der Firma Brainlab, bestehend aus einer Stereokamera und einem Monitor zur Visualisierung von Instrumenten. copyright: Brainlab AG, Quelle: [21]	20
4.7	Das Renaissance System auf der Wirbelsäule mit einer Führung zur Unterstützung der Platzierung von Pedikelschrauben. Quelle [63] copyright: Mazor Robotics	21
4.8	Das ROSA Spine System der Firma Zimmer Biomet, vorgestellt auf dem Deutschen Wirbelsäulenkongress 2016	21
4.9	Das ExcelsiusGPS System von Globus Medical, welches auf dem Deutschen Wirbelsäulen Kongress 2016 vorgestellt wurde	22

4.10	Prototyp der Active Cannula, aufgebaut aus ineinander liegenden vorgebogenen superelastischen Nitinol Röhrchen [97]	23
4.11	Der CardioARM, bestehend aus zylinderförmigen Verbindungsstücken mit Kugelgelenken (links) und die Basisstation mit Kabelzügen zur Steuerung des Armes [80]	24
4.12	Das I-SNAKE System mit integrierten Motoren (links) Quelle:[91], sowie das weiterentwickelte I2-SNAKE System das über Kabelzüge aktiviert wird, montiert an einem Roboterarm mit 2 Greifwerkzeugen an der Spitze (rechts) Quelle:[11]	25
5.1	Präoperative Röntgenaufnahme einer gekrümmten Wirbelsäule. Der zweite Wirbel (Lendenwirbel 4) berührt den unteren Wirbel rechtsseitig	27
5.2	Wirbelmodell der Lendenwirbelsäule mit eingebrachten Pedikelschrauben die mit einem Metallstab verbunden sind	28
5.3	Aufbau des Wirbelkörpers mit schematischer Darstellung der Pedikelschraubenplatzierung	29
5.4	a) und b) zeigen die schematische Querschnittansicht des Enddarmbereiches, wo der Enddarm vom angrenzenden Gewebe gelöst werden muss. Bild c) zeigt die Position von Trokarzugängen für Werkzeuge und Kameras. Am Bauchnabel wird ein Laparoskop eingeführt, das den gelben Bereich überwacht. Die endoskopische Ansicht dazu zeigt d). Quelle: STIFF-FLOP Projektdokumentation	30
6.1	Workflow zum Setzen von Pedikelschrauben mit der Option einer präoperativen Planung	33
6.2	Das chirurgische Assistenzsystem MODICAS, mit einer Führung für chirurgische Instrumente am Endeffektor [95]	35
6.3	Skizzierung der intraoperativ verwendeten Koordinatensysteme [87] .	36
6.4	Systemarchitektur des MODICAS Assistenzrobotersystems	37
6.5	Anbindung der Hardware Komponenten an die Planungssoftware . . .	38
6.6	Mevislab Netzwerk zum Laden von DICOM Bilddaten	40
6.7	Settings Panel des SoGVRVolumeRenderer Moduls zur Berechnung einer maßstabstreuen Darstellung durch Multiplikation der Transformationsmatrix (Formel: 6.1) durch die Aktivierung des Parameters „use world coordinate“	41
6.8	3D-Visualisierung eines DICOM Datensatzes mit dem eingezeichneten Patientenkoordinatensystem und der Ausrichtung des Patienten .	42

6.9	Mevislab Netzwerk zur Visualisierung der koronalen Schichtebene eines 3D-Bilddatensatzes	43
6.10	Parametrierung des „ <i>SoOrthoView2DModuls</i> “ und orthogonale 2D-Schichtansicht der koronalen Schichtansicht eines DICOM Datensatzes (links) und die Parametrierung des „ <i>SoView2DPosition</i> “ Moduls zum Erfassen der Mauszeigerposition und zum Hervorheben der aktuellen Schichtansichten (rechts)	44
6.11	Mevislab Netzwerk zum Hervorheben der aktuell angezeigten Schichtebenen und Parametrierung des „ <i>MPR</i> “ Moduls	45
6.12	3D-Visualisierung eines 3D-Datensatzes, in dem die aktuell angezeigten Schichtansichten farblich umrandet eingeblendet sind (a), axiale Schichtansicht (b) die blau umrandet dargestellt ist, koronale Schichtansicht (c) die grün umrandet dargestellt ist und die sagittale Schichtansicht (d) die rot umrandet dargestellt ist	46
6.13	Laden und Anzeigen eines 3D-Objektes in Mevislab (a), das Panel des „ <i>SoTransform</i> “ Moduls (b) und die gerenderte Darstellung des SoSeparator Moduls wo ein Schraubenmodell in Gelb dargestellt ist (c)	47
6.14	Positionierung eines Schraubenmodells (gelb) in der 3D-Ansicht des Bilddatensatzes (a) und in den Schichtansichten des Datensatzes (b), (c), (d)	50
6.15	Mevislab Netzwerk zur Berechnung des aktuellen Rotationsvektors und zur Generierung der Liste an geplanten Schraubenpositionen im XMarkerListContainer Modul, sowie das Panel zur Parametrierung des XMarkerListContainer Moduls	52
6.16	Stereokamera NDI Polaris VEGA und dessen Messvolumen mit der Lage des Koordinatenursprungs, Quelle: [76]	55
6.17	Benutzeroberfläche des Mevislab Moduls „ <i>NDIConnection</i> “ zur Parametrierung der Stereokamera (a) und zur Ausgabe der empfangenen Positionsdaten der Referenzkörper (b)	56
6.18	Referenzkörper am Roboter, sowie das Modell eines Werkzeugs zum Eindrehen der Schrauben mit den Koordinatensystemen an dem Flansch des Roboters	57
6.19	Mevislab Panel des aufgebauten HNI-Moduls zur Ansteuerung des Roboters [2]	59

6.20	Die reale Szene (links) und die Visualisierung des am Patienten angebrachten Referenzkörpers {ARB} (orange), sowie eines chirurgischen Zeigeelements {PRB} (grün) und der aktuellen Robotersystem Position {TTP} (rot) im 3D-Bilddatensatz (rechts)	61
6.21	Benutzeroberfläche zum Laden von Bildaufnahmen	62
6.22	Benutzeroberfläche zur Einstellung der Visualisierung des 3D-Bilddatensatzes	64
6.23	Benutzeroberfläche mit der Darstellung des Datensatzes zur Planung von Pedikelschrauben	65
6.24	Benutzeroberfläche zur Anbindung an das Robotersystem und zur Ansteuerung des neu angebrachten Navigationssystems	66
6.25	Zur Registrierung verwendeter Kalibrierkörper mit röntgenintransparenten Markerkugeln (blau) sowie einem Referenzkörper für das optische 3D-Lokalisierungssystem {KRB}	68
6.26	Setup zur Erfassung des verwendeten Kalibrierkörpers im Bildvolumen der Bildgebungsmodalität	69
6.27	Programmablauf zur Detektion der röntgenintransparenten Kugelmärkerzentren	70
6.28	Visualisierung der 3D Aufnahme des Kalibrierkörpers mit heller Darstellung hoher Intensitätswerte(a), sowie das Ergebnis des ersten Schrittes: Anwendung des Schwellwertfilters (b)	71
6.29	Mevislab Programm zur Detektion der röntgenintransparenten Markerpunkte in der 3D-Aufnahme des Kalibrierkörpers	72
6.30	Die im 2. Schritt detektierten Markerbereiche im Bilddatensatz der Kalibrierkörperaufnahme (a), sowie die im 3. Schritt bestimmten Märkerzentren die als grüne Kugeln dargestellt sind (b)	73
6.31	Einsatz des erstellten „ <i>PCL_matching</i> “ Moduls zur Registrierung der erkannten Märkerpunkte mit den aus der Geometrie bekannten Märkerpunkten	75
6.32	Kalibrierkörperaufnahme mit den detektierten Märkerpunkten (grün) und die aus der Geometrie bekannten Märkerpunkten (rot), die nach der erfolgreichen Registrierung überlagert sind	77
6.33	Benutzeroberfläche zur Ansteuerung des Navigationssystems und zum Erfassen der Referenzkörperpositionen des Kalibrierkörpers {KRB} und der Bildgebungsmodalität {CRB} sowie zur Aktivierung des DICOM Receiver Moduls zum Empfangen von DICOM Bilddatensätzen von der Bildgebungsmodalität	79

6.34	Benutzeroberfläche zum Laden und Visualisieren des DICOM Bild- datensatzes der Kalibrierkörperaufnahme, sowie zum Laden der aus der Geometrie bekannten Röntgenmarkerpositionen	80
6.35	Benutzeroberfläche zur Detektion der Kugelmarkerzentren in der Ka- librierkörperaufnahme. Die detektierten Kugelbereiche sind in rot vi- sualisiert und die Kugelmarkerzentren als grüne Kugelobjekte.	81
6.36	Benutzeroberfläche zur Ausführung der Punktregistrierung und zum Abspeichern der Ergebnis Transformation. Die aus der Geometrie be- kannten Markerpositionen (lila) werden mit den detektierten Marker- positionen (grün) überlagert.	81
6.37	Intraoperative Bildgebungsmodalität mit dem angebrachten Refer- enzkörper {CRB} und der Patient mit dem Patientenreferenzkörper {ARB} im Bildvolumen der Bildgebungsmodalität {CBVol} sowie das optische Navigationssystem	82
6.38	Das verwendete Wirbelmodell (oben), visualisierte 3D-Darstellung ei- ner CT-Aufnahme (unten links) und die visualisierte 3D-Darstellung einer C-Bogen Aufnahme (unten rechts)	84
6.39	3D-Darstellung der segmentierten Wirbelkörper der CT Aufnahme (oben links), sowie einer Querschnittansicht (oben rechts) und die 3D-Darstellung der segmentierten Wirbelkörper einer C-Bogen Auf- nahme (unten links) sowie einer Querschnittansicht (unten rechts) [93]	86
6.40	Die ausgewählten Wirbelkörperbereiche in der C-Bogen Aufnahme (links) und der CT Aufnahme (rechts). Die einzelnen Wirbelkörper sind farblich hervorgehoben. [93]	87
6.41	Parametrierung des „MERIT“ Moduls zur Registrierung von zwei 3D-Bilddatensätzen [93]	88
6.42	Ergebnis der Überlagerung einer CT Aufnahme eines Wirbelmodells (rot) mit einer C-Bogen Aufnahme (grün), erstellt mit dem „MERIT“ Modul [93]	90
6.43	„3D3DMatching“ Makro Modul und das Parameterpanel mit den übergebenen Visualisierungseinstellungen	91
6.44	3D-Ansicht und Ebenen Darstellung des Matchingergebnisses nach einer Feinjustierung über die ergänzten Schaltflächen	93
6.45	Benutzeroberfläche zum Start des 3D-3D Matchings und zur Visuali- sierung des Ergebnisses durch die Übertragung der im präoperativen Datensatz geplanten Schraubenpositionen (links) in die intraoperati- ve C-Bogen Aufnahme (rechts)	94

6.46	Erstelltes Modell mit zwei Bohrungen und das darin platzierte Zeigeinstrument, das vom Lokalisiersystem lokalisiert wird	96
6.47	3D-Datensatz des Modells mit einer geplanten Schraubenposition als Zielposition (gelb) und dem visualisierten Zeigeinstrument (grün) . .	97
6.48	Verwendetes Wirbelmodell und das Robotersystem, platziert über dem Metallstift, der als Zielposition geplant wurde	100
6.49	Planung einer Schraubenposition (gelb) im intraoperativen 3D-Bilddatensatz des Wirbelmodells mit der eingeblendeten Spitze des Werkzeugs am Roboter (rot) der an der Zielposition positioniert wurde . .	101
6.50	Planung einer Schraubenposition (gelb) im präoperativen CT Bilddatensatz des Wirbelmodells	103
6.51	Planung einer Schraubenposition (gelb) im präoperativen 3D-Bilddatensatz des Wirbelmodells mit der eingeblendeten Spitze des Werkzeugs am Roboter (rot)	104
6.52	Nach dem Matching mit der präoperativen Planung übertragene Schraubenposition (blau) im intraoperativen C-Bogen Bilddatensatz des Wirbelmodells mit der eingeblendeten Spitze des Werkzeugs am Roboter (rot)	105
7.1	Erste Version des STIFF-FLOP Armes, bestehend aus 2 Arm Segmenten mit einer gewobenen äusseren Hülle. Quelle: Projekt Newsletter	107
7.2	Aufbau eines STIFF-FLOP Manipulator Moduls. Die Abbildung (a) visualisiert den schematischen Querschnitt des Moduls. Die Abbildung (b) zeigt die schematische Darstellung des Moduls in der Ruheposition, bei Druckerhöhung in einer Luftkammer und bei einer Druckerhöhung in allen Luftkammern. Die Abbildung (c) zeigt ein STIFF-FLOP Modul und Abbildung (d) zeigt die Biegung eines Moduls. Quelle:[3]	108
7.3	STIFF-FLOP Arm bestehend aus 2 Modulen, montiert an einem Schunk Roboterarm über einem Phantommodell (links). Optische Referenzkörper an der STIFF-FLOP Arm Basis und an der eingesetzten Endoskopkamera (unten rechts), werden von der Stereokamera (oben rechts) überwacht.	110
7.4	Softwarestruktur der implementierten ROS Module zur Positionsbestimmung des STIFF-FLOP Armes [64]	112
7.5	Modell des Versteifungselements für die Endoskopkamera mit Haltern für Markerkugeln (links) und das gefertigte Element an der Endoskopkamera (rechts) mit dem festgelegten Koordinatensystem, Quelle: [64]	114

7.6	STIFF-FLOP Arm an dem Basiselement, das mit dem Schunk Roboter verbunden wird. An der Basis sind Markerkugeln für eine Detektion mit der Stereokamera angebracht. [43]	115
7.7	Schritte der Bildvorverarbeitung im Programmablauf der <i>Intensity</i> ROS-Node	116
7.8	Kamerakalibrierung mit einem Schachbrettmuster in ROS [64]	117
7.9	Endoskopische Aufnahme des weiterentwickelten STIFF-FLOP Armes im Phantommodell	119
7.10	Programmablauf zur Erstellung der R-Tabelle und der anschließenden Online Detektion	120
7.11	Kontur des STIFF-FLOP Armes aus der Abbildung 7.1 [29]	121
7.12	Beispielhafte Darstellung der Erstellung des Akkumulator Feldes (rechts) zur Bestimmung des Objektschwerpunktes in dem untersuchten Videobild (links) [29]	121
7.13	Mit dem SIFT Algorithmus detektierte Merkmalspunkte des STIFF-FLOP Armes aus der Abbildung 7.1, hervorgehoben durch rote Kreise [29]	123
7.14	Visualisierung verschiedener Musterpunkte entsprechend ihrer Charakterisierung, wobei rote Quadrate die erste Klasse und blaue Kreise die zweite Klasse repräsentieren [44]	125
7.15	Visualisierung der Abstandsberechnung zwischen den Supportvektoren auf der unteren Grenze \vec{x}_- und der oberen Grenze \vec{x}_+ [44]	126
7.16	Programmablauf zum Training der SVM-Methode	129
7.17	Segmentierung des STIFF-FLOP Armes mit der Grabcut Methode: (a) originales Videobild, in dem der Bereich mit dem STIFF-FLOP Arm ausgewählt wird (grünes Quadrat), (b) automatisch segmentierte Teile des Originalbildes, (c) Benutzereingaben zur Markierung von Bereichen die zum STIFF-FLOP Arm gehören (rot), sowie Bereiche die zum Hintergrund gehören (blau), (d) Ergebnis der Segmentierung [61]	130
7.18	Aufteilung eines Videobildes in Teilstücke zur Erstellung der Trainingsmatrix und der Labelmatrix	131
7.19	Erkannte Teilstücke des STIFF-FLOP Armes, gekennzeichnet mit grünem Rahmen, nach Ausführung der Detektionsfunktion	134

7.20	Detektionsszenario 1 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]	136
7.21	Detektionsszenario 2 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]	137
7.22	Detektionsszenario 3 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]	138
7.23	Detektionsszenario 4 zeigt den STIFF-FLOP Arm in einem Phantommodell des Abdomens: Die mit der polynomialen Kernelfunktion detektierten Teilstücke sind grün umrandet(a), das gefilterte Ergebnis von (a) ist in der Bildmaske (b) dargestellt. Die unter Verwendung der radialen Basisfunktion detektierten Teilstücke sind im Bild (c) grün umrahmt, das gefilterte Ergebnis von (c) ist in der Bildmaske (d) abgebildet. [61]	139
7.24	Programmablauf der „predict_svm“ Funktion zur Online Erkennung des STIFF-FLOP Armes mit der SVM Methode	140
7.25	Programmablauf der „Tubedetect“ Ros-Node zur Detektion des STIFF-FLOP Armes im Videobild und zur Bestimmung der Position im Videobild [64]	141
7.26	Mit der SVM Methode detektierte Bildmaske (oben rechts) der Videoszene (unten), in welcher die Mittellinie (rot) und die Höhe des STIFF-FLOP Armes (cyan) dargestellt sind	142

7.27	Schritte zur Detektion des Ringmarkers am Verbindungselement eines STIFF-FLOP Moduls (a), segmentierter Ringmarker(b), Kontur des segmentierten Ringmarkers mit einer Aufteilung in einen linken (blau) und einen rechten (rot) Randbereich (d), Bestimmung der passenden Ellipse (e), Originalbild mit den überlagerten Ellipsen (f) [43]	145
A.1	Berechnung und Speicherung der Schraubenposition in dem XMarkerListContainer Modul, sowie Berechnung und Speicherung einer gematchten Schraubenposition in dem XMarkerListContainer2 Modul. Daraufhin werden die Schraubenmodelle im addScrews Modul geladen und visualisiert.	i
A.2	Laden und Darstellen des Patienten Referenzkörpers (ARB), der aktuellen Roboter Werkzeugspitzenposition (TTP), der aktuellen Zeiginstrumentposition (PRB), sowie einer gegebenenfalls vorab gespeicherten Zeiginstrumentposition (saved PRB)	ii
A.3	Aufbau des Makro Moduls mit den Schaltflächen zur Feinjustierung der Schraubenposition für die koronale Schichtansicht, die sagittale Schichtansicht und die axiale Schichtansicht	iii
A.4	Aufbau des Makro Moduls „ScrewSeparator“ zum Laden und visualisieren einer Schraubenposition, mit den in der XMarkerList gespeicherten Planungsdaten	iv
A.5	Durch das Python Script automatisch erstelltes Netzwerk zur Visualisierung von 4 geplanten Schraubenmodellen	iv
C.1	Schritt1: Visualisierung und Einfärbung des präoperativen Datensatzes in rot	vii
C.2	Schritt2: Visualisierung und Einfärbung des intraoperativen Datensatzes in grün	viii
C.3	Schritt3: Vorpositionierung der Datensätze zur Korrektur der Ausrichtung	viii
C.4	Schritt4: Zuschnitt des Intraoperativen Datensatzes zur Eingrenzung der relevanten Bildbereiche	ix
C.5	Schritt5: Zuschnitt des preoperativen Datensatzes zur Eingrenzung der relevanten Bildbereiche	ix
C.6	Schritt6: Auswahl des für das Matching verwendeten Wirbelkörpers in beiden Bilddatensätzen	x
C.7	Schritt7: Automatische Registrierung mit dem MERIT Modul	x
C.8	3D-Visualisierung des Matching Ergebnisses	xi

Tabellenverzeichnis

6.1	Die zur Registrierung verwendeten Markerpunkte ${}^{CBV^{ol}}\vec{Markerpos}(i)$ und ${}^{CRB}\vec{Markerpos}(i)$	76
6.2	Resultierende Abweichung $\epsilon(\vec{i})$ der einzelnen Markerpunktepaare nach der erfolgten Registrierung	78
6.3	Translations- und Rotationsabweichung zwischen der geplanten Schraubenposition und der gemessenen Zeigeinstrumentposition	98
6.4	Positionsabweichung zwischen der in der C-Bogen Aufnahme geplanten Schraubenposition und der gemessenen Werkzeugspitzenposition .	102
6.5	Translationsabweichung zwischen der in der präoperativen Aufnahme geplanten Schraubenposition und der gemessenen Werkzeugspitzenposition	105
7.1	Format der R-Tabelle der generalisierten Hough Transformation zur Beschreibung eines Modells des gesuchten Objektes	120
7.2	Versuchsreihe zur Analyse des berücksichtigten Vordergrundanteils für das Training der SVM-Methode am Beispiel der Abbildung 7.23(b)	135

A Mevislab Funktionen der Planungssoftware

In diesem Abschnitt werden Mevislab Funktionen der Planungssoftware abgebildet, auf die im Text verwiesen wurde.

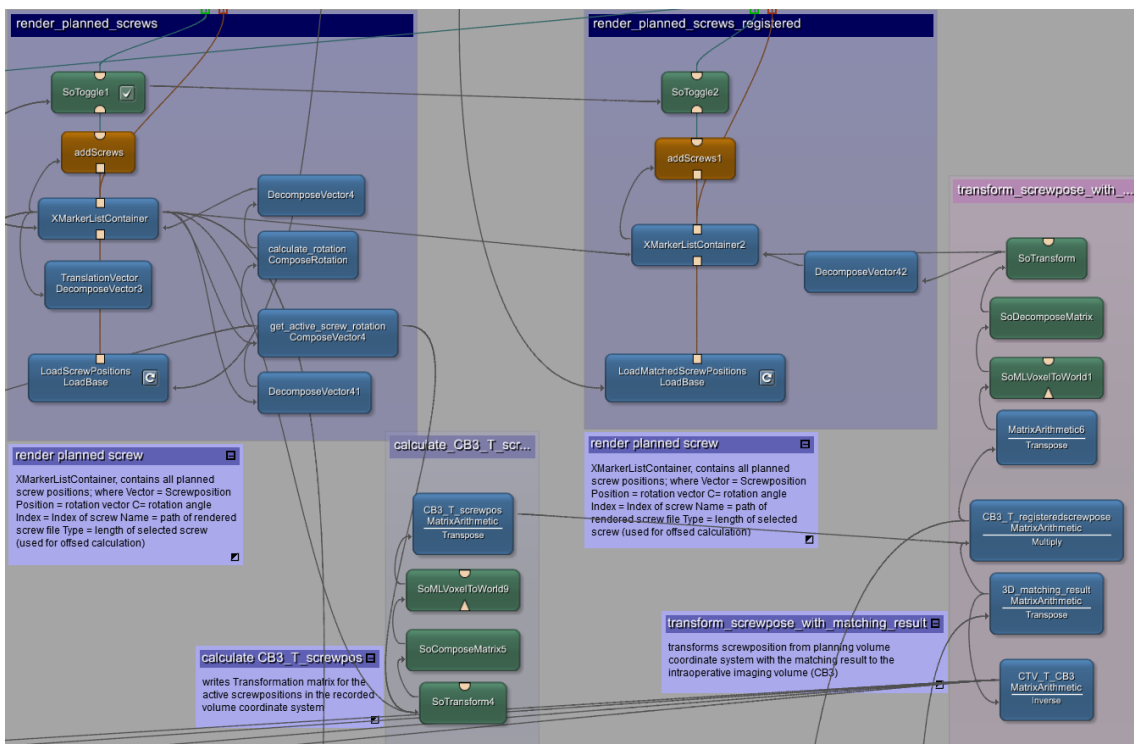


Abbildung A.1: Berechnung und Speicherung der Schraubenposition in dem XMarkerListContainer Modul, sowie Berechnung und Speicherung einer gematchten Schraubenposition in dem XMarkerListContainer2 Modul. Daraufhin werden die Schraubenmodelle im addScrews Modul geladen und visualisiert.

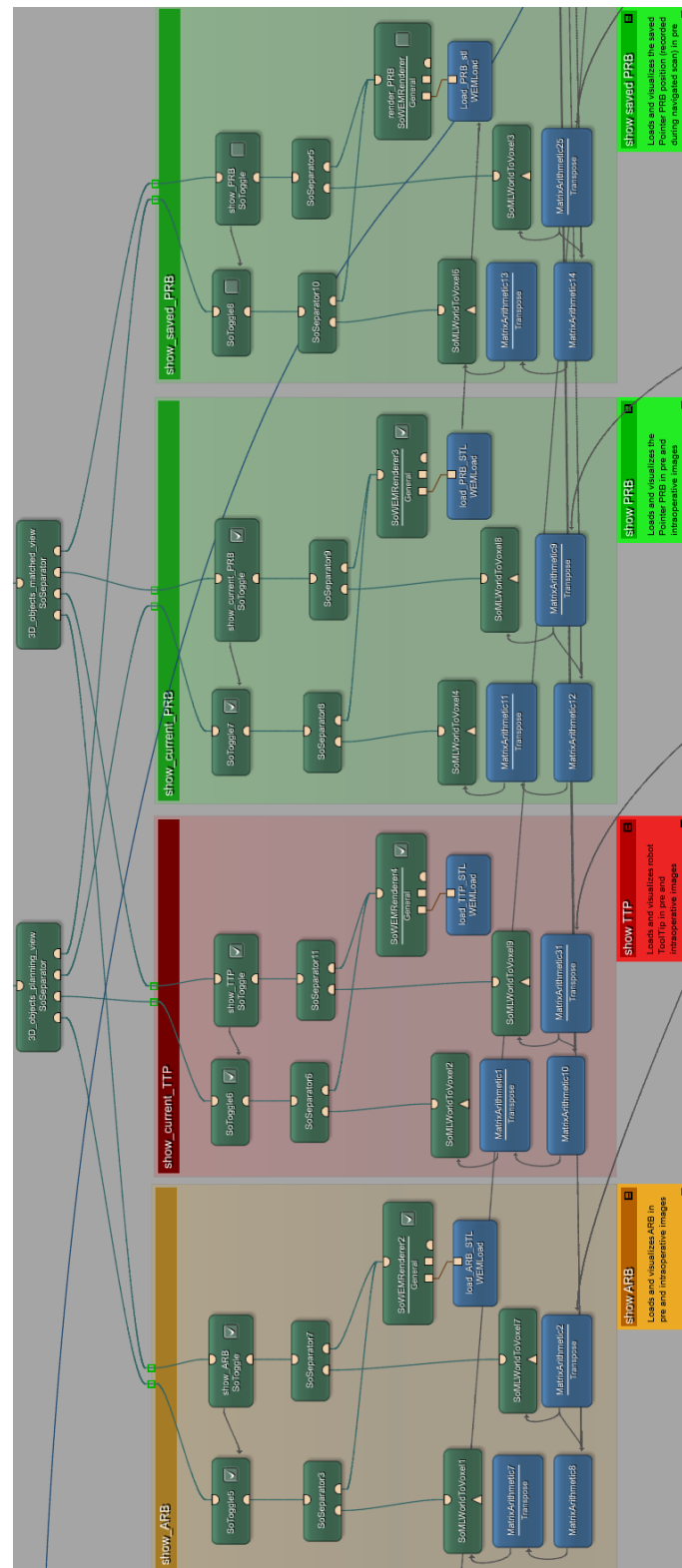


Abbildung A.2: Laden und Darstellen des Patienten Referenzkörpers (ARB), der aktuellen Roboter Werkzeugspitzenposition (TTP), der aktuellen Zeigeinstrumentposition (PRB), sowie einer gegebenenfalls vorab gespeicherten Zeigeinstrumentposition (saved PRB)

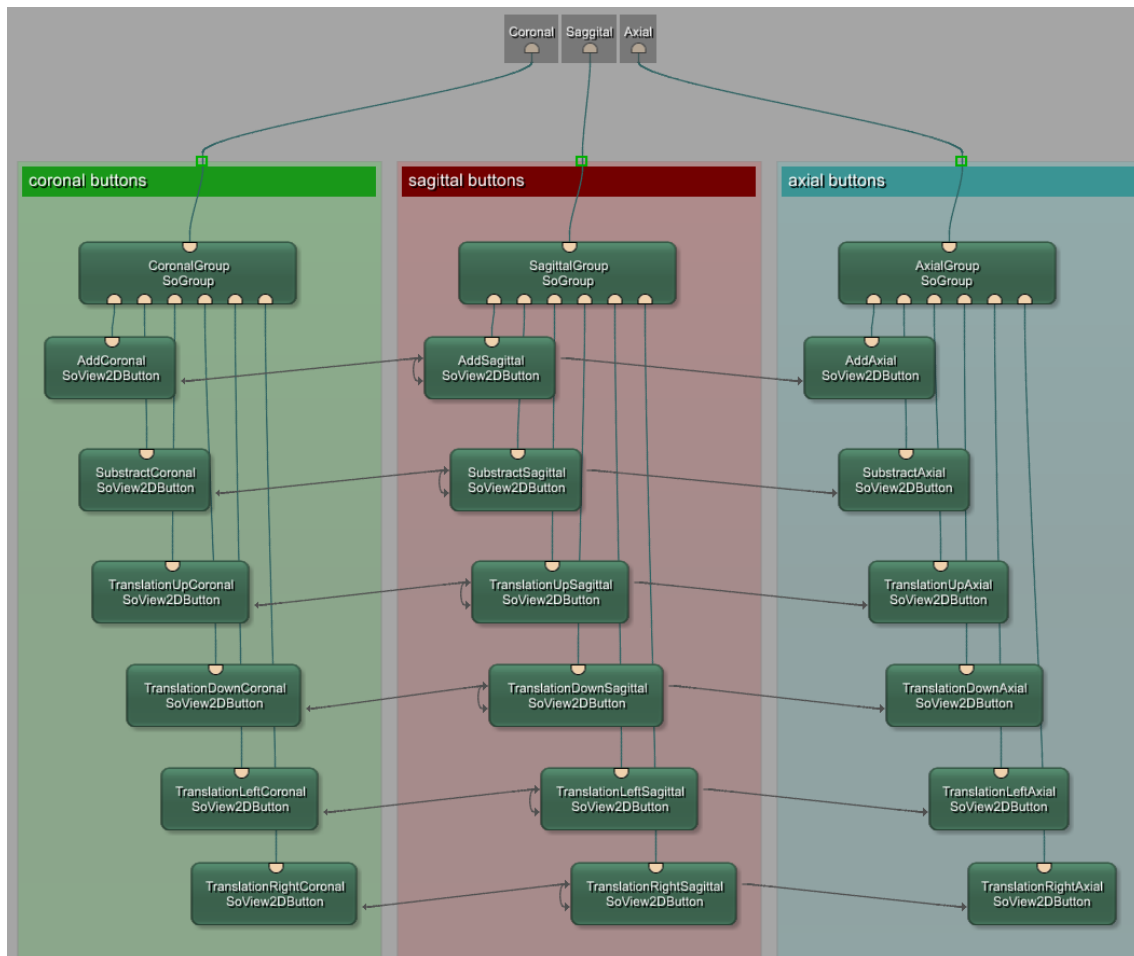


Abbildung A.3: Aufbau des Makro Moduls mit den Schaltflächen zur Feinjustierung der Schraubenposition für die koronale Schichtansicht, die sagittale Schichtansicht und die axiale Schichtansicht

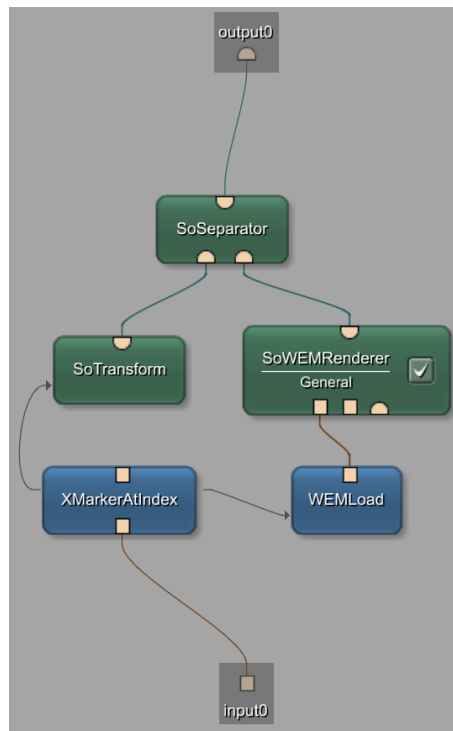


Abbildung A.4: Aufbau des Makro Moduls „ScrewSeparator“ zum Laden und visualisieren einer Schraubenposition, mit den in der XMarkerList gespeicherten Planungsdaten

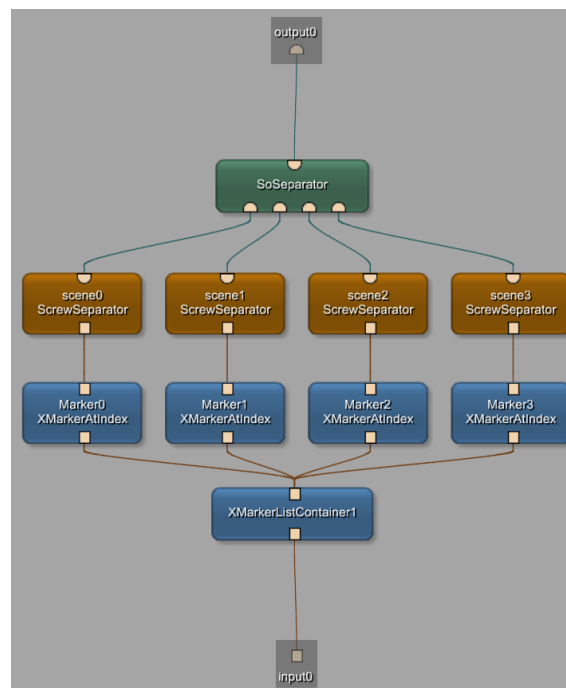


Abbildung A.5: Durch das Python Script automatisch erstelltes Netzwerk zur Visualisierung von 4 geplanten Schraubenmodellen

B Mevislab Modul zur Registrierung von Punktdatensätzen

In diesem Kapitel wird der Quellcode des erstellten Mevislab Moduls „PCL_matching“ abgebildet. Das Modul verwendet die Methode der Einheitsquaternionen [49] zur Bestimmung der Transformation zwischen 2 Punktdatensätzen. Die Methode wurde in einer vorhergehenden Arbeit in der eingebundenen „q_solver“ Klasse implementiert [55].

```
1 #include "mlPCL_matching.h"
2 #include <iostream>
3 #include <cmath>
4 #include "solver-q.h"
5
6 ML_START_NAMESPACE
7
8 //! Implements code for the runtime type system of the ML
9 ML_MODULE_CLASS_SOURCE(PCL_matching, Module);
10 //-----
11 PCL_matching::PCL_matching() : Module(0, 0)
12 {
13     // Suppress calls of handleNotification on field changes to
14     // avoid side effects during initialization phase.
15     handleNotificationOff();
16     // Add fields to the module and set their values.
17     //input fields:
18     _source_pointsFld = addBase("source_points", NULL);
19     _source_pointsFld->addAllowedType<XMarkerList>();
20     _target_pointsFld = addBase("target_points", NULL);
21     _target_pointsFld->addAllowedType<XMarkerList>();
22     _start_computationFld = addTrigger("start_computation");
23     //output fields
24     _source_T_targetFld = addMatrix4("source_T_target", Matrix4());
25     _source_T_target_rotationFld = addMatrix3("source_T_target_rotation");
26     _source_T_target_translationFld = addVector3("source_T_target_translation");
27     // Reactivate calls of handleNotification on field changes.
28     handleNotificationOn();
29 }
30 //-----
31 void PCL_matching::handleNotification(Field* field)
32 {
33     // run method if input values change
34     if (field == _start_computationFld)
35         run_matching();
36 }
37
38 void PCL_matching::activateAttachments()
39 {
40     // Update members to new field state here.
```

```
41 // Call super class functionality to enable notification handling again.
42 Module::activateAttachments();
43 }
44
45 void PCL_matching::run_matching()
46 {
47     // Load input fields
48     XMarkerList *_source_baseValue = (XMarkerList *)_source_pointsFld->getValue();
49     XMarkerList *_target_baseValue = (XMarkerList *)_target_pointsFld->getValue();
50     std::vector<Eigen::Vector3d> ref; //reference point list
51     std::vector<Eigen::Vector3d> mes; //measured point list
52
53     if (_source_baseValue && BASE_IS_A(_source_baseValue, XMarkerList)) {
54         for (int i = 0; i<int(_source_baseValue->getSize()); ++i) {
55             // Get marker position and transform it
56             XMarker source_marker = _source_baseValue->at(i);
57             ref.push_back(Eigen::Vector3d(source_marker.x(), source_marker.y(), source_marker.z()));
58             XMarker target_marker = _target_baseValue->at(i);
59             mes.push_back(Eigen::Vector3d(target_marker.x(), target_marker.y(), target_marker.z()));
60         }
61     }
62     //run unit quaternion solver
63     PFSolverQ solver_q(ref, mes);
64     solver_q.Solve();
65
66     //get rotation matrix result and write it in transformation matrix
67     Matrix3 result_rotation;
68     Matrix4 result_transformation;
69     for (int i = 0; i < 3; i++) {
70         for (int j = 0; j < 3; j++) {
71             result_rotation[j][i] = result_transformation[j][i] = solver_q.GetRotation()((i * 3 + j));
72         }
73     }
74     //add translationvector to transformation matrix
75     result_transformation[0][3] = solver_q.GetTranslation()[0];
76     result_transformation[1][3] = solver_q.GetTranslation()[1];
77     result_transformation[2][3] = solver_q.GetTranslation()[2];
78     result_transformation[3][3] = 1;
79
80     //update output field with rotation matrix
81     _source_T_target_rotationFld->updateMatrix3Value(result_rotation);
82     //update output field with translation vector
83     _source_T_target_translationFld->updateVector3Value({ solver_q.GetTranslation()[0], solver_q.
84         GetTranslation()[1], solver_q.GetTranslation()[2] });
85     //update output field with transformation matrix
86     _source_T_targetFld->updateMatrix4Value(result_transformation);
87 }
88 ML_END_NAMESPACE
```

Quellcode B.1: Mevislab Modul zur Registrierung von 2 Punktdatensätzen

C Mevislab Makro Modul zur Registrierung von multimodalen Bilddatensätzen

In diesem Abschnitt wird die Registrierung zweier 3D Bilddatensätze am Beispiel der im Abschnitt 6.4.2 dargestellten CT-Aufnahme eines Wirbelmodells mit der 3D C-Bogen Aufnahme des selben Modells aufgezeigt. Hierbei wurde das im Rahmen der betreuten studentischen Arbeit [93] erstellte und im Verlauf der Arbeit ergänzte Mevislab 3D-3DMatching Modul verwendet. Die Abbildungen zeigen die erstellte Benutzeroberfläche mit den einzelnen Schritten zur Durchführung des Matchings.

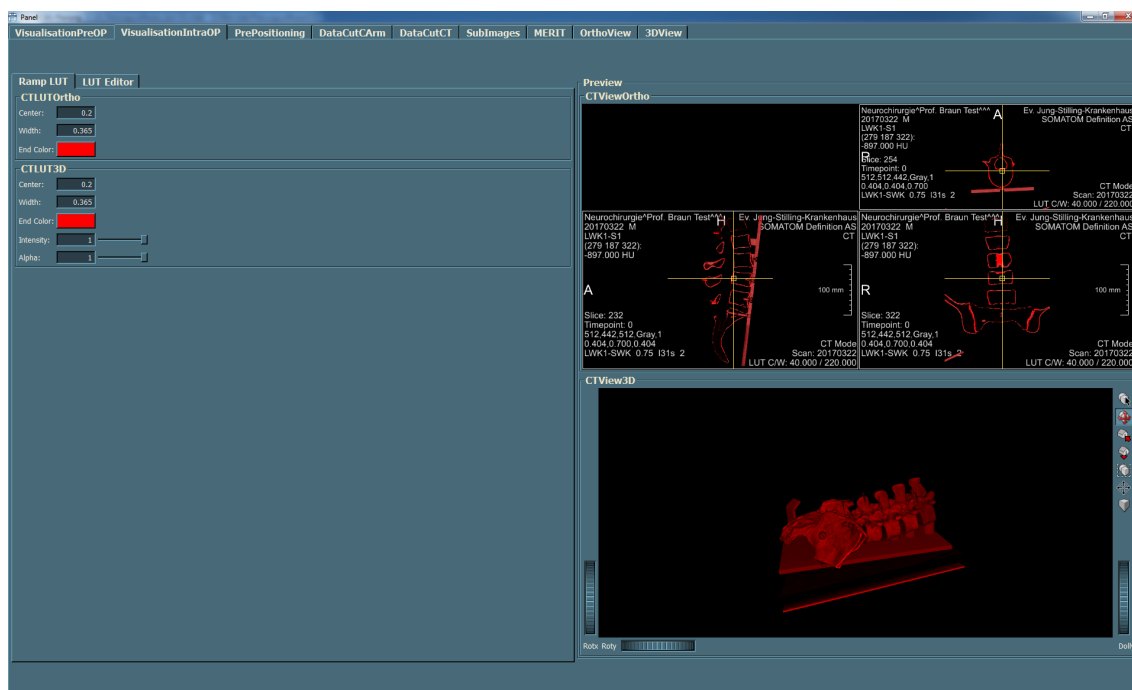


Abbildung C.1: Schritt1: Visualisierung und Einfärbung des präoperativen Datensatzes in rot

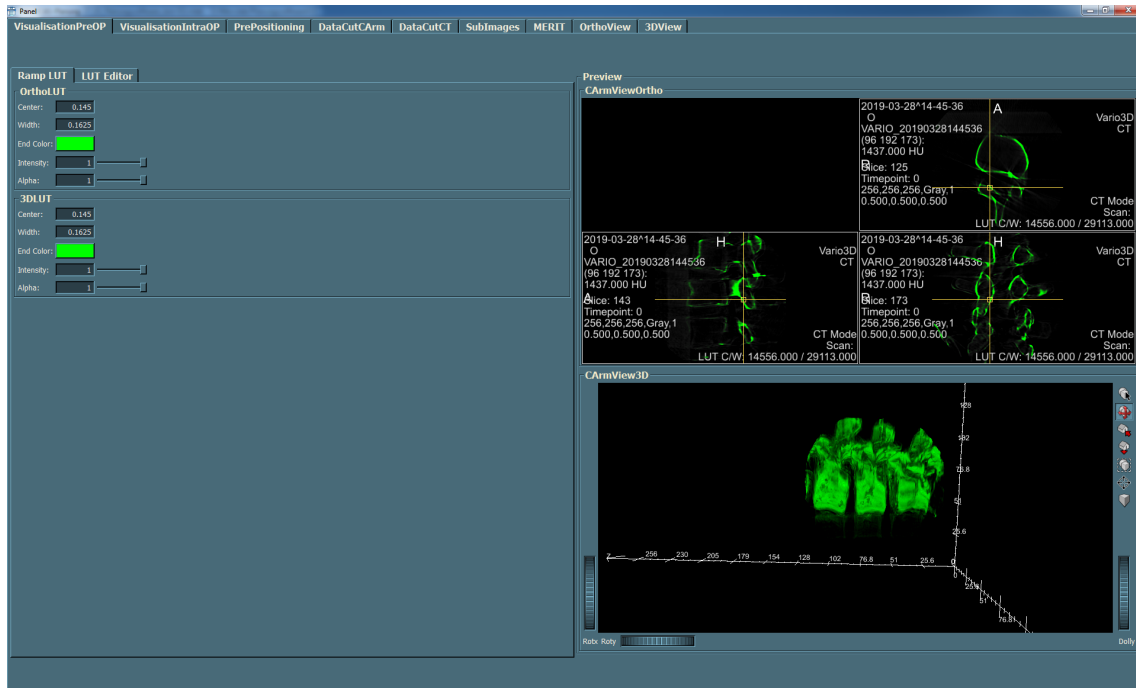


Abbildung C.2: Schritt2: Visualisierung und Einfärbung des intraoperativen Datensatzes in grün

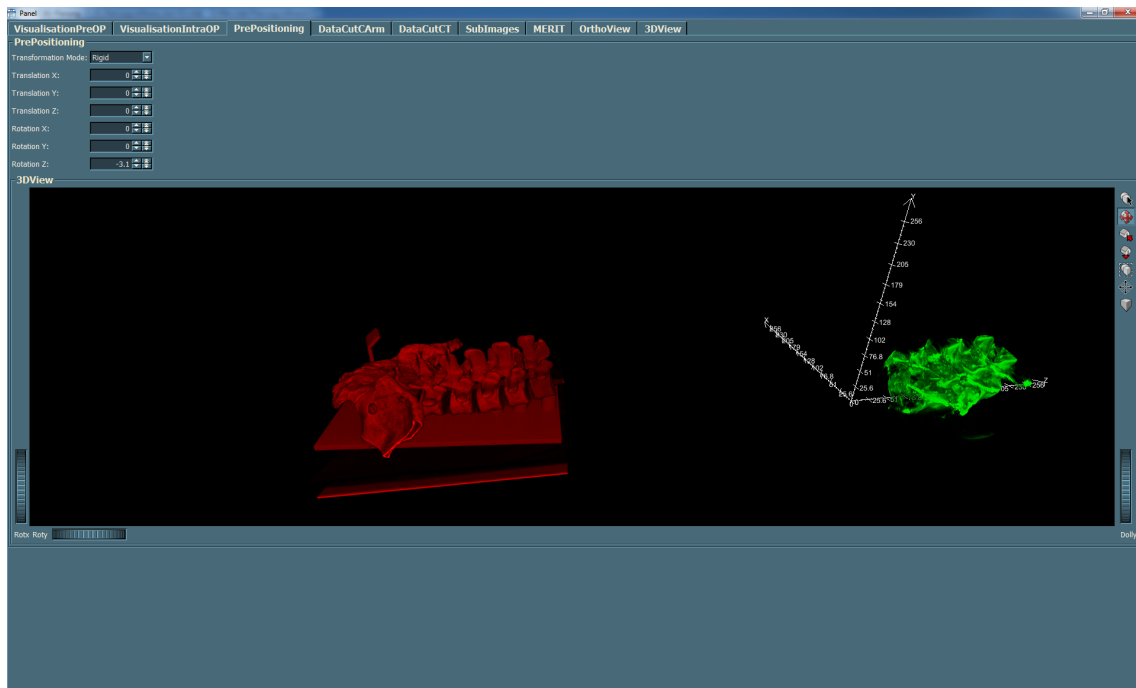


Abbildung C.3: Schritt3: Vorpositionierung der Datensätze zur Korrektur der Ausrichtung

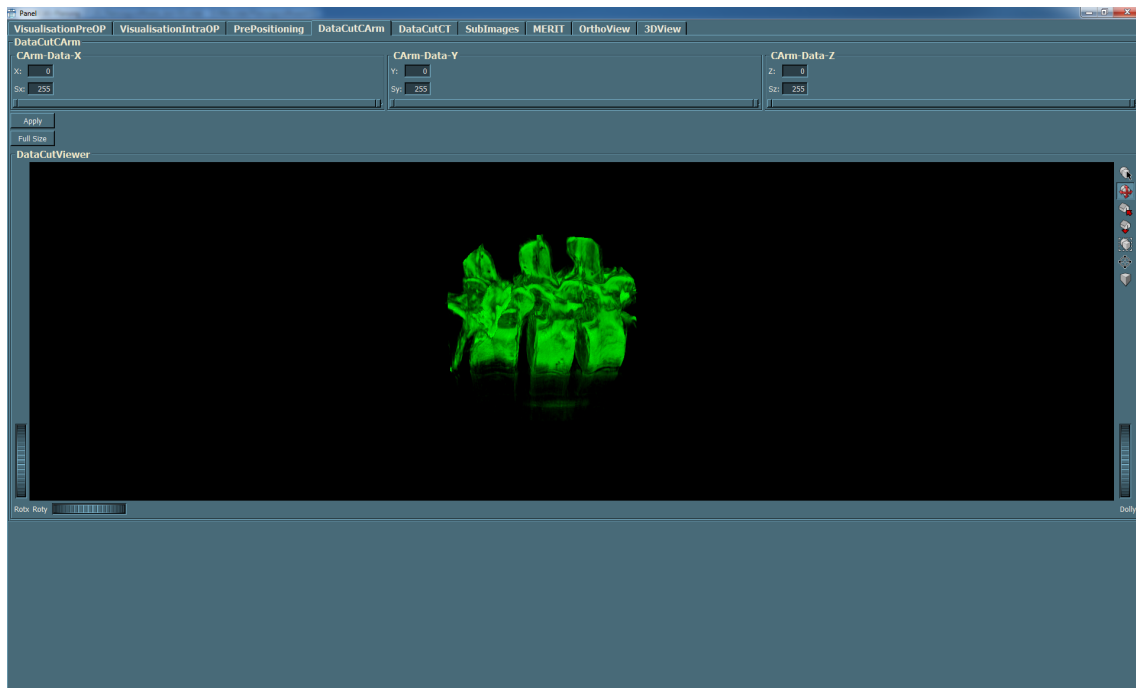


Abbildung C.4: Schritt4: Zuschnitt des Intraoperativen Datensatzes zur Eingrenzung der relevanten Bildbereiche

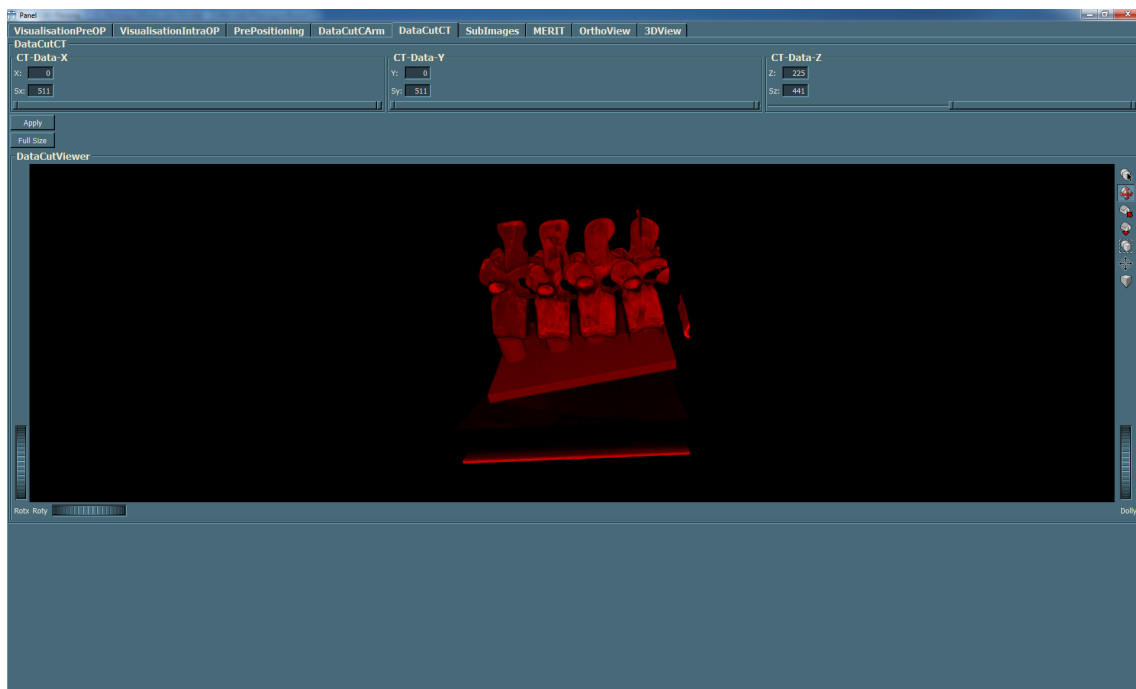


Abbildung C.5: Schritt5: Zuschnitt des preoperativen Datensatzes zur Eingrenzung der relevanten Bildbereiche

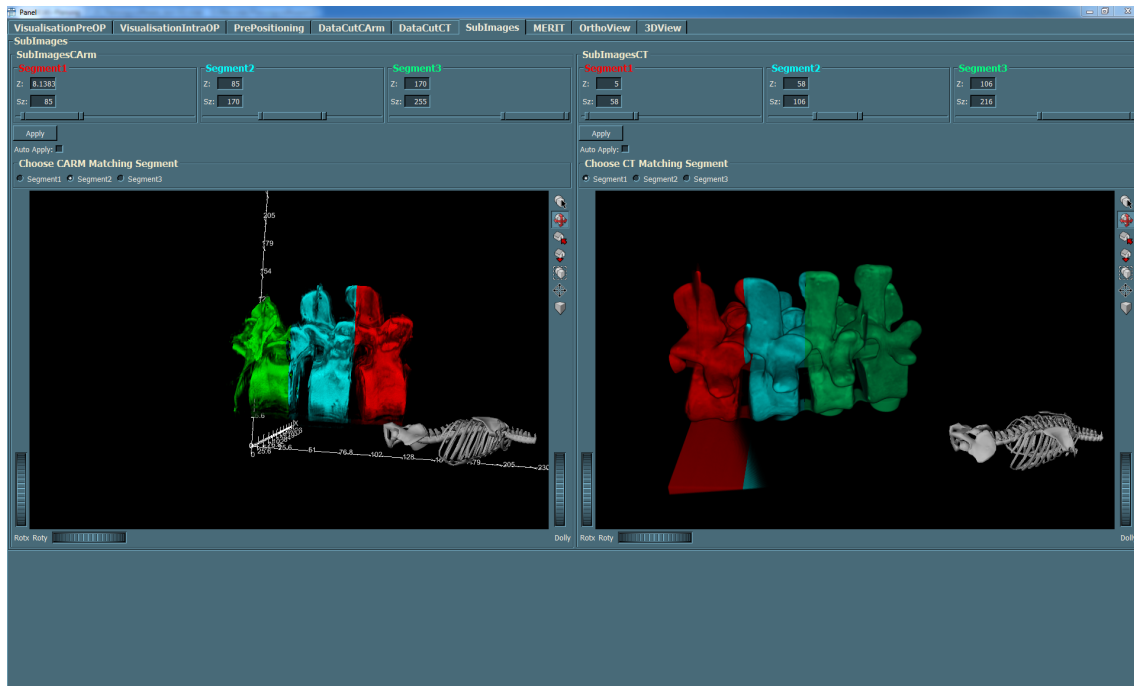


Abbildung C.6: Schritt6: Auswahl des für das Matching verwendeten Wirbelkörpers in beiden Bilddatensätzen

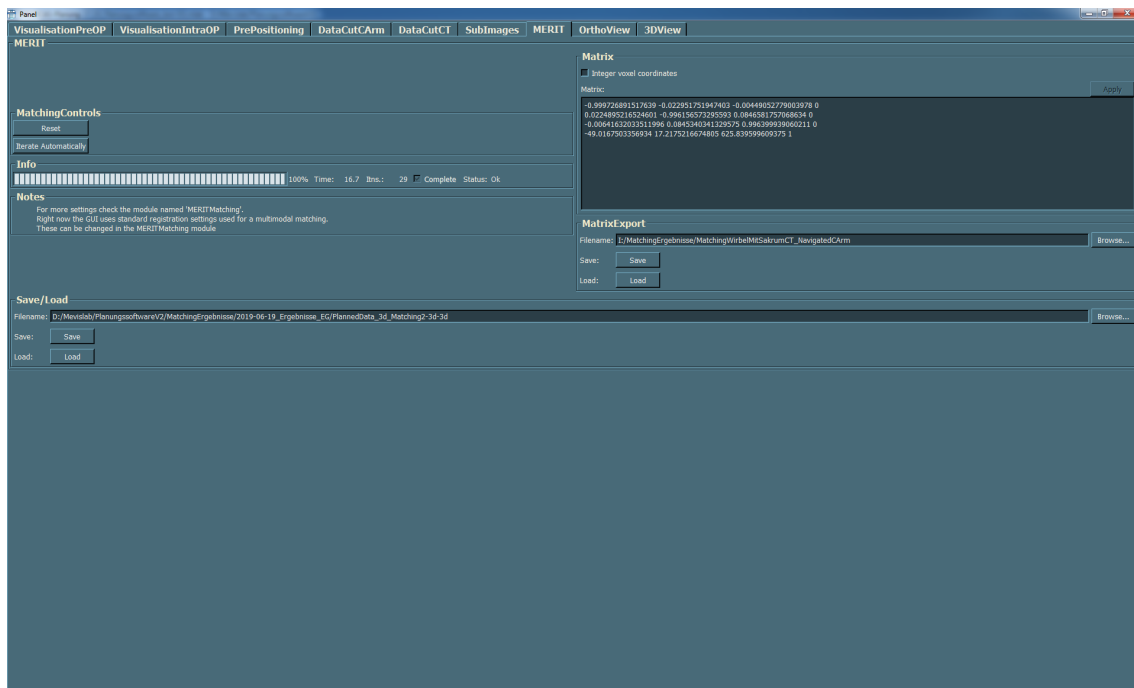


Abbildung C.7: Schritt7: Automatische Registrierung mit dem MERIT Modul

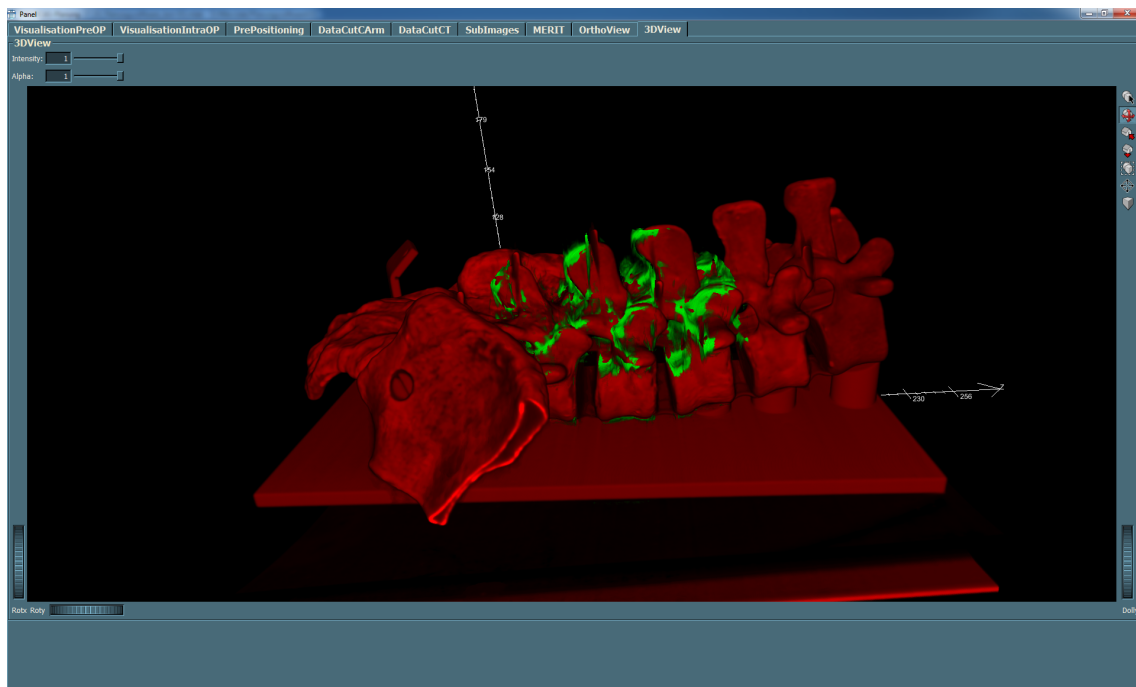


Abbildung C.8: 3D-Visualisierung des Matching Ergebnisses

D Funktionen zur Detektion des STIFF-FLOP Armes durch den Einsatz von Support Vector Machines

In diesem Abschnitt werden die implementierten Funktionen zur Detektion des STIFF-FLOP Armes durch den Einsatz von Support Vector Machines aufgezeigt. Die Funktionen wurden in Zusammenarbeit mit den studentischen Arbeiten [29] [61] entwickelt. Der Aufruf der Funktionen ist im Abschnitt 7.3.4 erläutert.

Die Funktion „breakdown_Img“ teilt ein übergebenes Trainingsbild in Teilstücke und legt diese im Zielverzeichnis ab.

```
1 void SupVecMac::breakdown_Img(std::string File_name, std::string dst_folder, int enum_offset, int
   percentage, int choice, int *fno)
2 {
3     cv::Mat image = cv::imread(File_name);
4     if( image.empty())
5     {
6         std::cout<<"Can't load image "<<std::endl;
7     }
8     else
9     {
10        std::vector<cv::Mat> Small_images;
11        Small_images.clear();
12        cv::Mat temporary_Mat;
13        cv::Rect temporary_Rect = cv::Rect(0,0,0,0);
14        temporary_Rect.height = Height;
15        temporary_Rect.width = Width;
16        for (int j=0; j< image.rows-Height; j+=Height)
17        {
18            temporary_Rect.y = j;
19            for (int i=0; i< image.cols-Width; i+=Width)
20            {
21                temporary_Rect.x = i;
22                temporary_Mat = image(cv::Rect(temporary_Rect.x,temporary_Rect.y,temporary_Rect.width,
                temporary_Rect.height));
23                Small_images.push_back(temporary_Mat);
24            }
25        }
26        //A temporary name is created to support moving the images from small images folder to tube
        images folder
27        std::string fg_name="Train_samples_Tube/Image ";
28        std::string small="Small_Images/Image ";
29        std::stringstream temporary_name_fg,small_image;
30
31        //The text file can be used to track the black pixels present in the corresponding breakdown
        folder
```

```
32     //std::ofstream myfile;
33     //myfile.open ("pixel_values.txt");
34
35     std::stringstream temporary_name;
36     dst_folder += "/Image ";
37     int y=Small_images.size();
38     *fno=0; //the pointer stores the no. of files in the folder
39
40     for (int x=enum_offset; x<y+enum_offset; ++x)
41     {
42         temporary_name<<dst_folder<<x+1<<".png";
43         std::string name = temporary_name.str();
44         int nonblack=0; //In this variable the pixels of the new small image will be stored in
45         for(int i=0;i<Small_images[x-enum_offset].cols;i++)
46         {
47             for(int j=0;j<Small_images[x-enum_offset].rows;j++)
48             {
49                 if(Small_images[x-enum_offset].at<cv::Vec3b>(j,i)[0]!=0&&Small_images[x-
50                     enum_offset].at<cv::Vec3b>(j,i)[1]!=0&&Small_images[x-enum_offset].at<cv::
51                         Vec3b>(j,i)[2]!=0)
52                     nonblack=nonblack+1;
53             }
54         }
55         //The temporary name created above is suffixed with the file name
56         temporary_name_fg.str("");
57         small_image.str("");
58         temporary_name_fg<<fg_name<<x+1<<".png";
59         std::string fg_name1 = temporary_name_fg.str();
60         small_image<<small<<x+1<<".png";
61         std::string small1 = small_image.str();
62         const char * fg = fg_name1.c_str();
63         const char * small = small1.c_str();
64
65         /*following conditional check was added to ensure that only the desired images are moved
66            to the corresponding folder */
67         int pixel_value=(percentage*Height*Width/100);
68         if(nonblack>=pixel_value)
69         {
70             if(choice==1)
71             {
72                 cv::imwrite(name, Small_images[x-enum_offset]);
73                 *fno= *fno+1;}
74             else if(choice==2)
75             {rename(small,fg);
76                 *fno= *fno+1;}
77         }
78     }
79     temporary_name.str("");
80 }
81 //myfile.close();
82 }
```

Quellcode D.1: Funktion zur Aufteilung eines Trainingsbildes in Teilstücke

In der Funktion „train_and_save_svm“ werden die Teilstücke des Hintergrundes und des Vordergrundes aus den zuvor erstellten Verzeichnissen geladen und in eine Trai-

ningsmatrix abgelegt. Zusätzlich wird eine Labelmatrix erzeugt, die das Label (Vordergrund = +1; Hintergrund = -1) für jedes Teilstück enthält. Nach der Auswahl des Kerneltyps (Zeile 67) und der zugehörigen SVM Parameter (Zeile 69-71) wird die SVM Methode trainiert (Zeile 74). Als Ergebnis des Trainings wird ein Dokument mit den ermittelten Stützvektoren exportiert, welches für die Online Detektion verwendet wird.

```
1 void SupVecMac::train_and_save_svm(int *num_class_1_img, int *num_class_2_img, std::string
   class_1_folder, std::string class_2_folder)
2 {
3     int num_images = *num_class_1_img+ *num_class_2_img; //Number of pictures to create training
   matrix
4     int img_area = Width*Height;
5     cv::Mat training_mat(num_images, img_area, CV_32FC1); //Initialize training matrix
6
7     cv::Mat temporary_mat; //Read image files
8     cv::vector<cv::Mat> Images;
9     Images.clear();
10    std::stringstream sname;
11    std::string name;
12    class_1_folder += "/Image ";
13    for (int x=1; x<=40000; x++) //Load foreground Images
14    {
15        sname<<class_1_folder<<x<<".png";
16        name = sname.str();
17        temporary_mat = cv::imread(name);
18
19        if (!temporary_mat.empty())
20        { Images.push_back(temporary_mat); }
21        sname.str("");
22    }
23    std::cout<<std::endl<<Images.size()<<" Number of foreground Images"<<std::endl;
24    int Class_1_images = Images.size();
25
26    class_2_folder += "/Image ";
27    for (int x=1; x<=40000; x++) //Load background Images
28    {
29        sname<<class_2_folder<<x<<".png";
30        name = sname.str();
31        temporary_mat = cv::imread(name);
32
33        if (!temporary_mat.empty())
34        { Images.push_back(temporary_mat); }
35        sname.str("");
36    }
37    std::cout<<std::endl<<Images.size()<<" Total number of images"<<std::endl;
38    int Total_images = Images.size();
39
40    //Fill training matrix
41    for (int file_num=0; file_num<Images.size(); file_num++)
42    {
43        int ii = 0; //Current column in training Mat
44        for (int i=0; i<Images[file_num].rows; i++)
45        {
```



```
46     for (int j=0; j<Images[file_num].cols; j++)
47     {
48         training_mat.at<float>(file_num, ii++) = Images[file_num].at<uchar>(i,j);
49     }
50 }
51 }
52
53 //CREATE LABELS FOR EACH ROW IN THE TRAINING MATRIX
54 cv::Mat labels(num_images,1,CV_32FC1);
55 for (int x=0; x<Class_1_images; x++) // Pictures here of Class +1
56 { labels.at<float>(x, 0) = 1;}
57 for (int x=Class_1_images; x<Total_images; x++) // Pictures here of Class -1
58 { labels.at<float>(x, 0) = -1;}
59
60 cv::imwrite("Training Matrix.png", training_mat);
61 cv::imwrite("Label Matrix.png", labels);
62
63 //Set SVM Parameters
64 CvSVMParams params;
65 params.svm_type = CvSVM::C_SVC;
66 //params.kernel_type = CvSVM::POLY;
67 params.kernel_type = CvSVM::RBF;
68 //params.term_crit = cvTermCriteria(CV_TERMCRIT_ITER+CV_TERMCRIT_EPS, 1000, 1e-6);
69 params.degree=2;
70 params.C=1;//
71 params.gamma = 0.0000002;
72 //Train CvSVM object
73 CvSVM svm;
74 svm.train(training_mat, labels, cv::Mat(), cv::Mat(), params);
75 //Save data
76 svm.save("SVM_Data_POLY"); // export training file
77 }
```

Quellcode D.2: Funktion zum Training der Support Vector Machine Methode

Mit der Funktion „predict_svm“ wird der STIFF-FLOP Arm in einem übergebenen Bild unter Verwendung des Trainingsergebnisses detektiert. Hierfür wird das Bild in Teilstücke unterteilt, die mit der SVM Methode klassifiziert werden. Zur Überprüfung des Ergebnisses werden die detektierten Teilstücke durch eine Umrahmung hervorgehoben und daraus eine Kontur des detektierten STIFF-FLOP Armes bestimmt.

```
1 void SupVecMac::predict_svm_masked_filtered(std::string Img_file_name)
2 {
3     cv::Mat Original=cv::imread(Img_file_name);
4     cv::Mat New_Original=Original(cv::Rect(0,0,Original.cols-1, Original.rows-1)); //****
5     cv::Mat export_contour = cv::Mat(Original.rows,Original.cols, CV_8U, cv::Scalar(0));
6     Original.copyTo(New_Original);
7     CvSVM svm;
8     svm.load("SVM_Data_POLY"); //Load training file
9     int Height= 25;
10    int Width = 25;
11    if( Original.empty())
12    {
13        std::cout<<"Can't load image "<<std::endl;
```

```
14     }
15     else
16     {
17         cv::Mat temporary_Mat;
18         cv::Rect temporary_Rect = cv::Rect(0,0,0,0);
19         temporary_Rect.height = Height;
20         temporary_Rect.width = Width;
21
22         cv::Mat Small_1D(1,Height*Width, CV_32FC1); //1D Matrix that works as an Input to "predict"
           function
23
24         for (int j=0; j< Original.rows-Height; j+=Height)
25         {
26             temporary_Rect.y = j;
27             for (int i=0; i< Original.cols-Width; i+=Width)
28             {
29                 temporary_Rect.x = i;
30                 temporary_Mat = Original(cv::Rect(temporary_Rect.x,temporary_Rect.y,temporary_Rect.width,
31                     temporary_Rect.height));
32
33                 int ii = 0;
34                 for (int i=0; i<temporary_Mat.rows; i++)
35                 {
36                     for (int j=0; j<temporary_Mat.cols; j++)
37                     {
38                         Small_1D.at<float>(0, ii++) = temporary_Mat.at<uchar>(i,j); //Convert small images in 1
39                             D matrices to predict
40                     }
41                 }
42                 int classe = svm.predict(Small_1D);
43                 if (classe == 1) //If classe belongs to a texture of the STIFF-FLOP Arm, draw a filled
44                     rectangle
45                 {
46                     cv::rectangle(New_Original, temporary_Rect, cv::Scalar(0,255,0),CV_FILLED);
47                 }
48             }
49         }
50
51         //contour filtering on masked image
52         cv::Mat Processed;
53         cv::inRange(New_Original, cv::Scalar(0,254,0),cv::Scalar(2,255,2), Processed);
54         std::vector<std::vector<cv::Point> > contours;
55         std::vector<cv::Vec4i> hierarchy;
56         cv::findContours( Processed, contours, hierarchy, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, cv::
57             Point(0, 0) );
58         std::vector<float> Areas;
59
60         for( int i = 0; i < contours.size(); i++ )
61         {
62             if (cv::contourArea(contours[i])>=1)
63             {
64                 Areas.push_back( cv::contourArea(contours[i]));
65             }
66         }
67
68         double biggestArea;
69         biggestArea=Areas[0];
70         int Biggest_Contour=0;
```

```
66
67 //find largest contour
68 if (Areas.size()>0)
69 {
70     for( int i = 0; i< Areas.size(); i++ )
71     {
72         if (Areas[i]> biggestArea)
73         {
74             biggestArea=Areas[i];
75             Biggest_Contour=i;
76         }
77     }
78 }
79 std::vector<cv::Vec4i> hierarchy_2;
80
81 cv::drawContours(export_contour, contours, Biggest_Contour, cv::Scalar(255), -3, 8, hierarchy_2,
82     1, cv::Point(0,0));
83 contours.clear();
84 }
85 imwrite("Result_filtered.png",export_contour); //export filtered result image
86 }
```

Quellcode D.3: Funktion zur Erkennung des STIFF-FLOP Armes mit der SVM Methode unter Verwendung des Trainingsergebnisses

In der Funktion „result_comparison“ wird der im Bild detektierte STIFF-FLOP Arm mit dem segmentierten Vordergrundbild des Trainingsbildes verglichen um das Ergebnis der Detektion zu untersuchen.

```
1 void SupVecMac::result_comparison(std::string fg_File, std::string result_file_unfiltered, std::
2     string result_file_filtered)
3 {
4     //read foreground image from grabcut
5     cv::Mat fg_grabcut;
6     fg_grabcut=cv::imread(fg_File,1);
7     //read masked image from result
8     cv::Mat mk_result_unfiltered;
9     mk_result_unfiltered=cv::imread(result_file_unfiltered,1);
10    cv::Mat mk_result_filtered;
11    mk_result_filtered=cv::imread(result_file_filtered,1);
12
13    //if images were loaded correctly
14    if(!fg_grabcut.empty()&&!mk_result_unfiltered.empty()&&!mk_result_filtered.empty())
15    {
16        int nonblack_fg=0; //count nonblack pixels in the foreground image
17        for(int i=0;i<fg_grabcut.cols;i++)
18        {
19            for(int j=0;j<fg_grabcut.rows;j++)
20            {
21                if(fg_grabcut.at<cv::Vec3b>(j,i)[0]!=0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]!=0&&
22                    fg_grabcut.at<cv::Vec3b>(j,i)[2]!=0)
23                    nonblack_fg=nonblack_fg+1;
24            }
25        }
26        results[6]=nonblack_fg;
27    }
28 }
```

```
25
26 //compare the individual pixels of unfiltered image with fg
27 int false_detection_uf=0;
28 int fg_detected_uf=0;
29 int fg_notdetected_uf=0;
30 for(int i=0;i<fg_grabcut.cols;i++)
31 {
32     for(int j=0;j<fg_grabcut.rows;j++)
33     {
34         if((fg_grabcut.at<cv::Vec3b>(j,i)[0]==0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]==0&&
35             fg_grabcut.at<cv::Vec3b>(j,i)[2]==0) &&
36             (mk_result_unfiltered.at<cv::Vec3b>(j,i)[0]!=0&&mk_result_unfiltered
37             .at<cv::Vec3b>(j,i)[1]!=0&&mk_result_unfiltered.at<cv::Vec3b>(j
38             ,i)[2]!=0))
39             {false_detection_uf=false_detection_uf+1;}
40         else if((fg_grabcut.at<cv::Vec3b>(j,i)[0]!=0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]!=0&&
41             fg_grabcut.at<cv::Vec3b>(j,i)[2]!=0)&&
42             (mk_result_unfiltered.at<cv::Vec3b>(j,i)[0]==0&&mk_result_unfiltered
43             .at<cv::Vec3b>(j,i)[1]==0&&mk_result_unfiltered.at<cv::Vec3b>(j
44             ,i)[2]==0))
45             {fg_notdetected_uf=fg_notdetected_uf+1;}
46         else if((fg_grabcut.at<cv::Vec3b>(j,i)[0]!=0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]!=0&&
47             fg_grabcut.at<cv::Vec3b>(j,i)[2]!=0)&&
48             (mk_result_unfiltered.at<cv::Vec3b>(j,i)[0]!=0&&mk_result_unfiltered
49             .at<cv::Vec3b>(j,i)[1]!=0&&mk_result_unfiltered.at<cv::Vec3b>(j
50             ,i)[2]!=0))
51             {fg_detected_uf=fg_detected_uf+1;}
52     }
53 }
54
55 //compare the individual pixels of filtered image with fg
56 int false_detection_f=0;
57 int fg_detected_f=0;
58 int fg_notdetected_f=0;
59 for(int i=0;i<fg_grabcut.cols;i++)
60 {
61     for(int j=0;j<fg_grabcut.rows;j++)
62     {
63         if((fg_grabcut.at<cv::Vec3b>(j,i)[0]==0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]==0&&
64             fg_grabcut.at<cv::Vec3b>(j,i)[2]==0) &&
65             (mk_result_filtered.at<cv::Vec3b>(j,i)[0]!=0&&mk_result_filtered.at<
66             cv::Vec3b>(j,i)[1]!=0&&mk_result_filtered.at<cv::Vec3b>(j,i)
67             [2]!=0))
68             {false_detection_f=false_detection_f+1;}
69         else if((fg_grabcut.at<cv::Vec3b>(j,i)[0]!=0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]!=0&&
70             fg_grabcut.at<cv::Vec3b>(j,i)[2]!=0)&&
71             (mk_result_filtered.at<cv::Vec3b>(j,i)[0]==0&&mk_result_filtered.at<
72             cv::Vec3b>(j,i)[1]==0&&mk_result_filtered.at<cv::Vec3b>(j,i)
73             [2]==0))
74             {fg_notdetected_f=fg_notdetected_f+1;}
75         else if((fg_grabcut.at<cv::Vec3b>(j,i)[0]!=0&&fg_grabcut.at<cv::Vec3b>(j,i)[1]!=0&&
76             fg_grabcut.at<cv::Vec3b>(j,i)[2]!=0)&&
77             (mk_result_filtered.at<cv::Vec3b>(j,i)[0]!=0&&mk_result_filtered.at<
78             cv::Vec3b>(j,i)[1]!=0&&mk_result_filtered.at<cv::Vec3b>(j,i)
79             [2]!=0))
80             {fg_detected_f=fg_detected_f+1;}
81     }
82 }
```

```
64     }
65
66     float accuracy=((fg_detected_f*100)/(nonblack_fg));
67     float false_percent;
68     false_percent=((false_detection_f*100)/(nonblack_fg));
69     std::cout<<accuracy<<false_percent<<std::endl;
70     results[0]=false_detection_uf;
71     results[1]=fg_notdetected_uf;
72     results[2]=fg_detected_uf;
73     results[3]=false_detection_f;
74     results[4]=fg_notdetected_f;
75     results[5]=fg_detected_f;
76     //write the values to file and close the file
77     if(false_percent<5)
78         {results[7]=0;}
79     else
80         {results[7]=1;}
81     }
82 }
```

Quellcode D.4: Funktion zum Vergleich des Detektierten Ergebnisses mit dem segmentierten Vordergrundbild des STIFF-FLOP Armes