

Entwicklung eines optischen
Lageüberwachungssystems für einen
neuartigen flexiblen Roboter mit steuerbarer
Steifigkeit zur Durchführung minimal-
invasiver chirurgischer Eingriffe

DISSERTATION

zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften

vorgelegt von

Dipl.-Ing. Matthias Mende

geb. am 06.11.1974 in Wickede an der Ruhr

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen

Siegen 2016

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier.

Haftungsausschluss

Die folgenden Ausführungen sind nach bestem Wissen erstellt. Sie dienen rein universitären Zwecken und sollen ein prinzipielles Verständnis für die technischen Zusammenhänge vermitteln. Garantien können nicht gegeben werden. Haftungsansprüche gegen die Autoren sind grundsätzlich ausgeschlossen.

Hinweis zur Verwendung von Namen, Marken und Warenzeichen

Alle innerhalb dieses Textes genannten und ggf. durch Dritte geschützten Namen, Marken- und Warenzeichen werden anerkannt und unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer, auch wenn nicht explizit auf das Schutzrecht hingewiesen wird. Allein aufgrund der bloßen Nennung oder Abbildung werden keine Rechte an solchen Namen, Marken- und Warenzeichen beansprucht. Diese werden lediglich unter Beachtung der Interessen der Eigentümer zitiert.

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

Zweiter Gutachter
Dr.-Ing. habil. Christian Fedrowitz
Koenig & Bauer AG, Würzburg

Tag der mündlichen Prüfung
19. Januar 2018

Vorwort

Die vorliegende Doktorarbeit entstand im Rahmen des EU FP7 Projektes „Stiff-Flop“ (STIFFness controllable Flexible and Learn-able Manipulator for surgical OPERations; Challenge 2: Cognitive Systems, Interaction, Robotics; Grant No. 287728, Jan 2012 – Jan 2015), dessen Ziel in der Entwicklung eines flexiblen, aber dynamisch versteifbaren Manipulators für den Einsatz in der minimal-invasiven Chirurgie lag.

Auf Seiten der Universität Siegen wurde dieses Projekt von Herrn Univ.-Prof. Dr.-Ing. Hubert Roth betreut. Ihm gilt mein besonderer Dank – an seinem Institut habe ich so viel mehr gelernt, als im Hörsaal vermitteln werden kann. Insbesondere die Projektarbeit überall auf dem Globus hat mein Weltbild nachhaltig verändert und meinen Blickwinkel erweitert.

Bedanken möchte ich mich auch bei den Kollegen der verschiedenen Arbeitsgruppen für die freundschaftliche Arbeitsatmosphäre, viele wertvolle Anregungen und stete Hilfsbereitschaft, die wesentlich zur Erstellung dieser Arbeit beigetragen haben.

Zusammenfassung

Eine Sparte der modernen Robotik, die sich in den letzten Jahren außergewöhnlich dynamisch entwickelt hat, ist die Medizinrobotik. Besonders ergiebig ist der Bereich der minimal-invasiven Chirurgie (MIS). Diese bringt neben den Vorteilen für den Patienten (schnellere Rekonvaleszenz, bessere kosmetische Ergebnisse, erhöhter Komfort) auch einige technische Herausforderungen für den Chirurgen mit sich. Die minimal-invasive Chirurgie bietet dabei großes Potential für roboter-basierte Assistenzsysteme und stellt mittlerweile einen eigenen Forschungszweig dar, der kontinuierlich weiter ausgebaut wird.

Die meisten der verfügbaren Systeme bauen dabei konzeptionell auf adaptierten Industriepattformen auf. Die etablierte Technik ermöglicht hohe Präzision und häufig eine detaillierte Vorab-Planung der einzelnen Arbeitsabläufe. Dem gegenüber steht der oft hohe Platzbedarf, der auch den Arbeitsraum des OP-Teams erheblich einschränkt und nicht selten zu einer unergonomischen Arbeitshaltung führt. Die extremen Sicherheitsvorkehrungen, die auf die Verwendung motorisch angetriebener, starrer Kinematiken zurückzuführen sind, treiben die Kosten in die Höhe. Als weitere Folge sind die entwickelten Systeme häufig nicht kompatibel mit der Kernspinresonanzspektroskopie (NMR-Kompatibilität).

Einen völlig anderen Ansatz verfolgt das von der EU geförderte Forschungsprojekt „Stiff-Flop“. Die Basis bildet ein hochflexibler Silikonschlauch, welcher durch Druckluft gesteuert und zum Zweck der Ausübung von Kräften dynamisch versteift werden kann. Inspiriert wird dieser Ansatz von der Bewegung des Tintenfisches, welcher sich einerseits ohne eine innere Schale durch die engsten Öffnungen zwängen kann, andererseits aber auch in der Lage ist, Teile seines Körpers zu versteifen, um nach Objekten zu greifen. Ein solches weiches System ist zwar inhärent sicher, bringt aber eine Reihe neuer Herausforderungen mit sich. Gerade im Bereich der Sensorik erfordert die flexible Beschaffenheit des Systems völlig neue Herangehensweisen und Lösungsansätze.

Eine Teilaufgabe bildet die Entwicklung eines optischen Sensorsystems, welches, basierend auf vorhandenem Live-Videomaterial (Endoskopkamera des Chirurgen), die Lage des Stiff-Flop Armes überwacht und zur Optimierung seiner Positionsregelung beiträgt. Die größte Herausforderung liegt dabei in der zuverlässigen Erkennung des gesuchten flexiblen Manipulatorabschnittes. Sowohl die Beschaffenheit des Objektes selbst als auch die durch den Arbeitsraum und die verwendete Gerätetechnik vorgegebenen Beschränkungen führen zu diversen Restriktionen bei der Umsetzung des Vorhabens. Erschwert wird die Aufgabe, weil ohne die Verwendung eines Stereokamerasystems aus der Kameraprojektion Rückschlüsse auf die Pose des detektierten Objektes im Raum gezogen werden müssen.

Etablierte Verfahren sind im untersuchten Szenario nicht anwendbar, da diese im Allgemeinen entweder auf der Erkennung eines bekannten und angelernten Umrisses, einer eindeutigen, bekannten Textur oder starken Kontrastunterschieden von Vorder- und Hintergrund basieren. Keine

dieser Voraussetzungen ist hier gegeben: Der Manipulator ist durchgängig flexibel, kann seine Länge ändern und auch eine Texturerkennung ist schwierig umzusetzen, da gegebenenfalls angebrachte Muster mit zunehmender Krümmung des Armes deutliche Verzerrungen in der zu erkennenden Textur erhalten.

Aus diesem Grund erfolgt die Lageermittlung zweigleisig: Zum einen wird der Stiff-Flop Arm als Ganzes durch einen texturbasierten Ansatz, einer so genannten Support Vector Machine (SVM), detektiert und vermessen, zum anderen erfolgt eine Überwachung von am Prüfling angebrachten optischen ringförmigen Markern mittels eines neuartigen, auf einer modifizierten Kreiserkennung basierenden Algorithmus. Die Detektierung dieser Umlaufmarkierungen bietet verschiedene Vorteile. Zum einen bleibt die Abbildung bei hinreichend schmalen Markierungen annähernd verzerrungsfrei, zum anderen wirkt sich eine eventuell auftretende, systembedingte radiale Streckung des Armes ausschließlich auf die Genauigkeit der Entfernungsmessung zwischen Kameraebene und Mittelachse des Manipulators aus. Das vorgestellte Verfahren ist außerdem robust gegenüber Bildrauschen oder kleineren Glanzlichtern, solange die Kontur des Ringes nicht über größere Abschnitte unterbrochen ist.

Die Redundanz der beiden angewandten Verfahren bietet im Wesentlichen zwei Vorteile: Zum einen erlaubt sie eine Plausibilitätsprüfung der erkannten Markerpositionen, zum anderen ist die Berechnung einer stetigen Mittellinie möglich. Diese kann im Gegensatz zu den diskreten Ringmarkerpositionen zusätzlich für eine einfache Kollisionserkennung genutzt werden, falls 3D-Modelle des Arbeitsraumes vorhanden sind. Für die eigentliche Aufgabe, also die Überwachung der Regelung, spielt das aber eine untergeordnete Rolle.

In der Implementierungsphase hat sich herauskristallisiert, dass eine Parameteroptimierung mit nachfolgender erneuter Kalibrierung beziehungsweise Vermessung des optischen Aufbaus essentiell ist. Ausgehend von einer großen Anzahl teils abhängiger Parameter wurden Methoden und Werkzeuge konzipiert und implementiert, die einem Anwender eine strukturierte Vorgehensweise bei der Parameterauswahl ermöglichen. Basierend auf einer Darstellungsmethode von Prof. Alfred Inselberg von der Tel Aviv University, Israel, wurde eine intuitiv anwendbare grafische Oberfläche zur Erleichterung der Parametrierung des Lageermittlungssystems geschaffen, die ein einfaches Vergleichen verschiedener Parametersätze ermöglicht.

Abstract

One of the most dynamically evolving branches of modern robotics is the area of medical robotics. Especially in minimally invasive surgery these applications lead to several advantages for the patients (faster convalescence, better cosmetic results, higher comfort), but also to some technical challenges for the surgeon. The minimally invasive surgery offers great potential for robot-based assistance and now represents a separate branch of research that is expanding continuously.

Most of the available systems are based on adapted industrial manipulator platforms. The established technology allows high precision and often a detailed pre-planning of the individual work processes. This often contrasts with high space requirements, which considerably restrict also the working space of the surgical team and often results in a non-ergonomic working posture.

Electric driven devices used in surgery require extreme safety precautions raising the development and operating costs. As a further consequence the developed systems are often not compatible with the nuclear magnetic resonance spectroscopy (NMR-compatibility).

The EU funded project „Stiff-Flop“ follows a different approach. The basis here is a highly flexible silicon tube that can be maneuvered through narrow holes by applying pressure to channels inside and that can be stiffened dynamically to apply forces to its environment. The idea is taken from the octopus, which can squeeze through narrow passages and grab prey on the other hand. Such a system - although inherently secure - brings a new set of challenges. The flexible nature of the system requires completely new approaches and solutions, especially in the field of sensors.

One of the tasks is the development of an optical sensor system that calculates the pose of the Stiff-Flop arm based on the available video stream from the endoscopic camera in order to enhance its position control. The biggest challenge lies in the reliable detection of the desired object. The nature of the object itself and the conditions in the work space as well as the equipment available for MIS lead to several restrictions in implementation strategies.

In addition, the task has to be carried out without the usage of a stereo camera system. Position and orientation of the tube have to be determined based on the video stream of a single endoscopic camera.

Established methods are not applicable in the examined scenario, as these are generally based on either the detection of known and trained outline, a clear well-known texture or pronounced differences in contrast of foreground and background. None of these conditions is given here: The manipulator is throughout flexible, can change its length, and texture recognition is difficult to implement, since any fitted pattern gets distorted significantly with increasing curvature of the arm. Because of these reasons, the pose determination is done in two steps: In the first step the Stiff-Flop arm is detected with a texture-based approach by a so-called Support Vector Machine (SVM). The resulting outline is measured and used to estimate the spatial pose of the tube. The second step is the detection of optical circular markers with a modified circle detection algorithm.

The detection of these circulating markers has several advantages. Firstly, the figure remains almost distortion-free with sufficiently narrow markings; on the other hand, a possibly occurring, system-related radial extension of the arm solely affects the accuracy of the distance measurement between the camera plane and the central axis of the manipulator.

The presented method is also robust against noise or smaller highlights, as long as the contour of the ring is not interrupted over larger sections.

The redundancy of the two methods used essentially offers two advantages: First, it allows a plausibility check of the detected marker positions; on the other hand, the calculation of a continuous centre line is possible. In addition, this center line can be used for simple collision detection, if 3D models of the working space are provided. For the primary task this aspect plays a subordinate role.

During the implementation phase, it has turned out that a parameter optimization with subsequent recalibration or measurement of the optical setup is essential. Based on a large number of partially dependent parameters methods and tools have been designed and implemented allowing a user to identify optimal parameters with a structured approach. Based on a graphical representation method developed by Prof. Alfred Inselberg of Tel Aviv University, Israel, an intuitive user interface has been developed, allowing a simple comparison of different parameter sets.

Inhalt

1.	Einleitung	3
2.	Einordnung der Arbeit	5
3.	Ziele der Arbeit	16
4.	Problemdefinition und Lösungsansatz / Systemarchitektur	17
5.	Implementierung	20
5.1	Verwendete Hardware	20
5.1.1	Industrieroboter KUKA KR 16	20
5.1.2	Endoskop Richard Wolf EndoCam® HD 5509R2	21
5.1.3	Frame Grabber Blackmagic Intensity Pro	22
5.1.4	Trackingsystem Axios CamBar B2	22
5.2	Verwendete Software	23
5.2.1	ROS Robot Operating System	23
5.2.2	Bildverarbeitungsframework OpenCV	25
5.2.3	Punktwolkenverarbeitungsframework PCL	26
5.2.4	TF: Transformation Framework Bibliothek	26
5.2.5	Datenbank MariaDB	27
5.3	Softwarearchitektur	28
5.3.1	Einbindung der Endoskopkamera	31
5.3.2	Einbindung des Trackingsystems	46
5.3.3	Einbindung der Datenbank MariaDB	63
5.3.4	Einbindung des Roboters KUKA KR 16-2	64
5.3.5	Das Launchfile (Systemstart)	71

5.4	Lageermittlung des Prüflings	73
5.4.1	Videovorverarbeitung.....	74
5.4.2	Konnektordetektion: Markererkennung und -vermessung.....	84
5.4.3	Objekterkennung: Aktive Konturen / „Snakes“	93
5.4.4	Objekterkennung: Support Vector Machines (SVM).....	94
5.4.5	Objektvermessung.....	95
5.5	Vermessung der optischen Komponenten	103
5.5.1	Iterative Closest Point (ICP)	104
5.5.2	Aligning object templates to a point cloud.....	105
5.5.3	Auswertung der Ergebnisse.....	110
5.6	Parallelisierungsmöglichkeiten	110
5.7	Visualisierung und Anwenderinteraktion	112
6.	Überprüfung der optischen Positionsermittlung	114
7.	Zusammenfassung und Ausblick	117

Anhänge

A.	Parallele Koordinaten	A
B.	O-Notation	F
C.	Literaturverzeichnis	K
D.	Abbildungsverzeichnis	O
E.	Verwendete Abkürzungen	R
F.	Schlüsselwörter	S

1. Einleitung

Die Gesundheitswirtschaft ist eine der größten Wachstumsbranchen in der Bundesrepublik Deutschland. Im Zeitraum von 2007 bis 2013 ist die Bruttowertschöpfung in diesem Sektor jährlich im Schnitt um 3,5 Prozent gestiegen – deutlich schneller als die Gesamtwirtschaft mit 2,4% Wachstum (Bundesministerium für Gesundheit, 2015). Dabei zeichnet sich der Gesundheitsmarkt neben der Entwicklung von innovativen Arzneimitteln und neuen Behandlungsmethoden insbesondere durch die Erforschung und Entwicklung neuer Hightech-Produkte in der Medizintechnik aus.

Besonders ergiebig ist hier der Bereich der minimal-invasiven Chirurgie (MIS). Diese bringt neben den Vorteilen für den Patienten (schnellere Rekonvaleszenz, bessere kosmetische Ergebnisse, erhöhter Komfort) auch einige technische Herausforderungen für den Chirurgen mit sich. Die minimal-invasive Chirurgie bietet dabei großes Potential für roboter-basierte Assistenzsysteme und stellt mittlerweile einen eigenen Forschungszweig dar, der kontinuierlich weiter ausgebaut wird.

Die meisten der verfügbaren Systeme bauen konzeptionell auf adaptierten Industriepattformen auf. Die etablierte Technik ermöglicht hohe Präzision und häufig eine detaillierte Vorab-Planung der einzelnen Arbeitsabläufe. Dem gegenüber steht der oft hohe Platzbedarf, der auch den Arbeitsraum des OP-Teams erheblich einschränkt und nicht selten zu einer unergonomischen Arbeitshaltung führt. Die extremen Sicherheitsvorkehrungen, die auf die Verwendung motorisch angetriebener, starrer Kinematiken zurückzuführen sind, treiben die Kosten in die Höhe. Als weitere Folge sind die entwickelten Systeme häufig nicht kompatibel mit der Magnetresonanztomographie, da sie magnetische Materialien enthalten und somit nicht in die starken Felder des MRT-Gerätes gebracht werden dürfen (NMR-Kompatibilität). Die besonderen Vor- und Nachteile einiger prominenter Produkte dieser Gattung sollen exemplarisch in Kapitel 2 näher vorgestellt werden.

Einen völlig anderen Ansatz verfolgt das von der EU geförderte Forschungsprojekt „Stiff-Flop“. Die Basis bildet ein hochflexibler Silikonschlauch, welcher durch Druckluft gesteuert und dynamisch versteift werden kann. Ein solches weiches System ist zwar inhärent sicher, bringt aber eine Reihe neuer Herausforderungen mit sich. Gerade im Bereich der Sensorik erfordert die flexible Beschaffenheit des Systems völlig neue Herangehensweisen und Lösungsansätze. Diese Arbeit beschäftigt sich mit der optischen Überwachung eines weichen Manipulators. Die konkreten Ziele dieses unabhängigen Lageüberwachungssystems sowie die gegebenen Rahmenbedingungen werden in Kapitel 3 definiert.

Eine kurze Betrachtung der besonderen Schwierigkeiten, die es zur Erreichung der Ziele aus dem Weg zu räumen gilt, erfolgt in Kapitel 4.

Kapitel 5 widmet sich der Implementierung des Lageüberwachungssystems sowie der technischen Umsetzung der definierten Vorgaben. Obwohl es sich beim Stiff-Flop-System um eine völlig neuartige Technologie handelt, kommen in der verwendeten Infrastruktur hauptsächlich medizintechnische Standardkomponenten wie Endoskopkameras oder genormte, minimal-invasive, aufsetzbare Werkzeuge wie Greifer oder Hochfrequenzablationsgeräte (HF-Cutter) zum Einsatz.

Das gleiche gilt für den Aufbau, der zur Kalibrierung des Lageüberwachungssystems eingesetzt wird: Der verwendete KUKA KR16 Handhabungsroboter ist ebenfalls ein Serienprodukt aus der Fertigungsautomation. Auf der Softwareseite wurde in weiten Bereichen auf etablierte Standards zurückgegriffen - sowohl das verwendete Roboterbetriebssystem ROS als auch die Bildverarbeitungsbibliothek sind verbreitete Komponenten, die den aktuellen Stand der Technik repräsentieren und sich großer Beliebtheit erfreuen. In Kapitel 5.1 und 5.2 werden diese Teile der eingesetzten Infrastruktur kurz umrissen, Kapitel 5.3 beschreibt die zur Integration nötigen Schritte – also im Wesentlichen den Aufbau und die Programmierung der einzelnen ROS-Treiber-Nodes und die daraus resultierende Funktionalität der einzelnen Bausteine.

Basierend auf den Eingangsdaten der einzelnen Nodes erfolgt die eigentliche Lageermittlung des Stiff-Flop-Arms. In Kapitel 5.4 werden initial verschiedene Arten der Bildaufbereitung vorgestellt und auf ihre Tauglichkeit im vorgesehenen Einsatzgebiet hin bewertet. Die eigentliche Lageermittlung erfolgt zweigleisig: Zum einen wird der Stiff-Flop Arm als Ganzes durch einen texturbasierten Ansatz, einer so genannten Support Vector Machine (SVM), detektiert und vermessen, zum anderen erfolgt eine Überwachung von am Prüfling angebrachten optischen ringförmigen Markern mittels eines neuartigen, auf einer modifizierten Kreiserkennung basierenden Algorithmus.

Ein großes Problem stellen hierbei die Toleranzen der eingesetzten Gerätschaften dar. Eine nachträgliche Vermessung der Komponenten ist unerlässlich. Die eingesetzten Verfahren dokumentiert Kapitel 5.5.

In der Implementierungsphase hat sich herauskristallisiert, dass eine Parameteroptimierung mit nachfolgender erneuter Kalibrierung beziehungsweise Vermessung des optischen Aufbaus essentiell ist. Ausgehend von einer großen Anzahl teils abhängiger Parameter wurden Methoden und Werkzeuge konzipiert und implementiert, die einem Anwender eine strukturierte Vorgehensweise bei der Parameterauswahl ermöglichen. Basierend auf einer Darstellungsmethode von Prof. Alfred Inselberg von der Tel Aviv University, Israel, wurde eine intuitiv anwendbare grafische Oberfläche zur Erleichterung der Parametrierung des Lageermittlungssystems geschaffen, ein einfaches

Vergleichen verschiedener Parametersätze ermöglicht. Das resultierende Visualisierungstool ist Gegenstand von Kapitel 5.7.

Die Überprüfung der Resultate an einem Prototyp des Manipulators, gefolgt von einer kurzen Zusammenfassung der Ergebnisse dieser Arbeit, zeigt Kapitel 6.

2. Einordnung der Arbeit

Der Einsatz von Robotern prägt seit einigen Jahrzehnten die Arbeitsabläufe in der Industrie, und auch im Privatleben hält die „soziale Robotik“ – also Maschinen, die darauf ausgelegt sind, mit ihrem Umfeld zu interagieren und Menschen im Alltag zu unterstützen – langsam aber kontinuierlich Einzug.

Ein Bereich der modernen Robotik, der sich in den letzten Jahren besonders dynamisch entwickelt hat, ist die Medizinrobotik. Der erste Roboter, der ab 1983 an der University of British Columbia und dem Vancouver General Hospital entwickelt wurde und dann 1985 in größeren Testreihen erprobt wurde, ist der *Arthrobot*TM (Lechky, 1985). Dieser wurde entwickelt, um Chirurgen bei Arthroskopien (Gelenkspiegelungen) an den unteren Extremitäten zu unterstützen, indem er auf Anweisung des Arztes das Bein des Patienten neu positionierte.

Das erste vollautomatische System, das in Deutschland zum Einsatz kam, war der von Integrated Surgical Systems in enger Zusammenarbeit mit IBM entwickelte und 1992 der Öffentlichkeit präsentierte *ROBODOC*TM. Dieses basierte im Kern auf einem Industrieroboter der Firma Sanko-Seiki und wurde entwickelt, um eine höhere Passgenauigkeit beim Einsetzen einer Hüftprothese in den Femurschaft zu erzielen. Obwohl dieser Medizinroboter in Amerika keine Zulassung erhielt, bescheinigte ihm der TÜV Rheinland die technische Sicherheit, weshalb das System eine europaweite Zulassung erhielt und kurzzeitig für eine Explosion der Chirurgierobotereinsätze in der Europäischen Union sorgte. Einige Jahre später kam das System in Verruf, da bei einer nicht unbedeutenden Anzahl von Patienten systembedingt durch die nötige großflächigere Ablösung des Gesäßmuskels Spätfolgen auftraten und viele Kliniken auf Schadensersatz verklagt wurden. Die monatelange Präsenz in den Medien hinterließ bei den Patienten eine ausgeprägte Skepsis und sorgte dafür, dass *ROBODOC*TM wieder weitgehend aus den Krankenhäusern verschwand. Neue Systeme werden seit diesem Vorfall noch intensiver auf mögliche unbeabsichtigte Auswirkungen getestet; die Hürden einer Markteinführung sind durch die zuständigen Behörden deutlich erhöht worden.

Trotz dieser anfänglichen Rückschläge bleibt das Potential manipulatorbasierter Assistenzsysteme unbestritten. Insbesondere die Chirurgie besitzt großes Potential für den Einsatz von Robotersystemen. Das erklärte Ziel ist jedoch nicht, den Chirurgen zu ersetzen, sondern ihm zusätzliche Behandlungsmöglichkeiten zu eröffnen. Um die Akzeptanz beim Patienten zu erhöhen, wird außerdem auf Komfortfunktionen wie die Ausgabe von Statusmeldungen oder Handlungsempfehlungen via Sprache verzichtet, um den Eindruck zu vermeiden, dass nicht der Chirurg, sondern die Maschine die Arbeitsweise vorgibt.

Derzeit haben sich ohne Anspruch auf Vollständigkeit die verschiedensten Einsatzfelder für roboterbasierte medizinische Assistenzsysteme etabliert:

- Minimalinvasive Chirurgie
- Neurochirurgie
- Orthopädie
- Physiotherapie und Rehabilitation
- Strahlentherapie
- Bildgebung
- Ophthalmologie
- Urologie

Obwohl viele aktuelle Medizinroboter aus technischer Sicht in Teilbereichen vollautomatisiert arbeiten können, kommen sie fast ausschließlich als Assistenz- oder Teleoperationssysteme zum Einsatz. Die Basis bilden nicht selten Industrieroboter oder ein an deren Funktionsweise angelehnte spezialisierte Manipulatoren mit 6 Freiheitsgraden.

Die Gründe für den Einsatz von Manipulatoren sind dabei vielfältig:

- Genauigkeit

Medizinroboter sind hochgenau. Während es in der Automobilindustrie in erster Linie auf eine hohe Wiederholgenauigkeit ankommt, müssen Manipulatoren in der Medizin eine hohe Absolutgenauigkeit aufweisen. Diese Anfahrngenauigkeit liegt je nach Anwendungsfall zwischen Imm und wenigen μm und wird zum einen durch ausgesuchte Komponenten mit wenig Spiel als auch durch die Verwendung spezieller Kinematiken erreicht.

Weitere spezielle Eigenschaften können je nach Anwendungsfall implementiert werden. In der Mikrochirurgie werden beispielsweise die Bewegungen des Chirurgen noch einmal um einen hohen Faktor verkleinert, damit Kanülen sicher in winzige Arterien im Auge eingeführt werden können. Intelligente Regelungen sorgen hierbei zusätzlich für die

Reduktion des physiologischen Tremors des Operateurs, was eine weitere Genauigkeitserhöhung mit sich bringt.

In einem anderen Fall hilft die automatische Werkzeugnachführung bei Patienten- oder Organbewegungen. Innere Organe können sich allein durch die Atmung des Patienten um mehrere Zentimeter verschieben. Eine Überwachung dieser Bewegung und eine vollautomatische sowie hochdynamische Korrektur der Posen der Werkzeuge erhöht die Genauigkeit des Eingriffs und vereinfacht die Arbeit des Operateurs erheblich.

- Ausweitung der Sicherheitsvorkehrungen

Der Einsatz von Manipulatoren in der Medizin bringt häufig zusätzliche Sicherheit mit sich. So kann zum Beispiel im Vorfeld der erlaubte Arbeitsraum auf Basis von Bilddaten definiert und während des Eingriffs überwacht werden; eine ungewollte Verletzung sensibler Bereiche wird dadurch vermieden. Hinzu kommt, dass in vielen Fällen Kräfte und Momente, die auf den Körper wirken, überwacht und beschränkt werden können. Diese Option findet häufig in der Orthopädie Anwendung, um Kollateralschäden am Knochen zu vermeiden.

- Fingerfertigkeit

Die Fingerfertigkeit des Chirurgen ist bei jeder Operation ein entscheidender Faktor. Gerade bei minimalinvasiven Eingriffen wird diese aber häufig durch einen kleinen Aktionsraum mit vorgegebener Stoßrichtung auf die Probe gestellt. Flexible Teleoperationssysteme mit mehr als sechs Freiheitsgraden sowie eine gute haptische Schnittstelle, gepaart mit einer hochauflösenden 3D-Optik, helfen dabei, auch in beengten Platzverhältnissen die nötige Fingerfertigkeit zu erhalten und intuitives Arbeiten zu ermöglichen.

- Ausdauer

Operationen sind häufig langwierig. Je nach Anwendungsfall werden einzelne Werkzeuge wie beispielsweise Endoskope zu Beginn des Eingriffs positioniert und verharren oft mehrere Stunden in der gleichen Position. Ein menschlicher Assistent kann dabei ermüden und im schlimmsten Fall unbeabsichtigt die Position des Werkzeuges verändern. Ein Manipulator hingegen ist ermüdungsfrei und leidet nicht unter einer unergonomischen Haltung, wenn er lange Zeit in der gleichen Pose verharren muss. Handelt es sich beim montierten Werkzeug um eine Kamera, wird zudem das erfasste Bild deutlich ruhiger. Im Unterschied zu einem festen Stativ ist der Roboter jedoch nicht auf eine Position festgelegt, sondern kann seine Werkzeuge ähnlich einem menschlichen Assistenten frei im Raum positionieren.

- Unempfindlichkeit gegen schädigende Einflüsse

Einige chirurgische Eingriffe finden bei gleichzeitigem Einsatz von bildgebenden Verfahren wie Computertomographie oder Röntgen statt. Dem Mediziner bleibt - möchte er sich nicht selbst der Strahlung oder dem Magnetfeld aussetzen - nur der Einsatz eines Teleoperationssystems.

- Optimierung des Arbeitsraumes

Der verfügbare Arbeitsraum am Patienten ist gering. Das gilt ganz besonders für minimalinvasive Eingriffe: Neben den eigentlichen Werkzeugen müssen auch noch Kameras so positioniert werden, dass sie sich auf kleinstem Raum nicht gegenseitig behindern und ihre Bedienelemente jederzeit zugänglich sind. Das führt nicht selten zu einer eher unergonomischen Position, die der Chirurg teilweise über mehrere Stunden einnehmen muss. Durch den Einsatz von Telemanipulatoren kann der beengte Raum effizienter genutzt werden; abgesetzte Bedienkonsolen bieten zudem die Möglichkeit, entspannter zu arbeiten – davon profitiert letztendlich auch der Patient.

- Online-Konsultation / Erweiterte Realität

Telemanipulatoren bieten zudem einen weiteren Vorteil: Bei Bedarf können externe Spezialisten ohne großen Aufwand konsultiert werden, indem deren Bedienkonsolen dem lokalen System via getunnelter Internetverbindung hinzugeschaltet werden. Auf diese Weise können Sie den Eingriff mit den Augen des lokalen Operators verfolgen und beratend tätig werden oder sogar unterstützend zur Hand gehen. Die Einblendung von zusätzlichen unterstützenden Daten (Augmented Reality / Erweiterte Realität), wie zum Beispiel der Überlagerung des Kamerabildes mit vorab generierten 3D Bilddaten oder im Vorfeld erstellten Planungsdaten bietet einen nicht zu unterschätzenden Mehrwert für Operateur und Patienten.

Nachdem die Vorteile manipulatorgestützter Assistenzsysteme zusammengefasst wurden, werden im Folgenden einige prominente zugelassene Medizinrobotersysteme und vielversprechende Forschungsprojekte in deren Umfeld exemplarisch kurz vorgestellt.

Orthopädie: Caspar™ (ortoMAQUET GmbH)

Das System *Caspar*™ (Computer Assisted Surgical Planning and Robotics) basiert auf einem Stäubli-Industrieroboter mit 6 Freiheitsgraden und wurde weitgehend vom *ROBODOC*™ inspiriert. Das System verfügte über eine eigene 3D-Planungsstation. Die Präzision der Planung lag im Bereich von Zehntelmillimetern, was das Orthopädie-System für passgenaue Fräsungen für Hüft- und

Knieprothesen wertvoll machte. Für den Eingriff mussten dem Patienten zunächst zwei Schrauben am Oberschenkel oberhalb des Knies eingesetzt werden. Diese Marker dienten als Orientierungspunkte und wurden für die Registrierung der 3D-Daten aus dem Computertomographen benötigt. *Caspar*TM erfreute sich unter den Orthopäden großer Beliebtheit – bis die Herstellerfirma wegen zu geringen Umsatzes liquidiert wurde.

Orthopädie: AcrobotTM (Acrobot Company Ltd)

*Acrobot*TM wird ebenfalls in der Orthopädie eingesetzt, allerdings mit unterschiedlichen Zielsetzungen. Der Name steht für „Active Constraint Robot“ und verdeutlicht die Aufgabe des Systems: Der Chirurg kann sein Werkzeug im vorher zu definierenden Arbeitsraum frei bewegen. Erreicht er jedoch dessen Grenzen oder definierte „No Fly Zones“, greift die Robotersteuerung ein und verhindert aktiv ein Überschreiten der definierten Grenzen. Eine autonome Bewegung des Roboters ist nicht vorgesehen. Das ursprüngliche Anwendungsfeld des *Acrobot*TM ist der Einsatz von Knieendoprothesen, er ist jedoch universell einsetzbar.

Radiotherapie: CyberKnifeTM (Accuracy Inc.)

Das *CyberKnife*TM-System kombiniert ein modernes Bildortungssystem mit einem Präzisionsroboter aus der Automobilindustrie, an dessen Handflansch ein leichtes und kompaktes Bestrahlungsgerät befestigt ist. Sein Ziel ist die Behandlung von gut abgrenzbaren Tumoren im Bereich des Gehirnes, der Wirbelsäule, des Rückenmarkes, der Lunge, Leber und Niere oder Prostata.

Mit Hilfe des computergesteuerten Bildortungssystems kann das *CyberKnife*TM den Tumor während der gesamten Behandlung orten und kleinere Patientenbewegungen automatisch ausgleichen. Es entfällt die Notwendigkeit, den Körper mit Hilfe externer Fixierungen zu befestigen oder eine Narkose durchführen zu müssen. Außerdem besteht die Möglichkeit, auch bei Atembewegungen dem Tumor mit dem Behandlungsstrahl punktgenau zu folgen. Eine Vergrößerung des Bestrahlungsbereiches - und somit die Zerstörung gesunden Gewebes - ist somit unnötig.

Telediagnostik: ReMeDi (Bristol Robotics Laboratory, University of the West of England)

Das Projekt *ReMeDi* hat sich zum Ziel gesetzt, dem Ärzte- und Spezialistenmangel, der vor allem in weniger entwickelten Ländern deutlich wird, durch ein Teleuntersuchungssystem entgegenzuwirken. Die *ReMeDi*-Einheit kann dabei eine Reihe physischer Untersuchungen durchführen. Dazu gehören zum jetzigen Zeitpunkt Palpation und Ultraschalluntersuchung, weitere Methoden sind geplant.

Bei jeder dieser Anwendungen geht *ReMeDi* über die klassischen Telepräsenz-Konzepte hinaus: Das Ziel ist, durch ein Multisensorkonzept und die Entwicklung einer intelligenten Nutzerschnittstelle ein Assistenzsystem zu schaffen, das dem Arzt kontextabhängig und proaktiv Unterstützung bei der

Untersuchung und gegebenenfalls auch bei der Behandlung anbietet. Da der Arzt auf qualifizierte Rückmeldung des Patienten angewiesen ist, gehört auch ein Telekonferenzsystem zum Aufbau.

In der entfernteren Zukunft ist ein Expertenmodul geplant, das autonome Untersuchungen von Patienten ermöglichen soll.

Interventionelle Radiologie: Sensei™ (Hansen Medical), CorPath™ (Corindus),

Innomotion™ (Innomedic), iSYS 1™ (iSYS Medizintechnik)

Die Interventionelle Radiologie ist ein relativ junges Teilgebiet der diagnostischen Radiologie. Sie dient nicht der Diagnostik, sondern stellt im Gegensatz zu dieser eine Behandlungsmethode dar. Der Grundgedanke ist die Abwicklung von therapeutischen Eingriffen mit visueller Unterstützung. Der Eingriff selbst erfolgt per Teleoperation; ein direkter händischer Eingriff verbietet sich auf Grund der radioaktiven Belastung durch das CT sowie des geringen verfügbaren Raumes um den Patienten. Bei den oben genannten Systemen handelt es sich ausschließlich um kleine, die Aufnahme nicht störende, fernbedienbare Kinematiken, die entweder am Röntgentisch oder dem Patienten selbst befestigt werden und mit in den Computertomographen eingeführt werden, um zum Beispiel bei laufender Aufnahme eine Kanüle in sensiblen Bereichen zu positionieren.

Minimalinvasive Chirurgie: da Vinci® (Intuitive Surgical),

SPORT™ Surgical System (Titan Medical Inc.)

RAVEN (University of Washington / UC Santa Cruz)

Das 2004 eingeführte *da Vinci®* System der Firma Intuitive Surgical ist ein universell einsetzbares, ursprünglich für Herzoperationen entwickeltes Telemanipulationssystem für minimalinvasive Eingriffe, das seine Stärken aber besonders im Urologiebereich ausspielt und deshalb mittlerweile vor allem dort eingesetzt wird. Es zeichnet sich vor allem durch seine hohe Präzision aus, wodurch sowohl Anzahl als auch Größe der nötigen Einschnitte minimiert werden können. Obwohl sich das System großer Beliebtheit erfreut, wird die vom Chirurgen notwendige Haltung an der Bedienkonsole von vielen Medizinern als suboptimal empfunden. Andere Systeme, wie das *SPORT™* Surgical System der Firma Titan Medical Inc., versuchen die Vorteile des *da Vinci®* Systems aufzugreifen und mit verbesserter Ergonomie für den Chirurgen zu verbinden. Für Forscher steht außerdem das quelltextoffene System *RAVEN*, dessen Entwicklung von der amerikanischen National Science Foundation gefördert wurde, für eigene Forschungsprojekte zur Verfügung. *RAVEN* hat keine Zulassung als Medizinprodukt und dient ausschließlich Forschungszwecken, was sich auch im Preis widerspiegelt.



Abbildung 1: da Vinci® System der Firma Intuitive Surgical; Quelle: Intuitive Surgical

Die bisher vorgestellten Systeme ähneln sich alle in ihrem grundlegenden Aufbau: Sie basieren auf einem Industrieroboter oder einer optimierten Variante, sind diskret aus Gelenken und Verbindungsstücken aufgebaut und haben zumeist sechs Freiheitsgrade. Dadurch ist es möglich, jede Position im Arbeitsraum in jeder möglichen Orientierung – mit Ausnahme der Vorgabe des Quadranten, zu erreichen. Dabei kann aber kein Einfluss auf den Verlauf des Armes genommen werden.

Diese Beschränkung haben kontinuumsmechanische Systeme nicht. Diese Systeme zeichnen sich dadurch aus, dass sie nicht aus starren Grundkörpern bestehen, die durch Gelenke verbunden sind, sondern ihre Bewegung durch Deformation erreichen. Sie sind in der Lage, ihren Zielbereich anzusteuern und dabei empfindliche Bereiche zu umfahren. Dieser Vorteil kommt zum Beispiel bei der Entfernung von intrazerebralen Blutungen zum Tragen: Eine „steuerbare Nadel“ wird durch eine Öffnung im Cranium eingeführt und sucht sich ihren Weg zum Blutgerinnsel, wobei sensible Areale umgangen werden und somit Kollateralschäden vermieden werden.

Die „Active Cannula“ der Vanderbilt University wurde speziell für diesen Anwendungsfall mit Fördergeldern der National Science Foundation und Geldern des Deutschen Akademischen Austauschdienstes (DAAD) entwickelt.

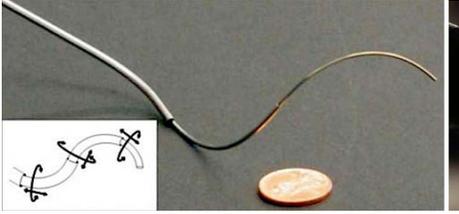


Abbildung 2: Active Cannula der Vanderbilt University; Quelle: David Comber, Vanderbilt University



Abbildung 3: CardioARM der Carnegie Mellon University; Quelle: Howie Choset

Das System besteht aus einer Reihe passgenau ineinander liegender, unterschiedlich vorgebener,

flexibler Röhren, die durch eine berechnete Serie von Verdrehungen und Verschiebungen gegeneinander dazu genutzt werden kann, einem kurvigen Pfad im Raum zu folgen. Das System wurde speziell für die transnasale Chirurgie entwickelt und kann nach Auskunft seiner Entwickler 83%-92% des hämorrhagen Volumens auf eine sichere und effektive Weise entfernen. Abbildung 2 zeigt den Aufbau der aktiven Kanüle sowie ihren Einsatz bei der Entfernung eines intrazerebralen Blutgerinnsels. Obwohl durch diese Methode bereits ein hohes Maß an Flexibilität erreicht wird, bietet das System nicht die Möglichkeit des nachträglichen Veränderns des Kurvenverlaufs. Außerdem ist die Anzahl der aufeinander folgenden Biegungen durch die Anzahl der Kanülen begrenzt; die Kurvenform kann nur innerhalb definierter Grenzen frei gewählt werden.

Eine Verbesserung bieten verschiedene von der Natur inspirierte Forschungsprojekte, deren Ziel es ist, die Bewegungsmöglichkeiten einer Schlange, eines Elefantenrüssels oder sogar eines Oktopusses abzubilden.

Ein Beispiel hierfür ist der *CardioARM* (Abbildung 3 auf der folgenden Seite) der Carnegie Mellon University, Pittsburgh. Im Gegensatz zum deformierbaren *Stiff-Flop Arm* besteht er aus einer Serie von verbundenen starren zylinderförmigen Objekten um einen flexiblen Tunnel, durch den katheterbasierte Werkzeuge eingeführt werden können. Diese Bauart ermöglicht unbegrenzte, aber kontrollierbare Flexibilität. Der so genannte Kopf der Schlange wird dabei durch einen Joystick gesteuert, der restliche Körper folgt dem einmal eingeschlagenen Pfad.

Ein wichtiges Kriterium für die Zulassung sämtlicher medizinischer Systeme ist die Gewährleistung der Sicherheit des Patienten bei Fehlfunktion oder Ausfall dieser Produkte. Häufig erfolgt die Sicherstellung durch redundante Sensoren, Aktoren und/oder Recheneinheiten. Eine beliebte Methode ist außerdem die Einführung einer übergeordneten Kontrollinstanz, die autonom und unabhängig vom regulären System über dieses wacht und dessen Entscheidungen kontinuierlich

überprüft. Im Falle der Medizinrobotik sind weiterhin zusätzliche Mechanismen zu implementieren, die dem Roboter im Falle einer Fehlfunktion ermöglichen, seinen Arbeitsraum zu verlassen, ohne den Patienten oder den Mediziner zu gefährden. Besonders einfach ist diesen Vorgaben mit so genannten „weichen“ Robotern zu entsprechen. Nachgiebiges Verhalten kann zwar auch mit einem Industrieroboter über aufwendige Sensorik und Steuermechanismen realisiert werden - einfacher ist jedoch die Verwendung von Konstruktionen, die von Natur aus weich und somit im Fehlerfall sicher sind.

Zu dieser Gruppe gehört auch der in dieser Arbeit verwendete Stiff-Flop Arm des gleichnamigen Forschungsprojektes unter der Leitung des King's College in London. Die Zielsetzung war die Entwicklung eines weichen und extrem flexiblen Silikonarmes, der durch eine 12mm Trokarhülse passt und sich zur Ausübung von Kräften an einzelnen Stellen dynamisch versteifen lässt. Gesteuert wird der Arm dabei über eine Anzahl von Kammern im Innern des Silikonkörpers, die bei Beaufschlagung mit Druckluft beliebige Verformungen bis hin zur Verlängerung des gesamten Armes hervorrufen können. Einen frühen Prototyp zeigt Abbildung 4.

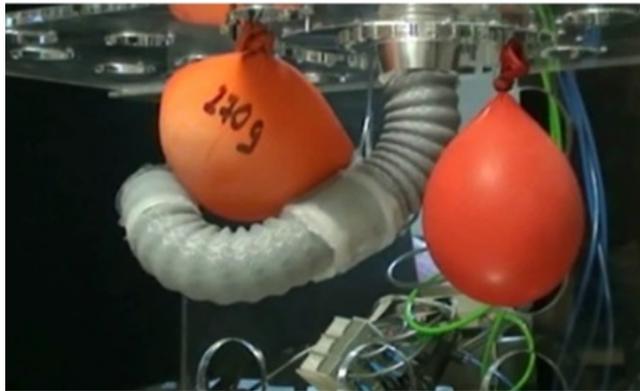


Abbildung 4: Früher Prototyp des Stiff-Flop Armes; Quelle: SSSA, Project Newsletter

Die Entwicklung von diesen weichen, kontinuumsmechanischen Manipulatoren bringt eine Reihe von interdisziplinären Herausforderungen mit sich. Unabhängig von der Erkundung der optimalen Materialstrukturen und der dazugehörigen Fertigungstechnik müssen neue Ansätze für Steuerung und Regelung gefunden, angepasste und möglichst flexible Sensoren entwickelt und eine der Aufgabe angepasste, möglichst intuitive Benutzerschnittstelle erarbeitet werden. Eine Teilaufgabe bildet die Entwicklung eines optischen Sensorsystems, das basierend auf vorhandenem Live-Videomaterial (Endoskopkamera des Chirurgen) die Lage des Stiff-Flop Armes überwacht und zur Optimierung seiner Positionsregelung beiträgt. Die größte Herausforderung liegt dabei in der zuverlässigen Erkennung des gesuchten Objektes.

Die Medizin verwendet mittlerweile eine Reihe von Standardverfahren, die eine optische Objekterkennung nutzen, um den Anwender bei der Entscheidungsfindung oder der Durchführung einer Aktion zu unterstützen. Hierzu gehören beispielsweise die automatische Erkennung von Markern zur Registrierung von Patientenaufnahmen, die Detektion von Tumoren in CT- oder Spektralaufnahmen oder auch die vermeintlich einfache Aufgabe des Zählens von Bakterien in einem Biofilm. Hierbei können verschiedene Segmentierungsverfahren zum Einsatz kommen:

- Pixelorientierte Verfahren

Bei den Pixelorientierten Verfahren wird die Entscheidung, ob ein Bildpunkt zum Objekt gehört oder nicht, ausschließlich vom Bildpunkt selber abhängig gemacht. Die Bildpunkte in der Nachbarschaft bleiben im Allgemeinen ohne Beachtung. Beispiele sind die Segmentierung auf der Basis von Farbwerten oder die Erkennung eines angeleuchteten Gegenstandes vor dunklem Hintergrund.

- Kantenorientierte Verfahren

Die Detektion von Objekten mittels kantenorientierter Verfahren verläuft meist mehrstufig. In einem ersten Schritt werden Kantensegmente mittels einfacher Vorgehensweisen wie dem *Sobel*- oder *Laplace-Operator* sowie einer *Gradientensuche* identifiziert, die aber meist noch keine geschlossenen Kurven darstellen und mittels eines Kantenverfolgungsalgorithmus komplettiert werden müssen.

- Regionenorientierte Verfahren

Bei den regionenorientierten Verfahren wird das Bild als Ganzes betrachtet. Ein gern verwendetes Verfahren ist das so genannte *Region Growing*, bei dem das Ursprungsbild zuerst gleichmäßig in Zellen aufgeteilt wird. Nachbarzellen werden dann miteinander verglichen und bei ähnlichen (vorher zu definierenden) Parametern miteinander verschmolzen.

- Modellbasierte Verfahren

Diese Verfahren setzen a priori Wissen über den gesuchten Körper ein, um diesen im Bild zu lokalisieren. Ein prominenter Vertreter dieser Verfahren ist die *Hough-Transformation*, die sich zum Auffinden von parametrisierbaren geometrischen Objekten in Gradientenbildern eignet. Für jeden dieser im Gradientenbild als Kante identifizierten Bildpunkte werden alle möglichen Parametersätze, die zu der zu suchenden Geometrie passen und deren Teil der betrachtete Kantenpunkt sein könnte, in den Hough-Raum übertragen. Jedem Punkt im Hough-Raum entspricht somit ein geometrisches Objekt im Bildraum. In einem zweiten Schritt wird der Hough-Raum nach Häufungen durchsucht; Maxima entsprechen dabei mit hoher Wahrscheinlichkeit dem Parametersatz des gesuchten Objektes. Als Beispiel bieten sich für einen Kreis Mittelpunkt und Radius als Parameter an.

- Texturorientierte Verfahren

Texturorientierte Verfahren bewerten nicht die Form oder Farbe eines Objektes, sondern dessen Oberflächenbeschaffenheit bzw. Struktur. Dadurch ergibt sich häufig die Möglichkeit der Objektklassifizierung. Ein effektives Beispiel für einen texturbasierten Algorithmus sind die so genannten Support Vector Machines (SVM). Diese klassifizieren kleine quadratische Bildausschnitte als „Objekt“ oder „Hintergrund“ und ermöglichen dadurch eine grobe Lageermittlung des gesuchten Körpers. Damit eine SVM tätig werden kann, muss sie im Vorfeld mit Bildausschnitten trainiert werden, die eindeutig einer der beiden Klassen „Objekt“ oder „Hintergrund“ zugeordnet werden können. Dieser Schritt erfolgt mit Hilfe von zahlreichen Beispielen beider Klassen, die entweder von Hand oder teilautomatisiert erstellt werden können. Abbildung 5 zeigt eine erfolgte Segmentierung auf der Basis des GrabCut-Algorithmus: Links ist das Ausgangsmaterial zu sehen. Der Anwender markiert grob Bereiche, die zum Vordergrund gehören. Der Algorithmus schätzt die Farbverteilung von Vorder- und Hintergrund mit Hilfe eines Gaussian Mixture Models (GMM, Methode zum Clustering) und ermöglicht so die Herauslösung des Vordergrundes. Im mittleren Bild ist die erfolgreiche Herauslösung des Roboters zu sehen, im rechten Bild wurde der gleiche Algorithmus auf den hinter dem Roboter stehenden Mann angewandt.

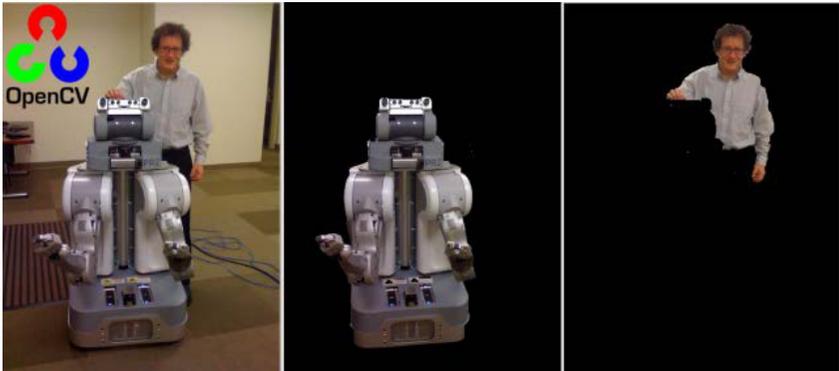


Abbildung 5: Beispiel für eine Segmentierung auf der Basis des GrabCut-Algorithmus, Quelle: WillogGarage.com

Sämtliche dieser Verfahren stellen verschiedene Anforderungen an das verwendete Bildmaterial. Außerdem wurde jeder Algorithmus für einen speziellen Anwendungsfall entwickelt und besitzt ausgeprägte Stärken und Schwächen. Aus diesem Grund muss in einem ersten Schritt der konkrete Anwendungsfall genau betrachtet werden, bevor es möglich ist, einen Lösungsansatz zu entwerfen.

3. Ziele der Arbeit

Die Ziele dieser Arbeit werden hauptsächlich durch ihre Einbettung in das EU FP7 Projekt Stiff-Flop (STIFFness controllable Flexible and Learn-able Manipulator for surgical Operations; Challenge 2: Cognitive Systems, Interaction, Robotics; Grant No. 287728) definiert.

Ziel des Projektes ist es, einen flexiblen Roboterarm für minimal-invasive chirurgische Eingriffe zu entwickeln, der sich flexibel bewegen, biegen, strecken und abschnittsweise versteifen kann. Inspiriert wird dieser Ansatz von der Bewegung des Oktopusses, welcher sich einerseits ohne eine innere Schale durch die engsten Öffnungen zwingen kann, andererseits aber auch in der Lage ist, Teile seines Körpers zu versteifen, um nach Objekten zu greifen.

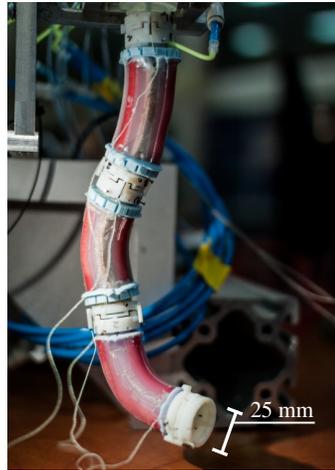


Abbildung 6: Stiff-Flop Arm mit 3 Segmenten; Quelle: PIAP, Project Newsletter

In der modernen Endoskopie und der roboterassistierten Chirurgie können diese Eigenschaften eines Werkzeuges sehr nützlich sein, da es bedingt durch den minimal invasiven Zugang, den Mangel an haptischem Feedback und Komplikationen mit steifen Robotern, die in einem begrenzten Raum zwischen den Organen operieren, Einschränkungen gibt.

Der Aufbau des hochflexiblen, aber dynamisch versteifbaren Stiff-Flop Armes erfordert die Entwicklung neuartiger verteilter Sensoren (Druck, Steifigkeit, Position, Biegung) und den Entwurf neuer Regelungs- und Lernstrategien für hyperredundante Robotersysteme.

Zur Überwachung und Optimierung der Regelung des Stiff-Flop Armes sollen unter anderem die Videodaten eines zusätzlich zum Manipulator einzuführenden Endoskops ausgewertet werden, dessen Hauptaufgabe darin besteht, dem Operateur einen besseren Blick auf den Arbeitsraum zu geben. Basierend auf dem Datenstrom dieses Endoskops soll die Position des Stiff-Flop Armes im 3D-Raum bestimmt werden.

Die Herausforderung besteht dabei in der robusten Detektion eines flexiblen Armes, welcher Form, Größe und Lage verändern kann und somit mit eingeführten Segmentierungsverfahren nur schwer zu erfassen ist. Hinzu kommt die Aufgabe, ohne Verwendung eines Stereokamerasystems aus der Kameraprojektion des Endoskops Rückschlüsse auf die Pose im Raum zu schließen.

Zielsetzung dieser Arbeit ist das Design und die Implementierung eines optischen Sensors, der verlässlich in der Lage ist, die Position eines kontinuumsmechanischen Roboters im Abdomen

basierend auf Videodaten einer Endoskopkamera zu bestimmen und somit zur Optimierung der Regelung und zur Überwachung verbotener Bereiche beizutragen.

Eine besondere Herausforderung ergibt sich durch die im Arbeitsraum vorherrschenden Bedingungen: Der zu detektierende Manipulator ist mit hoher Wahrscheinlichkeit durch Körperflüssigkeiten verschmutzt oder durch andere Strukturen teilverdeckt. Außerdem ergeben sich durch die Beleuchtungsanordnung in Kombination mit durch Feuchtigkeit reflektierenden Oberflächen Probleme in Form von Glanzlichtern im Videomaterial. Die durch Dämpfe und Sekrete kontinuierlich verschmutzende Linse des Endoskops verkompliziert ebenfalls die Ausgangslage. Alle diese Punkte müssen bei der Auswahl geeigneter Bildverarbeitungsalgorithmen berücksichtigt werden.

4. Problemdefinition und Lösungsansatz / Systemarchitektur

Der erste Schritt der Bildanalyse zur Verwirklichung des maschinellen Sehens ist üblicherweise die Unterteilung des Bildmaterials in Vorder- und Hintergrund (Segmentierung).

Die Auswahl eines geeigneten Segmentierungsverfahrens unterliegt diversen durch das Objekt und das Videomaterial vorgegebenen Beschränkungen:

- Pixelbasierte Verfahren sind auf Grund des unbekanntes und nicht beeinflussbaren Hintergrundes nur sehr begrenzt einsetzbar. Es kann keine Eigenschaft des Vordergrundes definiert werden, welche ihn in jedem Fall vom Hintergrund unterscheidbar macht.
- Kantenorientierte Verfahren setzen allgemein ein gutes Kontrastverhältnis voraus, welches auf Grund der ungünstigen Beleuchtungssituation im vorliegenden Fall selten gegeben ist.
- Regionenorientierte Verfahren kommen in der Medizintechnik häufig zum Einsatz, sind aber auf Grund ihrer Komplexität nicht auf Videodatenströme anwendbar.
- Modellbasierte Verfahren wie zum Beispiel die Hough-Transformation sind effektiv, wenn es sich bei dem gesuchten Objekt um eine geometrische Grundform handelt. Da die Dimension des Hough-Raumes aber direkt die Anzahl der zur eindeutigen Definition der Objektform nötigen Parameter widerspiegelt, wächst die Komplexität des Algorithmus exponentiell mit der Anzahl der Parameter. Die für einen Silikon Schlauch nötigen Parameter lassen eine Verwendung des Algorithmus zur Detektion des Stiff-Flop Armes nicht zu.

- Texturorientierte Verfahren eignen sich nicht nur zur Detektion bekannter Objekte, sondern auch zur Klassifizierung einer Szene in Vorder- und Hintergrund. Die Form des Vordergrundes spielt dabei weitgehend keine Rolle. Außerdem sind texturorientierte Verfahren relativ robust gegen verrauschtes Eingangsmaterial oder Teilverdeckung des gesuchten Objektes. Alle diese Eigenschaften qualifizieren diese Art der Algorithmen für den Einsatz im chirurgischen Umfeld.

Die Verwendung von texturorientierten Verfahren führt zuverlässig zur Erkennung des Objektes, bedingt durch eine nötige Rasterung des Ausgangsmaterials ist die mögliche Positionsbestimmung jedoch nicht hinreichend genau. Aus diesem Grund kommen zusätzlich Marker auf dem Stiff-Flop Arm zum Einsatz; das resultierende Erkennungssystem arbeitet also zweigleisig.

Für die Detektion von Markern auf dem Stiff-Flop Arm stellen die kantenbasierten Verfahren mit allen ihren Einschränkungen im vorliegenden Arbeitsumfeld einen brauchbaren Ansatz dar; für die Erkennung seiner Mittellinie sind sie suboptimal. Hier eignet sich besonders ein texturorientiertes Verfahren. Mit dem Ziel, eine automatisierte Erkennungsmethode für die Mittellinie des Stiff-Flop Armes mit hoher Performance zu entwerfen, fiel die Entscheidung auf ein Mustererkennungs- und Klassifizierungsverfahren basierend auf Support Vector Machines. Eine Herausforderung, mit der prinzipiell jeder Objekterkennungsalgorithmus zu kämpfen hat, sind schlecht ausgeleuchtete Szenen. Dieses Problem liegt auch im betrachteten Anwendungsfall vor: Bedingt durch die Bauart der Endoskopkamera sind Lichtquelle und optische Achse der Kamera koaxial, was zum verstärkten Auftreten von Glanzlichtern führt. Die zu implementierenden Objekterkennungsalgorithmen müssen auf diese besondere Situation hin optimiert und gegen die schädlichen Einflüsse von Glanzlichtern gehärtet werden. Für diesen Zweck werden verschiedene Verfahren zur Glanzlichtreduktion implementiert und evaluiert.

Ebenfalls durch den Aufbau der Endoskopkamera bedingt kommt es immer wieder zu Veränderungen der optischen Achse. Das liegt hauptsächlich an der Bauweise des Konnektors zwischen Kameraobjektiv und Endoskopschaft, der die beiden Elemente nur durch Federdruck zusammenhält. Außerdem haben die Linsen des Endoskopschaftes ein wenig Spiel: Bei passgenau eingefügten Linsen könnte es im Autoklav durch die unterschiedlichen Ausdehnungskoeffizienten von Stahl und Glas zu Beschädigungen kommen. Eine extrinsische Kamerakalibrierung ist also nötig und dient außerdem der Ermittlung der Präzision des fertigen Gesamtsystems. Um die Ergebnisse reproduzierbar zu halten, kommt bei der Kalibrierung ein Industrieroboter zum Einsatz, der den Prüfling automatisch an festgelegte Positionen im Arbeitsraum bewegt und dort vom Bildverarbeitungsalgorithmus vermessen lässt. Durch diesen Automatismus lassen sich im Idealfall nicht nur die extrinsischen Kameraparameter bestimmen, sondern auch Kenngrößen der Bilderkennungsalgorithmen optimieren.

Die praktische Vorgehensweise bei der Implementierung des Lageüberwachungssystems orientiert sich – wo möglich – am V-Modell. Dabei handelt es sich um ein Verfahrensmodell in der Softwareentwicklung, bei dem der Entwicklungsprozess ebenso wie das Vorgehen zur Qualitätssicherung in einzelnen Phasen organisiert ist (s. Abbildung 7).

Das V-Modell stellt im Kern eine Erweiterung des Wasserfallmodells dar: Die Ergebnisse einer Phase sind bindende Vorgaben für die nächsttiefere Projektphase. Der linke, orangene Ast beinhaltet die Festlegung der Spezifikationen und schließt mit der Realisierungsphase ab.

Den spezifizierenden Phasen stehen jeweils Testphasen im gelben, nach oben führenden Ast gegenüber. In der Darstellung ergibt sich ein charakteristisches „V“, woraus sich der Modellname ableitet. Ein Vorteil dieser Vorgehensweise liegt unter Anderem in der frühzeitigen Offenlegung konzeptioneller Schwachstellen oder Fehler.

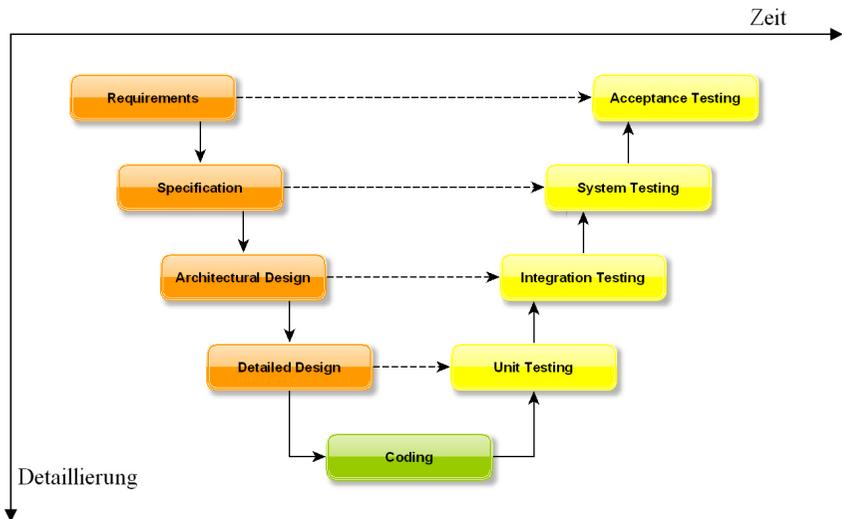


Abbildung 7: Vorgehensweise nach dem V-Modell

Die Aktionen der beiden oberen Detaillierungsschichten erfolgten in enger Kooperation mit dem Stiff-Flop Konsortium. In Arbeitsgruppen wurden die Anforderungen definiert und ein Lastenheft mit Spezifikationen erstellt. Basierend auf diesem Lastenheft wurden Architektur und detailliertes Design ausgearbeitet, welche im weiteren Verlauf näher beschrieben werden.

Da sich auch der Stiff-Flop Arm noch in einem frühen Forschungsstadium befand, wurde in vielen Fällen auf HIL (Hardware In the Loop) bzw. SIL (Software In the Loop) Ansätze zurückgegriffen, um unabhängiges Arbeiten der einzelnen Teams sicherzustellen. Halbjahresweise wurden Integrationstreffen veranstaltet, bei denen die Zusammenarbeit der einzelnen Module überprüft

wurde ("Integration testing"). Bei diesen Gelegenheiten wurde auch das Gesamtsystem auf Funktionalität überprüft ("System Testing").

Die Ergebnisse wurden regelmäßig durch ein Konsortium von Chirurgen evaluiert ("Acceptance testing"). Diese hatten während der gesamten Entwicklungsphase Zugriff auf alle Module und erhielten abschließend die Möglichkeit, das Gesamtsystem in sowohl künstlichen Umgebungen als auch an verschiedenen konservierten menschlichen Leichnamen zu erproben.

5. Implementierung

5.1 Verwendete Hardware

5.1.1 Industrieroboter KUKA KR 16

Die Positionierung des Stiff-Flop-Arms im Arbeitsraum zur Kalibrierung des Messsystems erfolgt mit Hilfe eines KUKA KR 16-2 Industrieroboters. Durch die Verwendung des KR 16-2 sind nicht nur automatische Optimierungsabläufe möglich, sondern auch eine Reproduzierbarkeit der Ergebnisse sichergestellt, da der Roboter über eine als sehr gut zu bezeichnende absolute Wiederholgenauigkeit ($\leq \pm 0,05$ mm) verfügt.



Abbildung 8: KUKA KR16 Industrieroboter

Die Anfahrgenauigkeit des Systems ist leider nicht bekannt – entsprechende Daten werden von KUKA nur für Manipulatoren mit dem Kürzel „HA“ (High Accuracy) zur Verfügung gestellt. Diese Baureihen verfügen über speziell selektierte Komponenten wie Getriebe und Achsen, die sich durch besonders kleines Spiel auszeichnen. Da im Versuchsaufbau bereits ein 3D-Trackingsystem (Axios

CamBar B2) vorhanden ist, kann dieses zur experimentellen Bestimmung der Fehlergrenzen im definierten Arbeitsraum genutzt werden. Der Roboter selber hat eine maximale Reichweite von 1610mm, welche aber im aktuellen Aufbau nicht ausgeschöpft wird. Das Gleiche gilt für die maximale Traglast von 16kg.

5.1.2 Endoskop Richard Wolf EndoCam® HD 5509R2

Als bildgebendes System in der Kette der verwendeten Hardware kommt die Kamera „HD EndoCam® 5509R2“ der Firma Richard Wolf (<http://www.richard-wolf.com>) zum Einsatz. Diese zeichnet sich durch eine hohe Auflösung von 1280 x 1024 Bildpunkten (Medical HD, Format 5:4) sowie eine Vielzahl von analogen und digitalen Schnittstellen aus, von denen sich insbesondere die HDMI-Schnittstellen zur Weiterverarbeitung der Daten eignen.

Über ein Zoom-Objektiv (C-Mount) mit einer Brennweite von 21mm-36mm wird ein Endoskopschaft mit einem Öffnungswinkel von 30° angeschlossen, der außerdem über Lichtleiterfasern für den Transport von Kaltlicht zur Ausleuchtung des Testkörpers verfügt.

Bei der angeschlossenen Lichtquelle handelt es sich um einen 300 Watt Xenon Strahler vom Typ Richard Wolf High Performance LP 5132. Die Lichtleistung kann stufenlos reguliert werden. Um die Farbtemperatur über den kompletten Stellbereich konstant zu halten, kommt dabei eine mechanische Blende zum Einsatz, welche bauartbedingt auch nur von Hand justiert werden kann.



Abbildung 9: Richard Wolf EndoCam mit Kaltlichtquelle

5.1.3 Frame Grabber Blackmagic Intensity Pro

Das von der Endoskopkamera digital zur Verfügung gestellte Videosignal (HDMI) muss zur weiteren Bearbeitung in die Workstation eingelesen werden. Das geschieht mit Hilfe eines Framegrabbers, welcher die digitale Bildinformation des Videodatenstroms entgegennimmt und im Hauptspeicher ablegt – in diesem Fall dem Modell „Intensity Pro“ der Firma Blackmagic Design (<http://www.blackmagicdesign.com>). Die nachrüstbare Steckkarte ermöglicht auf einfache Art und Weise die Erfassung und Bearbeitung von Videodaten in SD, HD und Ultra HD-Qualität. Sie verfügt sowohl über eingebaute HDMI Eingänge als auch einen HDMI Ausgang, welcher die Ausgabe des verarbeiteten Videosignals auf einem Kontrollmonitor ermöglicht. Für eventuell vorhandenes analoges Video- und Audioequipment ist eine Kabelpeitsche im Lieferumfang enthalten, mit deren Hilfe YUV-, FBAS- und S-Video-Signale verarbeitet werden können. Die Blackmagic Intensity Pro arbeitet mit unkomprimiertem SD oder HD Video sowie Kompressionscodecs wie ProRes, DVCPHD, DVCPHD 50, JPEG, HDV und DV.

Ein weiterer Vorteil und in diesem Fall ausschlaggebendes Kriterium für die Anschaffung der Intensity Pro ist das frei verfügbare und hinreichend dokumentierte „Decklink“ Software Development Kit (Blackmagic Design, 2011). Dieses stellt nicht nur Funktionen zur hardwarenahen Kontrolle des Framegrabbers zur Verfügung, sondern bietet dem Entwickler auch Schnittstellen für die einfache Durchführung von Standardaufgaben wie der Konvertierung von Videodaten. Unterstützt werden aktuell nicht nur die Betriebssysteme Microsoft Windows und Mac OS X, sondern auch aktuelle Linux-Distributionen wie das hier verwendete Ubuntu 12.04 LTS.

5.1.4 Trackingsystem Axios CamBar B2

3D-Messsysteme sind heute eine wesentliche Komponente von Systemen in der computerunterstützten Chirurgie. Mit Ihnen können räumliche Positionen von chirurgischen Instrumenten in Bezug auf die Anatomie des Patienten vermessen und visualisiert werden, wodurch für den Chirurgen meist präzisere Eingriffe möglich sind.

Im chirurgischen Bereich sind aktuell (Stand 2014) hauptsächlich optische Systeme im Einsatz; magnetische Tracker sind auf Grund ihrer geringen Baugröße und einfachen Handhabung im Vormarsch. Letztere haben aber immer noch das Manko der leichten Beeinflussbarkeit durch elektrisch leitfähige Materialien.

In unserem Aufbau kommt ein optoelektronisches Messsystem der Firma Axios zum Einsatz. Das Stereokameramodell Cambar B2 arbeitet auf der Basis der Nahbereichsphotogrammetrie: Ausgehend von Digitalaufnahmen des gleichen Arbeitsraumes aus unterschiedlichen Perspektiven wird die

räumliche Position vorab definierter Objekte bestimmt. Die CamBar B2 zeichnet sich dabei durch hohe Genauigkeit (3D Positioniergenauigkeit eines Punktfeldes 0,25mm) und einen großen Messbereich (Messabstand 800 - 2500mm) aus. Die Messungen erfolgen mit einer Frequenz von 20Hz bei voller Auflösung.

Im Gegensatz zu Produkten der Marktbegleiter erfolgt die Verarbeitung der Messdaten nicht in der Kamera selber, sondern auf einem über Ethernet verbundenen Rechner. In der Praxis ergeben sich dadurch keine Nachteile.

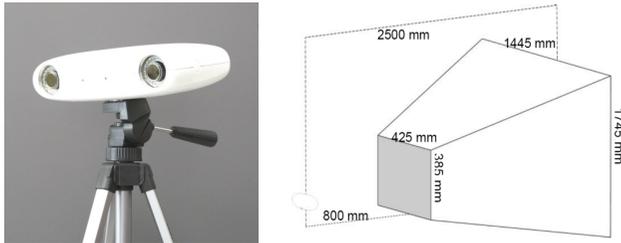


Abbildung 10: CamBar B2 (M) mit Abmessungen des erreichbaren Messvolumens, (AXIOS 3D@ Services GmbH, 2012)

Die CamBar B2 wird im Versuchsaufbau dazu verwendet, sowohl Position als auch Orientierung der Endoskopkamera zu ermitteln, um auf diesem Wege eine Transformation der durch den Bildverarbeitungsalgorithmus gewonnenen Positionsdaten des Stiff-Flop Armes in das Weltkoordinatensystem sicherzustellen. Außerdem wurde auch die Basis des flexiblen Arms selbst mit reflektierenden Markern versehen, um dem Kalibrieralgorithmus die exakte (reale) Position des Koordinatenursprungs des Manipulators zur Verfügung zu stellen.

5.2 Verwendete Software

5.2.1 ROS Robot Operating System

ROS ist nach eigenen Angaben ein Open Source Meta-Betriebssystem für Roboter (<http://www.ros.org>). Das Hauptaugenmerk liegt dabei auf der Wiederverwendung von Quellcode in der Roboterforschung und -entwicklung sowie der einfachen Einbindung von eigenen Bibliotheken. Diese können dabei in Python, C++ oder Lisp implementiert sein - Schnittstellen für Java und Lua befinden sich aktuell (August 2014) im experimentellen Zustand.

ROS stellt verschiedene Dienste zur Verfügung, unter Anderem

- Hardwareabstraktion / Gerätetreiber
- Transparente Interprozesskommunikation über Hardwaregrenzen hinweg
- Paketmanagement
- Diverse Werkzeuge und Bibliotheken zur einfachen Einbindung von eigenem Code
- Umfangreiche Simulationswerkzeuge

ROS verbindet individuelle Komponenten (Nodes) auf eine flexible Art und Weise und ermöglicht dadurch nicht nur die Wiederverwendung bestehenden Codes, sondern auch eine verteilte Entwicklung und Erprobung neuer Komponenten. Von besonderem Vorteil hat sich dabei die Möglichkeit erwiesen, Kommunikation zwischen den Nodes aufzuzeichnen und wieder in das System einzuspielen. Damit kann nicht vorhandene Hardware simuliert und außerdem eine Reproduzierbarkeit gewonnener Erkenntnisse sichergestellt werden.

Die Kommunikation der einzelnen Nodes erfolgt synchron über Services, asynchron über Topics oder über zentral zur Verfügung gestellte Datenspeicher wie den Parameter Server.

Die folgende Übersicht der ROS Strukturen wurde (Foote, Ros Wiki - Concepts, 2013) entnommen:

- *Nodes: Knoten sind Prozesse, welche Berechnungen durchführen. ROS ist modular ausgelegt, deshalb arbeiten in einem Robotersystem normalerweise viele Knoten zusammen. Ein Knoten setzt auf einer ROS client library wie roscpp oder rospy auf. Knoten kommunizieren direkt untereinander, die Verbindungsinformationen erhalten sie vom Master. Das Standardkommunikationsprotokoll ist TCPROS.*
- *Master: Über den ROS Master können Namen registriert oder nachgeschlagen werden. Ohne den Master wären die Knoten nicht in der Lage zu kommunizieren.*
- *Parameter Server: Der Parameterserver ist Teil des Masters und speichert Daten zu einem Schlüssel an zentraler Stelle.*
- *Messages: Knoten kommunizieren über Nachrichten (messages). Eine Nachricht ist eine Datenstruktur mit typisierten Feldern. Standard Datentypen (Integer, Float, Boolean) und Arrays derselben werden unterstützt. Nachrichten können beliebig verschachtelte Strukturen enthalten.*
- *Topics: Nachrichten werden über einen Publisher-Subscriber Mechanismus verteilt. Ein Knoten veröffentlicht Nachrichten zu einem Thema (publish to a topic) identifiziert über einen Namen. Ein anderer Knoten, welcher an diesen Nachrichten interessiert ist, abonniert das Thema (subscribe). Es können mehrere Knoten gleichzeitig als Publisher und Subscriber*

fungieren und ein Knoten kann mehrere Themen veröffentlichen oder abonnieren. Allgemein gesagt wissen Publisher und Subscriber nichts voneinander. So werden Erzeugung und Konsum von Informationen entkoppelt.

- *Services: Der Publisher-Subscriber Mechanismus erlaubt eine flexible, einseitige n-zu-n Kommunikation. Diese ist jedoch für eine Anfrage-Antwort-Interaktion zwischen den Knoten nicht geeignet. Für diese Art der Kommunikation bietet ein Knoten einen Service an, identifiziert durch einen Namen. Der aufrufende Knoten sendet eine Anfrage und wartet auf die Antwort.*
- *Bags: Bags sind zur Ablage und Wiedergabe von ROS messages. Mit Bags können z.B. Sensordaten aufgezeichnet werden, welche für die Entwicklung nötig sind, aber nur mit großem Aufwand aufgezeichnet werden können.*

Aktuell (Stand August 2014) läuft ROS hauptsächlich auf Unix-basierten Plattformen und wird in erster Linie unter Ubuntu und MacOS X getestet. Andere Portierungen befinden sich im experimentellen Status und werden von einzelnen Nutzergruppen weiterentwickelt.

ROS verfügt über eine große Entwicklergemeinschaft mit einem daraus resultierenden riesigen Pool an Treiber-Nodes für jede erdenkliche Hardware, sowie Implementierungen von Algorithmen für die verschiedensten Roboteraufgaben.

5.2.2 Bildverarbeitungsframework OpenCV

OpenCV (Open Computer Vision, <http://www.opencv.org>) ist eine in C und C++ implementierte, plattformunabhängige Softwarebibliothek zur Echtzeitverarbeitung von (Video-)Bildern. Sie wurde ursprünglich vom Intel Russia Research Center entwickelt. Heute wird die Weiterentwicklung hauptsächlich von den Firmen Willow Garage und Itseez unterstützt und koordiniert.

Der Funktionsumfang umfasst Algorithmen zur Bildsegmentierung, Objekt-, Bewegungs- und Gestenerkennung sowie verschiedenen Ansätzen zur Lösung von speziellen Problemen der Bildverarbeitung. Dabei profitiert OpenCV von einer großen Entwicklergemeinschaft. Diese bietet eine gute Unterstützung im Problemfall und hält die Bibliothek auf dem neuesten Forschungsstand der jeweiligen Fachgebiete. Durch einheitliche Schnittstellen ist die Wiederverwertbarkeit von vorhandenem Quelltext – gegebenenfalls mit kleineren Anpassungen – jederzeit möglich.

Da Bildverarbeitung im Allgemeinen eine hohe Rechenleistung benötigt, wurde bei der Implementierung der Bibliothek viel Wert auf die Effizienz der Algorithmen und des Programmcodes im Allgemeinen gelegt. OpenCV verwendet wenn möglich die Intel Integrated Performance Primitives (IPP) und unterstützt im breiten Umfang Multi-Threading. Viele Routinen

sind ebenfalls in einer Version für die Bearbeitung der Daten auf der GPU vorhanden. Unterstützt wird in diesem Fall die NVIDIA-Architektur CUDA.

OpenCV ist unter der BSD Lizenz frei verfügbar. Es ist außerdem fester Bestandteil jeder ROS-Installation.

5.2.3 Punktwolkenverarbeitungsframework PCL

Die Point Cloud Library (PCL, <http://www.pointclouds.org>) ist eine umfangreiche Bibliothek zur Verarbeitung von 3D-Bilddaten und n-dimensionalen Punktwolken. Enthalten sind zahlreiche, dem aktuellen Forschungsstand entsprechende Algorithmen zur Filterung, Segmentierung und Registrierung von Punktwolken sowie zur Oberflächenrekonstruktion aus Elementen der Punktwolke.

Mit dem Aufkommen von neuer, kostengünstiger 3D-Sensorik wie z.B. der Microsoft Kinect Kamera gewinnt 3D-Wahrnehmung im Bereich der mobilen Robotik in der Forschung, aber auch im Heimbereich immer mehr Bedeutung. Ihr Einsatz wird auch im Bereich der Handhabungsrobotik häufiger: Anwendungen wie der „Griff in die Kiste“ (die roboterbasierte Vereinzelung chaotisch bereitgestellter Objekte) eines Industrieroboters sind ohne umfangreiche Kenntnisse über die Beschaffenheit der Umgebung so gut wie unmöglich.

PCL kann im Rahmen der BSD Lizenz frei verwendet werden, auch für kommerzielle Zwecke. Eine detaillierte Übersicht über Funktionalität und Einsatzgebiete liefert (Radu Bogdan Rusu, 9-13 May 2011).

5.2.4 TF: Transformation Framework Bibliothek

Koordinatentransformation ist sowohl bei mobilen Robotern als auch im verwendeten Versuchsaufbau eine wesentliche Aufgabe.

Die meisten Sensoren liefern Daten in ihrem eigenen Bezugssystem. Dem Anwender ist lediglich die Position der einzelnen Sensoren zueinander bekannt. Über geeignete Koordinatentransformationen können die Sensordaten in ein einheitliches Bezugssystem überführt werden. Sollten die einzelnen Sensoren ihre Positionen

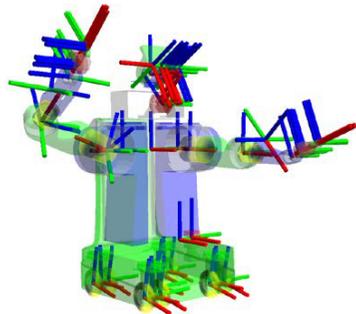


Abbildung 11: Koordinatensysteme eines mobilen Roboters, Quelle: (Foote, *tf: The Transform Library*, 2013)

während der Laufzeit relativ zu einander verändern, muss auch dieses erfasst und berücksichtigt werden. Die einzelnen Transformationen werden mit einem Zeitstempel versehen und die jeweils zum Zeitpunkt der Sensordatenaufnahme gültige für die Transformation verwendet. Dieser Aspekt ist besonders bei Sensordaten relevant, deren Vorverarbeitung eine gewisse Zeit in Anspruch nimmt: Auch wenn sich der Sensor während der Verarbeitungszeit bewegt hat, darf sich kein Einfluss auf das Transformationsergebnis ergeben.

Diese fehlerträchtige Aufgabe übernimmt im ROS die tf-Bibliothek (Foote, tf:The Transform Library, 2013). Auf Grund ihrer Struktur ist sie robust gegenüber asynchronen Eingaben und kann auch mit schwankenden Latenzzeiten umgehen. Die einzelnen Koordinatensysteme werden zu diesem Zweck in einer Baumstruktur abgelegt; die Transformationsmatrizen von einem Knoten zu einem übergeordneten Knoten sind bekannt. Außerdem wird für jeden Knoten eine Zeitspanne hinterlegt, für welche seine Transformation gültig ist.

Möchte man jetzt einen Sensorwert von einem Koordinatensystem in ein anderes transformieren, läuft ein Modul der tf-Bibliothek (der so genannte Listener) ausgehend von diesen beiden Koordinatensystemen im Baum nach oben, bis ein gemeinsamer Knoten gefunden wurde. Die Gesamttransformationsmatrix ergibt sich aus dem Produkt der auf dem Weg zum übergeordneten Knoten gefundenen inversen Transformationsmatrizen sowie der auf dem weiteren Weg wieder nach unten gefundenen hinterlegten Matrizen.

Diese Vorgehensweise hat sich als sehr effizient herausgestellt. Durchsucht man einen beliebigen Graphen ergibt sich die Laufzeit zu $O(E+N \cdot \log(N))$, wobei E die Anzahl der Kanten und N die Anzahl der Knoten darstellt. Durch die Verwaltung der Transformationsdaten in einer Baumstruktur verkürzt sich die Laufzeit auf $O(D)$; in diesem Fall stellt D die Tiefe des Baumes dar. Ein weiterer Vorteil liegt bei der einfachen Erweiterbarkeit des Graphen durch Hinzufügen neuer Koordinatensysteme. Eine kurze Einführung in die hier verwendete O-Notation findet sich in Appendix B: O-Notation.

5.2.5 Datenbank MariaDB

Für die Datenhaltung des Systems ist eine NoSQL-Datenbank vorgesehen. Obwohl die erwartete Datenmenge einen solchen Schritt nicht zwingend notwendig macht, wurde dieser Weg aus verschiedenen Gründen als vorteilhaft erachtet:

- Entsprechende Datenbanken sowie intuitiv zu nutzende Bibliotheken sind frei verfügbar
- Der Datenpool ist skalierbar, falls veränderte Vorgaben dies nötig machen sollten
- Ein einheitliches und weit verbreitetes Interface stellt den Datenaustausch zwischen den verschiedensten Programmpaketen her

Zum Einsatz kommt MariaDB (<https://mariadb.com/>), ein freier Ableger von Oracles SQL-Datenbank MySQL mit erweiterter Funktionalität. Sämtliche Schnittstellen, Bibliotheken sowie APIs der weit verbreiteten MySQL-Datenbank werden dabei unterstützt.

MariaDB unterstützt – wie MySQL auch – eine Vielzahl von Storage Engines, zum Beispiel MyISAM, InnoDB, XtraDB und einige mehr. Neu hinzugekommen und bevorzugte Wahl ist ARIA, welche im Gegensatz zu den vorher genannten Engines nicht relational, sondern objektorientiert arbeitet. Das heißt, dass sämtliche Daten und Funktionen gemeinsam in Objekten abgelegt werden, wobei Objekten Daten oder Eigenschaften vererbt werden können. Im Unterschied dazu werden in relationalen Datenbanken Daten als Zeilen in Tabellen verwaltet, wobei beliebige Beziehungen zwischen den Daten möglich sind.

Da es sich beim aktuellen Anwendungsfall um ein von der Datenmenge her kleineres Projekt handelt, werden sich voraussichtlich keine Engstellen beim Datendurchsatz ergeben und auch Aspekte wie sichere Transaktionen, Sicherung im laufenden Betrieb und Auditsicherheit keine Rolle spielen. Auf eine Abwägung der Vor- und Nachteile der einzelnen Storage Engines wird deshalb verzichtet und die Präferenz der Datenbank verwendet (ARIA Storage Engine).

5.3 Softwarearchitektur

Das aktuell implementierte Vision-System besteht aus zwei Schichten: Der kontinuierlichen Erkennung des Stiff-Flop Armes im Videodatenstrom der Endoskopkamera sowie seiner Positionsberechnung im Arbeitsraum und der offline arbeitenden Kalibrierung. Beides ist in Abbildung 12 auf der folgenden Seite zu sehen.

Zur Lageermittlung des Stiff-Flop Armes wird das Videosignal der Endoskopkamera von einem Framegrabber aufgezeichnet. Die zugehörige ROS Node „Intensity“ nimmt die Daten entgegen. Nach der Konvertierung in einen zur Weiterverarbeitung geeigneten Farbraum wird jedes Einzelbild auf der Basis der vorab ermittelten Kameraparameter entzerrt und – falls gewünscht – weitere Farbkorrekturen vorgenommen sowie die Auflösung zur Reduzierung der Systemlast verringert. Das auf diese Art aufbereitete Videosignal wird danach zur weiteren Verwendung im ROS-System publiziert.

Der nächste Arbeitsschritt ist die Detektierung des Armes im Videodatenstrom sowie die Ermittlung seiner Position im Raum bezogen auf das Kamerakoordinatensystem. Für den Regelalgorithmus ist dabei die Position der Basiskoordinatensysteme jedes einzelnen Armmoduls zu bestimmen. Für das angestrebte Ziel der Beachtung von so genannten „no fly zones“ im Arbeitsraum des Armes – also von Bereichen, die für den Arm tabu sind - sowie der Schaffung einer zusätzlichen Kontrollinstanz

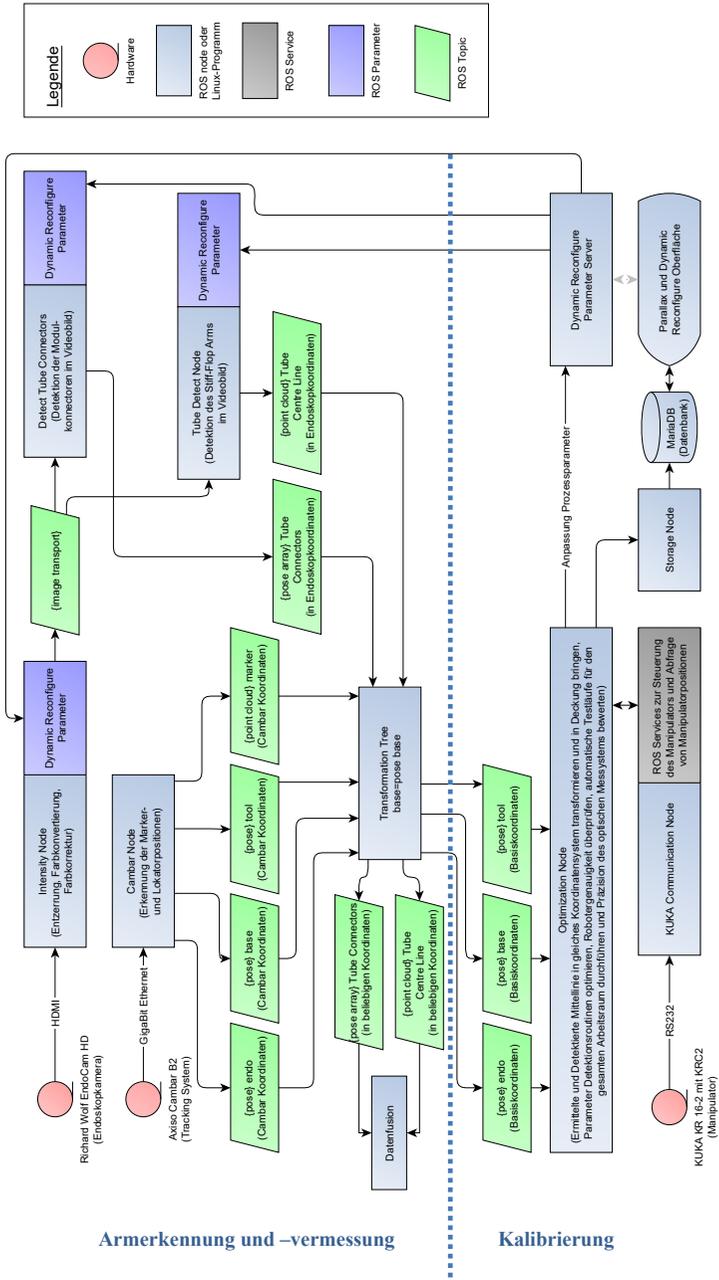


Abbildung 12: Das Vision-System mit Armerkennung und Vermessung (linker Teil) und Kalibrierung (rechter Teil)

zur Überwachung der Regelung ist es außerdem unerlässlich, die Mittellinie des Armes kontinuierlich zu bestimmen und zu verfolgen.

Für die Bestimmung der Basiskoordinaten jedes einzelnen Armmoduls ist die Node „Detect Tube Connectors“ zuständig. Der von der Intensity-Node gesendete Videodatenstrom wird nach farbigen, ringförmigen Markern um die Konnektoren der Armmodule durchsucht und deren konvexe Hülle bestimmt. Nach Anwendung verschiedener Filter werden die Parameter einer Ellipse ermittelt, die die verbleibende Kontur möglichst exakt beschreibt. Diese Ellipse stellt letztendlich die Abbildung eines Kreises im Raum auf die Bildebene der Kamera dar, aus der die Position des Mittelpunktes des Ursprungskreises rückgerechnet werden kann. Eine Aussage über die mögliche Ausrichtung des Kreises ist möglich, aber nicht eindeutig.

Die vom Algorithmus ermittelten Positionsdaten werden publiziert und stehen dem System direkt in Endoskopkoordinaten oder nach Transformation durch die tf-Node im Basiskoordinatensystem des Stiff-Flop Armes zur Verfügung.

Die zweite Teilaufgabe erfolgt durch die ROS Node „Tube Detect“. Hier wird das Videomaterial durch eine Support Vector Machine (SVM) analysiert und die Bereiche, in denen Teile des Armes erkannt wurden, in eine Maske geschrieben. Ausgehend von dieser Maske werden X- und Y-Position des Armes (bezogen auf das Kamerakoordinatensystem) ermittelt.

Da der Durchmesser des Armes bekannt ist, kann von der Breite des detektierten Armes auf seine Entfernung zur Kamera und somit auf seine Z-Position geschlossen werden. Eine genaue Berechnung ist hier leider nicht möglich, da bereits kleinste Detektionsfehler einen größeren Versatz in Z-Richtung nach sich ziehen. Eine leichte Verbesserung bringt die Anwendung diverser Filter sowie die abschließende Kalibrierung des Systems. Eine genaue Ermittlung der Entfernung ist nur durch Verwendung eines Stereoendoskops möglich, welches nicht zur Verfügung stand.

Die vom Vermessungsalgorithmus ermittelten Lagedaten werden wiederum publiziert und eine Punktwolke, welche die Mittellinie des Armes abbildet, zur Verfügung gestellt. Diese Punktwolke muss in einem weiteren Schritt in das Koordinatensystem des Stiff-Flop Armes konvertiert werden, um dessen Regelung als interpretierbarer Istwert zur Verfügung zu stehen.

Die Konvertierung zwischen Koordinatensystemen erfolgt unter Zuhilfenahme des fertigen ROS-Paketes „tf“, welches die Basiskoordinatensysteme von Arm, Roboterbasis und –werkzeug sowie Endoskop entgegennimmt und in Form eines Transformationsbaumes ablegt und verwaltet. Andere Komponenten im System können diesen Baum nutzen, um Koordinaten von einem Bezugssystem in ein anderes zu transformieren.

Zur Erstellung des Transformationsbaumes müssen aktuelle Informationen über die relative Lage und Orientierung der Basiskoordinatensysteme zueinander vorhanden sein. Diese werden vom optischen Trackingsystem ermittelt und in den Transformationsbaum eingefügt. Die Punktwolke der Einzelmarker der Lokatoren wird ebenfalls in der ROS Umgebung veröffentlicht, allerdings zurzeit nicht verwendet – sie dient aktuell nur der Ermittlung möglicher Fehlerquellen.

Vor dem ersten Einsatz muss das Vision System kalibriert werden. Auf Grund des verwendeten aufwendigen Linsensystems kann trotz extrem kleiner Fertigungstoleranzen leider nicht auf eine Vermessung verzichtet werden. Mit dem hier verwendeten Aufbau können Fehlergrenzen des Vermessungssystems bestimmt und die Genauigkeit der ermittelten Armposition deutlich erhöht werden.

Zu diesem Zweck wird ein vermessener Stiff-Flop Arm mittels eines Industrieroboters an bekannten Stellen im Arbeitsraum positioniert und diese Position mit der vom Vision System ermittelten Position verglichen. Die dabei verwendeten Point Cloud Matching Algorithmen der Point Cloud Library (PCL) liefern Korrekturdaten für die einzelnen Positionen im Arbeitsraum, welche für die Optimierung der Parameter des Vision-Systems verwendet werden. Außerdem werden sämtliche Korrekturdaten in einer Datenbank abgelegt, um räumliche Verteilungen im Arbeitsraum erkennen und gegebenenfalls über eine Korrekturmatrix ausgleichen zu können. Die Auswertung erfolgt dabei in parallelen Koordinaten nach (Inselberg, 2009).

Für die eigentliche Kommunikation mit dem Roboter ist die Node „KUKA Connector“ zuständig. Sie baut eine Verbindung zur KUKA Robotersteuerung auf, nimmt Kommandos von anderen ROS Nodes entgegen, prüft diese auf Plausibilität und leitet sie dann an den Roboter weiter. Gegebenenfalls erhaltene Antworten werden im Anschluss an die aufrufende Node zurückübermittelt.

5.3.1 Einbindung der Endoskopkamera

Der Stiff-Flop Arm wird während des Prozesses von der EndoCam HD aufgenommen. Diese liefert das Videomaterial via HDMI-Schnittstelle an einen Frame Grabber (Blackmagic Intensity Pro, <http://www.blackmagic.com>). Im Rahmen dieser Arbeit wurde eine ROS-Node entwickelt, die die Daten entgegennimmt, in ein passendes Format umwandelt, die durch die Linsenparameter der Kamera entstandenen Verzeichnungen herausrechnet und das resultierende Video entweder komprimiert oder unkomprimiert im ROS-Netzwerk zur Verfügung stellt. Einen detaillierten Ablaufplan zeigt Abbildung 13.

Wird die Node gestartet, muss sie sich zuerst beim ROS Master (dem so genannten Core) anmelden. Im weiteren Verlauf werden die angebotenen Services und Publisher initialisiert sowie die aktuell zu verwendenden Parameter vom ROS Parameterserver abgerufen. Zu diesen Parametern gehören die zu sendende Auflösung, Verwendung eines Farbkorrekturalgorithmus sowie ein Parameter zum Verwerfen jedes n-ten Bildes zur Reduzierung der Systemlast. Der nächste Schritt ist die Initialisierung der Intensity Pro Steckkarte und – bei Erfolg – das Laden der Kamerakalibrierungsdaten. Sollten keine Kalibrierungsdaten vorhanden sein, wird auf Standardwerte zurückgegriffen und der Benutzer entsprechend gewarnt.

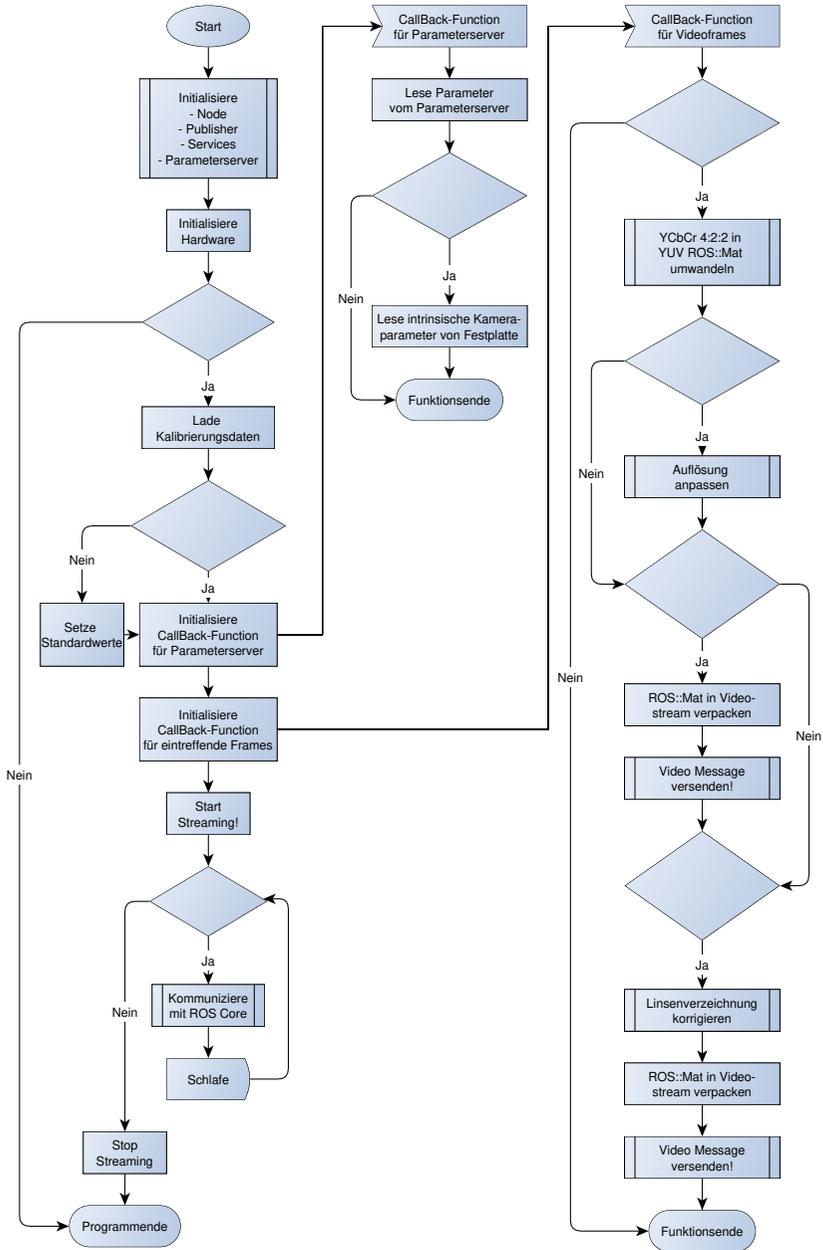


Abbildung 13: Programmablaufplan der Intensity-Node

Des Weiteren werden eine Callback-Funktion für den ROS Parameterserver sowie eine weitere für den Intensity-Treiber eingerichtet, bevor das Hauptprogramm sich in einer Endlosschleife auf die Kommunikation mit dem Master konzentriert.

Die Parameterserver-Callback-Funktion wird immer dann aufgerufen, wenn sich am Parametersatz der Node etwas ändert. So kann zum Beispiel der Benutzer über die „Dynamic Reconfigure“¹ Oberfläche zur Laufzeit Einfluss auf das Verhalten der Treiber-Node nehmen. Bei Aufruf wird der Funktion das komplette zuvor definierte Parameterset übergeben und sämtliche Werte werden lokal gespeichert. Sollte sich die gewünschte Auflösung des Videodatenstroms geändert haben, werden die bisher verwendeten Kalibrierungsdaten der Kamera gelöscht und ein entsprechender Satz für die neue Auflösung geladen. Obwohl sich weder an der Optik noch am Bildaufnehmer etwas geändert hat, macht diese Vorgehensweise Sinn, da jetzt der Algorithmus zur Entfernung der Verzeichnung nach der Reduzierung der Auflösung verwendet werden kann und somit die Rechenlast reduziert wird. Die Intensity-Callback-Funktion wird immer dann aufgerufen, wenn im Speicher des Frame Grabbers ein neues Videobild zur Abholung bereitliegt. Zuerst wird geprüft, ob das aktuelle Bild verarbeitet oder verworfen werden soll – letzteres kann vom Benutzer zur Reduzierung der Systemlast veranlasst werden. Im nächsten Schritt wird der Bildspeicher ausgelesen und die Daten in ein ROS::Mat Objekt umgewandelt.

Das vom Intensity-Treiber (Decklink) gelieferte Videobild liegt im Format 8BitYCbCr 4:2:2 vor.

Dieses ist mit dem bekannteren YUV Farbmodell, welches ähnlich dem Farbdreieck von einer linearen Addition der Farbzeize ausgeht, verwandt. Das YUV Farbmodell definiert die Farbinformation durch den Farbanteil (Chrominanz, definiert durch die Komponenten U=Blau-Gelb-Chrominanz und V=Rot-Grün-Chrominanz) sowie die Lichtstärke (Luminanz, Y).

Zum praktischen Einsatz kam dieses System erstmalig bei der Entwicklung des Farbfernsehens. Die Trennung von Helligkeit und Farbanteil ermöglicht eine vollständige Abwärtskompatibilität zu älteren Schwarz-Weiß-Geräten ohne die Notwendigkeit einer Bandbreitenerhöhung.

Analog dazu wird das YPbPr Farbmodell definiert und in der Norm ITU-R BT.601 (CCIR Comité Consultatif International des Radiocom, 1994) beschrieben. Mit dem Einzug der Digitaltechnik wurde die Norm um das gleichwertige YCbCr für digitale Übertragung erweitert. Abbildung 14 zeigt eine Abbildung der Parameter Cb und Cr auf den RGB-Farbraum für eine Helligkeitsebene von $Y=0,5$.

¹ ROS-Paket zur dynamischen Veränderung von Node-Parametern, welches auch eine Oberfläche zur Interaktion mit dem Nutzer zur Verfügung stellt. (http://wiki.ros.org/dynamic_reconfigure)

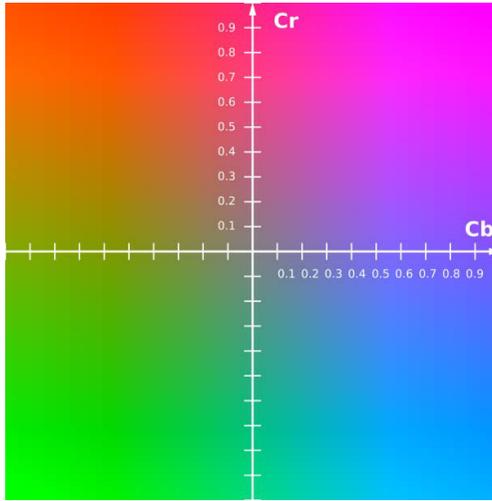


Abbildung 14: CbCr-Fläche des YCbCr-Farbmodells im RGB-Farbraum auf einer Helligkeitsebene von $Y = 0,5$.
 Autor: Simon A. Eugster (CC BY-SA 3.0)

YCbCr wird häufig fälschlicher Weise mit YUV gleichgesetzt. Trotz enger Verwandtschaft gibt es Unterschiede in der Bewertung der Luminanz. Sowohl bei YCbCr als auch bei YPbPr spannen die Chrominanzparameter eine symmetrische Fläche in der Farbebene auf. Beim ursprünglichen YUV wird die U Komponente mit einem geringeren Pegel verwendet als die V Komponente. Durch diesen Kunstgriff können YUV und YCbCr/YPbPr in einander überführt werden, es ergeben sich bei der verwendeten Quadraturmodulation jedoch geringere Auswirkungen auf das Luminanzsignal:

$$\begin{aligned} U &\approx 0,872021 \cdot P_b \\ V &\approx 1,229951 \cdot P_r \end{aligned} \tag{5.1}$$

Der im Projekt verwendete Frame Grabber liefert die einzelnen Parameter mit einer Tiefe von 8 Bit - allerdings erfolgt eine so genannte Farbunterabtastung (chroma subsampling, Abbildung 15) um die Übertragungsbandbreite und den nötigen Speicherbedarf zu reduzieren. Man spricht von einer Farbunterabtastung, wenn im Gegensatz zur Abtastung der Luminanz nicht jedes Pixel separat erfasst, sondern bei benachbarten Bildpunkten von annähernder Farbgleichheit ausgegangen wird. In der Praxis haben sich verschiedene Formen der Farbunterabtastung (also der Bündelung der Farbinformation) durchgesetzt. Zur Festlegung des verwendeten Formats wird normalerweise eine Gruppe aus 3 Ziffern verwendet. Ist ein Alphakanal enthalten kann eine vierte Ziffer folgen.

In unserem Fall kommt das 4:2:2 Verfahren zum Einsatz, welches ursprünglich aus dem analogen Farbfernsehstandard NTSC entstand (Poynton, 2003). Hier existiert ein Unterschied zwischen horizontaler und vertikaler Unterabtastung: Die Abtastrate in horizontaler Richtung ist nur halb so groß. Eine spürbare Qualitätsverringering ergibt sich daraus nicht. Das menschliche Auge kann Helligkeitsunterschiede zwischen zwei Pixeln viel deutlicher wahrnehmen als Unterschiede im Farbton.

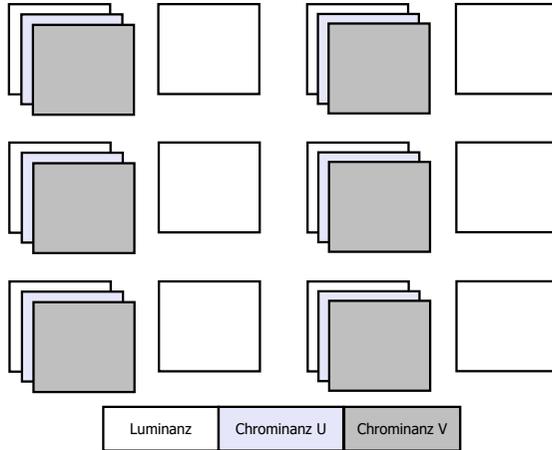


Abbildung 15: Chroma Subsampling

Die Darstellung im Speicher erfolgt gemäß ITU-R BT.601 in Blöcken von vier 8-Bit vorzeichenlosen Zahlen, welche als sog. „32-bit little-endian word“ übertragen werden und für die Darstellung von zwei auf einander folgenden Bildpunkten verantwortlich sind. Rein rechnerisch ergibt sich somit der Speicherverbrauch von 16 Bit pro Bildpunkt.

Details ergeben sich aus Abbildung 16, entnommen aus (Blackmagic Design, 2011):



Abbildung 16: Organisation der Bilddaten im Speicher, Autor (Blackmagic Design, 2011)

Der erste Schritt der Aufarbeitung der Rohdaten liegt in der Konvertierung in ein ROS-kompatibles Format – in diesem Fall in ein ROS::Mat Objekt. Die Farbanteile Y_0 , Y_1 , C_r und C_b werden ausgelesen, gewichtet und nach folgenden Formeln zu RGB-Daten gewandelt (Blackmagic Design, 2011):

$$R_1 = 1.164 \cdot (Y_0 - 16) + 1.793 \cdot (C_r - 128) \quad (5.2)$$

$$G_1 = 1.164 \cdot (Y_0 - 16) - 0.534 \cdot (C_r - 128) - 0.213 \cdot (C_b - 128)$$

$$B_1 = 1.164 \cdot (Y_0 - 16) + 2.115 \cdot (C_b - 128)$$

$$R_2 = 1.164 \cdot (Y_1 - 16) + 1.793 \cdot (C_r - 128)$$

$$G_2 = 1.164 \cdot (Y_1 - 16) - 0.534 \cdot (C_r - 128) - 0.213 \cdot (C_b - 128)$$

$$B_2 = 1.164 \cdot (Y_1 - 16) + 2.115 \cdot (C_b - 128)$$

Im Speicher werden die Daten in 3 Farbkanälen mit einer Tiefe von 8bit ohne Vorzeichen (CV_8UC3) abgelegt. Im Einzelfall kann es dazu kommen, dass die errechneten Farbanteile den gültigen Wertebereich zwischen 0 und 255 verlassen. In diesem Fall wird der errechnete Wert auf die Grenzen des erlaubten Intervalls verschoben.

Dieser relativ rechenaufwendige Schritt der Rohdatenkonvertierung kann parallel abgearbeitet werden. Zu diesem Zweck werden die Bildschirmzeilen in Blöcken zusammengefasst, welche jeweils einen separaten Thread zur Bearbeitung übergeben werden. Auf diese Weise lassen sich auch hochauflösende Bilder zeiteffizient konvertieren und später entzerren. Die Anzahl der Threads wurde empirisch auf eine Anzahl begrenzt, die der Hälfte der Prozessorkerne entspricht – in unserem Fall sind es sechs Bildabschnitte, die parallel konvertiert werden.

Sollte vom Anwender eine Größenanpassung erwünscht sein, erfolgt diese im nächsten und letzten Schritt vor der Entzerrung des Bildes. Diese Entzerrung ist nötig, da der später anzuwendende Vermessungsalgorithmus auf eine möglichst exakte 2D Projektion der realen Welt zurückgreifen muss, um möglichst genaue Abmessungen und Distanzen aus der Projektion ableiten zu können. Für den rechnerischen Vorgang der Entzerrung benötigen wir die Kameramatrix \mathbf{K} sowie den Entzerrungsvektor \mathbf{D} . Soll zusätzlich eine Rektifizierung erfolgen wird zusätzlich die Rotationsmatrix \mathbf{R} verwendet. In diversen Algorithmen erscheint noch die Projektionsmatrix \mathbf{P} – sie dient der Abbildung von Punkten des 3D-Raumes auf dem rektifizierten Bild.

Die einzelnen Parameter werden im Folgenden kurz erklärt.

Ausgehend vom idealisierten Modell der Lochkamera lässt sich zunächst die Kameramatrix \mathbf{K} ableiten.

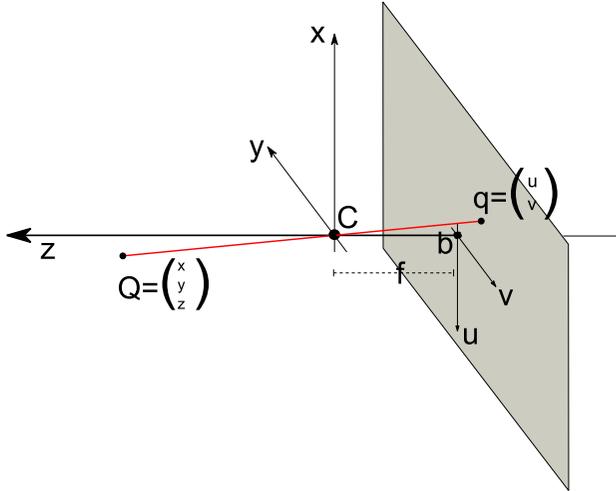


Abbildung 17: Funktionsweise einer Lochkamera

Die Lochblende – auch das optische Zentrum C genannt – liegt im Ursprung des Kamerakoordinatensystems. Senkrecht zur Bildebene steht die optische Achse. Diese ist im Allgemeinen mit der z -Achse identisch. x - und y -Achse liegen parallel zur Bildebene. Verlängert man die z -Achse durch die Lochblende in Richtung der Bildebene, markiert sie dort den Bildmittelpunkt und somit den Ursprung des Bildkoordinatensystems, dessen Achsen hier mit u und v bezeichnet sind. Der Abstand vom optischen Zentrum C zur Bildebene wird Brennweite f genannt.

Ein Punkt $Q=(x, y, z)^T$ aus dem dreidimensionalen Koordinatensystem der Kamera in der realen Welt wird auf den zweidimensionalen Punkt $q=(u, v)^T$ des Bildaufnehmers abgebildet. Das Abbild des Objektes wird dabei um 180° gedreht. In Abbildung 17 wurde dieser Effekt durch ein gedrehtes Bildkoordinatensystem berücksichtigt, um die abgeleiteten Formeln nicht unnötig zu verkomplizieren. Basierend auf dem Strahlensatz wird die Abbildung um den von Brennweite und Abstand des aufgenommenen Punktes zum optischen Zentrum definierten Faktor skaliert:

$$\begin{aligned} u &= \frac{f}{z} \cdot x \\ v &= \frac{f}{z} \cdot y \end{aligned} \tag{5.3}$$

Das entspricht nach (Hartley, 2008) einer Abbildung vom euklidischen \mathbf{R}^3 nach \mathbf{R}^2 :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{fx}{z} \\ \frac{fy}{z} \end{pmatrix} \quad (5.4)$$

Wählt man eine Darstellung in homogenen Koordinaten, kann die Projektion auch als Matrixmultiplikation ausgedrückt werden:

$$Q_h = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 \\ f & 0 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = q_h \quad (5.5)$$

Diese Darstellungsform bietet unter anderem den Vorteil, dass man neben der Projektion eine konstante Verschiebung des Bildkoordinatensystems festlegen kann – üblicherweise möchte man den Bildhauptpunkt b in der linken oberen Ecke des Monitors vorfinden.

Erweitert man Gleichung (5.5) um die Verschiebung $b=(b_u, b_v)^T$ bekommt man

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fx + b_u z \\ fy + b_v z \\ z \end{pmatrix} = \begin{bmatrix} f & b_u & 0 \\ f & b_v & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (5.6)$$

In der Literatur ist häufig auch noch ein Faktor s (skew, Schrägheit/Scherung) zu finden, der der Tatsache Rechnung trägt, dass durch Fertigungstoleranzen realer Kameras gegebenenfalls die optische Achse nicht orthogonal zur Bildebene steht. Meistens ist s annähernd gleich null.

Die resultierende Matrix

$$\mathbf{K} = \begin{bmatrix} f & s & b_u \\ & f & b_v \\ & & 1 \end{bmatrix} \quad (5.7)$$

wird Kamerakalibriermatrix oder kurz Kameramatrix genannt und enthält sämtliche intrinsischen Parameter (f , b_u und b_v sowie ggf. die Skalierungen m_u und m_v , s. unten), die sich – unabhängig von etwaigen Kamerabewegungen – nicht ändern.

Im Zeitalter der digitalen Fotografie möchte man meistens das dreidimensionale Koordinatensystem der realen Welt in Koordinaten diskreter Pixel abbilden. Dieser Vorgang wird dadurch erschwert, dass die Bildpunkte auf einem CCD-Chip nicht in jedem Fall als quadratisch anzunehmen sind.



Abbildung 18: Aufnahme der Pixelmatrix eines CCD-Chips unter dem Mikroskop (3000x), Pixelgröße $\approx 2,5\mu\text{m}$, Autor: Beuth Hochschule für Technik, Labor für Gerätetechnik, Optik und Sensorik, Berlin

Aus diesem Grund müssen zwei verschiedene Skalierungsfaktoren m_u und m_v für die Größe eines Bildpunktes auf dem Bildaufnehmer berücksichtigt werden, siehe (Hartley, 2008):

$$\mathbf{K} = \begin{bmatrix} f \cdot m_u & s & b_u \cdot m_u \\ & f \cdot m_v & b_v \cdot m_v \\ & & 1 \end{bmatrix} = \begin{bmatrix} a_u & s & q_{uc} \\ & a_v & q_{vc} \\ & & 1 \end{bmatrix} \quad (5.8)$$

Bisher wurden sämtliche Objektkoordinaten in Kamerakoordinaten beschrieben. In der Realität möchte man häufig ein Weltkoordinatensystem nutzen, welches nicht im optischen Zentrum der Kamera liegt.

Zur Projektion von Weltkoordinaten auf die Bildebene ist dabei ein Zwischenschritt nötig, in dem die Objektkoordinaten in das Kamerakoordinatensystem transformiert werden. Dieser Zwischenschritt besteht aus einer Rotation \mathbf{R} um den Ursprung sowie einer Translation \mathbf{C} :

$$q_h = \mathbf{KR} \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \end{bmatrix} Q_h \quad (5.9)$$

Der „statische“ Teil dieser Projektion wird in der Literatur zur Projektionsmatrix \mathbf{P} zusammengefasst:

$$\mathbf{P} = \mathbf{KR} \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \end{bmatrix} \quad (5.10)$$

Sämtliche Betrachtungen bezogen sich bisher auf das idealisierte Lochkameramodell. In der Praxis scheitert eine Lochkamera an der viel zu geringen Lichtmenge, die die Lochblende passiert. Aus diesem Grund ist der Einsatz von Linsen unverzichtbar, welche die Aufgabe haben, parallel einfallende Lichtstrahlen konvergent zu machen und in einem Brennpunkt zu bündeln. Leider bringt der Einsatz von Linsen den negativen Effekt der Verzerrung der Abbildung auf der Bildebene mit sich. Man spricht in diesem Zusammenhang von der so genannten „Linsenverzeichnung“. Hochwertige Objektive versuchen, die Verzeichnung durch eine Anordnung hochwertiger Linsen möglichst gering zu halten. Gerade bei eher preiswerten (und in unserem Fall kleinen) Systemen ist eine rechnerische Korrektur der Verzeichnung aber unerlässlich. In der Praxis treten fast ausschließlich radiale und tangentiale Verzeichnungen auf.

Die radiale Verzeichnung ist ein rotationssymmetrischer Vergrößerungsfehler um das so genannte optische Zentrum der Linse, welcher mit wachsendem Abstand zu diesem Punkt zunimmt. Wird das Objekt zu den Bildrändern hin vergrößert dargestellt, spricht man von kissenförmiger Verzeichnung. Scheint das Objekt am Bildrand zu schrumpfen, liegt eine tonnenförmige Verzeichnung vor, wie Abbildung 19 veranschaulicht.

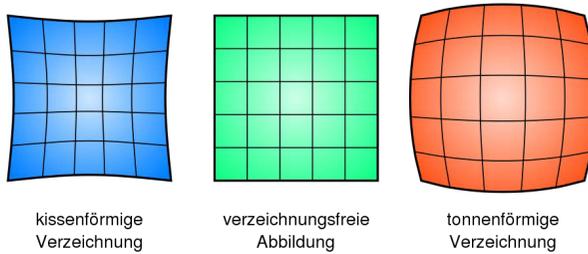


Abbildung 19: Geometrische Verzeichnung, Grafik: Joachim Bäcker (CC BY-SA 3.0)

Vereinfacht ausgedrückt kann man sagen, dass der verzerrte Bildpunkt entlang einer Geraden durch den Bildhauptpunkt und den korrekt abgebildeten Punkt verschoben wurde. Um diese Verschiebung rückgängig zu machen, wird ein Korrekturfaktor c eingeführt, der abhängig ist vom euklidischen Abstand des Punktes in der Bildebene zum Bildhauptpunkt b , hier mit r bezeichnet.

$$\hat{\mathbf{q}} = \mathbf{q} \cdot c(r) \quad (5.11)$$

Dieser Korrekturfaktor $c(r)$ wird häufig in einer Taylorreihe angenähert, um die Datenhaltung zu optimieren und effiziente Algorithmen zur Verarbeitung der Information zu ermöglichen. Unter Berücksichtigung des Rechenaufwandes sowie des geringen Einflusses höherer Glieder erscheint es ausreichend, die Entwicklung nach dem zweiten Glied abubrechen, sofern nicht mit extremen Verzerrungen, wie zum Beispiel bei einer Fisheye-Linse, zu rechnen ist. Nach (Hartley, 2008) werden somit nur Parameter k_1 und k_2 sowie gegebenenfalls k_3 zur Beschreibung der radialen Linsenverzeichnung benötigt:

$$c(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 \quad (5.12)$$

Die tangentielle Verzeichnung entsteht durch eine nicht perfekt zur Bildebene ausgerichtete Linse; ihr Mittelpunkt liegt nicht auf der optischen Achse. Mögliche Auswirkungen zeigt Abbildung 20:

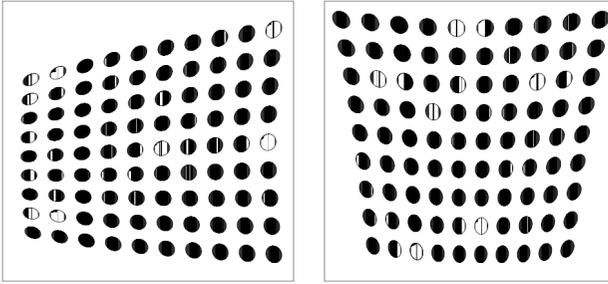


Abbildung 20: Tangentiale Linsenverzeichnung, entnommen aus (Christoph Stiller, 2009)

In diesem Fall ist die Verzeichnung nicht direkt vom Abstand des Bildpunktes zum Bildhauptpunkt abhängig; vielmehr wird ein Kreis um den Bildhauptpunkt durch den Bildpunkt gelegt, eine Tangente an den Kreis durch den Bildpunkt konstruiert und dann dieser Bildpunkt entlang der Tangente verschoben, siehe Abbildung 21:

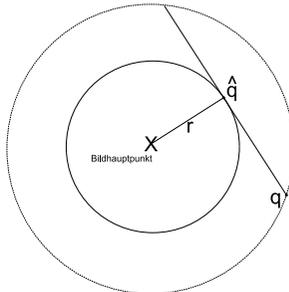


Abbildung 21: Tangentiale Verzerrung

Eine Korrektur der tangentialen Verzeichnung errechnet sich nach (Bradski & Kaehler, 2008) über den Term

$$\hat{\mathbf{q}} = \mathbf{q} + \begin{pmatrix} 2p_1uv + p_2(r^2 + 2u^2) \\ p_1(r^2 + 2v^2) + 2p_2uv \end{pmatrix} \quad (5.13)$$

Die Gleichungen (5.11) bis (5.13) werden in der Literatur gerne zum „plumb bob“ genannten und für die meisten Kameras ausreichenden Verzerrungsmodell zusammengefasst:

$$\hat{\mathbf{q}} = \mathbf{q} \cdot \left((1 + k_1r + k_2r^2 + k_3r^3) + \begin{pmatrix} 2p_1uv + p_2(r^2 + 2u^2) \\ p_1(r^2 + 2v^2) + 2p_2uv \end{pmatrix} \right) \quad (5.14)$$

Insgesamt werden also fünf Parameter benötigt, um die gängigsten Linsenverzerrungen korrigieren zu können. Diese werden in OpenCV zum Vektor \mathbf{D} zusammengefasst:

$$\mathbf{D} = (k_1, k_2, p_1, p_2, k_3)^T \quad (5.15)$$

Zur Bestimmung der oben festgelegten Parameter existieren verschiedene Ansätze. Gebräuchlich ist zum Beispiel der Ansatz von (Tsai, 1986), da er einen guten Mittelweg zwischen Exaktheit und Rechenaufwand darstellt. Tsai nutzt die Methode der kleinsten Quadrate um möglichst viele Parameter zu bestimmen. Dabei geht er in vier Schritten vor:

- Umrechnung von Welt- in Kamerakoordinaten
- Berechnung des Bildkoordinatensystems
- Vergleich des durch die Linse verzeichneten Bildes mit dem Original zur Ermittlung der Verzeichnungsparameter
- Abbildung des metrischen Koordinatensystems in ein Pixelkoordinatensystem

Ein entscheidender Nachteil dieser Vorgehensweise ist, dass alle 5 Faktoren des \mathbf{D} -Vektors initial bekannt sein müssen.

OpenCV nutzt deshalb eine Kombination aus verschiedenen eingeführten Verfahren. Zur Bestimmung der Kameraparameter f , b_u und b_v wird der Algorithmus von (Zhang, 2000) verwendet. Ähnlich wie beim Verfahren von Tsai wird als Kalibrieremuster eine Ebene verwendet (üblicherweise ein Schachbrettmuster, da es mathematisch leicht zu beschreiben ist und seine hohen Kontraste durch die üblichen Bildverarbeitungsalgorithmen leicht erkannt werden), welches allerdings in mindestens 10 verschiedenen – möglichst unterschiedliche - Orientierungen aufgenommen werden muss. Das Verfahren nutzt nun eine Homographie mit 8 Freiheitsgraden zur Beschreibung der jeweiligen Verhältnisse der aufgenommenen Bilder zur Kalibrierebene, welche mit einer reduzierten Variante des DLT-Algorithmus (Direct Linear Transformation, siehe (Dubrofsky, 2009)) bestimmt werden kann. Danach wird die Genauigkeit der ermittelten Parameter durch eine nichtlineare Optimierung erhöht.

Obwohl das Verfahren nach Zhang in der Lage ist, Linsenverzeichnungen zu modellieren, greift OpenCV in diesem Bereich das Modell von (Brown, 1971) zurück, welches die meisten Linsenverzeichnungen auf eine radiale und eine tangential Verzeichnung zurückführt. Zur Anwendung kommen ebenfalls verschiedene interne Algorithmen zur Detektion der Eckpunkte des Schachbrettmusters im Subpixelbereich, auf welche hier nicht weiter eingegangen werden soll.

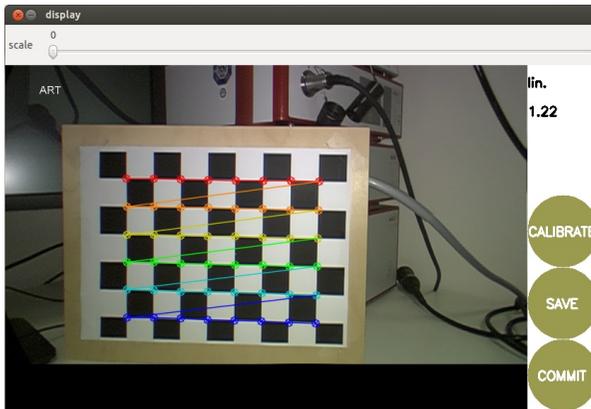


Abbildung 22: Kamerakalibrierung mit OpenCV

Durch eine Kalibrierung mit Hilfe der OpenCV-eigenen Tools konnten folgende Parameter für die verwendete EndoCam 5590 ermittelt werden:

```
# Camera intrinsics
[image]
width
1920
height
1080
[EndoCam 5590]
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
```

Tabelle 1: Kamera- und Linsenparameter der EndoCam 5590

In der resultierenden Projektionsmatrix P' wird bereits die Linsenverzeichnung berücksichtigt.

Verwendet wird ein Schachbrett mit 8×10 Feldern, welche je eine Kantenlänge von 2,5 cm aufweisen (s. Abbildung 22, Seite 27, oben). Die Anzahl der zur Kalibrierung nutzbaren Punkte ergibt sich somit zu $7 \times 9 = 63$.

In der Theorie reicht ein Schachbrett mit 3×3 Feldern, welches in mindestens zwei verschiedenen Posen aufgenommen wird. In der Praxis kommen Kalibriermuster der verwendeten Größe und in mindestens 10 unterschiedlichen Posen zur Anwendung, um Rauschen zu eliminieren und eine allgemein höhere Genauigkeit zu erzielen. Eine Detaillierte Erklärung liefert (Bradski & Kaehler, 2008).

Wie man den ermittelten Parametern $p_1 (= -0.00093)$ und $p_2 (= -0.00011)$ der Matrix D entnehmen kann, ist die tangentielle Verzeichnung nahezu zu vernachlässigen. Das ist der hohen Präzision bei der Fertigung der Kamera zu verdanken. Etwas überrascht haben die Koeffizienten q_{uc} ($=1028.25149$) und q_{vc} ($=483.48426$) der Kameramatrix K . Zu erwarten gewesen wären hier Werte, die in etwa der Hälfte der Kameraauflösung entsprechen, also $q_{uc} \approx 960$ und $q_{vc} \approx 540$. Es lässt sich vermuten, dass dieser Versatz aus den Toleranzen der komplizierten Linsenanordnung im Schaft des Endoskops resultiert. Diese Toleranzen sind notwendig, da die Linsen auch bei großen Temperaturschwankungen – zum Beispiel im Autoklaven – keinen Schaden nehmen dürfen. Außerdem ist die Verbindung zwischen Kameramodul und Endoskopschaft nicht besonders steif. Auf dieses Problem wird später noch im Detail eingegangen.

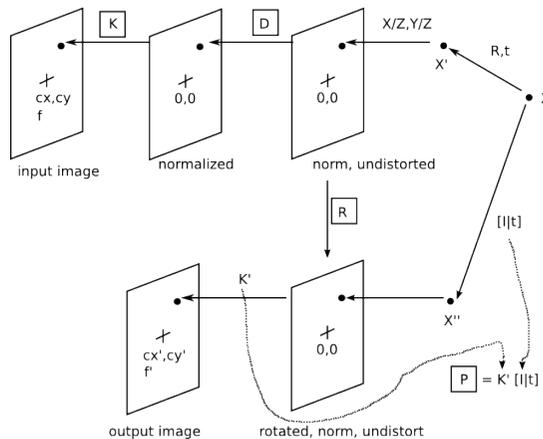


Abbildung 23: Abfolge der einzelnen Bildverarbeitungsschritte zum ausgerichteten, normalisierten und entzerrten Bild
 Entnommen ROS Wiki (http://wiki.ros.org/image_pipeline/CameraInfo, 04.03.2015), Creative Commons Attribution 3.0

Nach der Ermittlung der intrinsischen und extrinsischen Parameter wird das zuvor konvertierte Bild entzerrt. Die einzelnen Teilschritte der Bildaufbereitung sowie die dazu nötigen Parameter sind in *Abbildung 23* dargestellt.

Um diesen Vorgang effizient zu gestalten kommt wieder ein Multithreading-Ansatz zum Tragen; das Bild wird zeilenweise in Blöcke zusammengefasst und jeweils einem Thread zur Bearbeitung übergeben. Danach erfolgt der Versand des Bildmaterials via `ROS::image_transport`. Diese Bibliothek stellt neben dem Rohmaterial auch verschiedene komprimierte Datenkanäle zur Verfügung, welche bei der Übertragung an externe Recheneinheiten von Vorteil sein können. Im Projekt wurde auf die Verwendung von komprimiertem Material verzichtet, da der Datenstrom zum einen ausschließlich lokal weiterverarbeitet wurde und zum anderen möglichst unverfälschte Bilder für den Vermessungsalgorithmus zur Erreichung größtmöglicher Präzision nötig waren.

5.3.2 Einbindung des Trackingsystems

Die CamBar B2 verwendet im Versuchsaufbau zur Lokalisierung von Objekten im Messvolumen retroreflektierende Kugeln, so genannte „passive Marker“. In diesem Fall handelt es sich um die Variante SORT-P der Firma Atesos medical AG in Aarau (CH, <http://atesos.ch/>). Nach Angaben des Herstellers zeichnen sich diese Marker durch eine hohe Präzision aus, da die reflektierende Oberfläche von einer Kunststoffkuppel geschützt wird und somit weder Flüssigkeiten noch mechanische Beschädigungen die Reflektionen vermindern können.



Abbildung 24: SORT-P Targets der Firma Atesos medical AG, Bild: Hersteller

Das verwendete Trackingsystem ermittelt die Position der Marker im Raum und stellt sie dem Anwender über eine API zur Verfügung.

Zur Ermittlung von Posen medizinischer Objekte – also ihrer Position und Orientierung im Raum – werden mehrere Marker zu einer definierten Anordnung mit genau bekannter Geometrie zusammengefasst. Diese „Lokatoren“ oder auch „Rigid Bodies“ müssen verschiedenen Vorgaben

genügen, um ihre eindeutige Identifizierung zu ermöglichen. Um diese Vorgaben zu verstehen, muss zuerst die grundlegende Arbeitsweise eines photogrammetrisch arbeitenden Trackingsystems betrachtet werden.

Das 3D-Trackingsystem sendet Lichtsignale im infraroten Bereich (Wellenlänge 810nm) aus, welche von den Markern reflektiert werden. Diese Marker erscheinen als kreisförmige Körper auf den Bildern der beiden integrierten und synchronisierten Kameras. Nacheinander müssen nun zwei Probleme gelöst werden (Faugeras, 1995):

- Korrespondenzproblem: In beiden Kamerabildern müssen dieselben Elemente erkannt und einander zugeordnet werden. Gängige Verfahren zur Ermittlung von Punktpaaren sind das Zuordnen gleicher Intensitäten mittels Korrelation oder die Detektion von gleichen Merkmalen wie Kreisen, Rechtecken oder Linien. Eine interessante Einführung liefert (Tönnies, 2009).
- Rekonstruktionsproblem: Ermittlung der Raumkoordinaten der im ersten Schritt gefundenen korrespondierenden Markerpaare.

Das Lösung des ersten Problems - die Suche kann Punktpaaren in den beiden Bildern – kann durch Anwendung der Epipolargeometrie deutlich vereinfacht und beschleunigt werden. Die Grundlage für diese Betrachtungsweise lieferte bereits sehr früh (Hauck, 1883) – Verbreitung fanden seine Ergebnisse allerdings erst mit dem Aufkommen der digitalen Photographie etwa 100 Jahre später.

Für eine Einführung in die Epipolargeometrie müssen zuerst einige Begriffe definiert werden.

Basis für Abbildung 25 auf der folgenden Seite ist das Modell der Lochkamera (2 Stück) – für das bessere Verständnis wurden allerdings die Bildebenen vor das Projektionszentrum gespiegelt. Diese Vorgehensweise hat auf die rechnerische Betrachtung keinen Einfluss.

Die Projektion des Punktes X auf der Bildebene 1 (Bildpunkt X_L) lässt keinen eindeutigen Rückschluss auf die Position von X im Raum zu: Sämtliche mögliche Positionen von X befinden sich auf einer Geraden ausgehend vom Projektionszentrum der Kamera 1 durch den Bildpunkt X_L . Die Projektion dieser Geraden auf die Bildebene der Kamera 2 ergibt wiederum eine Gerade, die so genannte Epipolarlinie. Das Ganze gilt natürlich auch in umgekehrter Richtung: Der Projektion des Punktes X auf die Bildebene der Kamera 2 ($=X_R$) lässt sich eine Epipolarlinie im Bild 1 zuordnen.

Dieser Zusammenhang erleichtert das Auffinden korrespondierender Punktepaare X_L und X_R erheblich, da diese nicht mehr im gesamten Bild, sondern lediglich in den korrespondierenden Epipolarlinien gesucht werden müssen.

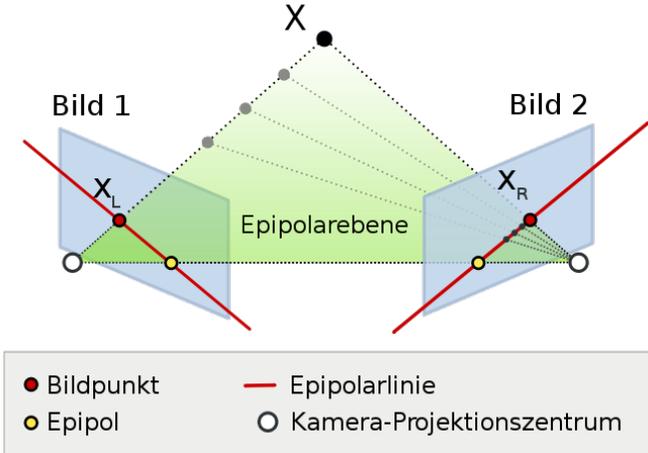


Abbildung 25: Begriffe der Epipolargeometrie, Autor:Arne Nordmann (CC BY-SA 3.0)

Sämtliche Epipolarlinien eines Bildes schneiden sich im so genannten Epipol. Der Epipol eines Bildes ist außerdem die Abbildung des Projektionszentrums der jeweils anderen Kamera. Je nach Anordnung der Kameras kann sich ein Epipol auch außerhalb des eigentlichen Bildausschnitts befinden. Die Ebene, welche die beiden Projektionszentren der Kameras und der aufgenommene Punkt X aufspannen, wird Epipolarebene genannt.

Für die Anwendung der Epipolargeometrie müssen die Positionen der Kameras zu einander nicht bekannt sein. Benötigt man relative Positionen können diese aus den Aufnahmen eines Referenzkörpers hergeleitet und das Stereokamerasystem dadurch kalibriert werden.

Den Zusammenhang zwischen dem Punkt $X_L=(x_L, y_L)^T$ und $X_R=(x_R, y_R)^T$ stellt die so genannte Epipolargleichung her. Hierbei greift man wieder auf die Darstellungsform der homogenen Koordinaten zurück (Faugeras, Luong, & Maybank, 1992):

$$\begin{pmatrix} x_R & y_R & 1 \end{pmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} = \begin{pmatrix} x_R & y_R & 1 \end{pmatrix} \mathbf{F} \begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} = 0 \quad (5.16)$$

\mathbf{F} wird hierbei als Fundamentalmatrix bezeichnet. Mit ihrer Hilfe kann zu einem Punkt X_L im linken Bild die dazugehörige Epipolarlinie l_R im rechten Bild berechnet werden. Das wird deutlich, wenn man in (5.16) den Term $\mathbf{F}X_L$ für einen fixen Punkt X_{LP} ausrechnet:

$$\begin{pmatrix} x_R & y_R & 1 \end{pmatrix} \begin{pmatrix} x_{LP}f_{11} + y_{LP}f_{12} + f_{13} \\ x_{LP}f_{21} + y_{LP}f_{22} + f_{23} \\ x_{LP}f_{31} + y_{LP}f_{32} + f_{33} \end{pmatrix} = \begin{pmatrix} x_R & y_R & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = 0 = ax_R + by_R + c \quad (5.17)$$

Die resultierende Geradengleichung (5.17) definiert die Epipolarlinie des Punktes X_L in der Bildebene der rechten Kamera und kann im nächsten Schritt dazu beitragen, die Elemente der Fundamentalmatrix \mathbf{F} zu ermitteln.

Im Prinzip lässt sich die Fundamentalmatrix bei bekannter Kalibrierung der beiden Kameras direkt aus beiden Projektionsmatrizen und einem Projektionszentrum berechnen. Stehen keine Kalibrierungsdaten zur Verfügung, kann der umgekehrte Weg beschritten werden: Aus einer Anzahl von Punktkorrespondenzen kann die Fundamentalmatrix berechnet werden. Die einzelnen Schritte werden im Folgenden erläutert.

Sämtliche Epipolarlinien aus (5.17) schneiden sich im Epipol (x_{er}, y_{er}) der rechten Bildebene. Als direkte Folgerung lässt sich daraus ableiten, dass (x_{er}, y_{er}) ein Element jeder Epipolarlinie des rechten Bildes ist und sich somit durch die Epipolargleichung Rückschlüsse auf die Fundamentalmatrix \mathbf{F} ziehen lassen:

$$\begin{pmatrix} x_L & y_L & 1 \end{pmatrix} \mathbf{F} \begin{pmatrix} x_{er} \\ y_{er} \\ 1 \end{pmatrix} = 0 \quad (5.18)$$

Da Gleichung (5.18) für beliebige Bildpunkte des linken Bildes gilt, lässt sich folgern, dass der Epipol (x_{er}, y_{er}) ein Eigenvektor der Fundamentalmatrix \mathbf{F} mit dem Eigenwert 0 ist.

$$\mathbf{F} \begin{pmatrix} x_{er} \\ y_{er} \\ 1 \end{pmatrix} = 0 \quad (5.19)$$

\mathbf{F} ist eine 3×3 Matrix, besitzt aber nur 7 Freiheitsgrade. Das liegt zum einen am bereits erwähnten Eigenwert von 0, zum anderen daran, dass \mathbf{F} beliebig skalierbar sein muss.

Zur Bestimmung der Fundamentalmatrix existieren verschiedene Ansätze. Die meisten Verfahren bestehen aus folgenden Schritten:

- Finden zusammengehöriger Punktepaare (z.B. durch SIFT)
- Häufig: Normalisierung der Eingangsbilder / waagerechte Ausrichtung der Epipolarlinien
- Aus jedem Punktepaar ergibt sich eine Bedingung für \mathbf{F}

In der Praxis hat sich in den letzten Jahren der so genannte *8-Punkt-Algorithmus* nach (Longuet-Higgins, 1981) durchgesetzt, obwohl er relativ anfällig gegen Messungenauigkeiten und falsche Korrespondenzen ist. Sein Vorteil liegt in der einfachen Implementierung und der daraus resultierenden Geschwindigkeit, welcher bei der hier verwendeten hohen Auflösung essentielle Bedeutung zukommt. Voraussetzung für seine Anwendung ist das Vorhandensein von mindestens 8 linear unabhängigen (also nicht auf einer Linie liegenden) Punktepaaren in der Darstellungsform der normierten homogenen Koordinaten.

Zur Bestimmung der Fundamentalmatrix aus der Menge der normierten homogenen Koordinaten \hat{x}_R und \hat{x}_L wird die Epipolargleichung ausmultipliziert:

$$\hat{\mathbf{x}}_R^T \mathbf{F} \hat{\mathbf{x}}_L = 0 \tag{5.20}$$

$$(\hat{x}_R \hat{x}_L f_1 + \hat{x}_R \hat{y}_L f_2 + \hat{x}_R f_3 + \hat{y}_R \hat{x}_L f_4 + \hat{y}_R \hat{y}_L f_5 + \hat{y}_R f_6 + \hat{x}_L f_7 + \hat{y}_L f_8 + f_9) = 0$$

oder in vektorieller Schreibweise

$$\begin{pmatrix} \hat{x}_R \hat{x}_L & \hat{x}_R \hat{y}_L & \hat{x}_R & \hat{y}_R \hat{x}_L & \hat{y}_R \hat{y}_L & \hat{y}_R & \hat{x}_L & \hat{y}_L & 1 \end{pmatrix} \cdot \mathbf{f} = \mathbf{A} \cdot \mathbf{f} = 0 \tag{5.21}$$

Aus 8 Punkten kann das folgende homogene Gleichungssystem aufgestellt werden (zweiter Index [i..n]=Punktnummer)

$$\mathbf{A} = \begin{bmatrix} \hat{x}_{R_1} \hat{x}_{L_1} & \hat{y}_{R_1} \hat{x}_{L_1} & \hat{x}_{L_1} & \hat{x}_{R_1} \hat{y}_{L_1} & \hat{y}_{R_1} \hat{y}_{L_1} & \hat{y}_{L_1} & \hat{x}_{R_1} & \hat{y}_{R_1} & 1 \\ \vdots & \vdots \\ \hat{x}_{R_n} \hat{x}_{L_n} & \hat{y}_{R_n} \hat{x}_{L_n} & \hat{x}_{L_n} & \hat{x}_{R_n} \hat{y}_{L_n} & \hat{y}_{R_n} \hat{y}_{L_n} & \hat{y}_{L_n} & \hat{x}_{R_n} & \hat{y}_{R_n} & 1 \end{bmatrix} \quad \text{und} \quad \mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_9 \end{bmatrix}$$

Berechnet man die Fundamentalmatrix

$$\hat{\mathbf{F}} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \tag{5.22}$$

jetzt mit der Methode der kleinsten Fehlerquadrate, fällt auf, dass in den meisten Fällen $\hat{\mathbf{F}}$ nicht wie gefordert singular wird. Das liegt im Allgemeinen an den Messfehlern der ermittelten Punktepaare. Besitzt $\hat{\mathbf{F}}$ aber den vollen Rang, schneiden sich die Epipolarlinien nicht. Als Lösungsansatz kann $\hat{\mathbf{F}}$ durch eine Matrix $\hat{\mathbf{F}}'$ ersetzt werden, die die Singularitätsbedingung erfüllt und $\hat{\mathbf{F}}$ möglichst ähnlich ist. Vorschläge für die schnelle Ermittlung von $\hat{\mathbf{F}}'$ kommen unter anderem von (Tsai & Huang, 1984), auf Details wird an dieser Stelle verzichtet.

Eventuell auftretende Fehlzuordnungen müssen vor der Verwendung des 8-Punkte-Algorithmus mittels eines geeigneten Verfahrens zur Identifizierung von Ausreißern eliminiert werden. Zu diesem Zweck bietet sich der häufig verwendete RANSAC-Algorithmus an.

RANSAC (Random Sample Consensus), (Fischler & Bolles, 1981) ist ein robuster und effizienter Algorithmus zur Filterung von Ausreißern aus automatischen Messungen, welcher häufig in der Umgebung des maschinellen Sehens zur Anwendung kommt. Voraussetzung für seine Anwendung ist, dass mehr Datenpunkte vorliegen, als zur Bestimmung der Modellparameter notwendig sind.

Basierend auf diesen Messdaten durchläuft der Algorithmus folgende Schritte:

1. Zufällige Auswahl von exakt so vielen Datenpunkten, wie zur Berechnung der Modellparameter nötig sind.
2. Ermittlung der Modellparameter basierend auf den ausgewählten Datenpunkten
3. Bestimmung der Teilmenge der Datenpunkte (Consensus Set), deren Abstand zur Modellkurve kleiner ist als ein festzulegender Grenzwert. Enthält das Consensus Set eine bestimmte Menge an Elementen, wird davon ausgegangen, dass bei den ausgewählten Datenpunkten bestenfalls wenige Ausreißer vorhanden waren und dass es sich bei den ermittelten Modellparametern um gute Werte handelt.
4. Wiederholung von Schritt 1-3

Die Schritte 1-4 werden eine vorher festzulegende Anzahl mal durchlaufen. Im Anschluss wird diejenige Teilmenge gewählt, welche die meisten Punkte enthält.

Kann die Höhe des Ausreißeranteils im Vorfeld abgeschätzt werden, kann der Algorithmus auch nach Schritt 3 vorzeitig abgebrochen werden, wenn das Consensus Set eine akzeptable Größe erreicht hat. Diese Variante des RANSAC-Algorithmus wird als „präemptives RANSAC“ bezeichnet.

Im vorliegenden Fall – also zur Filterung der Ausreißer zur Bestimmung der Fundamentalmatrix - heißt das konkret:

1. Zufällige Auswahl von 8 Datenpunkten
2. Bestimmung von \mathbf{F} aus den gewählten Korrespondenzen mittels des 8-Punkt-Algorithmus
3. Bestimmung der Datenpunkte, für die gilt $\tilde{\mathbf{x}}_R^T \mathbf{F} \tilde{\mathbf{x}}_L \leq \varepsilon$. Der Schwellwert ε ist notwendig, da auf Grund von Rechenungenauigkeiten die Epipolargleichung quasi nie exakt erfüllt werden kann.
4. Wiederholung von Schritt 1–3, bis die zufällig ausgewählten Datenpunkte mit ausreichender Wahrscheinlichkeit keine Fehler enthalten.

Anschließend wird die Fundamentalmatrix \mathbf{F} unter Verwendung des Consensus Sets mit der größten Menge an Datenpunkten und des 8-Punkt-Algorithmus bestimmt. Gegebenenfalls kann darauf hin eine weitere Korrespondenzanalyse durchgeführt werden, bei der die berechnete Fundamentalmatrix zum Einsatz kommt und ein niedrigerer Schwellwert Anwendung findet.

Diese letzten beiden Schritte können iterativ wiederholt werden, bis die Zahl der korrespondierenden Bildpunkte stabil ist.

Ist die Fundamentalmatrix bekannt, ermöglicht sie die Rektifizierung der beiden Ausgangsbilder. Die Findung weiterer Punktpaare durch Intensitätsschwankungen entlang der Epipolarlinien reduziert sich dadurch auf den Vergleich von Intensitätsschwankungen entlang der Bildschirmzeilen, wie in Abbildung 26 zu sehen ist.

Die Lösung des zweiten Hauptproblems der Photogrammetrie - der Ermittlung der Raumkoordinaten der im ersten Schritt gefundenen korrespondierenden Markerpaare – erfolgt durch Rückrechnung mit Hilfe der Fundamentalmatrix in homogenen Koordinaten.

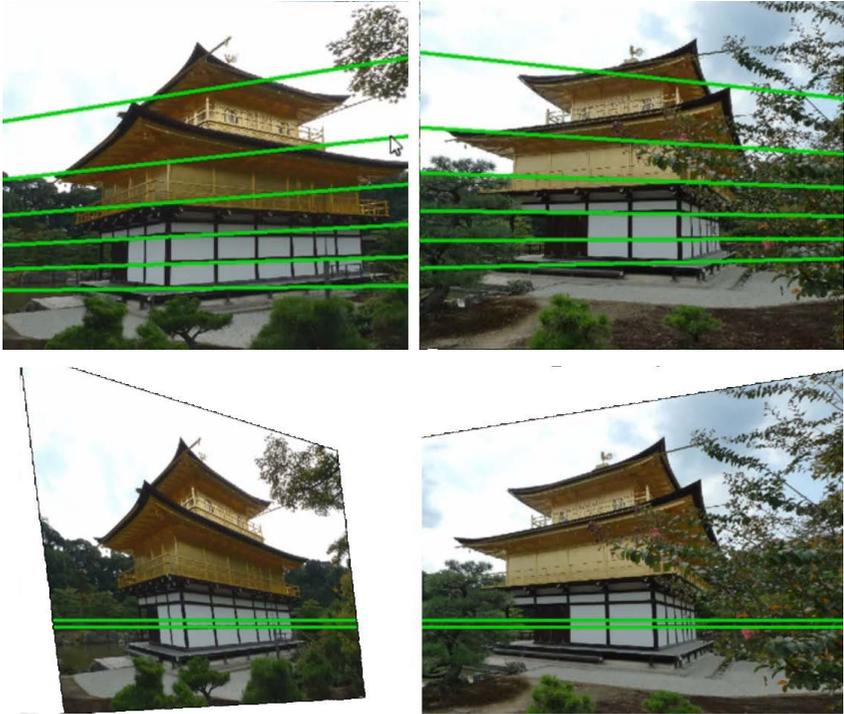


Abbildung 26; Oben: unrektilifizierte Ausgangsbilder mit Epipolarlinien, Unten: Bildpaar nach der Rektifizierung, Autor: Rich Radke (CC BY-SA 3.0)

Die voran erläuterten Schritte dienen der Kalibrierung des Stereokamerasystems sowie der Ermittlung von 3D-Koordinaten der sich im Bildbereich befindlichen retroreflektierenden Marker. Zur Berechnung der Posen der sich im Arbeitsraum befindenden Objekte müssen dem Tracking System so genannte Lokatoren angelehrt werden, welche dann über geeignete Matching-Verfahren in der Punktwolke der detektierten Marker identifiziert werden können.

Die Lokatoren müssen verschiedenen Mindestanforderungen gerecht werden, um vom Trackingsystem zuverlässig erkannt zu werden. Zur Erreichung der maximalen Präzision sind zusätzliche Anforderungen zu stellen.

Laut Hersteller müssen/sollen Lokatoren folgende Parameter einhalten:

	Mindestanforderung (die angegebene Messgenauigkeit kann ggf. nicht eingehalten werden)	Empfehlung
Mindestanzahl von Marken	3	>= 4
Form und Oberfläche der Marken	Kugel, Kreis (retroreflektierend)	Kugel (retroreflektierend)
Mindestabstand zwischen Marken	Zweifacher Minimal-Durchmesser der Marken (siehe „Technische Beschreibung“)	> Dreifacher Minimal-Durchmesser der Marken
Streckendifferenz zwischen Marken zu anderen Strecken im Lokator	> 1 mm	> 2 mm
Geometrische Bedingungen	- Marken dürfen nicht auf einer Geraden liegen	- Marken mit möglichst homogener räumlicher Anordnung

Tabelle 2: Anforderungen an Lokatoren, Quelle: AXIOS 3D Services GmbH

Außerdem gilt nach (AXIOS 3D® Services GmbH, 2012):

- Ist die Anordnung der Punkte eines Rigid Bodys nahe einer Geraden, so verschlechtert sich die ermittelte Rotationsinformation. Liegen alle zu messenden Punkte auf einer Geraden, ist demnach keine Rotationsinformation um die Längsachse erhältlich.
- Je größer und homogener die räumliche Ausdehnung der Punkte eines Rigid Bodys ist, desto gleichmäßiger ist die Genauigkeit bei der Ermittlung der Positionsinformationen.

Um den Fehler des Trackingsystems möglichst gering zu halten, sollten nicht nur die Mindestanforderungen eingehalten werden; dem Design der Lokatoren ist besondere Aufmerksamkeit zu widmen.

Nach (Fitzpatrick, West, & Maurer, 1998) gibt es bei sämtlichen optischen Trackingsystemen drei unterschiedliche durch das Verfahren an sich begründete Fehlerarten:

- FLE: Fiducial Localization Error
Der FLE beschreibt die Abweichung, die bei der Lokalisierung eines einzelnen Markers gemacht wird, und ist die Distanz zwischen tatsächlicher und gemessener Position des Markers

$$FLE = \sqrt{3}\sigma \quad (5.23)$$

σ = Standardabweichung

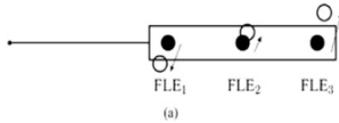


Abbildung 27: FLE (Fiducial Localization Error), (Fitzpatrick, West, & Maurer, 1998)

- FRE: Fiducial Registration Error

Lässt sich die gemessene Geometrie des Lokators nicht exakt mit der gespeicherten in Deckung bringen – zum Beispiel durch Messfehler beim Anlernen oder durch Ungenauigkeiten bei der Produktion – wird bei der Registrierung des Lokators ein weiterer Fehler gemacht. Dieser FRE genannte Fehler entspricht dem quadratischen Mittelwert des Fehlers der Referenzmarken zwischen Bildraum und korrektem Wert im physikalischen Raum. Er wirkt sich auf Position und Orientierung der gemessenen Lokatorposition aus.

$$\langle FRE \rangle^2 = \left(1 - \frac{2}{N}\right) \langle FLE \rangle^2 \quad (5.24)$$

N = Anzahl der Marker

$\langle FLE \rangle$ = Erwartungswert des Fehlers eines Einzelmarkers

$\langle FRE \rangle$ = Erwartungswert des „Fiducial Registration Errors“, also des Fehlers nach der Registrierung des Lokators

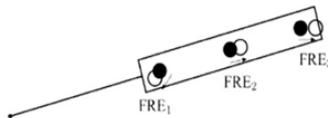


Abbildung 28: FRE (Fiducial Registration Error), (Fitzpatrick, West, & Maurer, 1998)

- TRE: Target Registration Error

Der TRE bezieht sich ähnlich dem FRE auf die Abweichung von realer und gemessener Pose des Lokators. Allerdings wird der Fehler hier nicht relativ zum Schwerpunkt der Markerwolke, sondern bezogen auf das definierte Koordinatensystem der Lokatorgeometrie angegeben. Mit steigender Entfernung des Ursprungs dieses Koordinatensystems zum Schwerpunkt der dazugehörigen Marker vergrößert sich der TRE.

$$\langle TRE \rangle_{ges}^2 = \underbrace{\frac{\langle FLE \rangle^2}{N}}_{\text{translatorischer Anteil}} + \underbrace{\frac{\langle FLE \rangle^2}{3N} \sum_{k=1}^3 \frac{d_k^2}{f_k^2}}_{\text{rotatorischer Anteil}} = \frac{\langle FLE \rangle^2}{N} \left(1 + \frac{1}{3} \sum_{k=1}^3 \frac{d_k^2}{f_k^2} \right) \quad (5.25)$$

N = Anzahl der Marker

d = Distanz vom Ursprung zum Zentrum des Lokators

f = Distanz der Marker zu den Ursprungsachsen des Lokators

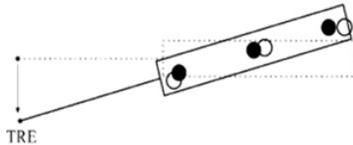


Abbildung 29: TRE (Target Registration Error), (Fitzpatrick, West, & Maurer, 1998)

(Adam, 2009) hat in seiner Arbeit den Einfluss der Referenzgebergeometrie und der Markereigenschaften auf die statische Genauigkeit optischer Navigationssysteme untersucht und ist dabei zu folgenden Erkenntnissen gekommen:

- Bei der Gestaltung eines Rigid Body für optische Lokalisiersysteme ist es von Vorteil, sich an einem gleichseitigen Dreieck bzw. einem Tetraeder zu orientieren, da so die höchsten Genauigkeiten in Hinsicht auf das Orientierungsrauschen in den gemessenen Posen realisierbar sind.
- Der Rigid Body sollte dabei ein Fläche-Umfang Verhältnis von $\frac{A}{U} > 1,5 \frac{\text{cm}^2}{\text{cm}}$ besitzen.
- Das Positionsrauschen lässt sich durch einen vierten Marker erheblich reduzieren. Zusätzlich bringt ein vierter Marker den Vorteil, dass der Rigid Body robuster gegenüber Teilverdeckungen wird. Jeder weitere Marker hat jedoch keine nennenswerte Auswirkung mehr auf die Genauigkeit. Es wird lediglich weitere Redundanz gegenüber Verdeckungen hinzugefügt, was jedoch zu Lasten der Systemtzeit geht.
- Um die Robustheit eines Rigid Body unter Neigungswinkeln zu erhöhen, sollte mit dem vierten Marker nach Möglichkeit ein Tetraeder aufgespannt werden.

Neben diesen grundlegenden Vorgaben zum optimalen Lokator-Design können auch Umgebungsbedingungen wie Art und Stärke der Beleuchtung, Stärke und Homogenität der vom Marker reflektierten Strahlung etc. Auswirkungen auf die Lokalisiergenauigkeit haben; Untersuchungen speziell für retroreflektierende Marker scheinen noch nicht durchgeführt worden zu sein.

Ebenfalls Auswirkungen auf die erreichbare Genauigkeit hat die Position des Lokators im Messvolumen des Kamerasystems. (Pieck, 2007) und (Wiles, Thompson, & Frantz, 2004) haben in ihren Arbeiten das dynamische Fehlverhalten innerhalb des Messvolumens untersucht und ein stark anisotropes Verhalten ihrer Tracking Systeme speziell in z-Richtung festgestellt. In Ebenen parallel zur Stereokamera ist der Fehler relativ homogen, nimmt aber mit steigendem Abstand dieser Ebenen zur Kamera zu. Dieser Vorgang ist auch bei dem von uns verwendeten System der Firma Axios zu vermuten, da er systembedingt ist. Als Resultat wird im Versuchsaufbau darauf geachtet, dass sich bei sämtlichen Messungen die Lokatoren im kameranahen Bereich des Messvolumens befinden.

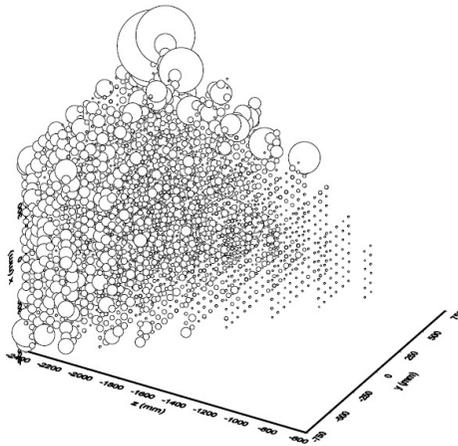


Abbildung 30: Anisotroper Verlauf des dynamischen Fehlers, (Wiles, Thompson, & Frantz, 2004)

Basierend auf den zuvor genannten Erkenntnissen wurden im Rahmen dieser Arbeit drei Lokatoren entworfen und angefertigt. In einigen Punkten musste dabei leider vom theoretisch optimalen Design abgewichen werden, um die für die eigenen Zwecke nötige Funktionalität zu erreichen.

1. Lokator zur Feststellung der Kamera-Pose

Der am Endoskop zu befestigende Lokator erfüllt zwei wesentliche Aufgaben:

- Er dient der Ermittlung des Bezugskoordinatensystems des ermittelten Polygonzuges, der die Zentrumslinie des Stiff-Flop Armes beschreibt. Der Lokator definiert sich über insgesamt sieben Marker, welche auf Höhe der Optik in unterschiedlichen Abständen und auf unterschiedlichen Ebenen um die z-Achse der Endoskopkamera angeordnet sind. Diese hohe Anzahl der Marker kommt zum Einsatz, da speziell beim Endoskop häufig mit einer Teilverdeckung der Marker zu rechnen ist. Aus diesem Grund wurde mehrfache Redundanz geschaffen.
- Er sorgt für eine sichere Arretierung und Versteifung des Schnellverschlusses zwischen Kameraoptik und Endoskopschaft. Dieser Schnellverschluss weist leider nicht die notwendige Steifigkeit auf, um den relativ schweren Endoskopschaft starr mit der Kameraoptik zu verbinden. Dreht man zum Beispiel das Endoskop um seine Längsachse, verschiebt sich das Kamerabild zusätzlich zur Rotation. Einer vermessenen Position des Stiff-Flop Armes im Kamerabild lässt sich dadurch keiner eindeutigen Position im Raum zuordnen. Aus diesem Grund wurden sowohl Optik als auch Schaft umfasst und gegeneinander verspannt. Als Material kam hochfestes Aluminium zum Einsatz, da es sowohl verformungsstabil als auch extrem leicht ist. Ein Nachteil ist die relativ niedrige Bruchzähigkeit, was hier jedoch nicht zum Tragen kommt.

Abbildung 31 zeigt sowohl den Konstruktionsentwurf in Autodesk Inventor 2015 als auch den gefertigten und am Endoskop montierten Lokator:



Abbildung 31: Lokator zur Feststellung der Kamera-Pose

2. Lokator zur Feststellung der Pose des Stiff-Flop Armes / der Manipulatorpose zu Zwecken der Kalibrierung

Abbildung 32 zeigt die am Roboterflansch montierbare Aufnahme für den Stiff-Flop Arm. Grundplatte sowie die diversen Sockel wurden aus Aluminium gefertigt, um das Gewicht gering zu halten. Da keine Informationen zur Anfahrergenauigkeit des Manipulators vorliegen, wurden ebenfalls Marker in Form eines dreidimensionalen Simplex (ungleichseitiger Tetraeder) angebracht, um eine Abschätzung der Fehlergrenzen vornehmen zu können.



Abbildung 32: Lokator zur Feststellung der Manipulatorpose

Um ungewollte Reflexionen der vom Trackingsystem ausgesandten Infrarotstrahlung zu vermeiden, wurde das glänzende Aluminium der Trägerplatte flächig mit einer Velourfolie überzogen.

Obwohl durch die ausgesprochene Leichtbauweise und die hohe Traglast des Manipulators (16kg direkt am Flansch) keinerlei Probleme für den Roboter zu erwarten sind, wurde der Vollständigkeit halber eine Lastanalyse mittels KUKA.LOAD durchgeführt. KUKA.LOAD „ist ein Werkzeug zur Auslegung der Last an einem KUKA Roboter bzw. zur Auswahl des geeigneten Roboters bei einer speziellen Last“ (KUKA Roboter GmbH). Verwendet wurde Version 5.0. Die Lastdaten wurden mit Hilfe der Konstruktionssoftware ermittelt.

Sowohl die statische als auch die dynamische Auslastung der einzelnen Achsen liegt (wie erwartet) weit unterhalb der jeweiligen Maxima. Detaillierte Ergebnisse der Lastanalyse finden sich in Abbildung 33.

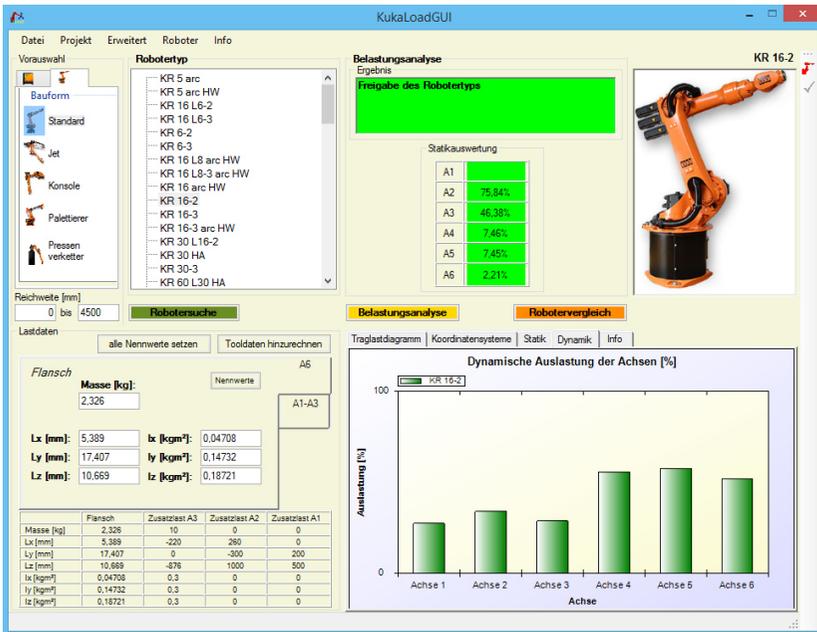


Abbildung 33: Lastanalyse des Roboterwerkzeugs mit Kuka Load 5.0

3. Lokator für die Definition des Basiskoordinatensystems des Industrieroboters

Um die Koordinatensysteme des Industrieroboters und des Trackingsystems ineinander überführen zu können, musste eine gemeinsame Basis geschaffen werden. Zu diesem Zweck wurde eine solide Aluminiumplatte sowohl mit optischen Markern als auch Aluminiumstiften versehen, welche vom Roboter eingelesen und als Werkstückkoordinatensystem verwendet werden können. Die Maße dieser Basisplatte wurden großzügig gewählt, da zum einen Gewicht in diesem Fall keine Rolle spielt und sich größere Abstände sowohl der Lokatoren als auch der Aluminiumstifte zueinander positiv auf die Genauigkeit auswirken. Für die Anordnung der optischen Marker wurde wie bei den anderen Lokatoren auch ein dreidimensionales Simplex gewählt, um den Körper robuster gegen Verschattung zu machen und den Target Registration Error zu minimieren. Auch dieser Lokator wurde wieder flächig mit Velourfolie überzogen, um störende Reflektionen der Infrarotstrahlung des Trackingsystems zu vermeiden. Das Resultat zeigt Abbildung 34.

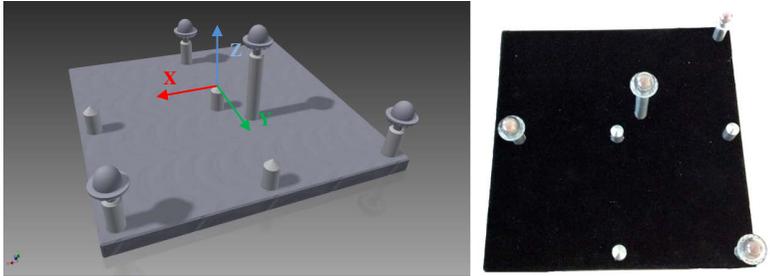


Abbildung 34: Lokator zur Definition des Basiskoordinatensystems des Industrieroboters

```

<locatordefinitions version="1" >
  <locator id="example_locator" >
    <parameter>
      <min_targets>3</min_targets>
      <matching_threshold>1.0</matching_threshold>
    </parameter>

    <targets>
      <target id="1" >
        <position>
          <x>-47.0</x>
          <y>63.95</y>
          <z>10.30</z>
        </position>
      </target>

      <target id="2" >
        <position>
          <x>45.0</x>
          <y>65.95</y>
          <z>10.3</z>
        </position>
      </target>

      <target id="3" >
        <position>
          <x>3.0</x>
          <y>111.95</y>
          <z>50.3</z>
        </position>
      </target>
    </targets>
  </locator>
</locatordefinitions>

```

Tabelle 3: Beispieldefinition eines Lokators

Nach dem Design und der Fertigung der Lokatoren müssen diese dem Trackingsystem bekannt gemacht werden. Der Kamera liegt zu diesem Zweck ein Programm bei, dessen Anwendung im Detail in (AXIOS 3D® Services GmbH, 2012) beschrieben ist. Diese Software erstellt eine statische Aufnahme aller im Sichtbereich der Kamera verfügbaren Marker. Im Anschluss wird der Anwender gebeten, sämtliche Marker auszuwählen, die zu dem neu zu definierenden Lokator gehören. Dessen Ursprung liegt im ersten ausgewählten Marker. Die x -Achse läuft vom ersten Punkt durch den zweiten; der dritte Punkt spannt die x - y -Ebene auf. Für unseren Anwendungsfall ist es jedoch von Vorteil, den Ursprung des Lokatorkoordinatensystems an die jeweilige Aufgabe angepasst außerhalb eines Markers zu platzieren. Zu diesem Zweck wurden die generierten XML-Lokatordefinitionen angepasst und mit Werten aus den Konstruktionsplänen gefüllt. Tabelle 3 zeigt beispielhaft den Aufbau einer Lokatordefinition, Details sind in (AXIOS 3D® Services GmbH, 2012) nachzulesen.

Um die Daten des Trackingsystems in der ROS-Umgebung zur Verfügung zu stellen, wurde eine ROS-Node entwickelt, die auf der Basis der von der Firma Axios zur Verfügung gestellten Metrology-Bibliothek Marker in den Bildern des Trackingsystems erfasst und die Posen von vorher definierten Lokatoren ermittelt und publiziert.

Im ersten Schritt meldet sich die Node beim ROS Master an. Im Anschluss werden die Publisher initialisiert sowie die aktuell zu verwendenden Parameter vom ROS Parameterserver abgerufen. Über diese Parameter kann festgelegt werden, ob die ermittelten Markerpositionen in Form einer Punktwolke (ROS::PointCloud) publiziert werden sollen oder ob sich der Treiber zur Verringerung der Systemlast auf die Posen der Lokatoren beschränken soll (`_pointcloud:=true/false`). Ist der Aufbau eines TF-Trees auf der Basis der Lokatorposen gewünscht, kann dies ebenfalls unter Bekanntgabe des zu verwendenden Basisknotens erreicht werden.

Der nächste Schritt ist das Einlesen der Lokatordefinitionen und die Initialisierung sämtlicher Publisher der Node. Jeder in der Definitionsdatei hinterlegte Lokator erhält sein eigens Topic mit dem dazugehörigen Publisher.

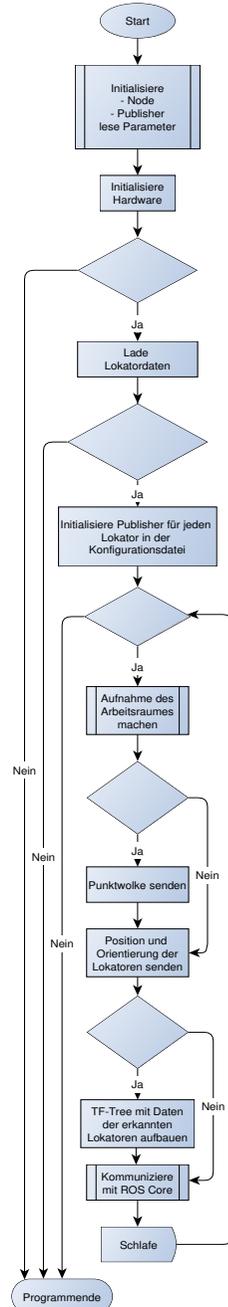


Abbildung 35: Programmablaufplan der CamBar-Node

In einer Endlosschleife werden jetzt Aufnahmen vom Arbeitsraum gemacht, von der Metrology-Bibliothek ausgewertet und die erhaltenen Informationen gemäß den Vorgaben der beim Start übergebenen Parameter weiterverarbeitet.

Zuerst wird die Markerpunktwolke – wenn gewünscht – publiziert. Danach folgen die Posen der Lokatoren auf ihren jeweiligen Topics. Wurde der Node beim Start der Name eines Lokators mitgeteilt (`_root:={Lokatorname}`), wird ein TF-Tree mit dem genannten Lokator als Wurzel und den restlichen Lokatoren als Blätter aufgebaut.

5.3.3 Einbindung der Datenbank MariaDB

Die gewünschte Reproduzierbarkeit der Vermessungsergebnisse sowie eine gewünschte Offline-Auswertbarkeit und –Optimierbarkeit der ermittelten umfangreichen Parameter und Daten erfordert ein Konzept der effizienten, adaptierbaren und vor allem transparenten Datenhaltung. Aus diesem Grund und der Problematik, dass der Umfang der anfallenden Daten während der Implementierungsphase nicht abzuschätzen war, wurde auf die freie Datenbank MariaDB zurückgegriffen. Die Anbindung erfolgt über eine eigene ROS-Node, die eine beliebige Anzahl unterschiedlicher Topics abonniert und den Inhalt der erhaltenen Messages in einer oder mehrerer festzulegender Datenstrukturen der Datenbank ablegt.

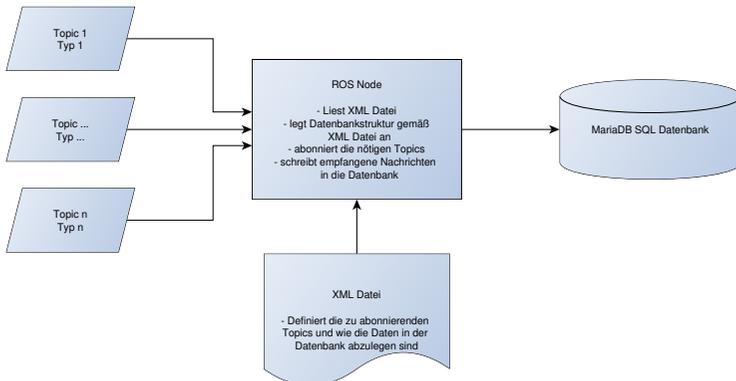


Abbildung 36: Aufbau der ROS Datenbank-Node

Die Konfiguration der Node erfolgt über eine in Tabelle 4 dargestellte XML-Datei. Hier werden die Verbindungsparameter zum Datenbankserver festgelegt sowie gestaffelt nach Datenbank, Tabelle und Tabellenspalte der abzulegende Parameter definiert, wobei auf die korrekte Angabe des Datentyps zu achten ist. Eine Plausibilitätsabfrage erfolgt auf dieser Ebene nicht.

Eine Einspeisung von Datenbankinhalten in das ROS-System ist zum aktuellen Zeitpunkt nicht vorgesehen, da die weiterverarbeitenden Systembestandteile (wie die Parallax-Visualisierung) über eigene Datenbankanbindungen via ODBC verfügen.

```
<Database_Connection_Settings IP="%%" user="%%" password="%%">
  <database name="%%">
    <table name="Tabelle 1">
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 1</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 2</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 3</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 4</column>
      ...
    </table>
    <table name="Tabelle 2">
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 1</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 2</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 3</column>
      <column datatype="%%" rostopic="%%" roselement="%%">Spaltenname 4</column>
      ...
    </table>
    ...
  </database>
  ...
</Database_Settings>
```

Tabelle 4: Aufbau der Konfigurationsdatei der ROS Datenbank-Node

5.3.4 Einbindung des Roboters KUKA KR 16-2

Der verwendete Industrieroboter KUKA KR 16-2 wird in der ROS-Umgebung in Form einer Treiber-Node abgebildet, welche Services zur Steuerung und zum Auslesen der Roboterparameter zur Verfügung stellt. Der Aufruf eines Services entspricht der Ausführung eines Befehls durch die Robotersteuerung. Benötigt werden Fahrbefehle (LIN und PTP) sowie Abfragemöglichkeiten zur Ermittlung der aktuellen TCP-Position sowie der Gelenkstellungen des Manipulators. Zusätzlich erscheinen Services zur Limitierung der Bahn- und Gelenkgeschwindigkeiten sinnvoll.

Zur Erreichung dieses Zieles existieren prinzipiell mehrere Vorgehensweisen, von denen sich leider die meisten zum Zeitpunkt dieser Arbeit (2015) als impraktikabel erwiesen haben. Der Vollständigkeit halber werden sie hier kurz vorgestellt:

- Verwendung von ROS Industrial

ROS Industrial ist ein Open-Source Projekt, welches die weitreichenden Fähigkeiten von ROS auf Fertigungsstraßen in der Industrie ausweitet, in dem es Bibliotheken, Werkzeuge und Treiber für industriell verwendete Hardware – wie zum Beispiel Handhabungsroboter -

zur Verfügung stellt. Dabei wird die ursprüngliche Robotersteuerung mit allen ihren systemnahen Sicherheitsmechanismen nicht ersetzt, sondern um die high-level Funktionalität von ROS ergänzt. Durch die Nutzung der vorhandenen standardisierten Schnittstellen wird die plattformunabhängige Entwicklung von übergeordneten Algorithmen, wie zum Beispiel Bahnplanern, Ablaufsteuerungen oder Kollisionsvermeidungsroutinen, erleichtert.

Abbildung 37 zeigt einen Überblick über die verschiedenen Schichten von ROS Industrial. Wie man sieht, gliedern sich die neuen Bestandteile nahtlos in die bestehende ROS-Architektur ein. Sämtliche Module funktionieren hardwareunabhängig und lassen sich auf Grund ihrer transparenten Strukturen einfach erweitern oder austauschen. Einzige Ausnahme bildet die unterste Schicht, der so genannte „Controller Layer“. Dieser kommt einem spezifischen Treiber gleich und ist somit herstellerabhängig.

Die Firma KUKA hat bereits seit längerer Zeit einen Controller Layer für ihre Manipulatoren angekündigt, allerdings ist dieser auch nach mehreren Jahren der Entwicklung noch nicht für die hier eingesetzte Robotersteuerung verfügbar. Eine Eigenentwicklung wäre viel zu umfangreich, da neben der reinen Bahnplanung auch alle Sicherheitsmechanismen implementiert werden müssen. Dieser an sich besonders elegante Weg der Anbindung des KR 16-2 an ROS ist momentan also noch nicht gangbar.

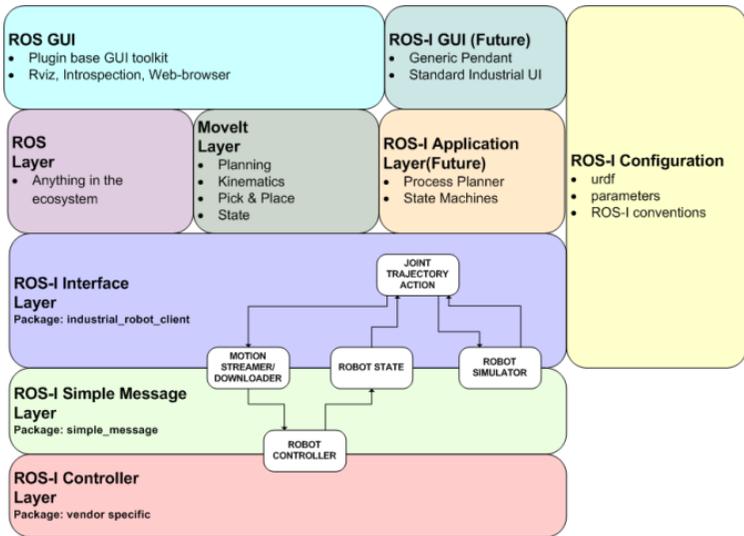


Abbildung 37: Darstellung des ROS-Industrial Schichtenmodells (GvdHoon, 2015)

- KUKA OPC Technologiepaket

Die OPC-Technologie (OLE for Process Control) ist ein offener Schnittstellenstandard, der einen einfachen Datenaustausch zwischen den Geräten im Fertigungsbereich und den PC-Applikationen zur Überwachung bzw. Planung ermöglicht. KUKA stellt einen OPC-Server in Form eines separat zu erwerbenden Technologiepaketes für die Steuerungseinheit KR C2 zur Verfügung, welcher den externen Zugriff auf System- und Nutzervariablen der Steuerung via Ethernet ermöglicht und somit weitreichende Fernsteuerungsmöglichkeiten bietet. Leider liegen die Kosten für diese Option im unteren fünfstelligen Bereich – aus diesem Grund wurde von einem Einsatz abgesehen.

- Einfache Kommunikation über die serielle Schnittstelle

Eine einfache Möglichkeit der Kommunikation, die sich mit geringem Aufwand und ohne zusätzliche Kosten realisieren lässt, ist der Einsatz der seriellen Schnittstelle des Steuerungs-PCs der Robotersteuerung. Durch geeignete Konfiguration lässt sich auf diese Schnittstelle auch aus der Echtzeitumgebung der Ablaufsteuerung zugreifen, Details dazu liefert das Handbuch des KR16 (KUKA Robot Group, 2007). Eine Fernsteuerung des Roboters im klassischen Sinn ist auf diesem Weg nicht zu erreichen, allerdings können vordefinierte Programmblöcke abhängig von Parametern aus der seriellen Kommunikation durchlaufen werden. Da im vorliegenden Einsatzfall nur wenig Funktionalität abgebildet und in ROS zur Verfügung gestellt werden muss, erscheint dieser Weg als ausreichend flexibel und gut geeignet, um mit vertretbarem Aufwand ein Maximum an Nutzen zu erreichen.

Ein weiterer Vorteil liegt darin, dass die ursprüngliche Robotersteuerung (Ablaufsteuerung, Bahnplanung etc.) nicht verändert werden muss und somit die vorhandenen Sicherheitsmechanismen intakt bleiben.

Als Nachteil konnte die relativ hohe und nicht determinierte Übermittlungsdauer der Kommandos von der ROS Node zur KUKA Ablaufsteuerung identifiziert werden. Diese resultiert daraus, dass auf Seiten der Ablaufsteuerung die Daten der Schnittstelle erst vom Windows-Teil des Rechners entgegen genommen werden und dann erst in festen Intervallen zum Echtzeitteil weitergeleitet werden. Die dadurch vereinzelt auftretenden hohen Latenzzeiten können jedoch bei dem hier angedachten Einsatzzweck vernachlässigt werden.

Um die Kommunikationsstruktur sowie die daraus resultierenden Probleme zu verstehen, muss man zuerst einen Blick auf den Aufbau der Robotersteuerung werfen. Der einzelnen Elemente sowie der mechanische Aufbau sind in Abbildung 38 dargestellt:

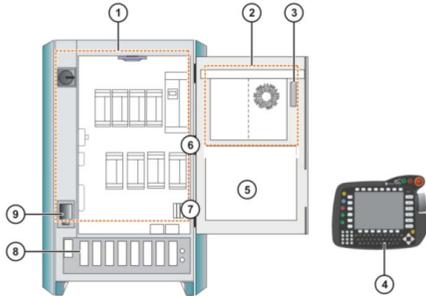


Abbildung 38: Aufbau der Robotersteuerung KR C2 (KUKA Roboter GmbH, 2010)

1. Leistungsteil
2. Steuerungs-PC (Windows XP & VxWorks als Gastsystem)
3. KCP-Koppler Bedien- und Anzeigeelemente (Option)
4. KCP (KUKA Control Panel, Handbediengerät)
5. Kundeneinbauraum
6. Sicherheitslogik (ESC)
7. KCP-Koppler-Karte (Option)
8. Anschlussfeld
9. Servicesteckdose (Option)

Die Schnittstelle zwischen Leistungselektronik und Anwender stellt ein Industrie-PC mit den Betriebssystemen Windows XP sowie VxWorks als Gastsystem dar. Sämtliche zeitkritischen Funktionen, wie die Steuerung der Roboterachsen, Bahnplanungsalgorithmen sowie die Ablaufsteuerung, laufen unter dem Echtzeitbetriebssystem VxWorks, da nur hier eine determinierte Antwortzeit des PCs auf Anfragen des Leistungsteils sowie der Sicherheitskomponenten gewährleistet ist. Die Kommunikation mit dem Anwender übernimmt das Hostsystem (Windows XP).

Eine komplette Übersicht der Aufgaben des PCs kann (KUKA Roboter GmbH, 2010) entnommen werden:

Der PC übernimmt mit seinen gesteckten Komponenten alle Funktionen der Robotersteuerung.

- Windows-Bedienoberfläche mit Visualisierung und Eingabe
- Programmerstellung, -Korrektur, -Archivierung, -Pflege
- Ablaufsteuerung
- Bahnplanung
- Ansteuerung des Antriebskreises
- Überwachung
- Teile des ESC-Sicherheitskreises
- Kommunikation mit externer Peripherie (andere Steuerungen, Leitrechner, PCs, Netzwerk)

Im System der Robotersteuerung existieren insgesamt drei serielle Schnittstellen nach dem RS-232 Standard (Com1 – Com3). Die ersten beiden werden systemintern verwendet. Der zur freien Nutzung verfügbare Com3-Port ist standardmäßig der Windows-Umgebung zugeordnet und muss explizit zur Nutzung unter VxWorks (und somit der Ablaufsteuerung) freigegeben werden. Das

geschieht auf Seiten der Windows-Umgebung in der Datei HW_INF.INI im Abschnitt [SERIAL] (KUKA Roboter GmbH, 2007):

```
...  
[SERIAL]  
:ENABLE: COM is accessible by robot (vxWorks)  
:DISABLE: COM is accessible by Win95  
:CONSOLE: for developer only  
COM3=ENABLE ;[ENABLE, DISABLE, CONSOLE]  
...
```

Änderungen werden erst bei einem Neustart des Systems übernommen – alternativ kann die Konfiguration über das KUKA Control Panel (KCP) im Administratormodus erneut eingelesen werden.

Die eigentliche Konfiguration der Schnittstellenparameter erfolgt ebenfalls unter Windows, jedoch in der Datei SERIAL.INI im Abschnitt [COM3] (KUKA Roboter GmbH, 2007):

```
...  
[COM3]  
BAUD=57600  
CHAR_LEN=8 ; 7,8  
STOP_BIT=1 ; 1,2 at time not changeable  
PARITY=0 ; EVEN=2, ODD=1, NONE=0  
PROC=4 ; 3964R=1, SRVT=2, WTC=3, XONXOFF=4  
...
```

Ist die Robotersteuerung konfiguriert, können die Programme sowohl auf der PC Seite (ROS) als auch auf der Robotersteuerung (KRL) in Betrieb genommen werden. Abbildung 39 beinhaltet sowohl den Programmablaufplan der ROS Node (blau) als auch den des Pendants auf der Roboterseite (rot).

Die ROS Node beginnt mit ihrer Initialisierung und Anmeldung beim ROS Master. Da die Implementierung der einzelnen Roboterbefehle über ROS Services erfolgen soll, müssen diese ebenfalls dem Master bekannt gegeben werden. Im Anschluss erfolgt die Initialisierung der seriellen Schnittstelle. Danach geht das Hauptprogramm in den Ruhemodus und erwacht nur periodisch, um sich der Anwesenheit des Masters zu überzeugen.

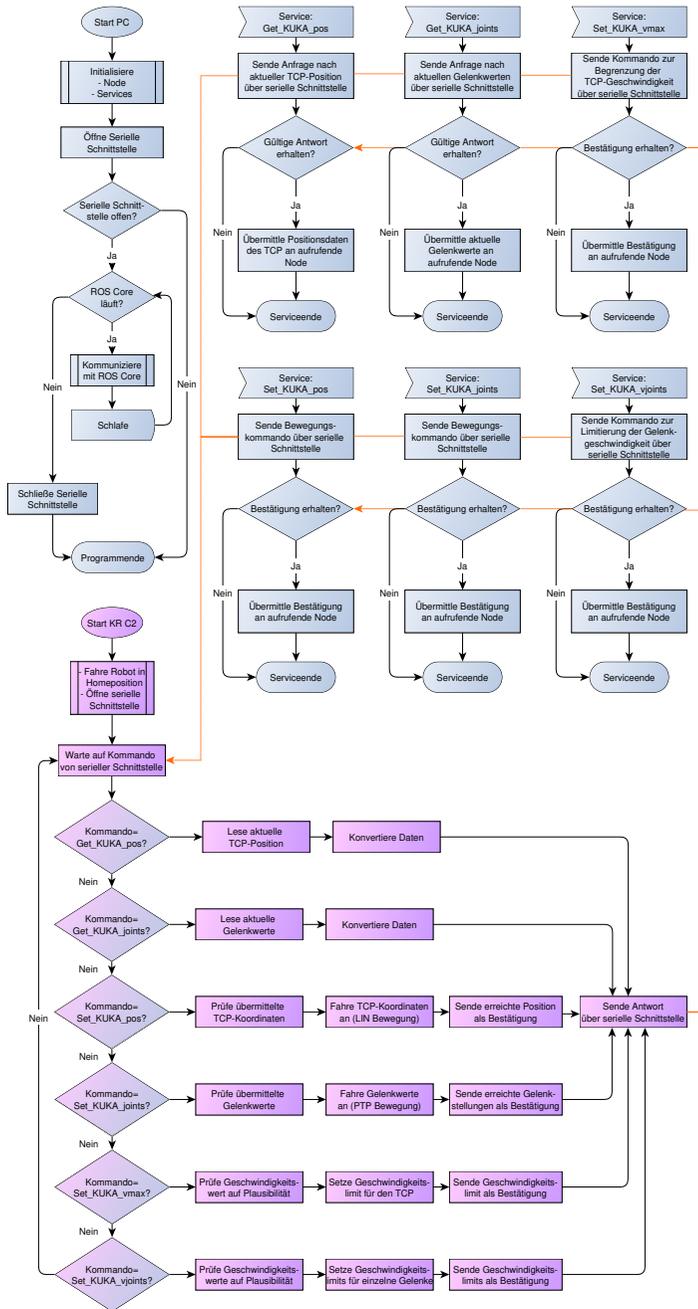


Abbildung 39: Programmablaufplan der KUKA-node sowie des Pendants auf Seiten der Robotersteuerung

Möchte man eine Bewegung des Roboters auslösen, genügt der Aufruf eines dessen Kommandos zugeordneten Services unter Angabe der benötigten Parameter (zum Beispiel Koordinaten im Arbeitsraum des Roboters). In der Servicerroutine werden die übergebenen Parameter –falls möglich– auf Schlüssigkeit geprüft und anschließend in ein geeignetes Format für die Datenübertragung gebracht. Da sich die interne Darstellung von Gleitkommawerten in C++ und KRL stark unterscheidet und außerdem sämtliche Parameter in einer Zeile übertragen werden müssen, wurde als Datenaustauschformat auf Textvariablen zurückgegriffen. Den Anfang dieser Zeichenkette bildet eine Kennung für den jeweils vom Roboter auszuführenden Befehl, danach kommen die durch Kommata separierten Parameter. Der zusammengestellte Datensatz wird danach über die serielle Schnittstelle übertragen und auf eine Antwort gewartet. Enthält diese Antwort Rückgabewerte, werden diese in Gleitkommawerte zurückkonvertiert und vor der Beendigung des Services an die aufrufende Stelle übermittelt.

Auf der Seite der Robotersteuerung wird während der Initialisierungsphase die serielle Schnittstelle geöffnet und der Roboter per PTP-Fahrplanweisung in eine vordefinierte Startposition gefahren. Dieses ist nötig, da zum einen die Robotersteuerung als erste Fahrplanweisung immer eine PTP-Bewegung erwartet und zum anderen nur bei identischen Startpositionen identische Anfahrwege während des Programms garantiert werden können.

Danach wartet die Robotersteuerung auf Daten von der seriellen Schnittstelle. Abhängig von der Kennung am Anfang der empfangenen Zeichenkette werden dann verschiedene Unterprogramme aufgerufen, die die empfangenen Parameter in ein geeignetes Datenformat (meist Gleitkomma) konvertieren und das gewünschte Roboterkommando ausführen. Sollten Rückgabewerte vorhanden sein, werden diese wieder in Text konvertiert und über die serielle Schnittstelle zurückgesendet – andernfalls erfolgt eine Quittung ohne Parameter.

Im Einzelnen wurden bisher folgende Kommandos implementiert:

ROS Service	KRL Kommando	Parameter	Rückgabewerte
set_KUKA_pos	LIN	x, y, z, a, b, c	result=true/false
set_KUKA_joints	PTP	a1, a2, a3, a4, a5, a6	result=true/false
get_KUKA_pos	Auslesen \$POS_ACT	---	x, y, z, a, b, c
get_KUKA_joints	Auslesen \$AXIS_ACT	---	a1, a2, a3, a4, a5, a6
set_KUKA_vmax	Setzen von: \$ACC_AXIS, \$VEL_AXIS	vcp, vori1, vori2, acp, aori1, aori2	result=true/false

set_KUKA_vjoint	Setzen von: \$VEL.CP, \$VEL.ORI1, \$VEL.ORI2, \$ACC.CP, \$ACC.ORI1, \$ACC.ORI2	vmax	result=true/false
get_KUKA_ping	---	---	result=true/false

Tabelle 5: Implementierte Roboterkommandos

Unter Verwendung der oben genannten ROS Services kann der Roboter sowohl von der Kommandozeile aus durch Nutzung des Befehls „rosservice call {servicename} {parameter}“ als auch aus eigenen ROS Nodes heraus gesteuert werden.

5.3.5 Das Launchfile (Systemstart)

Komplexe Systeme wie der Stiff-Flop Manipulator beinhalten üblicherweise eine größere Anzahl miteinander kommunizierender Programmknoten, von denen jeder über einen eigenen Parametersatz verfügt, der gegebenenfalls beim Start der Node initialisiert werden muss. ROS stellt zum vereinfachten Starten des Gesamtsystems einen Mechanismus zur Verfügung, der über so genannte Launch Files konfiguriert werden kann. Diese Launch Files sind nicht nur Listen der zu startenden Nodes, sondern XML-Struktur, die auch die Definition und Abfrage von Abhängigkeiten der einzelnen Programmknoten untereinander oder der verwendeten Hardware sowie die Initialisierung von Laufzeitparametern ermöglichen. Sollte zum Zeitpunkt der Ausführung kein ROS-Kern verfügbar sein, wird dieser automatisch gestartet.

Tabelle 6 zeigt den Aufbau des verwendeten Launchfiles. Über das XML-Element <param/> werden vorab einzelne Werte des Parameterservers initialisiert. Die einzelnen aufzurufenden Nodes werden durch das Element <node/> festgelegt. Diese müssen neben dem Namen der Node auch dessen übergeordnetes Paket sowie ein Laufzeitname als Attribute enthalten. Falls nötig, können hier auch das Arbeitsverzeichnis oder andere benutzerdefinierte Argumente übergeben werden. Eine Besonderheit stellt das Element <remap/> dar; es dient der Umleitung eines von der Node abonnierten oder veröffentlichten Datenkanals. Auf die Bedeutung der einzelnen Parameter wird an dieser Stelle nicht eingegangen – diese sind ausführlich im Quellcode der einzelnen Nodes beschrieben.

```

<launch>

  <!-- GLOBAL PARAMETER INITIALIZATION -->
  <param name="/intensity/resolution" value="0"/>

  <!-- STARTING DRIVER NODES -->
  <node pkg="intensity" type="intensity" name="intensity"/>
  <node pkg="cambar_b2" type="cambar_b2" name="cambar_b2" cwd="node"
    args="_root:=SF HOLDER"/>
  <remap from="/cambar_smoother_endo/input" to="/cambar_b2/pose_endo">
  <remap from="/cambar_smoother_endo/output" to="/cambar_smoother_endo/pose_endo">
  <node pkg="cambar_smoother" type="cambar_smoother" name="cambar_smoother_endo">

  <!-- STARTING DETECTION NODES -->
  <node pkg="tubedetect" type="tubedetect" name="tubedetect" output="screen"/>
  <node pkg="DetectTubeConnector" type="DetectTubeConnector"
    name="DetectTubeConnector" output="screen" args="image:=/camera/image
    camera_info:=/camera/camera_info"/>

  <!-- ESTABLISH ADDITIONAL CONNECTIONS (STATIC) IN TF TREE -->
  <node pkg="tf" type="static_transform_publisher" name="world_to_sfbase"
    args="0 0 0 0 0 1 world SF_BASE 100"/>
  <node pkg="sf_tf_flex" type="sf_tf_flex" name="sf_tf_flex_endo_camera1"
    args="_leaf1:=ENDO _leaf2:=camera1" output="screen"/>
  ...

  <!-- STARTING SF MISPLACEMENT SECURITY ALERT -->
  <node pkg="sf_tf_alert" type="sf_tf_alert" name="sf_tf_alert"
    args="_pcl:=/DetectTubeConnector/connector_positions_projected
    _pc2:=/projected_cloud"/>

  <!-- STARTING KUKA INETRFACE FOR CALIBRATION PURPOSES -->
  <node pkg="KUKA_connector" type="KUKA_connector" name="KUKA_connector"
    output="screen"/>

  <!-- STARTING USER INTERFACE -->
  <node pkg="rqt_reconfigure" type="rqt_reconfigure" name="rqt_reconfigure" />
  <node pkg="rqt_rviz" type="rqt_rviz" name="rviz">
</launch>

```

Tabelle 6: Aufbau des Stiff-Flop Vision - Launchfiles

5.4 Lageermittlung des Prüflings

Die Lageermittlung des Stiff-Flop Armes stellt auf Grund verschiedener Besonderheiten eine Herausforderung dar, der mit eingeführten Methoden der Objekterkennung nur schwer begegnet werden kann.

Die Verwendung von effizienten kantenorientierten Verfahren scheidet in weiten Teilen aus, da weder Form noch Lage des Armes im Videobild bekannt sind. Die Verwendung von Markern schafft hier leider nur bedingt Abhilfe, da diese nur auf den steifen Verbindungsstücken des Armes angebracht werden können – ein Aufdruck auf der flexiblen Silikonhaut ließe die Marker bei Bewegung des Armes nicht mehr zuverlässig erkennbar werden. Beschränkt man die optische Detektion aber auf die Konnektoren des Armes, ist eine verlässliche Überwachung der geforderten „No Fly Zones“, also von für den Arm gesperrten Gebieten, nicht umsetzbar und könnte bestenfalls durch interpolierte Daten erfolgen. Die für Medizinprodukte dieser Klasse geforderte Redundanz wäre damit nicht erreicht.

Parallel zur Detektion der Marker auf den Konnektoren kommt deshalb ein texturorientiertes Verfahren zum Einsatz. Eine so genannte „Support Vector Machine (SVM)“ klassifiziert kleine quadratische Bildausschnitte als „Objekt“ oder „Hintergrund“ und ermöglicht dadurch eine grobe Lageermittlung des gesuchten Objektes. Durch die nötige Rasterung kommt es zum Verlust von Genauigkeit – zur Überwachung von gesperrten Gebieten ist dieser Ansatz aber tauglich. Ebenfalls getestet wurde die Verwendung von so genannten Aktiven Konturen. Auf Grund von mangelnder Performanz kommen diese nicht im aktuellen System zum Einsatz, werden aber der Vollständigkeit halber im weiteren Verlauf ebenfalls kurz skizziert.

Die Effizienz sämtlicher Verfahren steht und fällt mit der Qualität des eingehenden Videomaterials und dabei insbesondere der gleichmäßigen und glanzlichtfreien Ausleuchtung der Szene. Gerade der letzte Punkt führt jedoch immer wieder zu Problemen: Durch den konstruktiven Aufbau des Endoskops liegt die Lichtquelle direkt an der optischen Achse des Bildaufnehmers. Die daraus resultierenden Glanzlichter können nur durch eine Reduzierung der Helligkeit vermindert werden. Diese muss mechanisch und manuell erfolgen, was eine automatische Nachführung dieses Faktors leider verhindert. Die mangelhafte Ausleuchtung dunklerer Bereiche verringert zwar die Qualität der Detektion, ist aber eher für den Mediziner als für die automatische Objekterkennung ein Problem.

Die getesteten und letztendlich verwendeten Algorithmen werden in den nächsten Kapiteln näher beschrieben.

5.4.1 Videovorverarbeitung

Ein großes Problem bei der Akquise des Bildmaterials ist der relativ geringe Dynamikumfang der Kamera im Vergleich zum menschlichen Auge. Liegt der Wert des Auges bei etwa 14 Blendenstufen (inklusive Adaption noch deutlich höher), erreicht eine Standardkamera in etwa 11 Blendenstufen.

Im technischen Umfeld wird der Dynamikbereich häufig in der logarithmischen Einheit dB angegeben. Der Wert in dB drückt dabei aus, um welchen Faktor der höchste Helligkeitswert I_h größer ist als der geringste Helligkeitswert I_d :

$$D = 20 \cdot \lg\left(\frac{I_h}{I_d}\right) \text{dB} \quad (5.26)$$

Ein typischer Bildsensor liegt bei etwa 60dB, das Auge erreicht innerhalb einer Szene etwa einen Wert von 100dB. Für einige Einsatzzwecke sind hochdynamische Bildsensoren verfügbar – allerdings leiden diese unter anderen Einschränkungen wie längerer Belichtungszeit oder geringer Auflösung. Da diese Sensoren häufig logarithmische Kennlinien haben, eignen sie sich außerdem nur sehr bedingt für eine realistische Abbildung der Szene.

Der eingeschränkte Dynamikumfang des Videomaterials führt in der Praxis häufig zu unter- oder überbelichteten Bereichen, in denen weder Farbe noch Struktur eines Objektes erkennbar sind. Bei industriellen Anwendungen wird deshalb besonderer Wert auf die perfekte Ausleuchtung der Szene gesetzt. Da diese Möglichkeit im chirurgischen Einsatz leider weitestgehend entfällt, müssen andere Lösungen gefunden werden.

Zur Erprobung kam unter anderem ein stark vereinfachter HDR (High Dynamic Range)-Algorithmus sowie ein Ansatz zur Reduzierung von Glanzlichtern durch Transformation in einen anderen Farbraum mit anschließender Glättung der Extrema.

Der HDR-Ansatz ist vor allem aus der Amateurfotographie bekannt. Hier werden bei komplexen Lichtverhältnissen Belichtungsreihen von statischen Objekten aufgenommen. Diese bestehen meist aus einer Folge von 3 Bildern, von denen eines leicht unterbelichtet wird, eines eine gute mittlere Helligkeit hat und ein weiteres eine leichte Überbelichtung aufweist. Hat man Zugang zu den Rohdaten des Bildsensors und hat dieser eine lineare Kennlinie – also einen linearen Zusammenhang zwischen eingefallener Lichtleistung und Ausgangssignal – können die Intensitätswerte der Einzelaufnahmen direkt miteinander verkettet werden und somit zu einem Bild mit höherem Dynamikumfang zusammengestellt werden. Meist hat man jedoch nur Zugriff auf die bereits in der Kamera vorverarbeiteten Daten. Ein linearer Zusammenhang zwischen eintreffender Lichtleistung und Helligkeitsinformation eines Bildpunktes ist hier leider nicht mehr gegeben, da die Kameralogik das Ausgabebild für das menschliche Auge optimiert.

Da diese Optimierung auch zu einer firmeneigenen Produktdifferenzierung beiträgt und deshalb von den Herstellern als Betriebsgeheimnis eingestuft wird, erhält der Anwender leider keine Information über die verwendeten Kennlinien, so dass diese in einem aufwändigen Prozess aus den Bilderreihen errechnet werden müssen. Ist die Kennlinie bekannt, können über ihre Umkehrfunktion die realen Strahlungswerte ermittelt werden, aus denen dann ein HDR Bild zusammengesetzt werden kann. Häufige Verwendung findet das Verfahren von (Debevec & Malik, 1997). Dieses Verfahren liefert gute Ergebnisse und ist relativ einfach zu implementieren.

Der errechnete hohe Dynamikumfang führt spätestens dann zu Problemen, wenn das Bildmaterial auf einem Anzeigegerät mit geringerem Dynamikumfang dargestellt werden soll. Hier kommen Tone Mapping Verfahren zum Einsatz, die die errechneten Werte auf den darstellbaren Bereich abbilden. Lineare Funktionen sind dabei ungeeignet, da die Möglichkeit des menschlichen Auges, feine Helligkeitsunterschiede wahrzunehmen, mit abnehmender Beleuchtung zunimmt. In dunkleren Bereichen können also feine Nuancen eher wahrgenommen werden als in hellen Arealen. Die meisten Tone Mapping Verfahren verwenden daher an die Helligkeitsempfindung des menschlichen Auges angepasste logarithmische Funktionen. Einen guten Überblick über Tone Mapping Verfahren bietet unter anderem (Devlin, Chalmers, Wilkie, & Purgathofer, 2002).

Für die Anwendung auf Videodatenströme ist das oben genannte Verfahren jedoch auf Grund seiner relativ hohen Rechenlast eher ungeeignet. Aus diesem Grund wurde eine stark vereinfachte Variante implementiert, die den Dynamikumfang des Videobildes erweitert und dabei leichte Kontrastreduzierungen in Kauf nimmt. Im Folgenden wird der Ablauf der Dynamikerweiterung an Hand eines Beispiels näher erläutert.

Notwendig sind – wie auch bei HDR Verfahren – zwei Aufnahmen der gleichen Szene mit unterschiedlichen Belichtungszeiten. Im folgenden Fall müssen die Bilder im HSV-Farbraum vorliegen.

Das erste, helle Bild **H** wird so lang (über-)belichtet, dass nur ein sehr geringer Prozentsatz der Pixel unterbelichtet ist, also einen Helligkeitswert von weniger als 20 aufweist. Die Belichtungsdauer wird dabei kontinuierlich über einen P-Regler nachgeführt, dessen Parameter empirisch ermittelt wurden.

Das zweite, dunkle Bild **D** hingegen wird zur Erfassung der hellen Bereiche deutlich kürzer belichtet; hier darf nur ein geringer Anteil an Pixeln einen Helligkeitswert größer 250 aufweisen. Als Resultat beinhaltet diese Aufnahme meist größere unterbelichtete Bereiche. Auch hier erfolgt die Nachführung der Belichtungsdauer über einen einfachen P-Regler.

$$\mathbf{H}_{\text{Helligkeit}} = \begin{bmatrix} 56 & 56 & 120 & 255 & 255 \\ 52 & 64 & 255 & 255 & 255 \\ 53 & 56 & 98 & 255 & 234 \\ 49 & 57 & 76 & 98 & 78 \end{bmatrix} \quad (5.27)$$

$$\mathbf{D}_{\text{Helligkeit}} = \begin{bmatrix} 11 & 11 & 75 & 224 & 216 \\ 7 & 19 & 222 & 253 & 217 \\ 8 & 11 & 53 & 214 & 189 \\ 4 & 12 & 31 & 53 & 33 \end{bmatrix}$$

Aus der Helligkeitsmatrix des überbelichteten Bildes \mathbf{H} wird jetzt eine Maske der Bereiche erstellt, die an der oberen Grenze der möglichen Helligkeitswerte liegen (roter Rahmen). Innerhalb dieses Bereiches wird jetzt im unterbelichteten Bild \mathbf{D} nach dem niedrigsten Helligkeitswert m_{\min} gesucht. In Beispiel handelt es sich um die grün schraffierte „214“

Dieser Minimalwert wird jetzt vom Helligkeitskanal des Bildes \mathbf{D} subtrahiert und ergibt einen Summanden ($\mathbf{D}^*_{\text{Helligkeit}}$) der späteren Helligkeitsmatrix des in der Dynamik erweiterten Bildes \mathbf{B} . Eventuell auftretende negative Werte werden dabei auf 0 begrenzt:

$$\mathbf{D}_{\text{Helligkeit}} - m_{\min} = \begin{bmatrix} 0 & 0 & 0 & 10 & 2 \\ 0 & 0 & 8 & 39 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{D}^*_{\text{Helligkeit}} \quad (5.28)$$

Addiert man $\mathbf{D}^*_{\text{Helligkeit}}$ jetzt zum Helligkeitskanal $\mathbf{H}_{\text{Helligkeit}}$ des dunklen Bildes, erhält man eine Helligkeitsmatrix $\mathbf{B}_{\text{Helligkeit}}$, die den vollen Dynamikumfang des ursprünglichen Bildes abdeckt:

$$\mathbf{D}^*_{\text{Helligkeit}} + \mathbf{H}_{\text{Helligkeit}} = \begin{bmatrix} 56 & 56 & 120 & 265 & 257 \\ 52 & 64 & 263 & 294 & 258 \\ 53 & 56 & 98 & 255 & 234 \\ 49 & 57 & 76 & 98 & 78 \end{bmatrix} = \mathbf{B}^*_{\text{Helligkeit}} \quad (5.29)$$

Damit \mathbf{B}^* weiterhin auf gängigen Komponenten darstellbar bleibt, müssen die Helligkeitswerte im nächsten Schritt auf den anzeigbaren Bereich $[0..255]$ abgebildet werden. Um eine möglichst gute

Helligkeitsverteilung zu erreichen wird dazu der Helligkeitswert h_{\max} des hellsten Bildpunktes ermittelt und zur Ermittlung eines Anpassungsfaktors herangezogen.

Das Resultat liefert den Helligkeitskanal des fertigen Bildes \mathbf{B} :

$$\frac{\mathbf{B}^*_{\text{Helligkeit}}}{h_{\max}} * 255 = \mathbf{B}_{\text{Helligkeit}} \quad (5.30)$$

10	10	65	194	187
6	16	193	219	188
7	10	46	186	164
3	10	27	46	29

Sowohl der Farbwert- als auch der Sättigungskanal des resultierenden Bildes werden im nächsten Schritt aus den entsprechenden Kanälen der Ausgangsbilder zusammengesetzt. Basierend auf dem Helligkeitskanal des resultierenden Bildes werden in besonders dunklen Bereichen Farb- und Sättigungswert des helleren Bildes 1:1 übernommen, um das Rauschen möglichst gering zu halten. In den helleren Zonen kommen beide Werte vom dunkleren Bild, da die Farbabstufungen des Kamerabildes nahe der Überbelichtung stark an Qualität verlieren. In einem schmalen Bereich mittlerer Helligkeit wird ein fließender Übergang von Werten des helleren zu Werten des dunkleren Bildes erzeugt.

Das Resultat zeigt Abbildung 40. Links ist die stark belichtete Aufnahme zu sehen, in der Mitte unterbelichtete. Auf der rechten Seite befindet sich das zur Weiterverarbeitung bereitgestellte resultierende Bild:



Abbildung 40: Ergebnis des HDR Algorithmus: Links unterbelichtetes Bild, Mitte überbelichtetes Bild, Rechts kombinierte Ansicht

Die Komplexität des Algorithmus liegt bei $O(n)$ - wobei n der Anzahl der Bildpunkte entspricht. Die Laufzeit ist somit direkt von der gewählten Auflösung des Videomaterials abhängig.

Obwohl der Algorithmus durchaus brauchbare Ergebnisse liefert, scheitert seine Verwendung leider an der vorhandenen Infrastruktur: Die Änderung der Belichtungsdauer der verwendeten Kamera ist ein zeitraubender Vorgang. Wird nach jeder Aufnahme die Belichtungsdauer angepasst, pendeln sich die Aufzeichnungsraten bei etwa 2 Bildern/Sekunde ein. Eine derart niedrige Bildfrequenz ist für den vorliegenden Anwendungsfall undenkbar, weshalb dieser Ansatz nicht weiter verfolgt wurde.

In der Literatur gibt es eine große Anzahl an Veröffentlichungen, die sich mit der nachträglichen Eliminierung von Glanzlichtern in Digitalaufnahmen befasst. (Shuqair, 2014) hat in seiner Arbeit 13 dieser Algorithmen mit einander verglichen und auf ihre Tauglichkeit für den Einsatz im Stoff-Flop Projekt untersucht. Zwei Algorithmen kamen dabei in die engere Auswahl und werden im Folgenden kurz vorgestellt.

5.4.1.1 Algorithmus nach Shen & Cai

Der erste vielversprechende Ansatz beruht auf der Arbeit von (Shen & Cai, 2009) und versucht, Bildpunkte in Spiegelungen und diffuse Reflexionen zu unterteilen. Er arbeitet bildpunktbasierend und kommt ohne die Ermittlung von Beziehungen zwischen den einzelnen Bildpunkten aus. Eine Segmentierung des Ausgangsmaterials ist ebenfalls nicht nötig. Allerdings sind einige Voraussetzungen für den Algorithmus zu erfüllen:

- Das dichromatische Reflexionsmodell ist anwendbar
- Die Übertragungsfunktion der Kamera ist linear, d.h. es existiert ein linearer Zusammenhang zwischen eintreffender Lichtleistung und Helligkeitsinformation eines Bildpunktes (Bei Verwendung von komprimiertem Videomaterial besteht häufig ein nichtlinearer Zusammenhang)
- Die verwendete Lichtquelle ist weiß

In einem ersten Verarbeitungsschritt werden die Bildpunkte nach den kleinsten auftretenden Farbwerten im R, G und B Kanal durchsucht. Die gefundenen Minima werden vom jeweiligen Farbkanal des gesamten Bildes abgezogen. Das Bild wird dadurch insgesamt dunkler und verliert an Sättigung, weist aber keine Glanzlichter im herkömmlichen Sinne mehr auf. Um diesen Farbigkeitsverlust auszugleichen, errechnen Shen & Cai ein so genanntes „Modified Specular Free Image“

Wie vorab erwähnt, ist eine Voraussetzung für die Anwendbarkeit ein lineares Verhalten der Kamera. Während des Aufnahmeprozesses wird das einfallende Licht jedes Bildpunktes über die Belichtungsdauer integriert. Dabei ist $v_i(x, y)$ der Ausgabewert der Kamera für den Farbkanal i an der Stelle (x, y) , $s_i(\lambda)$ ist die Empfindlichkeit der Kamera bei einer bestimmten Wellenlänge λ und $I(\lambda)$ ist die Lichtleistung des Leuchtmittels bei der Wellenlänge λ .

Integriert wird über die komplette Bandbreite des sichtbaren Lichts:

$$v_i(x, y) = \int f(\lambda, x, y) I(\lambda) s_i(\lambda) d\lambda \quad (5.31)$$

$f(\lambda, x, y)$ stellt dabei die Radianz (Flächenhelligkeit) basierend auf dem dichromatischen Reflexionsmodell nach (Shafer, 1984) dar:

$$f(\lambda, i, e, g) = f_s(\lambda, i, e, g) + f_d(\lambda, i, e, g) \quad (5.32)$$

Dieses folgt dem Ansatz, dass sich die von einem Körper reflektierte Strahlung in Grenzschichtreflexion (surface reflection s) und diffuse Körperreflexion (Index b) aufteilen lässt.

λ ist dabei weiterhin die betrachtete Wellenlänge des auftreffenden Lichtes, i , e und g stehen für die Winkel zwischen dem Licht und der Flächennormalen bzw. zwischen Lichtquelle und Empfänger. Die genaue Winkeldefinition ist in Abbildung 41 zu sehen:

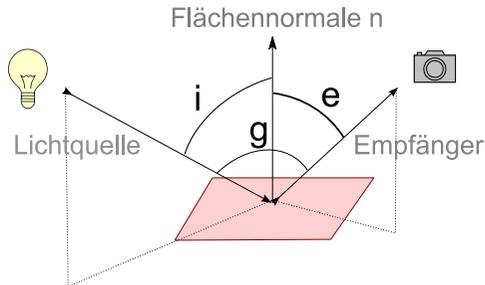


Abbildung 41: Definition Photometrischer Winkel

Spiegelung und diffuse Reflexion lassen sich weiterhin nach (Shafer, 1984) in weitere Faktoren zerlegen:

- Die Komposition (composition) ist eine relative, spektrale Energieverteilungsfunktion und ist nur von der Wellenlänge abhängig, nicht aber von der Geometrie
- Die Stärke (magnitude) definiert einen geometrischen Skalierungsfaktor, der ausschließlich von der Geometrie abhängt, nicht aber von der Wellenlänge

Basierend darauf kann Gleichung (5.32) umgeschrieben werden in

$$f(\lambda, i, e, g) = m_s(i, e, g) \cdot c_s(\lambda) + m_b(i, e, g) \cdot c_b(\lambda) \quad (5.33)$$

Da davon auszugehen ist, dass die spektrale Strahlungsverteilung von Spiegelung und Lichtquelle identisch ist, kann (5.33) nach (Shen & Cai, 2009) umgeschrieben werden in

$$f(\lambda, x, y) = m_s(i, e, g) \cdot c_{ss} + m_b(i, e, g) \cdot c_b(\lambda) \quad (5.34)$$

Setzt man nun (5.34) in (5.31) ein und ersetzt die Position (x, y) durch den Pixel (p) erhält man eine Gleichung, die die Helligkeit eines Pixels in Spiegelungsanteile und Anteile diffuser Reflexion zerlegt:

$$\begin{aligned} v_i(p) &= m_b(p) \int_{\lambda} c_b(\lambda) \cdot l(\lambda) \cdot s_i(\lambda) d\lambda + m_s(p) \cdot c_{ss} \int_{\lambda} l(\lambda) \cdot s_i(\lambda) d\lambda \\ &= m_b(p) \cdot v_{b,i} + m_s(p) \cdot v_{s,i} \end{aligned} \quad (5.35)$$

Dabei steht $v_{b,i} = \int_{\lambda} c_b(\lambda) \cdot l(\lambda) \cdot s_i(\lambda) d\lambda$ für die Körperfarbe und $v_{s,i} = c_{ss} \int_{\lambda} l(\lambda) \cdot s_i(\lambda) d\lambda$ für die Farbe der Lichtquelle, welche in unserem Fall weiß ist. Da der Ansatz von (Shen & Cai, 2009) vorsieht, die Farbwerte der einzelnen Pixel zu normalisieren und auf einem Bereich von $[0..255]$ abzubilden, gilt weiterhin

$$m_s(p) \cdot v_{s,i} = 255 \cdot m_s(p) = m_{ss}(p) \quad (5.36)$$

und Gleichung (5.35) vereinfacht sich zu

$$v_i(p) = m_b(p) \cdot v_{b,i} + m_{ss}(p) \quad (5.37)$$

Das grenzschichtreflexionsfreie Bild erhält man nun durch die Subtraktion der Minima jedes Farbkanals von jedem seiner Pixel:

$$v_{SF,i}(p) = v_i(p) - v_{\min}(p) \quad (5.38)$$

Dabei ist $v_{\min}(p) = \min_i \{v_i(p)\}$.

$v_{\min}(p)$ kann auch durch das dichromatische Reflexionsmodell beschrieben werden:

$$v_{\min}(p) = \min_i \{v_i(p)\} = m_b(p) \cdot v_{d,\min} + m_{ss}(p) \quad (5.39)$$

Hier gilt dann $v_{b,\min}(p) = \min_i \{v_{b,i}(p)\}$

Wendet man die oben genannten Erkenntnisse auf Gleichung (5.38) an, führt das zu

$$v_{SF,i}(p) = m_b(p) \cdot (v_{b,i} - v_{b,\min}) \quad (5.40)$$

Hier fällt auf, dass das glanzlichtfreie Bild zum einen unabhängig vom geometrischen Faktor m_s ist und zum anderen dunkler ist als das Originalbild, da von sämtlichen Farbkanälen die Minimalwerte abgezogen werden.

(Shen & Cai, 2009) stellen jetzt eine Beziehung zwischen dem zuvor berechneten glanzlichtfreien Bild und der Komponente der diffusen Körperreflektion her:

$$v_{dif,i}(p) = v_i(p) - m_{ss}(p) = v_{SF,i}(p) + v_{\min} - m_{ss}(p) = v_{SF,i}(p) + \tau_s(p) \quad (5.41)$$

wobei $\tau_s(p) = v_{\min} - m_{ss}(p)$ und somit ebenso wie $m_{ss}(p)$ unbekannt ist.

Damit das glanzlichtfreie Bild der diffusen Komponente des Originals möglichst nahe kommt, ist es ratsam, auch im endgültigen Bild den Offset $\tau(p)$ mit einzubeziehen:

$$v_{MSF,i}(p) = v_{SF,i}(p) + \tau(p) = m_b(p) \cdot (v_{b,i} - v_{b,\min}) + \tau(p) \quad (5.42)$$

Der Offset $\tau(p)$ ist dabei nicht konstant, sondern bildpunktabhängig.

Die Farbwert $c_{MSF,i}$ des modifizierten glanzlichtfreien Bildes ergibt sich nun zu

$$c_{MSF,i}(p) = \frac{v_{MSF,i}}{\sum_i v_{MSF,i}} = \frac{m_b(p) \cdot (v_{b,i} - v_{b,\min}) + \tau(p)}{m_b(p) \sum_i (v_{b,i} - v_{b,\min}) + 3\tau(p)} \quad (5.43)$$

Dabei fällt auf, dass bei $\tau(p) \neq 0$ die Farbwerte nicht mehr unabhängig von $m_b(p)$ sind. (Shen & Cai, 2009) schlagen daher vor, den Offset $\tau(p)$ in Abhängigkeit von einem Schwellwerte t_v zu bestimmen, der die Grenze zwischen Glanzlichtern und heller diffuser Reflexion festlegt.

t_v hängt dabei von μ_v und σ_v ab, welche den Durchschnitt und die Standardabweichung aller Pixel zu $v_{\min}(p)$ darstellen, sowie von η ab. η ist ein Faktor, der die Häufigkeit von Glanzlichtern im Bild widerspiegelt und für die meisten Bilder mit $\eta = 0.5$ passend dimensioniert ist.

$$t_v = \mu_v + \eta\sigma_v \quad (5.44)$$

Um nun zwischen Glanzlichtern und diffuser Reflexion zu unterscheiden, wird einfach der minimale Farbwert mit dem Schwellwert verglichen. Ist der Farbwert kleiner als der Schwellwert, handelt es sich um diffuse Strahlung, andernfalls um Teile eines Glanzlichtes.

Basierend auf dieser Unterscheidung kann jetzt der Offset $\tau(p)$ berechnet werden:

$$\tau(p) = \begin{cases} t_v & , v_{\min}(p) > t_v \\ v_{\min}(p) & , v_{\min}(p) \leq t_v \end{cases} \quad (5.45)$$

Die Komplexität des Algorithmus liegt auch hier bei $O(n)$ - wobei n der Anzahl der Bildpunkte entspricht. Die Laufzeit ist also wie beim vorab vorgestellten Minimal-HDR Algorithmus direkt von der gewählten Auflösung des Videomaterials abhängig.

5.4.1.2 Algorithmus nach Miyazaki et al.

Der Ansatz von (Miyazaki, Tan, Hara, & Ikeuchi, 2003) ist ebenfalls pixelbasiert und kommt ohne Bildsegmentierung aus. Abhängigkeiten von Nachbarbildpunkten bleiben ebenfalls unbeachtet. Das Bild muss im RGB-Farbraum vorliegen. Ähnlich wie bei der zuletzt vorgestellten Methode gibt es auch hier einige Voraussetzungen, damit der Algorithmus effektiv arbeiten kann. Diese sind im Einzelnen:

- Die verwendete Lichtquelle ist weiß
- Die Übertragungsfunktion der Kamera ist linear, d.h. es existiert ein linearer Zusammenhang zwischen eintreffender Lichtleistung und Helligkeitsinformation eines Bildpunktes
- Eine komplette Sättigung aller drei Farbkanaäle darf nicht vorliegen
- Einschränkung: Glanzlichter auf reinweißen Körpern können nicht entfernt werden

Zuerst werden die Bildpunkte in einen anderen Farbraum, den so genannten „M Color Space“, transformiert:

$$\begin{bmatrix} m_1(p) \\ m_2(p) \\ m_3(p) \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} r(p) \\ g(p) \\ b(p) \end{bmatrix} \quad (5.46)$$

Dabei sind $r(p)$, $g(p)$ und $b(p)$ die einzelnen Farbwerte des Bildpunktes p im RGB-Farbraum; $m_1(p)$, $m_2(p)$ und $m_3(p)$ sind die entsprechenden Werte im Color Space.

Nach erfolgter Transformation wird im neuen Farbraum eine Bewertung der einzelnen Farbkanäle vorgenommen:

$$\begin{bmatrix} \hat{m}_1(p) \\ \hat{m}_2(p) \\ \hat{m}_3(p) \end{bmatrix} = \begin{bmatrix} m_1(p) \\ m_2(p) \\ a\sqrt{m_1(p)^2 + m_2(p)^2} \end{bmatrix} \quad (5.47)$$

Im Anschluss erfolgt die Rücktransformation in den RGB-Ausgangsfarbraum:

$$\begin{bmatrix} \hat{r}(p) \\ \hat{g}(p) \\ \hat{b}(p) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & 0 & 1 \\ -\frac{1}{3} & -\frac{1}{\sqrt{3}} & 1 \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} & 1 \end{bmatrix} \begin{bmatrix} \hat{m}_1(p) \\ \hat{m}_2(p) \\ \hat{m}_3(p) \end{bmatrix} \quad (5.48)$$

Die Komplexität des Algorithmus liegt auch hier wieder bei $O(n)$, womit die Laufzeit ebenfalls direkt von der gewählten Auflösung des Videomaterials abhängt.

Beide Verfahren wurden von Shuqair im Rahmen des Stiff-Flop Projektes implementiert und evaluiert. Dabei kamen sowohl computergenerierte Testbilder als auch Videobilder des Stiff-Flop Armes zum Einsatz. Die Ergebnisse beider Verfahren sind in Abbildung 42 in Form eines Beispiels zu sehen.

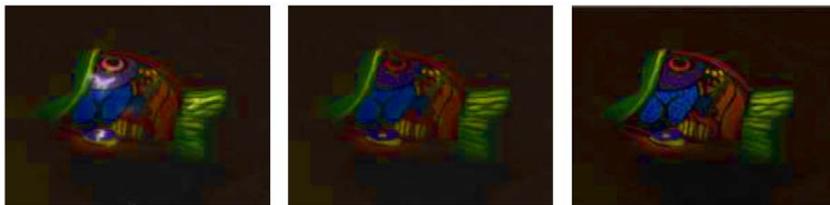


Abbildung 42: Ergebnisse der Glanzlichtentfernung. Links Ausgangsbild, Mitte Shen&Cai, Rechts Miyazaki. Quelle: (Shuqair, 2014)

Je nach Ausgangsmaterial liefern beide Verfahren sehr gute bis extrem schlechte Ergebnisse. Wie zu erwarten versagen beide Verfahren bei echten Überbelichtungen, bei denen die Farbwerte der entsprechenden Bereiche nicht nur nahe dem Weißwert liegen, sondern rein weiß sind.

Auf Grund der Unvorhersehbarkeit der Ergebnisse kommt im Stiff-Flop Projekt nach Implementierung der unterschiedlichen Methoden in die TubeDetect-Node - basierend auf der Arbeit von (Shuqair, 2014) – und umfangreichen Testreihen keines der vorgestellten Verfahren zum Einsatz.

5.4.2 Konnektordetektion: Markererkennung und -vermessung

Eine wichtige Eingangsgröße für die Regelung des Stiff-Flop Armes ist neben der Erfassung von Kräften und Momenten am Arm die Position der Konnektoren. Obwohl diese Position auch bei Anliegen externer Kräfte erstaunlich präzise aus einem Modell errechnet werden kann, besteht die Notwendigkeit, sie extern zu erfassen. Eine optische Detektion bietet sich an, da endoskopische Videodatenströme der Szene bereits vorliegen und durch eine Entkopplung vom restlichen System eine für Medizinprodukte dieser Kategorie nötige Redundanz erreicht wird.

Da die Mittellinie des Konnektors ermittelt werden soll, bietet sich ein kreisförmiger Marker koaxial um das jeweilige Verbindungsstück an. Wird dessen Farbe derart gewählt, dass sie in vivo möglichst nicht vorkommt, ist eine einfache Detektion ohne konturbasierte Formerkennung möglich. Aus den sichtbaren Ringsegmenten lassen sich Rückschlüsse über die Pose des umlaufenden Ringmarkers ziehen.

Der verwendete Ansatz beruht im Wesentlichen auf einer Idee von (Chen, Wu, & Wada, 2004), die in ihrer Arbeit die Kalibrierung von Stereokamerasystemen mit Hilfe von beliebigen komplaren Kreisen vorschlagen.

Basierend auf dieser Idee hat (Bader, 2014) eine ROS-Node entwickelt, die geschlossene Kreise im Bild detektiert und – bei bekanntem Durchmesser – ihre Lage im Raum ermittelt. Da der Code dieser

Node unter der BSD Open Source Lizenz frei verfügbar ist, konnten einige Teile davon für das Projekt übernommen und erweitert werden. Die einzelnen Schritte des Verfahrens werden im Folgenden kurz erläutert.

Abbildung 43(a) zeigt das Ausgangsbildmaterial. Zu sehen ist ein im 3D-Druck hergestellter Konnektor mit leuchtend grüner umlaufender Farbmarkierung.

Im ersten Arbeitsschritt wird durch Farbsegmentierung eine Maske erstellt, die nur noch die hellgrünen Bereiche des Ausgangsbildes enthält. Zu diesem Zweck wird das Bild in den HSV-Farbraum konvertiert und obere und untere Grenzwerte für Helligkeit, Sättigung und Farbwert festgelegt. Diese Grenzwerte können im laufenden Betrieb unter Verwendung des ROS Parameterservers über eine graphische Oberfläche (ROS dynamic reconfigure) angepasst werden, siehe Abbildung 44. Liegen die Werte eines Bildpunktes des Ausgangsbildes innerhalb der definierten Schranken, wird die Maske an dieser Stelle weiß eingefärbt, sonst bleibt sie schwarz. Das Resultat zeigt Abbildung 43(b).

Die generierte Maske liefert im zweiten Schritt die Grundlage für eine Konturerkennung nach (Suzuki & Abe, 1985). Dabei wird das Bild zeilenweise durchlaufen, bis der Algorithmus bedingt durch einen starken Intensitätsgradienten eine Kontur vermutet und dieser dann folgt. Erkannte Konturpunkte werden in einer baumförmigen Datenstruktur abgelegt, die auch das Verhältnis der Konturen zueinander abbildet: So kann festgestellt werden, ob eine Kontur innerhalb einer anderen Kontur liegt und so zum Beispiel die Position von Löchern in erkannten Körpern markiert. Ist die Eingabemaske auf Grund von verrauschtem Bildmaterial nicht ganz sauber, werden eventuell weitere unerwünschte Konturen erkannt. In vielen Fällen lassen sich diese durch die Festlegung von Schwellwerten für Konturlänge und Anzahl der enthaltenen Punkte eliminieren. Diese Schwellwerte lassen sich ebenfalls über die Nutzeroberfläche anpassen. Die verbliebene erkannte Kontur ist in Abbildung 43(c) zu sehen.

Da von der ursprünglichen Ringmarkierung sowohl die vordere als auch die hintere Kante als eine einzige geschlossene Kontur erkannt werden, muss ein geeigneter Weg gefunden werden, um diese Kanten in separaten Konturen abzubilden, bevor ihre Einzelpunkte der Ellipsenberechnung zugeführt werden. Das geht besonders effizient, wenn man die ursprüngliche Kontur in einen konvexen und einen konkaven Bereich aufteilt. In der praktischen Umsetzung wird dabei die Gesamtkontur von allen Defekten der konvexen Hülle – also konkaven Bereichen - befreit, indem die entsprechenden Punkte aus der Konturliste gelöscht werden. Die verbleibenden Punkte bilden die jeweils hintere Kante der Ringmarkierung (Abbildung 43(d), rote Kurve) ab. Die entfernten konkaven Bereiche werden auf der Basis ihrer Konturlänge bewertet, wobei der größte Bereich in fast allen Fällen zur vorderen Kante der Ringmarkierung führt (Abbildung 43(d), blaue Kurve). Sowohl die rote als auch die blaue Kurve werden im weiteren Verlauf wie Segmente zweier Ellipsen behandelt und können zu deren Parameterbestimmung herangezogen werden.

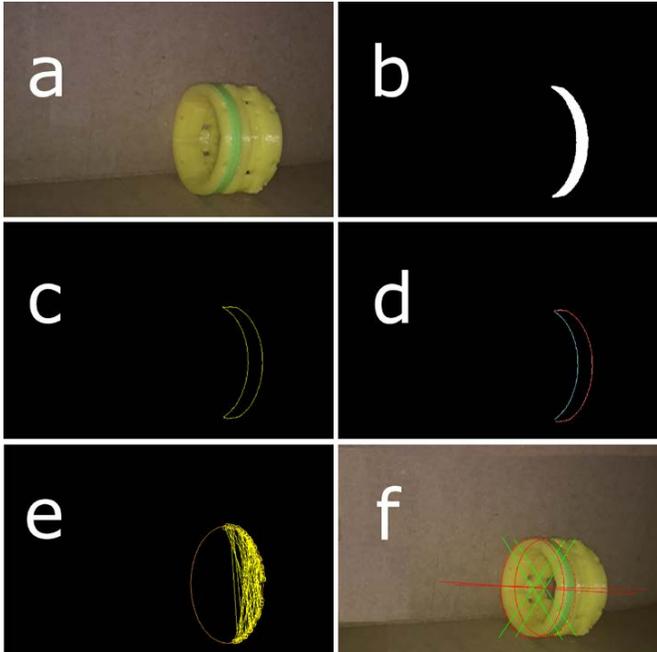


Abbildung 43: Teilschritte der Markererkennung: (a) Ausgangsbild, (b) Farbmaske, (c) Kontur, (d) Ellipsensegmente, (e) Detektierte Ellipse mit RANSAC, (f) Pose des erkannten Markers

Der ursprünglich von (Bader, 2014) entworfene Code nutzt die Methode der kleinsten Fehlerquadrate, um die Parameter einer Ellipse zu finden, die die vorgegebene Kontur möglichst exakt abbildet. Dieser Ansatz funktioniert sicher und effizient für geschlossene ellipsoide Konturen, versagt aber spätestens bei Teilverdeckungen der Ellipsensegmente.

Der deshalb gewählte Ansatz ist in Abbildung 43(e) zu sehen: Basierend auf dem RANSAC (Random Sample Consensus) Verfahren von (Fischler & Bolles, 1981) werden etwa 10% der Punkte einer Kontur zufällig ausgewählt und dem Algorithmus zur Parameterberechnung der Ellipse übergeben. Das Resultat wird mit sämtlichen Punkten des Ellipsensegments verglichen. Solange die Deckungsrate nicht einen definierten Schwellwert erreicht, wird die gewählte Punktuntermenge verworfen und zufällig eine neue bestimmt. Genügt das Ergebnis den Ansprüchen, gelten die ermittelten Ellipsenparameter als bestimmt. Auch dieser Schwellwert lässt sich zur Laufzeit über die Oberfläche in Abbildung 44 anpassen.

Die Berechnung der möglichen zur ermittelten Ellipse gehörenden Kreisposen ist prinzipiell auf verschiedene Weisen möglich. Der von (Chen, Wu, & Wada, 2004) verwendete Ansatz zeichnet sich dabei durch hohe Effektivität aus und wurde aus diesem Grund in der Implementierung von (Bader,

2014) übernommen. Dabei handelt es sich im Wesentlichen um eine dem Problem angepasste Hauptachsentransformation. Die Vorgehensweise ist im Folgenden kurz skizziert.

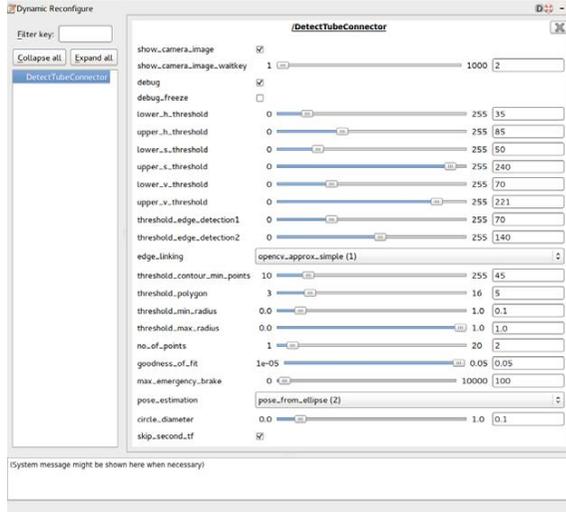


Abbildung 44: Konfigurationsoberfläche DetectTubeConnector

Jede im Videobild gefundene Ellipse kann durch die allgemeine Gleichung

$$Ax_e^2 + 2Bx_e y_e + Cy_e^2 + 2Dx_e + 2Ey_e + F = 0 \quad (5.49)$$

beschrieben werden. Dabei wird davon ausgegangen, dass sich der Ursprung der Ellipse im optischen Zentrum der Kamera befindet und die z-Achse der optischen Achse entspricht. Gleichung (5.49) lässt sich dabei auch in eine quadratische Form bringen:

$$\begin{pmatrix} x_e & y_e & 1 \end{pmatrix} \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = 0 \quad (5.50)$$

Stellen wir uns jetzt eine Linienschar vor, die sowohl durch das optische Zentrum der Kamera als auch die Ellipse geht, stellt diese einen schrägen elliptischen Kegel dar. Die Entfernung z der einzelnen Punkte zum optischen Zentrum entspricht dabei der Brennweite f der Kamera. In diesem Fall kann der Kegel wie folgt beschrieben werden:

$$\mathbf{P} = k(x_e \quad y_e \quad -f)^T \quad (5.51)$$

Der Skalierungsfaktor k beschreibt dabei die Entfernung vom Ursprung zu P .

Aus Gleichung (5.50) und (5.51) ergibt sich nun die Gleichung zur Beschreibung des schrägen elliptischen Kegels zu

$$\mathbf{P}^T \mathbf{Q} \mathbf{P} = 0; \quad \mathbf{Q} = \begin{pmatrix} A & B & -\frac{D}{f} \\ B & C & -\frac{E}{f} \\ -\frac{D}{f} & -\frac{E}{f} & \frac{F}{f^2} \end{pmatrix} \quad (5.52)$$

Der Ansatz von (Chen, Wu, & Wada, 2004) besteht jetzt darin, sich eine Hilfsebene mit dem ursprünglichen Kreis im Raum zu definieren. Der Koordinatenursprung dieser Ebene liegt ebenfalls im optischen Zentrum, die z -Achse entspricht jedoch der Flächennormalen der Kreisfläche. Der Kreis auf dieser Hilfsebene lässt sich dann durch folgenden Ausdruck beschreiben, wobei (x_0, y_0, z_0) den Ursprung beschreibt und r den Radius darstellt:

$$\begin{cases} (x - x_0)^2 + (y - y_0)^2 = r^2 \\ z = z_0 \end{cases} \quad (5.53)$$

Analog zu (5.52) lässt sich wieder eine Linienschar beschreiben, die sowohl durch das optische Zentrum der Kamera als auch den Kreis geht:

$$\mathbf{P}_C^T \mathbf{Q}_C \mathbf{P}_C = 0; \quad \mathbf{Q}_C = \begin{pmatrix} 1 & 0 & -\frac{x_0}{z_0} \\ 0 & 1 & -\frac{y_0}{z_0} \\ -\frac{x_0}{z_0} & -\frac{y_0}{z_0} & \frac{x_0^2 + y_0^2 - r^2}{z_0^2} \end{pmatrix} \quad (5.54)$$

Da der Ursprung der Koordinatensysteme beider Kegel im Ursprung des optischen Zentrums liegt, lässt sich das eine Koordinatensystem über eine einfache Drehung in das andere Koordinatensystem überführen:

$$\mathbf{P} = \mathbf{R}_c \mathbf{P}_c \quad (5.55)$$

Da $k\mathbf{Q}_c$ für alle $k \neq 0$ den gleichen Kegel wie \mathbf{Q}_c beschreibt, lässt sich basierend auf den Gleichungen (5.52), (5.54) und (5.55) folgende Beziehung aufstellen:

$$k\mathbf{R}_c^T \mathbf{Q} \mathbf{R}_c = \mathbf{Q}_c \quad (5.56)$$

Zur Ermittlung der gesuchten Kreisparameter müssen die Matrizen \mathbf{Q}_c und \mathbf{R}_c bekannt sein. Für deren Bestimmung ist es von Vorteil, wenn zuerst \mathbf{Q} in eine Diagonalform gebracht wird. Dazu werden vorab die Eigenwerte $\lambda_1, \lambda_2, \lambda_3$ und die normierten Eigenvektoren $\mathbf{v}_1 = (v_{1x} \ v_{1y} \ v_{1z})^T, \mathbf{v}_2 = (v_{2x} \ v_{2y} \ v_{2z})^T, \mathbf{v}_3 = (v_{3x} \ v_{3y} \ v_{3z})^T$ von \mathbf{Q} bestimmt. \mathbf{Q} lässt sich dann ausdrücken als

$$\mathbf{Q} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{-1} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T; \quad \begin{cases} \mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \lambda_3\} \\ \mathbf{V} = (\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3) \end{cases} \quad (5.57)$$

Die Verwendung von \mathbf{V}^T an Stelle von \mathbf{V}^{-1} wird durch die speziellen Eigenschaften der Orthonormalbasis ermöglicht, die durch die Normierung der Eigenvektoren entstanden ist.

Ersetzt man \mathbf{Q} in (5.56) durch das Äquivalent aus (5.57), erhält man

$$k\mathbf{R}^T \mathbf{\Lambda} \mathbf{R} = \mathbf{Q}_c \quad (5.58)$$

wobei gilt

$$\mathbf{R} = \begin{pmatrix} r_{1x} & r_{2x} & r_{3x} \\ r_{1y} & r_{2y} & r_{3y} \\ r_{1z} & r_{2z} & r_{3z} \end{pmatrix} = \mathbf{V}^T \mathbf{R}_c \quad (5.59)$$

Aus Gleichung (5.59) erhalten (Chen, Wu, & Wada, 2004) folgendes Gleichungssystem:

$$\begin{cases} \lambda_1(r_{1x}^2 - r_{2x}^2) + \lambda_2(r_{1y}^2 - r_{2y}^2) + \lambda_3(r_{1z}^2 - r_{2z}^2) = 0 \\ \lambda_1 r_{1x} r_{2x} + \lambda_2 r_{1y} r_{2y} + \lambda_3 r_{1z} r_{2z} = 0 \end{cases} \quad (5.60)$$

Auf Grund der besonderen Eigenschaften einer Diagonalmatrix lässt sich zusätzlich ohne Einschränkungen der Allgemeingültigkeit festlegen, dass

$$\begin{cases} \lambda_1 \lambda_2 > 0 \\ \lambda_1 \lambda_3 < 0 \\ |\lambda_1| \geq |\lambda_2| \end{cases} \quad (5.61)$$

Vereinfacht man jetzt Gleichung (5.60) und setzt $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ erhält man

$$\mathbf{R} = \begin{pmatrix} \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \cos(\alpha) & S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \sin(\alpha) & S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ \sin(\alpha) & -S_1 \cos(\alpha) & 0 \\ S_1 S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \cos(\alpha) & S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \sin(\alpha) & -S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix} \quad (5.62)$$

S_1 und S_2 stehen dabei für nicht definierte Vorzeichen. Setzt man \mathbf{R} jetzt in Gleichung (5.58) ein, ergeben sich erste Anhaltspunkte für die gesuchten Elemente von \mathbf{Q} aus (5.54):

$$\begin{cases} k = \lambda_2 \\ \frac{x_0}{z_0} = -S_2 \frac{\sqrt{(\lambda_1 - \lambda_2)(\lambda_2 - \lambda_3)} \cos(\alpha)}{\lambda_2} \\ \frac{y_0}{z_0} = -S_1 S_2 \frac{\sqrt{(\lambda_1 - \lambda_2)(\lambda_2 - \lambda_3)} \sin(\alpha)}{\lambda_2} \\ \frac{r^2}{z_0^2} = -\frac{\lambda_1 \lambda_3}{\lambda_2^2} \end{cases} \quad (5.63)$$

Da die z-Achse der Hilfsebene der Normalenvektor \mathbf{N} der Ebene ist, stellen (Chen, Wu, & Wada, 2004) basierend auf den Gleichungen (5.55), (5.59) und (5.63) das folgende Gleichungssystem auf, das sowohl \mathbf{N} als auch das Zentrum \mathbf{C} des projizierten Kreises beschreibt. Auch hier ist S_3 wieder ein unbestimmtes Vorzeichen:

$$\left. \begin{aligned}
 z_0 &= S_3 \frac{\lambda_2 r}{\sqrt{-\lambda_1 \lambda_3}} \\
 \mathbf{C} &= z_0 \mathbf{VR} \begin{pmatrix} x_0 \\ z_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} = z_0 \mathbf{V} \begin{pmatrix} S_2 \frac{\lambda_3}{\lambda_2} \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \frac{\lambda_1}{\lambda_2} \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix} \\
 \mathbf{N} &= \mathbf{VR} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{V} \begin{pmatrix} S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix}
 \end{aligned} \right\} \quad (5.64)$$

Da eine Fläche zwei Seiten hat, definieren wir \mathbf{N} derart, dass er zur Kamera hinzeigt. Da auch das Zentrum des Kreises \mathbf{C} vor der Kamera liegt, lässt sich folgende Beziehung aufstellen:

$$\begin{cases} \mathbf{N} \cdot (0 \ 0 \ 1)^T > 0 \\ \mathbf{C} \cdot (0 \ 0 \ 1)^T < 0 \end{cases} \quad (5.65)$$

Basierend darauf lassen sich zwei der drei unbestimmten Vorzeichen aus Gleichung (5.64) bestimmen, woraus sich zwei mögliche Sätze von \mathbf{N} und \mathbf{C} ergeben. Daraus folgt, dass prinzipiell zwei Kreise im Raum als möglicher Ausgang für die Projektion der detektierten Ellipsen existieren.

In der Konfigurationsoberfläche aus Abbildung 44 lässt sich festlegen, ob beide möglichen Ringposen an das System übermittelt werden, oder ob bereits in diesem Stadium eine Vorauswahl basierend auf a priori Wissen der Steuerungsdaten erfolgen soll. Zur weiteren Optimierung der Detektierung können außerdem verschiedene Filter zur Glättung der Ausgabewerte einbezogen werden.

Richtig parametrisiert läuft die Ringmarkerdetektion effizient und ressourcenschonend ab. Die Laufzeitkomplexität $O(n)$ hängt nach erfolgter Parameteroptimierung linear von der Anzahl n der erkannten Konturen ab. Abbildung 45 zeigt noch einmal den Ablaufplan der kompletten Node.

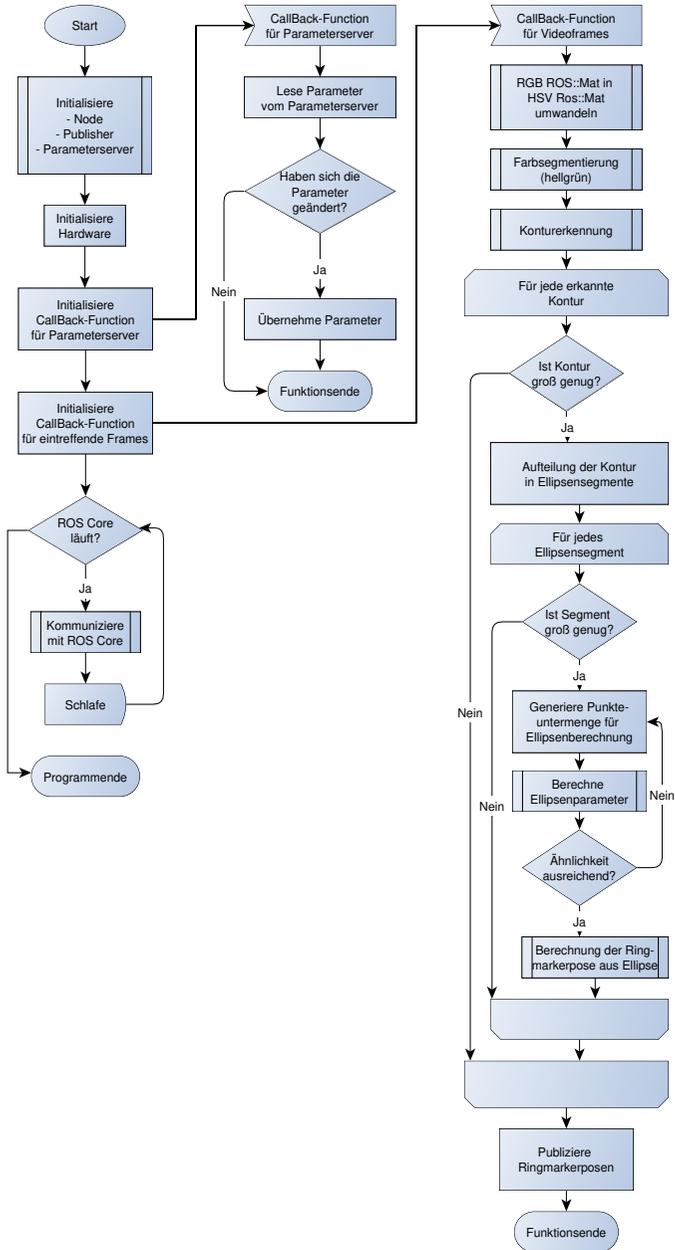


Abbildung 45: Programmablaufplan der DetectTubeConnector-Node

5.4.3 Objekterkennung: Aktive Konturen / „Snakes“

Nachdem in der Anfangsphase einfache und effiziente Segmentierungsalgorithmen auf der Basis von Kantendetektionsverfahren wie dem Canny-Algorithmus (Canny, Nov. 1986) oder der Nutzung des Sobel-Operators implementiert und evaluiert wurden, stand schnell fest, dass diese Verfahren eine gute und vor allem gleichmäßige Ausleuchtung benötigen, um zuverlässige Ergebnisse zu liefern. Sollte diese nicht gegeben sein, lassen sich die detektierten Kanten nicht mehr zuverlässig geschlossenen Bereichen zuordnen.

Die Verwendung von aktiven Konturen, so genannten „Snakes“, verspricht hier Abhilfe. Steht die grobe Lage durch einzelne Kontursegmente, die zum Beispiel mit Hilfe des Canny-Algorithmus ermittelt wurden, fest, lässt sich im entsprechenden Bereich eine parametrische, geschlossene Kurve definieren, die sämtliche Segmente umfasst. Diese Kurve korrigiert nun basierend auf so genannten gewichteten internen und externen Energien ihren Verlauf derart, dass die Summe der Energien minimal wird. Innere Energien sind dabei Parameter wie Kurvenlänge oder Kantenglattheit. Äußere Energien ergeben sich aus dem Helligkeitsgradienten der Bildvorlage.

Im Prinzip lässt sich der Vorgang vorstellen wie ein aufgeblasener, mit beliebigem Material gefüllter Luftballon, aus dem man die Luft langsam entweichen lässt: Durch die Spannung in seiner Oberfläche zieht er sich langsam zusammen, bis er von den gegenläufigen Energien des Materials in seinem Inneren gestoppt wird und die Energie des Gesamtsystems minimal ist. Im Gegensatz zum realen Luftballon lässt der Energieterm der aktiven Konturen je nach Aufbau aber auch ein weiteres Zusammenziehen an Hohlstellen zu.

Ein Beispiel aus der Anwendung im Bereich der Segmentierung medizinischen Bildmaterials zeigt Abbildung 46.



Abbildung 46: Beispiel für die Anwendung von Snakes auf Ultraschallbildern; Autor: (Gegler, 2005)

Durch die Fähigkeit, optische Lücken im Bildmaterial überbrücken zu können, stellen die aktiven Konturen ein effektives Mittel dar, eine geschlossene Kurve um ein detektiertes Objekt zu legen. Leider ist der Vorgang insbesondere bei suboptimal gewählten Startwerten sehr rechenintensiv und dadurch für den hier angestrebten Echtzeiteinsatz nicht zu verwenden, so dass eine Alternative gefunden werden musste. Diese findet sich bei der Methode der so genannten Support Vector Machines (SVM), die im Folgenden kurz beschrieben wird.

5.4.4 Objekterkennung: Support Vector Machines (SVM)

Da bei der Objekterkennung weder auf feste Konturen, noch optische Marker gesetzt werden kann, erfolgt die Detektion des Stiff-Flop Armes durch ein texturbasiertes Verfahren. Zum Einsatz kommt ein Large Margin Classifier, eine so genannte Support Vector Machine (SVM). Diese klassifiziert kleine quadratische Bildausschnitte als „Objekt“ oder „Hintergrund“ und ermöglicht dadurch eine grobe Lageermittlung des gesuchten Körpers.

Damit eine SVM tätig werden kann, muss sie im Vorfeld mit Bildausschnitten trainiert werden, die eindeutig einer der beiden Klassen „Objekt“ oder „Hintergrund“ zugeordnet werden können. Diese Ausschnitte stellen jeweils einen Vektor im n -dimensionalen Raum dar; beispielsweise ist bei einer gewählten Ausschnittgröße von 25×25 Bildpunkten $n=625$. Durch diesen Raum wird nun im Vorfeld eine Hyperebene derart gelegt, dass sie die beiden Klassen sauber voneinander trennt und dabei der Abstand derjenigen Vektoren, die der Hyperebene am nächsten liegen, zu dieser maximiert wird. Dieser möglichst breite Rand hilft dabei, Objekte, die nicht genau den Trainingsobjekten entsprechen, möglichst zuverlässig zu klassifizieren. Die Idee der Support-Vector-Machines geht dabei auf die Arbeit von (Vapnik & Chervonenkis, 1974) zurück.

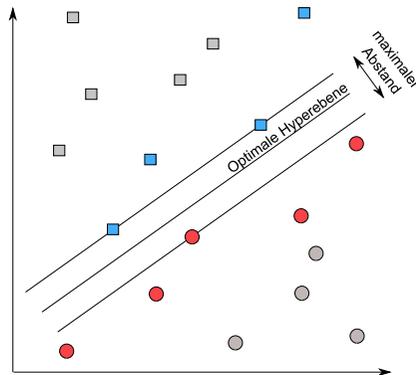


Abbildung 47: Hyperebene zur Separierung der Objektklassen einer SVM

Hyperebenen können nicht „verbogen“ werden. Aus diesem Grund ist die genannte Vorgehensweise nur praktikabel, wenn sämtliche Objekte linear trennbar sind. Ist dieses nicht der Fall, kommen so genannte Kernels zum Einsatz. Der aufgespannte Vektorraum wird so lange in einen Raum der nächsthöheren Ordnung transformiert, bis eine lineare Trennung möglich ist. Danach erfolgt die Rücktransformation in den ursprünglichen Vektorraum. Dieser Vorgang ist sehr rechenintensiv. Außerdem nehmen die gefundenen Hyperebenen bei Rücktransformation in den niedrigdimensionaleren Raum extrem an Komplexität zu. Diese Komplexitätssteigerung kann durch Verwendung geeigneter Kernelfunktionen in Grenzen gehalten werden.

Auf die konkrete Implementierung der Support Vector Machines soll hier auf Grund der hohen Komplexität nicht eingegangen werden.

Für die Beurteilung des Gesamtsystems soll aber nicht unerwähnt bleiben, dass die Laufzeitkomplexität $O(n)$ linear von der Anzahl n der Stützvektoren abhängt. Stützvektoren sind nur diejenigen Vektoren des Training-Sets, die für die Berechnung der Hyperebene relevant sind. In Abbildung 47 wurden sowohl Vorder- als auch Hintergrundstützvektoren farbig markiert. Die verworfenen grauen Vektoren stützen die Hyperebene nicht, deshalb werden sie nicht in die Stützvektordatenbank aufgenommen und tragen somit weder zur Qualität der Klassifizierung bei, noch wirken sie sich auf die Laufzeit des Algorithmus aus. Im Sinne einer guten Objekterkennung und eines effektiven Trainingszyklus sollte bei der Auswahl der Bildausschnitte also darauf geachtet werden, dass für beide Objektklassen möglichst viele stark unterschiedliche Bildausschnitte verwendet werden.

5.4.5 Objektvermessung

Ist die Lage des Objektes im Videobild erst einmal ermittelt, kann mit der Lageermittlung im Raum begonnen werden. Im Regelfall wird dazu ein Stereokamerasystem benötigt, dass im vorliegenden Szenario leider nicht verfügbar ist. Es wird daher versucht, die Pose des Stiff-Flop Armes im Raum zumindest möglichst exakt zu schätzen, um so die Überwachung der geforderten No-Fly-Zones - also für den Arm verbotener Bereiche - sicherstellen zu können.

Abbildung 48 zeigt auf der linken Seite das Ausgangsmaterial, also die vom SVM-Algorithmus ermittelte Lage des Armes. Die einzelnen Bildpunkte der Maske werden als Punktwolke interpretiert, durch deren Längsachse im ersten Schritt abschnittsweise ein Spline gelegt wird. Dieser Schritt dient der Glättung der Maske und somit auch der späteren Filterung der Ausgabewerte für die Mittelachse des Armes und ist notwendig, da zum einen gerade in den Randbereichen des zylindrischen Schlauches eine scharfe Trennung von Hintergrund und Silikonschicht durch Oberflächeneffekte nicht möglich ist und sich zum anderen durch die Anwendung des SVM-Verfahrens an sich eine starke Treppenbildung an den Kanten ergibt. Um den Rechenaufwand möglichst gering zu halten,

kommt nur jeder 10 Punkt sowohl in x- als auch in y-Richtung zum Tragen. Die resultierende Mittellinie ist auf der rechten Seite von Abbildung 48 zu sehen (rote Kurve).

Zur Ermittlung der Lage im Raum ist neben der Mittellinie auch der Durchmesser des detektierten Objektes relevant. Aus diesem Grund wird die Mittellinie von Anfang bis Ende in kleinen Schritten durchlaufen und für jeden Punkt der Durchmesser in Bildpunkten ermittelt. Dieser Vorgang erfolgt iterativ; zuerst wird eine zur Mittellinie im betrachteten Punkt orthogonale Gerade berechnet. Entlang dieser Gerade wird untersucht, welche Bildpunkte der Maske noch zum Objekt gehören und ab welcher Position der Hintergrund beginnt. Die ermittelten Durchmesser sind ebenfalls auf der rechten Seite von Abbildung 48 zu sehen (pinke Geraden).

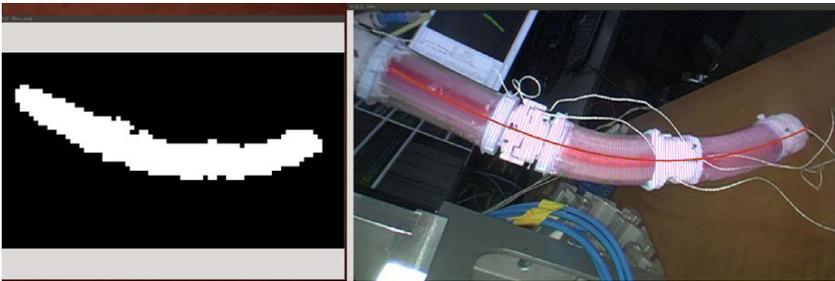


Abbildung 48: Objektvermessung; links: Vom SVM Algorithmus ermittelte Objektposition, rechts: Ermittelte Mittellinie des Stiff-Flop Armes mit eingezeichneten Radien; Autor: Erwin Gerz

Um von der Breite des Objektes in Bildpunkten auf dessen Entfernung zur Bildebene schließen zu können, ist es notwendig, sich erst über die Entfernung bei 100% Bildschirmfüllung Gedanken zu machen. Der Vorteil liegt dabei auch darin, dass die hierbei ermittelten Werte einfach empirisch überprüft werden können.

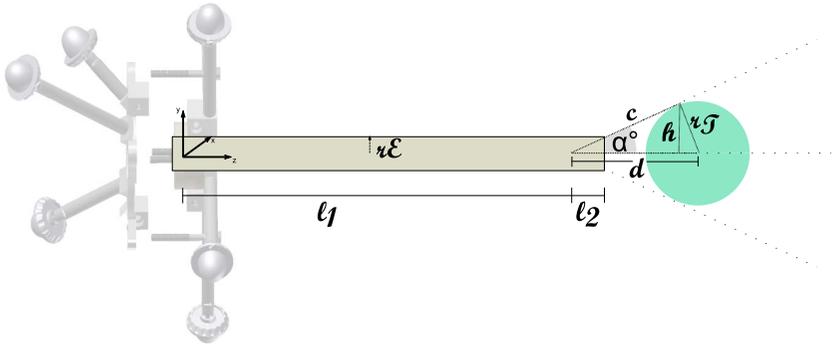


Abbildung 49: Geometrischer Aufbau der Endoskopkamera

Betrachtet man den Aufbau der Endoskopkamera in Abbildung 49, lassen sich einige geometrische Beziehungen aufstellen:

$$l_2 = \frac{r_E}{\sin(\alpha)} \quad (\text{Abstand Linse-Schaft}) \quad (5.66)$$

$$c = \frac{r_T}{\tan(\alpha)} = \sqrt{d^2 - r_T^2}$$

Aus (5.66) folgert direkt durch Auflösen nach dem Objektabstand d:

$$d = \sqrt{\frac{r_T^2 + r_T^2 * \tan^2(\alpha)}{\tan^2(\alpha)}} = \frac{r_T}{\tan(\alpha)} * \sqrt{1 + \tan^2(\alpha)} \quad (5.67)$$

Die wichtigsten Parameter sind aus dem Datenblatt des Endoskopschaftes bekannt:

$$r_E = 5\text{mm}$$

$$\alpha = 20^\circ$$

Hinzu kommt der Radius des Testkörpers:

$$r_T = 15\text{mm}$$

Weitere Parameter lassen sich durch Einsetzen in (5.66) sowie (5.67) ermitteln:

$$l_2 = 14,61\text{mm}$$

$$d = 43,85706\text{mm}$$

Sind diese Parameter bekannt, kann unkompliziert vom Durchmesser eines zylindrischen Objektes auf dessen Entfernung geschlossen werden:

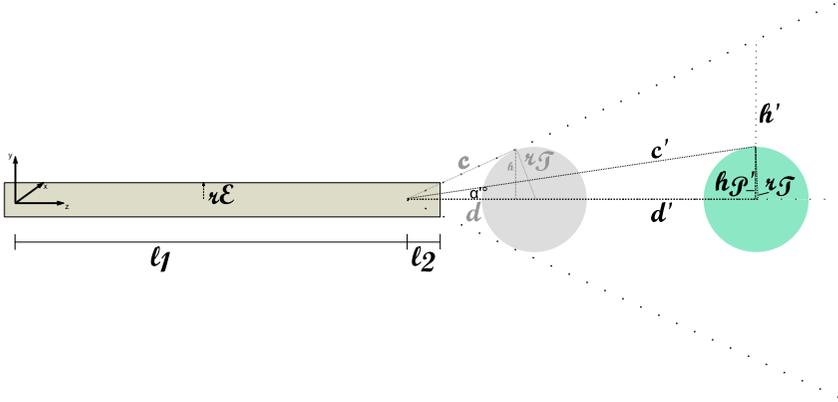


Abbildung 50: Geometrische Entfernungsberechnung eines Zylinders zur Bildebene

$$d' = \frac{\sqrt{r_T^2 + r_T^2 * \tan^2(\alpha')}}{\tan(\alpha')} = \frac{r_T}{\tan(\alpha')} * \sqrt{1 + \tan^2(\alpha')} \quad (5.68)$$

α' ist zu diesem Zeitpunkt unbekannt, kann aber über ein Verhältnis von $\frac{h'_p}{h'}$, also der relativen Bildschirmfüllung, ausgedrückt werden:

$$\tan(\alpha') = \frac{h'_p}{h'} * \tan(\alpha); \quad -\frac{\pi}{2} < \alpha' < \frac{\pi}{2} \quad (5.69)$$

Damit ergibt sich die Distanz eines Zylinders mit bekanntem Durchmesser bei ebenfalls bekannter Bildschirmfüllung zu:

$$\begin{aligned} d' &= \frac{r_T}{\frac{h'_p * \tan(\alpha)}{h'}} * \sqrt{1 + \left(\frac{h'_p * \tan(\alpha)}{h'}\right)^2} \quad (5.70) \\ &= \frac{r_T}{h'_p * \tan(\alpha)} * \sqrt{h'^2 + h'^2_p * \tan^2(\alpha)}; \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \\ &= \frac{r_T}{\tan(\alpha)} * \sqrt{\left(\frac{h'}{h'_p}\right)^2 + \tan^2(\alpha)}; \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \end{aligned}$$

Dabei ist $\frac{h'}{h'_p}$ die relative Bildschirmfüllung.

Über den Strahlensatz lässt sich die Entfernung eines Objektes direkt über die Bildschirmfüllung und somit seine Ausdehnung in Bildpunkten herleiten. Dabei wird eine quadratische Form der einzelnen Bildpunkte angenommen, was auf die verwendete Endoskopkamera zutrifft.

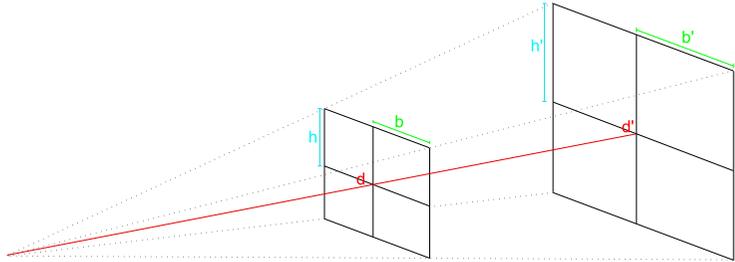


Abbildung 51: Darstellung der Parameter zur vereinfachten Abstandsberechnung mit Hilfe des Strahlensatzes

Bei voller Bildschirmfüllung in horizontaler Richtung, also $d=43,857$ mm, ergibt sich h zu

$$h=14,0953893118 \text{ mm}$$

Umgerechnet auf die maximale Bildschirmauflösung in vertikaler Richtung ergibt sich ein Umrechnungsfaktor von Höhe oder Breite des Objektes zu Bildschirmpixeln zu

$$q_h = q_b = \frac{540\text{px}}{14,0954\text{mm}} = 38,3104 \frac{\text{px}}{\text{mm}} \quad (5.71)$$

Als Resultat folgt direkt bei bekanntem Durchmesser r_T des Zylinders und relativer

Bildschirmfüllung von $\frac{h'}{h'_p}$ im Videobild:

$$d' = 41,2122 * \sqrt{\left(\frac{h'}{h'_p}\right)^2 + 0,132} \quad (5.72)$$

Da, wie bereits erwähnt, bei der Endoskopkamera von quadratischen Pixeln ausgegangen werden kann, gilt diese Formel nicht nur bei vertikaler Ausrichtung des Prüflings, sondern ist bei jeder Lage anwendbar.

Betrachtet man die Laufzeit des verwendeten Algorithmus, fällt auf, dass die eigentliche Vermessung des Testkörpers an einer beliebigen Stelle direkt von der Zeilenzahl der Kamera n als auch vom Bildschirmfüllverhältnis m abhängt: $O(n * f(m))$

Ist der Durchmesser des Zylinders erst einmal ermittelt, ergibt sich das Laufzeitverhalten der reinen Abstandsbestimmung zu $O(1)$ und ist somit als konstanter Faktor unkritisch. Das Gesamtlaufzeitverhalten verbleibt also bei $O(n * f(m))$.

Bereits im Vorfeld war abzusehen, dass die Verwendung einer einzelnen Endoskopkamera große Auswirkungen auf die erzielbare Präzision der Messung haben würde. Diese Vermutung bestätigt sich durch Testmessungen und Vergleich mit einem in seiner Position und Orientierung bekannten Testkörper. Das Resultat zeigt Abbildung 52. Die Visualisierung des Testkörpers erfolgt durch die gelbe Linie, die Ergebnisse des Vermessungsalgorithmus sind blau dargestellt.

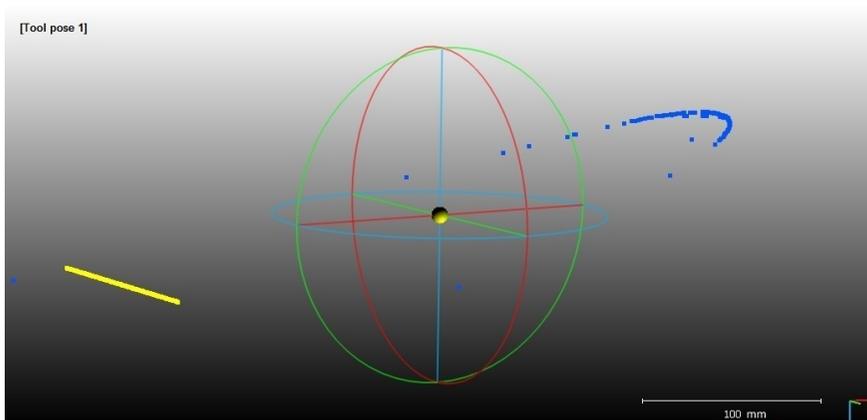


Abbildung 52: Ergebnisse des Vermessungsalgorithmus (blaue Punkte) - die reale Position des Stiff-Flop Armes ist gelb eingezeichnet; Quelle: (Elrazeq, 2015)

Auffällig hier ist – wie zu erwarten - die hohe Genauigkeit in X- und in Y-Richtung sowie extreme Verzerrungen in Z-Richtung. Hinzu kommt ein Offset (hauptsächlich in Z-Richtung), der durch das verwendete Zoom-Objektiv oder Fertigungstoleranzen des betrachteten Silikonschlauches hervorgerufen werden könnte. Zur genaueren Untersuchung der Lage erfolgt eine Abbildung in der X-Z-Ebene, siehe Abbildung 53.

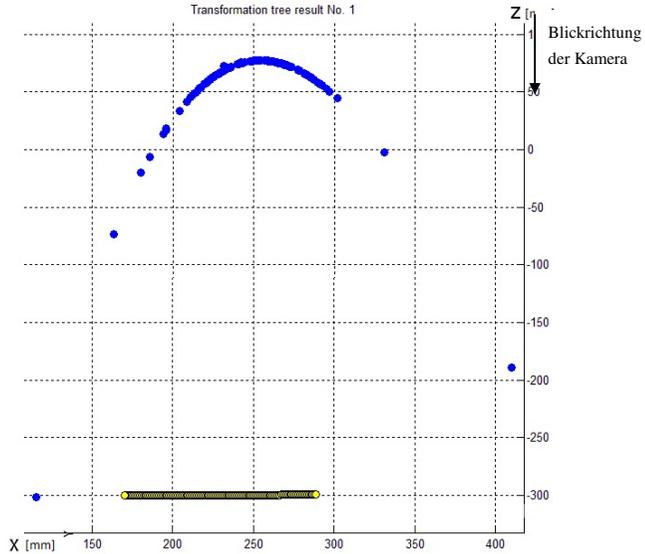


Abbildung 53: Ergebnisse des Vermessungsalgorithmus in der X-Z-Ebene (blaue Punkte) - die reale Position des Stiff-Flop Armes ist gelb eingezeichnet; Quelle: (Elrazeq, 2015)

Auffällig ist außerdem die in nahezu jeder Situation auftretende Biegung der erkannten Mittellinie an den äußeren Bereichen. Diese verläuft stets vom Betrachter weg. Dieses Verhalten ist bei nicht kalibrierten Kamerabildern zu erwarten; nach intensiver Prüfung konnte aber im vorliegenden Fall eine andere Fehlerquelle ausfindig gemacht werden: Bedingt durch die ungleichmäßige und hochgradig punktuelle Ausleuchtung der Szene werden die zentrumsfernen Bildbereiche kontrastärmer erfasst. Zusammen mit einem durch das komplexe Linsensystem bedingten Schärfeverlustes an der Peripherie des Bildes verliert der Segmentierungsalgorithmus mit zunehmendem Abstand vom Zentrum des Bildes einige Bildpunkte bei der Trennung von Vorder- und Hintergrund. Diese teils sehr geringen Abweichungen führen bei der Abstandsberechnung zu Fehlern von bis zu 5cm.

Der konstante Offset in X- und Y-Richtung lässt sich hauptsächlich auf die nachträglich versteifte Verbindung von Kameramodul, Objektiv und Linsensystem zurückführen (s. Seite 58). Es ist anzunehmen, dass eine Kalibrierungsmessung mit nachfolgender Feinjustierung diese Probleme beseitigt. Alternativ können die Kalibrierungsdaten dem Transformationsbaum (s. Seite 26) zur Verfügung gestellt werden. Der Vorteil dieser Vorgehensweise liegt darin, dass eventuelle Rekalibrierungen ohne Änderungen an der Mechanik erfolgen können. Die angewandte Vorgehensweise wird im nächsten Kapitel detailliert beschrieben.

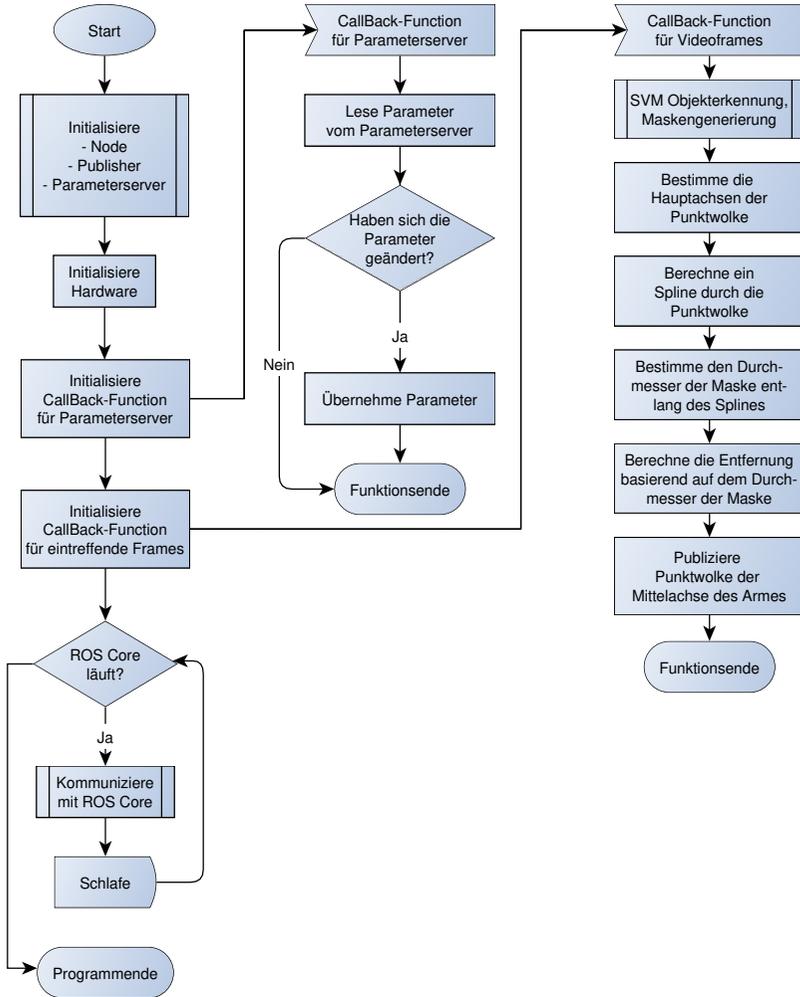


Abbildung 54: Programmablaufplan der Objektvermessungsnodes TubeDetect

5.5 Vermessung der optischen Komponenten

Die Kalibrierung des Systems erfolgt durch Vermessung eines Stiff-Flop Armes in einer Halterung, die innerhalb des Messvolumens des optischen Sensors positioniert wird. Da die Qualität der Positionsmessung voraussichtlich sowohl von der Entfernung des Objektes zur Kamerabildebene als auch von der Abweichung zur optischen Achse abhängt, erfolgt die Messwertaufnahme an einer Serie von definierten Positionen im Messvolumen. Über einen 3D-Registrierungsalgorithmus werden die ermittelten Mittellinien des Armes mit der bekannten Mittellinie verglichen und ein Korrekturwert errechnet, der dem Transformationsbaum des Systems (s. Seite 26) zur Verfügung gestellt wird.

Die Positionierung des Testmusters erfolgt durch einen KUKA KR16 Industrieroboter, wodurch automatisierte Messwertaufnahmen und vor allem reproduzierbare Ergebnisse möglich sind. Den Aufbau sowie die verwendeten Koordinatensysteme zeigt Abbildung 55.

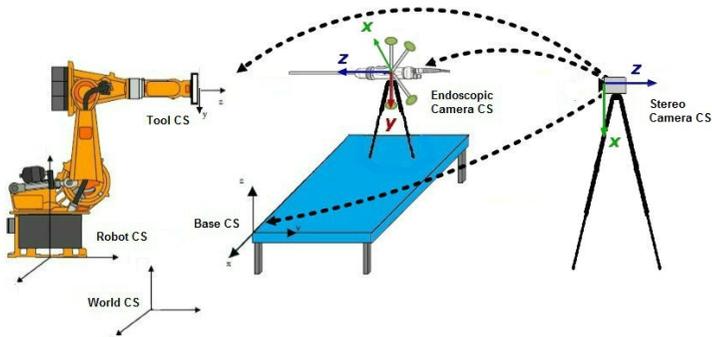


Abbildung 55: Aufbau des Kalibrierungssystems mit verwendeten Koordinatensystemen; Quelle: (Elrazeq, 2015)

Die nach dem in Kapitel 5.4.5 beschriebenen Verfahren ermittelte Mittellinie des Armes wird mittels eines Point Cloud Matching (PCM) Algorithmus mit der vom Cambar System ermittelten, korrekten Position verglichen. Implementiert und bewertet wurden zu diesem Zweck ein *Iterative Closest Point (ICP)* Ansatz wie in (Bellekens, Spruyt, Berkvens, & Weyn, 2014) beschrieben und die *Aligning object templates to a point cloud* Methode nach (Rusu, 2009).

5.5.1 Iterative Closest Point (ICP)

Der *Iterative Closest Point Algorithmus (ICP)* schätzt, basierend auf dem Aufbau zweier Punktwolken, die Parameter für eine Transformation, welche diese aufeinander abbildet. Dabei kommt die Methode der kleinsten Quadrate im Gauß-Markov-Modell (Gauss, 1821) zum Einsatz.

Ausgehend von einer Punktwolke $P\{p_i\}$ mit N_p Elementen und einer Referenzpunktwolke $X\{x_i\}$ mit N_x Elementen sollen die für eine Helmert-Transformation notwendigen Parameter bestimmt werden.

Die Helmert-Transformation (nach Friedrich Robert Helmert, 1843–1917) nutzt zur verzerrungsfreien Transformation von einem dreidimensionalen Koordinatensystem \mathbf{X} in ein anderes, ebenfalls dreidimensionales Koordinatensystem \mathbf{X}_T insgesamt 7 Parameter: Die Rotation um drei Raumachsen, die Verschiebung entlang dieser Achsen sowie einen Skalierungsfaktor.

$$\mathbf{X}_T = \mathbf{C} + \mu \mathbf{R} \mathbf{X} \quad (5.73)$$

\mathbf{C} = Translationsvektor

μ = Skalierungsfaktor

\mathbf{R} = Rotationsmatrix

Die Ermittlung der benötigten Parameter erfolgt durch die Minimierung der mittleren quadratischen Abweichung (mean square error, MSE). Der Skalierungsfaktor kann im vorliegenden Fall auf 1 gesetzt werden, da die Größe der Punktwolke und des Modells identisch sind.

$$MSE = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{x}_i - \mathbf{R}(p_i) - \mathbf{C}\|^2 \quad (5.74)$$

Nach einer Initialisierung der Startwerte ($\mathbf{P}_0 = \mathbf{P}$, $\mathbf{R}_0 = \mathbf{E}_3$, $\mathbf{T}_0 = \mathbf{0}$) durchläuft der ICP-Algorithmus für jede Iteration k folgende Schritte:

1. Finde für jeden Punkt p_i den Punkt aus X , der ihm am nächsten ist (Euklidischer Abstand)
2. Berechne die Translation \mathbf{C} und die Rotation \mathbf{R}
3. Wende diese Transformation auf \mathbf{P} an und ermittle die nächste Startpunktwolke:
4. $\mathbf{P}_{k+1} = \mathbf{R}_k(\mathbf{P}_k) + \mathbf{C}_k$
5. Wiederhole die vorangegangenen Schritte so lange, bis eine der folgenden Abbruchbedingungen eintritt:
 - Der MSE hat einen definierten Grenzwert unterschritten
 - Die Anzahl der Iterationen hat einen Grenzwert überschritten

- o Der Unterschied einer Transformation zu ihrem Vorgänger ist kleiner als ein festgelegter Grenzwert

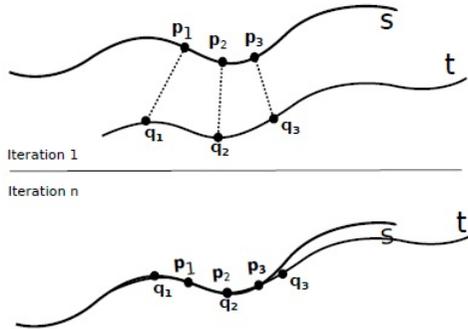


Abbildung 56: ICP Ausrichtung basierend auf einem Punkt-zu-Punkt-Ansatz; Quelle: (Bellekens, Spruyt, Berkvens, & Weyn, 2014)

Zur Suche des Nachbarpunktes mit dem geringsten Abstand stehen verschiedene Methoden zur Verfügung. Eine sequentielle Suche durch direkten Vergleich jedes einzelnen Abstandes ist einfach zu implementieren, aber extrem ineffizient: Die Laufzeitkomplexität liegt für etwa identisch große Punktwolken bei $O(N^2)$. Deutlich effizienter ist die Verwendung eines unbalancierten Suchbaumes. Beim k-dimensionalen Baum (k-d-Baum) handelt es sich um einen binären Suchbaum. Hier kommt die inhomogene Variante zum Einsatz, das heißt, dass die Daten lediglich in den Blättern der Baumstruktur gespeichert sind. Die Knoten tragen die Schlüssel, die mit dem Suchkriterium verglichen werden. Die Laufzeitkomplexität gestaltet sich hier deutlich günstiger zu $O(\log(N))$.

5.5.2 Aligning object templates to a point cloud

Die zweite implementierte Methode ist der „aligning object templates to a point cloud“ Algorithmus der Point Cloud Library, der im Wesentlichen auf einem automatischen Registrierungsansatz von (Rusu, 2009) beruht. Um Mehrdeutigkeiten zu vermeiden, kommen nicht nur Positionen im 3D-Raum, sondern üblicherweise zusätzliche Merkmale wie Intensität, Farbe etc. zum Einsatz. Je nach Datenquelle sind diese Informationen aber nicht verfügbar, so dass man sich um weitere Merkmale bemühen muss.

(Rusu, 2009) führt zu diesem Zweck das „Neighborhood“ Konzept ein:

1. Suche eine Punktmenge P^k in der Umgebung eines Punktes P_q
2. Ermittle die Abstände zwischen dem Punkt P_q und sämtlichen anderen Punkten in P^k
3. Die k nächstgelegenen Punkte werden nach ihrem Abstand zu P_q sortiert, beginnend vom kleinsten Wert
4. Der Positionsvektor von P_q wird erweitert auf $F(P_q, P^k)$

Über eine Abstandsfunktion lässt sich nun die Ähnlichkeit der Punkte mit erweiterten Merkmalen vergleichen. Das Neighbourhood-Konzept bewährt sich vor allem bei starren Bewegungen, variierender Punktdichte sowie verrauschten Datensätzen. Mit steigender Punktezahl nehmen jedoch sowohl Laufzeit als auch Speicherverbrauch exponentiell zu, so dass eine Einschränkung der „Nachbaranzahl“ sinnvoll erscheint. (Rusu, 2009) nimmt zu diesem Zweck nur noch Nachbarpunkte in den Merkmalsvektor auf, die in einem festzulegenden Umkreis von P_q liegen:

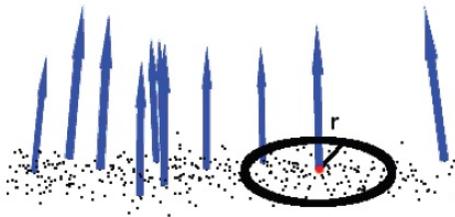


Abbildung 57: Radiusuche nach Nachbarpunkten zu P_q in P^k innerhalb des Radius r ; Quelle: (Rusu, 2009)

Um die Ausrichtung der beiden Punktwolken in den folgenden Schritten zu erleichtern und die Rechenkomplexität zu reduzieren, wird außerdem ein *Point Feature Histogram* (PFH) erstellt.

Ein PFH basiert im Ursprung auf den Verhältnissen aller Punkte der k -Nachbarschaft zu einander sowie ihren geschätzten Flächennormalen bei vollständiger Vernetzung:

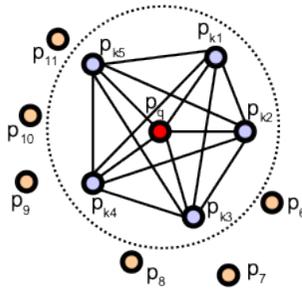


Abbildung 58: Vollständiges Netz der im Radius r gefundenen Punkte; Quelle: (Rusu, 2009)

Im nächsten Schritt wird ein festes Koordinatensystem definiert, um die Transformation von einem Punkt P_s zu einem Punkt P_t bzw. ihrer ermittelten Normalen n_s und n_t zu berechnen:

$$\begin{aligned}
 \mathbf{u} &= \mathbf{n}_s & (5.75) \\
 \mathbf{v} &= \mathbf{u} \times \frac{(\mathbf{p}_t - \mathbf{p}_s)}{\|\mathbf{p}_t - \mathbf{p}_s\|_2} \\
 \mathbf{w} &= \mathbf{u} \times \mathbf{v}
 \end{aligned}$$

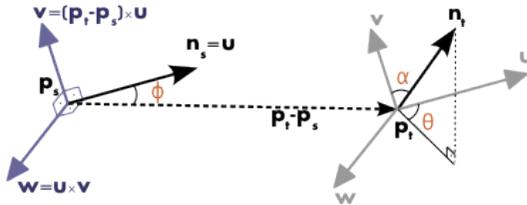


Abbildung 59: Transformationsvorschrift der Normalenvektoren; Quelle: (Rusu, 2009)

Durch die Nutzung dieses uvw-Koordinatensystems können die Unterschiede zwischen den Normalenvektoren durch einen Satz winkelbasierter Merkmale α, ϕ, θ sowie einem Abstand d ausgedrückt werden. Ein winkelbasiertes Merkmal (Feature) ist eine Kenngröße, die eine einfache Ähnlichkeitsbetrachtung zweier gegeneinander verdrehter Objekte ermöglicht. Zur Vereinfachung

der Rechnung werden dabei nicht die geometrischen Winkel verwendet, da sie für eine Ähnlichkeitsbetrachtung unerheblich sind:

(5.76)

$$d = \|p_t - p_s\|_2$$

$$\alpha = v \bullet n_t$$

$$\phi = u \bullet \frac{(p_t - p_s)}{d}$$

$$\theta = \arctan(w \bullet n_t, u \bullet n_t)$$

Durch diese Abbildung reduziert sich die Anzahl der Merkmale jeder Punktepaarung von 12 (pro Punkt 3xPosition und 3xOrientierung) auf insgesamt 4. Für die endgültige PFH Repräsentation werden sämtliche zuvor ermittelten Quadrupel in ein Histogramm abgelegt. Dabei wird jeder Wertebereich eines Merkmals in b Bereiche aufgeteilt. Jeder dieser b Bereiche wird wiederum in die gleiche Anzahl an Unterbereichen aufgeteilt. Jede dieser b^4 Klassen repräsentiert also genau einen Punkt mit bestimmten Eigenschaften.

Abbildung 60 zeigt exemplarisch Beispiele für PFH Darstellungen zweier Punkte in einer Szene:

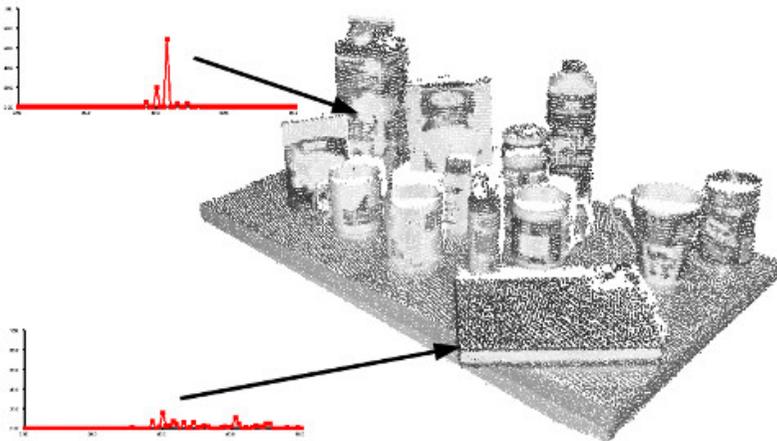


Abbildung 60: Beispiele für PFH Darstellungen zweier Punkte einer Szene; Quelle: (Rusu, 2009)

Die eigentliche Ausrichtung der Punktwolke erfolgt dann über die folgenden Schritte:

1. Der Algorithmus sucht nach ähnlichen Punktmerkmalen in Quelle und Ziel.
2. Im nächsten Schritt wird die Entfernung im PFH berechnet. Dabei kommt ein n-dimensionalen k-d-Baum zum Einsatz, um die Rechenlast möglichst gering zu halten.
3. Die k Punkte mit den ähnlichsten Histogrammen werden als Korrespondenzkandidaten ausgewählt.
4. Diese Schritte werden so lange wiederholt, bis keine weiteren sinnvollen Korrespondenzen mehr gefunden werden.
5. Die gefundenen Punktkorrespondenzen dienen als Grundlage zur Berechnung der Transformationsvorschrift

Die verschiedenen Parameter des Algorithmus wurden bei der Implementierung von (Elrazeq, 2015) wie folgt festgelegt:

- Flächennormalensuchradius: Dieser Parameter definiert das Kugelvolumen, in dem andere Punkte zur Erweiterung der Punktmerkmale herangezogen werden. Die Vergleichspunktwolke hat eine Ausdehnung von 12 cm an der weitesten Stelle. Der Abstand der Einzelpunkte liegt zwischen 1 mm und 10 mm. Aus diesem Grund wird der Suchradius auf 2 cm festgelegt, um eine möglichst hohe Detailtreue zu erreichen.
- Der Mindestabstand zwischen zwei Punkten wird zur Rechenleistungsreduktion auf 2 mm festgelegt.
- Maximale Distanz zwischen zwei korrespondierenden Punkten: Ist der Abstand zweier korrespondierender Punkte größer als ein Grenzwert, werden diese bei der Berechnung der Transformation nicht berücksichtigt. Der Wert kann im vorliegenden Fall beliebig groß gewählt werden, da sämtliche Punkte berücksichtigt werden sollen.
- Die maximale Iterationszahl wird auf 300 gesetzt, danach wird der Algorithmus mit den bis dato ermittelten Werten beendet.

Die Laufzeitkomplexität der Point-Feature-Histogramms ergibt sich zu $O(k^2)$, wobei k, wie oben beschrieben, der Anzahl der Punktkorrespondenzen entspricht.

5.5.3 Auswertung der Ergebnisse

Die zuvor beschriebenen Point Cloud Matching Verfahren liefern im Ergebnis eine Transformation zwischen korrekter und ermittelter Punktwolke sowie einen Qualitätsfaktor, der beschreibt, wie gut die eine Punktwolke in die andere überführt werden konnte. Dieser Faktor liefert die Grundlage für die Entscheidung, ob die Kalibrierfahrt des Roboters mit geänderten Parametern wiederholt wird, oder ob die Erkennung als präzise genug erachtet wird. Im letzten Fall wird die ermittelte Transformation dem TF-Tree hinzugefügt und steht ab diesem Zeitpunkt sämtlichen Vermessungsalgorithmen zur Verfügung.

5.6 Parallelisierungsmöglichkeiten

Zu Beginn des Projektes wurde intensiv über die Verwendung von Multithreading zur größtmöglichen Ausnutzung der 12 Prozessorkerne des Rechnersystems nachgedacht. Der Programmcode der Intensity Node wurde mit Hilfe der OpenMP-Bibliothek² in den rechenintensiveren Teilen parallelisiert. Andere Nodes boten auf Grund der Abhängigkeit der einzelnen Berechnungsschritte von einander weniger Möglichkeiten der Optimierung.

Mit fortlaufender Implementierung der unterschiedlichen Nodes wurde gänzlich auf eine Parallelisierung des Codes verzichtet. Die Gründe dafür sind vielfältig:

- Steigender Verwaltungsaufwand des Betriebssystems bei wachsender Anzahl an Threads: Aufteilung auf zwei Prozessorkerne bedeutet nicht die doppelte Ablaufgeschwindigkeit (Amdahl, 1967), s. Abbildung 61
- Arbeitsschritte der Algorithmen der meisten Nodes sind streng sequentiell aufgebaut und lassen sich auf Grund des hohen Synchronisierungsaufwandes schlecht parallelisieren
- Parallelisierter Code ist aufwändiger zu debuggen
- Durch eine Aufteilung der Gesamtfunktionalität des Vision Systems auf mehrere Nodes erfolgt eine native Lastverteilung auf sämtliche Cores; eine Aufteilung einzelner Nodes auf mehrere Prozessorkerne ist somit unnötig.

² Bibliothek zur einfachen Aufteilung von Programmcode auf mehrere Threads, <http://www.openmp.org>

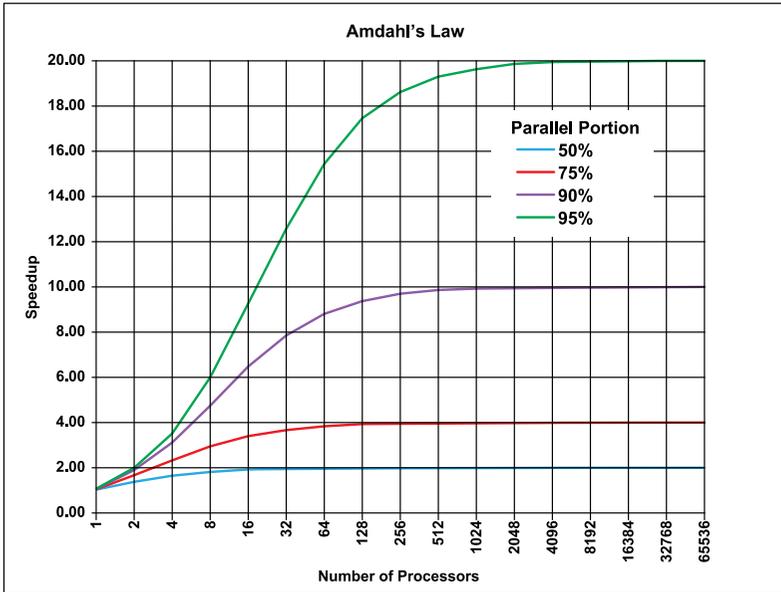


Abbildung 61: Amdahls Gesetz, Autor: Schwammerl-Bob (CC BY-SA 3.0)

Eigene Zeit- bzw. Auslastungsmessungen wurden nicht gemacht. Es wurde lediglich darauf geachtet, dass das Gesamtsystem die Bilder der Endoskopkamera kontinuierlich mit einer Bildrate von 25 Hz verarbeiten kann.

Sollte das Gesamtsystem bei steigender Komplexität der Bildverarbeitungsalgorithmen an seine Grenzen stoßen, lassen sich auf Grund der transparenten verteilten Strukturen der ROS Basis ohne zusätzlichen Aufwand einzelne Nodes auf zusätzliche Rechner verteilen, ohne dass der Nutzer unter Einschränkungen zu leiden hat. Bei der Aufteilung der Nodes auf Verschiedene PCs sollte bedacht werden, dass die Datenströme zwischen den Nodes eine zum Teil relative hohe Bandbreite benötigen. Aus diesem Grund sollten Nodes, die unkomprimierte Videodatenströme austauschen, möglichst auf der gleichen Plattform laufen. Sollte dieses nicht (mehr) möglich sein, sollte über die Verwendung von Netzwerkkomponenten nachgedacht werden, die Kanalbündelung („Link Aggregation“) unterstützen.

5.7 Visualisierung und Anwenderinteraktion

Die Qualität komplexer Bildverarbeitungsalgorithmen hängt von einer Vielzahl teils abhängiger Parameter ab. Eine vollständig autonome Ermittlung und Optimierung dieser Parameter ist prinzipiell möglich, führt aber auf Grund eines im Vorfeld nicht exakt definierbaren Gütekriteriums häufig zu einem auf den Anwendungsfall bezogenen suboptimalen Ergebnis. Sind die Abhängigkeiten der Parameter geeignet visualisiert, erkennt der Anwender komplexe Zusammenhänge häufig einfacher als eine Maschine. Das Problem liegt darin, einen Weg zu finden, die wesentlichen Zusammenhänge überschaubar und intuitiv erfassbar darzustellen. Die Tatsache, dass sich eine Visualisierung im klassischen Sinne auf drei Dimensionen beschränken muss, erfordert besondere Ideen und Denkansätze, um dem Anwender den Umgang mit großen Datenmengen zu erleichtern.

Einen interessanten Ansatz verfolgt (Inselberg, 2009): Er ordnet sämtliche Achsen des n -dimensionalen Raumes nicht rechtwinklig, sondern parallel in festem Abstand zu einander an. Auf die Details dieser Herangehensweise wird hier nicht eingegangen; sie sind in Anhang A: Parallele Koordinaten beschrieben.

Im Rahmen einer Diplomarbeit (Heidrich, 2009) entstand ein Visualisierungswerkzeug, das diese Technik verwendet. Daten können entweder aus einer Datei oder über die ODBC Schnittstelle aus beliebigen Datenbanken eingelesen werden. In diesem Fall ist eine zyklische Datenaktualisierung in beliebigen Zeitabständen möglich.

Zur übersichtlichen Aufbereitung der Daten bietet die Software verschiedene Möglichkeiten. Zum einen lässt sich die Anzeige auf beliebige relevante Datenkanäle begrenzen, zum anderen besteht die Möglichkeit, Datensätze von Hand oder nach definierbaren Kriterien einzufärben. Außerdem besteht die Möglichkeit, einfache Statistiken der eingelesenen Datensätze zu erstellen.

Ein Beispiel mit nur sehr wenigen Parametern findet sich in Abbildung 62; hier werden die Polygonzüge, die zu einem Datensatz mit HD-Videomaterial gehören, grün eingefärbt. Als Resultat fällt sofort ins Auge, dass der in Spalte 4-6 festgehaltene Positionsfehler der erkannten Ringmarker nicht von der Auflösung des Videomaterials abhängt. Basierend auf dieser Erkenntnis wurde die Videoaufzeichnung auf eine mittlere Auflösung begrenzt, um bei gleichwertigen Messergebnissen die Rechenlast zu reduzieren.

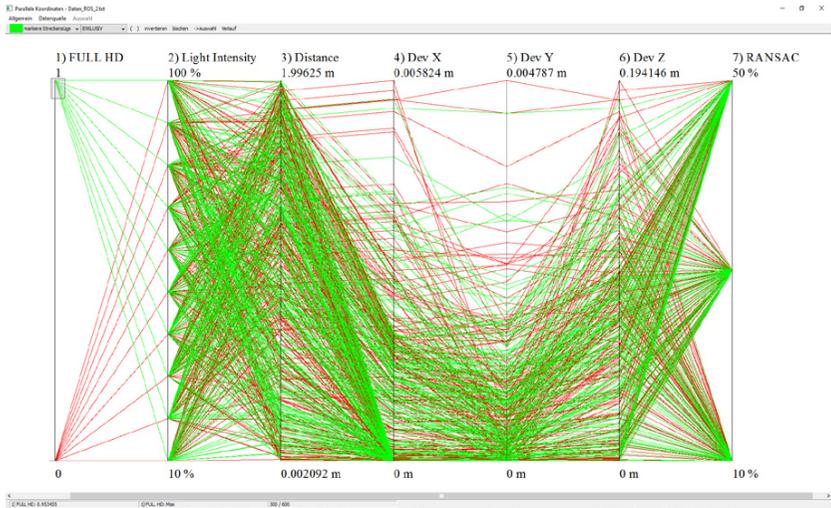


Abbildung 62: Visualisierung ausgewählter Parameter, farbige Markierung nach gewählter Auflösung

Abbildung 63 zeigt ein weiteres Beispiel: Hier wurde die Achse 2 (Lichtintensität) mit einer Regenbogenmarkierung belegt. Hierbei fällt auf, dass ab einer Lichtintensität von etwa 60% überhaupt keine Detektion mehr möglich ist. Der Grund dafür liegt in den dann verstärkt auftretenden, bauartbedingten Glanzlichtern, welche von dem geringen Abstand zwischen Licht- und Kameraoptik hervorgerufen werden.

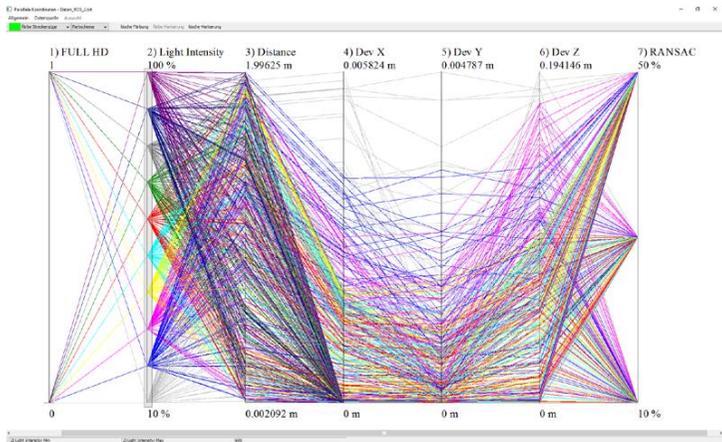


Abbildung 63: Visualisierung ausgewählter Parameter, farbige Markierung nach der Höhe der Lichtintensität der Kaltlichtquelle

Neben bedingten Formatierungen der jeweils einem Datensatz zugeordneten Polygonzüge bietet die Visualisierungssoftware auch die Möglichkeit einfacher statistischer Auswertungen, wie zum Beispiel eines in Abbildung 64 dargestellten Histogramms. Außerdem wurde ein Rückkanal im Konzept berücksichtigt, welcher allerdings auf Grund der vielen nur manuell anzupassenden Parameter nicht implementiert wurde. Der Anwender hat lediglich die Möglichkeit, über die ROS-eigene Dynamic Reconfigure Oberfläche Einfluss auf die Parameterwerte zu nehmen.

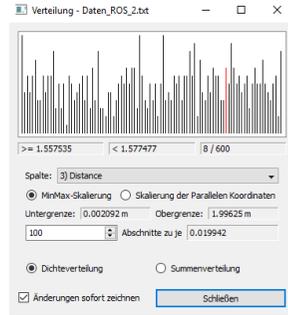


Abbildung 64: Beispiel für eine Histogrammauswertung

6. Überprüfung der optischen Positionsermittlung

Das Gesamtpaket der Algorithmen zur optischen Positionserkennung wurde anhand eines Versuchsaufbaus auf seine Wirksamkeit als Überwachungsinstanz der Regelung des Stiff-Flop Armes getestet.

Abbildung 65 zeigt einen zu diesem Zweck mit Ringmarkern versehenen und auf einen Lokator des optischen Trackingsystems montierten angetriebenen Prototyp des Armes, welcher von der Endoskopkamera aufgenommen wird.

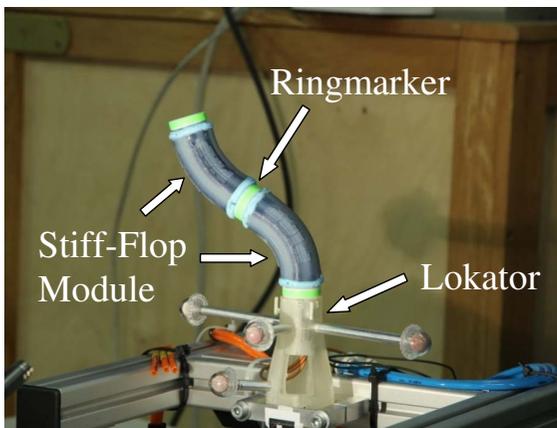


Abbildung 65: Integration der Systemkomponenten

Die Videobilder werden aufbereitet, vorverarbeitet und anschließend den Objekterkennungsalgorithmen zugeführt. Diese berechnen bei erfolgreicher Detektion des Armes und der Ringmarker deren Positionen im Raum und fügen diese in den Transformationsbaum des Gesamtsystems ein.

Eine übergeordnete Überwachungsinstanz berechnet kontinuierlich die Abweichung von Soll- und Istwerten und gibt Alarm, wenn sich diese außerhalb eines definierbaren Toleranzbandes bewegen.

Das Ergebnis wird in Abbildung 66 durch das ROS-Visualisierungstool rviz dargestellt. Zu sehen sind hier die Koordinatensysteme des Transformationsbaumes sowie ihre Abhängigkeiten (gelbe Pfeile). Der besseren Sichtbarkeit wegen wurden sowohl die Geometrie des Armes (grün) als auch die detektierten Ringmarkerposen (weiß) auf eine Ebene vor dem eigentlichen Koordinatenbaum projiziert. Durch diese Abbildung werden außerdem die Einflüsse der mangelhaften Erkennung in z-Richtung der Endoskopkamera eliminiert. Der dargestellte Zustand zeigt eine gute Übereinstimmung der Daten der Objekterkennung und der Steuerung - aus diesem Grund wird der Stiff-Flop Arm grün dargestellt.

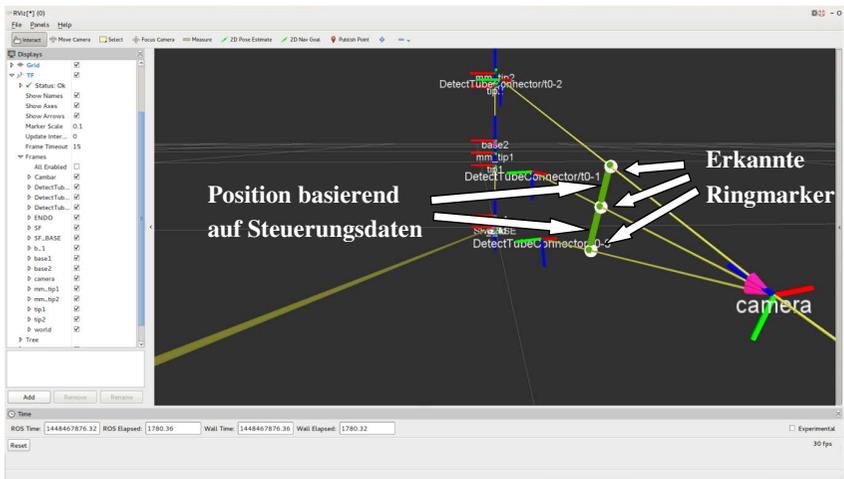


Abbildung 66: Darstellung des erweiterten Transformationsbaumes und des erkannten Stiff-Flop Volumens

Abbildung 67 zeigt noch einmal die komplette Versuchsanordnung. Der Stiff-Flop Arm befindet sich in Ruhelage, die internen Sensoren wurden deaktiviert. Die Positionsdaten der Steuerung stimmen mit den optisch ermittelten Werten überein, der Arm wird im Simulator grün visualisiert.

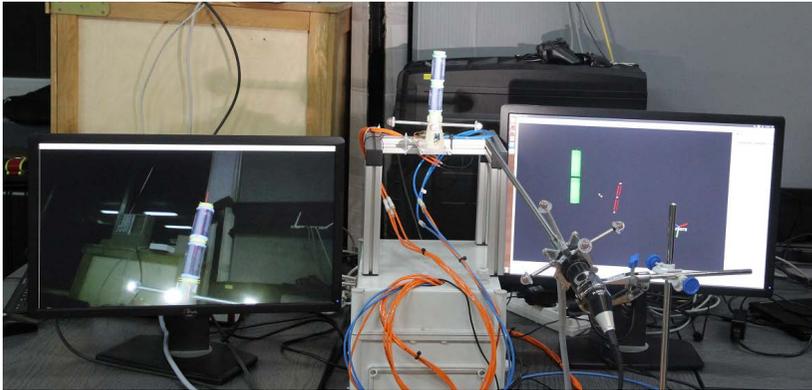


Abbildung 67: Aufbau des kompletten Versuchsystems. Positionsdaten von Bilderkennung und Regelung stimmen überein: Der Arm (linkes Modell im rechten Monitor) wird grün markiert.

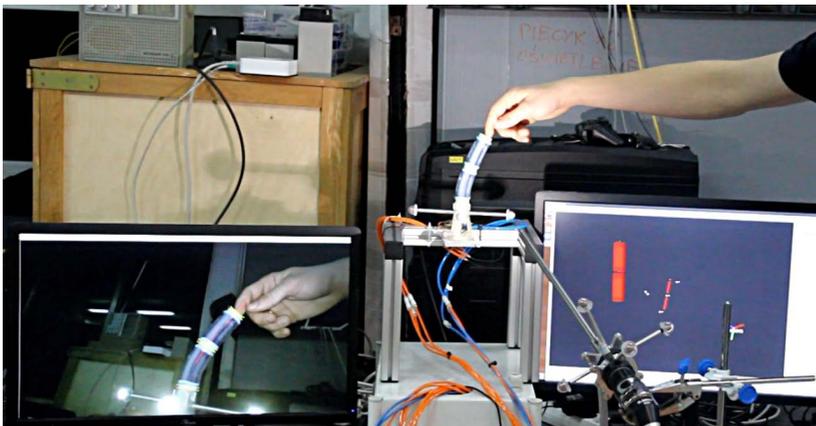


Abbildung 68: Der Arm wird manuell aus seiner Ruhelage bewegt. Die interne Sensorik wurde vorab deaktiviert. Die optische Positionsermittlung stellt eine Abweichung fest und gibt Alarm. Das Modell der Armes (linkes Modell im rechten Monitor) wird rot markiert.

Abbildung 68 zeigt die Auswirkungen einer manuellen Dislokation des Armes. Die Positionsdaten der Steuerung stimmen nicht mit den detektierten Werten überein. Als Resultat aus diesen Differenzen schlägt die übergeordnete Überwachungsinstanz Alarm, was durch eine rote Einfärbung des Simulationsmodells signalisiert wird.

7. Zusammenfassung und Ausblick

Mit der vorliegenden Arbeit wurde ein neues Verfahren zur optischen Lageüberwachung eines innovativen flexiblen Roboters mit steuerbarer Steifigkeit für minimal-invasive chirurgische Eingriffe (MIS) vorgestellt.

Etablierte Verfahren sind im untersuchten Szenario nicht anwendbar, da diese im Allgemeinen entweder auf der Erkennung eines bekannten und angelernten Umrisses, einer eindeutigen, bekannten Textur oder starken Kontrastunterschieden von Vorder- und Hintergrund basieren. Keine dieser Voraussetzungen ist hier gegeben: Der Manipulator ist durchgängig flexibel, kann seine Länge ändern und auch eine Texturerkennung ist schwierig umzusetzen, da gegebenenfalls angebrachte Muster mit zunehmender Krümmung des Armes deutliche Verzerrungen in der zu erkennenden Textur erhalten.

Eine reine Texturerkennung ist für die Lageermittlung außerdem nur dann einsetzbar, wenn sich der komplette Körper im Blickfeld der Kamera befindet. Sind ausschließlich Fragmente sichtbar, lässt sich kein Bezug zum Basiskoordinatensystem herstellen. Auch die in der Mobilien Robotik sehr beliebten Haar-like Features sind hier nur bedingt einsetzbar, da diese zur exakten Positionsbestimmung im Raum verzerrungsfrei abgebildet werden müssen.

Die Detektierung von ringförmigen Umlaufmarkierungen bietet hier verschiedene Vorteile. Zum einen bleibt die Abbildung bei hinreichend schmalen Markierungen annähernd verzerrungsfrei, zum anderen wirkt sich eine eventuell auftretende, systembedingte radiale Streckung des Armes ausschließlich auf die Genauigkeit der Entfernungsmessung zwischen Kameraebene und Mittelachse des Manipulators aus. Das vorgestellte Verfahren ist außerdem robust gegenüber Bildrauschen oder kleineren Glanzlichtern, solange die Kontur des Ringes nicht über größere Abschnitte unterbrochen ist.

Erreicht wird diese Zuverlässigkeit über ein mehrstufiges Verfahren. Erfolgt im ersten Schritt eine klassische Bildsegmentierung auf der Basis von Farbwerten, folgt im zweiten Schritt eine Vorselektion der aussagekräftigsten Kontursegmente.

Die nötige Einpassung einer Ellipse kann auf Grund der weitgehend verdeckten Kanten des Ringmarkers nicht auf klassische Weise über die Methode der kleinsten Fehlerquadrate erfolgen.

Techniken wie die Hough-Transformation bieten sich auf Grund der relativ hohen Parameterzahl einer Ellipse ($P=5$) ebenfalls nicht an, da Rechenlast und Speicherverbrauch exponentiell von der Parameteranzahl abhängen.

Der hier verwendete neue Ansatz benutzt einen modifizierten RANSAC-Algorithmus. Dabei werden zufällig ausgewählte Punkte der einzelnen Kontursegmente zur Findung der Parameter herangezogen. Beinhaltet die gefundene Ellipse einen definierten Teil der restlichen Punkte aller

Kontursegmente, werden die ermittelten Werte übernommen. Ist das Kriterium nicht erfüllt, erfolgt eine weitere Probe mit neuen Zufallspunkten.

Parallel zur Ringmarkerdetektierung erfolgt eine Segmentierung und Objektvermessung auf der Basis einer Support Vector Machine. Diese Redundanz bietet im Wesentlichen zwei Vorteile: Zum einen erlaubt sie eine Plausibilitätsprüfung der erkannten Markerpositionen, zum anderen ist die Berechnung einer stetigen Mittellinie möglich. Diese kann im Gegensatz zu den diskreten Ringmarkerpositionen zusätzlich für eine einfache Kollisionserkennung genutzt werden, falls 3D-Modelle des Arbeitsraumes vorhanden sind. Für die eigentliche Aufgabe, also die Überwachung der Regelung, spielt das aber eine untergeordnete Rolle.

Damit die vorab genannten Algorithmen effiziente und vor Allem zuverlässig arbeiten können, ist eine sauber ausgeleuchtete Szene, die im Idealfall frei von Glanzlichtern und Schlagschatten ist, die wichtigste Voraussetzung. Das resultierende Bildmaterial besitzt eine im relevanten Bereich weitgehend homogene Helligkeitsverteilung und dadurch bedingt ein geringes Rauschen. Da die verwendete Endoskoptechnik systembedingt nur eine punktförmige Ausleuchtung zulässt und diese punktförmige Lichtquelle außerdem durch ihre Positionierung nahe der optischen Achse der Kamera zwangsläufig zu Glanzlichtern und starken Schattenwürfen führt, ist eine nachträgliche Aufbereitung des Bildmaterials prinzipiell anzuraten.

Aus diesem Grund wurden ein vereinfachtes HDR (High Dynamic Range) Verfahren zur Tonwertspreizung entworfen, implementiert und evaluiert. In die Bewertung flossen ebenfalls verschiedene Verfahren zur Glanzlichtreduktion ein. Obwohl diese Verfahren bei einfachen Testszenen zu sichtbaren Verbesserungen führten, brachten sie keine sichtbare Steigerung der Erkennungsrate des Prüflings mit sich und wurden deshalb nicht in den aktuellen Prototypen übernommen.

Insgesamt lässt sich die Erkennungsgenauigkeit der verwendeten Algorithmen in der Kameraebene als durchaus akzeptabel bezeichnen. Die Abweichungen liegen hier bei optimal eingestellter Intensität der Lichtquelle im Submillimeterbereich. Die Verwendung der Entfernungswerte ist auf Grund der starken Schwankungen und unkalkulierbaren Abweichungen nahezu ausgeschlossen. Verfahrensbedingt sind hier jedoch keine besseren Werte erreichbar; für die Zuverlässige Ermittlung der Tiefeninformation ist ein Stereokamerasystem unerlässlich.

Ein Designwechsel des Manipulators kurz vor Projektende brachte zudem eine deutliche Verschlechterung der Erkennungsleistung der Support Vector Machine mit sich. Wurde anfänglich die Form des Armes durch einen außenliegenden Gewebeschlauch aus Nylon sichergestellt, erfolgt die Vermeidung des so genannten Ballooning-Effektes im aktuellen Prototypen durch Textilwicklungen um die innenliegenden Druckröhren. Dieser Wechsel auf eine glatte, texturlose Oberfläche verringert den Abstand der beiden Objektklassen der SVM deutlich. In Folge dessen steigt die Anzahl der falschen Zuordnungen immens an; eine Zuordnung wird insbesondere in den

Übergangsbereichen schwierig. Durch geeignete Filter kann dieser Effekt in der Kameraebene abgeschwächt werden – die Kalkulation der Entfernung hingegen leidet deutlich.

Abschließend lässt sich sagen, dass die verwendeten medizinischen Gerätschaften nur bedingt für die Lageermittlung eines flexiblen Manipulators geeignet sind. Erst im fortgeschrittenen Projektverlauf wurden einige Schwachstellen des zur Verfügung stehenden Materials deutlich. Die daraus resultierenden Probleme werden an dieser Stelle noch einmal kurz zusammengefasst:

- Das optische System der Endoskopkamera (Objektiv und Linsensystem im Schaft) ist nicht starr. Selbst eine leichte Drehung um die z-Achse verschiebt den Bildausschnitt entlang der beiden anderen Koordinatenachsen. Das liegt zum einen an der unpräzisen Schnellspannvorrichtung, mit der der Endoskopschaft auf dem Objektiv platziert wird, zum anderen an der spielbehafteten Montage der Linsen im Schaft selber. Dabei handelt es sich nicht um zu hohe Fertigungstoleranzen, sondern eine systembedingte Notwendigkeit: Wird das Endoskop sterilisiert, dehnen sich Linsen und Außenwand auf Grund unterschiedlicher Ausdehnungskoeffizienten der verwendeten Materialien unterschiedlich stark aus, was im schlimmsten Fall zu einer Beschädigung der feinen Präzisionslinsen führen kann. Abhilfe schafft hier eine mittlerweile erhältliche „Chip-On-Tip“ Lösung. Durch die Platzierung des Kamerasensors auf der Endoskopspitze entfallen das aufwendige Linsensystem sowie das Objektiv komplett; die Bildebene der Kamera ist fix mit dem Schaft verbunden. Laut Aussage des Herstellers bringt diese Anordnung außerdem eine verbesserte Bildschärfe mit sich.
- Die Vereinigung von Lichtleitern und Kameraoptik im selben Schaft führt unvermeidbar zu Schattenbildung und Glanzlichtern, denen nur durch eine deutliche Reduzierung der Lichtleistung entgegengewirkt werden kann. Dieses ist aber nur bis zu einem bestimmten Punkt möglich, da das zurückgeworfene Licht noch ein aufwendiges Linsensystem auf dem Weg zum Kamerasensor passieren muss und dabei wiederum an Intensität verliert. Das resultierende Bildrauschen erschwert zusätzlich eine saubere Detektion der Ringmarker. Eine „Chip-On-Tip“ Variante verhindert Letzteres, kann die Problematik der Glanzlichtbildung an sich aber nicht verhindern.
- Minimale Bewegungen der Kamera im Arbeitsraum erfordern auf Grund des geringen Abstandes zum Umfeld eine umfangreiche Nachführung der verschiedensten Parameter (Schärfe, Öffnungszeiten, Lichtstärke etc.). Die meisten dieser Einstellungen können dabei ausschließlich von Hand und zumeist nur mechanisch vorgenommen werden, was zu einer für den Bilderkennungsalgorithmus meist suboptimalen Arbeitsumgebung führt.

Für eine Optimierung der Vermessungsleistung ist die Zusammenarbeit mit den Herstellern der verwendeten Medizinprodukte unerlässlich. Dieser Schritt ist jedoch nicht so einfach umzusetzen: Erweiterte Funktionalität der Gerätschaften führt unweigerlich zu aufwendigeren Zulassungsverfahren; die entstehenden Mehrkosten sind nicht zu vernachlässigen.

A. Parallele Koordinaten

Eines der größten Probleme bei der Erhebung und Auswertung größerer Datenmengen (Data Mining) oder auch nur der Darstellung der Abhängigkeit vieler Parameter (Multivariate Probleme) liegt darin, einen Weg zu finden, die wesentlichen Zusammenhänge überschaubar und intuitiv erfassbar darzustellen. Etwa die Hälfte der sensorischen Nervenzellen des Menschen werden für die optische Wahrnehmung verwendet - eine grafische Präsentation bietet sich also an. Wie stellt man jedoch eine Gerade oder Fläche im fünf- oder sogar sechsdimensionalen Raum dar? Die Tatsache, dass sich eine Visualisierung im klassischen Sinne auf drei Dimensionen beschränken muss, erfordert neue Ideen und Denkansätze, um dem Anwender den Umgang mit großen Datenmengen zu erleichtern.

Einen alternativen Ansatz verfolgt Prof. Alfred Inselberg von der Tel Aviv University seit den sechziger Jahren. Er entwickelte über einige Jahre das System der *Parallelen Koordinaten* (kurz: \parallel -Koordinaten), welches er 1984 veröffentlichte (Inselberg, 2009). Der Grundgedanke liegt darin, sämtliche Achsen des n -dimensionalen Raumes nicht rechtwinklig, sondern parallel in festem Abstand zu einander anzuordnen. Eine wichtige Eigenschaft dieser Vorgehensweise ist, dass keinerlei Informationen durch die Transformation verloren gehen.

Ein Punkt \mathbf{P} im n -dimensionalen Raum wird in \parallel -Koordinaten zu einem Polygonzug mit n Stützpunkten, hier ein Beispiel für fünf Dimensionen:

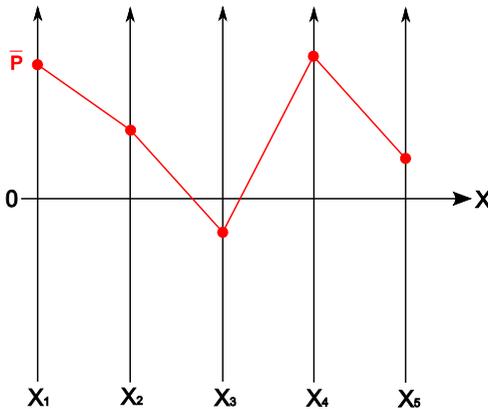


Abbildung 69: 5-dimensionaler Punkt P in \parallel -Koordinaten

Im weiteren Verlauf werden auf Grund der höheren Anschaulichkeit wo möglich zweidimensionale Beispiele verwendet.

Abbildung 70 zeigt die Darstellung des zweidimensionalen Punktes \mathbf{P} als Polygon mit zwei Stützpunkten in $\|\cdot\|$ -Koordinaten. Als Folgerung daraus wird eine Gerade im Raum durch eine Polygonschar in $\|\cdot\|$ -Koordinaten dargestellt, wie zu sehen in Abbildung 71:

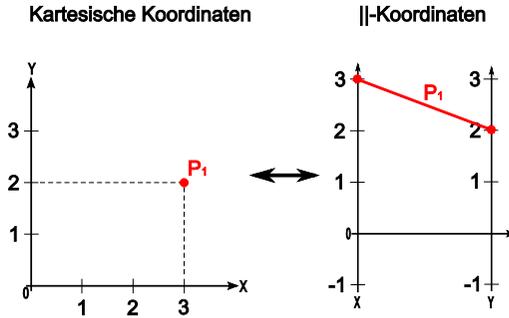


Abbildung 70: Transformation eines Punktes in $\|\cdot\|$ -Koordinaten

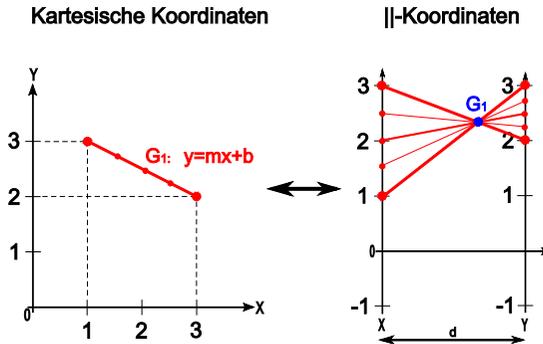


Abbildung 71: Transformation einer Geraden in $\|\cdot\|$ -Koordinaten

Auffällig ist in diesem Zusammenhang, dass sich sämtliche Geraden-Scharen in einem Punkt \mathbf{S} zwischen den eigentlichen Koordinatenachsen der $\|\cdot\|$ -Darstellung schneiden. Die Position dieses Punktes wird sowohl durch Steigung der Geraden im 2D-Raum als auch durch ihren y -Achsenabschnitt bestimmt.

Eine Gerade, die im kartesischen Raum durch $y(x) = mx + b$ definiert ist, wird in \parallel -Koordinaten durch einen Punkt

$$\mathbf{P} = \left(\frac{d}{1-m}, \frac{b}{1-m} \right), m \neq 1 \tag{7.77}$$

vollständig beschrieben. Dabei steht der Parameter d für den gewählten Abstand der Achsen im \parallel -System. \mathbf{P} liegt jedoch nicht zwingender Weise zwischen den jeweils auf einander folgenden \parallel -Achsenpaaren, sondern kann im Extremfall ($m=1$) sogar im Unendlichen liegen.

Geraden in Räumen mit $n > 2$ Dimensionen lassen sich \parallel auf zwei verschiedene Arten beschreiben. Die erste Möglichkeit ist die Aufstellung von Gleichungen mit benachbarten Variablen. Bei dieser Darstellungsform leitet sich der Wert jeder einzelnen Komponente von der vorangegangenen ab. Sie ist anschaulicher als die folgenden Darstellungsmöglichkeiten, aber nicht so einfach transformierbar.

$$\begin{aligned} l_{1,2} : x_2 &= m_2 x_1 + b_2 \\ l_{2,3} : x_3 &= m_3 x_2 + b_3 \\ &\vdots \\ l_{n-1,n} : x_n &= m_n x_{n-1} + b_n \end{aligned} \tag{7.78}$$

Es ergeben sich $n-1$ indizierte Punkte, die in ihrer Gesamtheit den Linienzug definieren. In geschlossener Schreibweise ergeben sich die Positionen durch

$$\mathbf{l}_{i-1,i} = ((i-2)(1-m_i) + 1, b_i, 1-m_i) \tag{7.79}$$

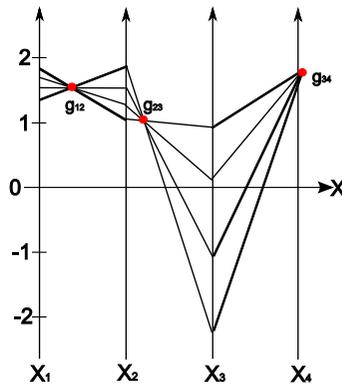


Abbildung 72: Darstellung einer Geraden, Variante 1

Alternativ bietet sich die Darstellung bezogen auf eine Basisvariable an. Da diese Form direkt aus der Parameterform abgeleitet werden kann, kommt sie häufiger zum Einsatz.

$$l_{1,2} : x_2 = m_2^1 x_1 + b_2^1 \quad (7.80)$$

$$l_{1,3} : x_3 = m_3^1 x_1 + b_3^1$$

⋮

$$l_{1,n} : x_n = m_n^1 x_1 + b_n^1$$

Es ergeben sich wiederum $n-1$ indizierte Punkte, die in ihrer Gesamtheit den Linienzug definieren. Die geschlossene Schreibweise lautet bei dieser Variante

$$l_{1,i} = (i-1, b_i^1, 1-m_i^1) \quad (7.81)$$

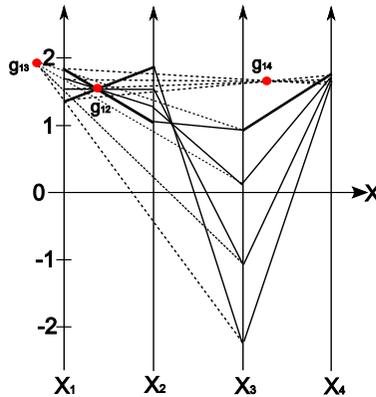


Abbildung 73: Darstellung einer Geraden, Variante 2

Der oben gezeigte Punkt-Linien-Zusammenhang wird von Inselberg als "Fundamentale Dualität" bezeichnet und ist die Grundlage für weitere Betrachtungen.

Ein ebenfalls unschätzbare Vorteil der \parallel -Koordinaten besteht darin, dass Rechenregeln und Betrachtungsweisen des 2D- oder 3D-Raumes durch einfache Kopier- und Verschiebeoperationen auf n -dimensionale Räume erweitert werden können. Die Relationen zwischen den einzelnen Dimensionen bleiben in jeder Situation und auch bei losgelöster Betrachtung erhalten. Das gilt natürlich auch im kartesischen Koordinatensystem, kann hier aber bei weitem nicht so intuitiv erfasst und Ergebnisse nicht so anschaulich interpretiert werden.

Abbildung 74 zeigt beispielhaft die Vorgehensweise bei der Konstruktion eines fünfdimensionalen Hyperwürfels.

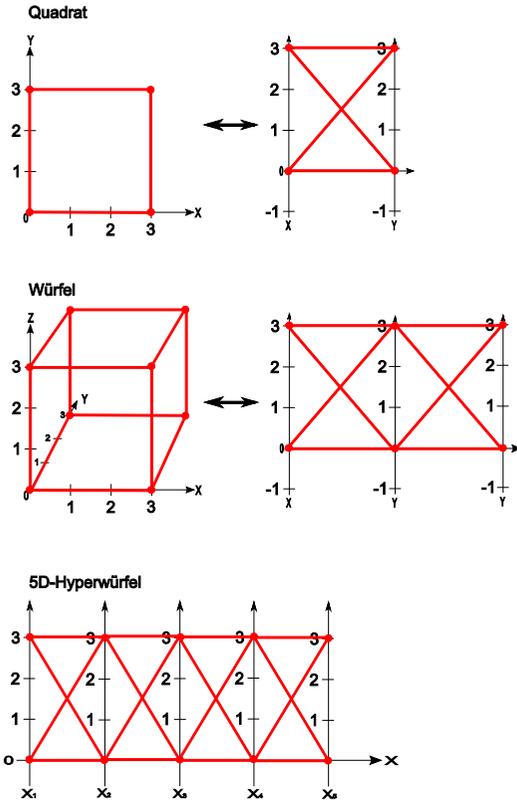


Abbildung 74: Konstruktion eines fünfdimensionalen Hyperwürfels

Im Folgenden werden kurz Abstandsbestimmungen und Schnittpunktberechnungen über einfache geometrische Zusammenhänge dargestellt.

Der einfachste Fall ist sicherlich die Abstandsberechnung zweier Punkte im n -dimensionalen Raum.

Abbildung 75 verdeutlicht, dass sich der Euklidische Abstand natürlich auch in $\|\cdot\|$ -Koordinaten direkt ermitteln lässt. Diese Erkenntnis ist trivial und soll hier nur dazu verwendet werden, einen

geeigneten Einstieg in die vereinfachte Abstandsberechnung zwischen Elementen im Raum zu schaffen.

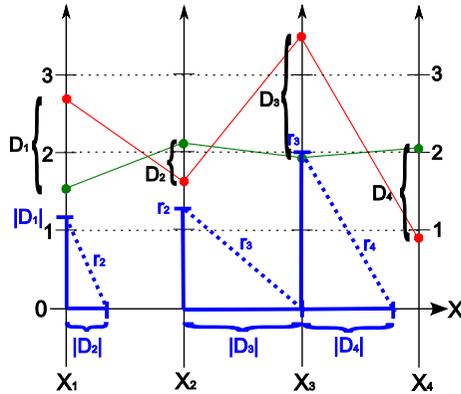


Abbildung 75: Euklidischer Abstand in $\|\cdot\|$ -Koordinaten

Der Abstand der beiden Punkte entlang der ersten Raumachse D_1 wird rechtwinklig zum Abstand der beiden Punkte entlang der zweiten Raumachse D_2 aufgetragen. Die Vorzeichen können dabei wie gewohnt vernachlässigt werden. Der nun ablesbare Abstand r_2 der Punkte in der Ebene, die von X_1 und X_2 aufgespannt wird, dient jetzt als Basis der Abstandsberechnung für die weitere Berechnung unter Einbeziehung der Achse X_3 . Der gesamte Vorgang lässt sich wie zu erwarten iterativ für beliebig viele Dimensionen wiederholen.

B. O-Notation

Die O-Notation (sprich "Groß-O-Notation") wurde 1894 vom Mathematiker Paul Bachmann zur allgemeinen Bewertung der Laufzeit eines mathematischen Algorithmus entwickelt - lange vor der Entwicklung der ersten elektronischen Rechenmaschinen.

Bekannt wurde die O-Notation durch den deutschen Zahlentheoretiker Edmund Landau. Gerade im deutschsprachigen Raum ist sie deshalb auch als Landau-Notation bekannt.

Die Landau-Notation findet Verwendung in der Komplexitätstheorie der Theoretischen Informatik. Die Komplexitätstheorie klassifiziert Algorithmen auf Grund der zur ihrer Lösung benötigten

Ressourcen - meist der nötigen Rechenzeit oder des benötigten Speicherplatzes. Sie stellt neben der Berechenbarkeitstheorie und der Theorie formaler Sprachen ein Kerngebiet der Theoretischen Informatik dar. Mit Hilfe der Komplexitätstheorie lässt sich abschätzen, ob ein theoretischer Algorithmus auch praktisch umsetzbar ist.

Die Landau-Notation beschreibt das asymptotische Verhalten von Funktionen und Folgen. Sie hilft insbesondere bei der Beurteilung, ob ein Algorithmus für eine große Menge n an Eingabeparametern noch geeignet ist bzw. erlaubt einen Effizienz-Vergleich zwischen verschiedenen Algorithmen für große n . In der modernen Informatik findet sie Verwendung bei der plattformunabhängigen Abschätzung der Laufzeit eines Algorithmus oder des dabei auftretenden Speicherverbrauchs.

Formal lassen sich die Landau-Symbole mittels Limes Superior und Limes Inferior definieren. Werden die Landau-Symbole zur Laufzeitabschätzung eines Algorithmus verwendet, ist vor allem die asymptotische obere Schranke interessant:

$$f \in O(g) \quad \text{wird definiert durch} \quad 0 \leq \limsup_{x \rightarrow a} \left| \frac{f(x)}{g(x)} \right| < \infty$$

Dabei bezeichnen f und g entweder

- Folgen reeller Zahlen, dann ist $x \in \mathbb{N}$ und der Grenzwert , oder
- reell wertige Funktionen der reellen Zahlen, dann ist $x \in \mathbb{R}$ und der Grenzwert aus den erweiterten reellen Zahlen $a \in \mathbb{R} \cup \{-\infty, +\infty\}$

Eine weitere Möglichkeit besteht darin, dass f und g reell wertige Funktionen beliebiger topologischer Räume sein können - allerdings wird diese Möglichkeit bei der Laufzeitanalyse eines Algorithmus nicht benötigt.

Eine besondere Betrachtung verdient das häufig im Zusammenhang mit der Landau-Notation verwendete Gleichheitszeichen. Hierbei handelt es sich nicht um eine Gleichheitsaussage, sondern lediglich um eine rein symbolische Schreibweise. Die Aussagen $f_1 = O(g(x))$ und $f_2 = O(g(x))$ bedeuten zum Beispiel nicht, dass f_1 und f_2 gleich sind - bei $O(g(x))$ handelt es sich eigentlich um eine Menge, die alle Funktionen enthält, die höchstens so schnell wachsen können wie $g(x)$. Formal korrekt müsste es heißen $f(x) \in O(g(x))$

In der Praxis geht man bei der Laufzeitabschätzung eines Algorithmus von drei Voraussetzungen aus:

- Ein Algorithmus benötigt Eingabedaten. Speziell bei der O-Notation wird jetzt das Verhalten dieses Algorithmus für eine große Menge n an Eingabewerten ($n \rightarrow \infty$) untersucht.
- Konstanten gelten als unwesentlich und können außer Acht gelassen werden. Da die Beurteilung unabhängig vom verwendeten Rechnermodell oder der eingesetzten Programmiersprache erfolgen soll, wird hier nur die Anzahl der nötigen Basisoperationen ausgewertet.
- Untersucht wird immer nur die obere Schranke an nötigen Basisoperationen, also der ungünstigste Fall. Die untere Schranke wird durch die sog. Ω -Notation definiert.

Damit man die asymptotische Laufzeitkomplexität eines Algorithmus auf einfache Weise ermitteln kann, teilt man die einzelnen Anweisungen in verschiedene Gruppen ein:

- Einfache Anweisungen
 Zu den einfachen Anweisungen gehören Deklarationen, Definitionen, Wertzuweisungen, Ein- und Ausgabeanweisungen, einfache Rechenoperationen und Sprunganweisungen. Die Laufzeit dieser Anweisungen ist eine hardwareabhängige Konstante und kann somit vernachlässigt werden. Die asymptotische Laufzeitkomplexität beträgt deshalb $O(1)$.
- Schleifen
 Eine Schleife führt meist die bereits oben aufgeführten einfachen Anweisungen in konstanter Zeit aus. Allerdings kann die Schleife selbst von der Anzahl der Eingabewerte n abhängig sein. In diesem Fall wird der Schleifenkörper mit der konstanten Laufzeit c insgesamt n mal durchlaufen, es gilt somit $c * n = O(n)$
 Da auch der Schleifenkörper von n abhängig sein kann, ist ggf. eine Verallgemeinerung sinnvoll. Besitzt der Schleifenkörper eine Laufzeitkomplexität von $O(f(n))$ und wird n mal durchlaufen, gilt für die komplette Schleife $O(n * f(n))$.
- Verzweigungen
 Im Falle von Verzweigungen wird das Laufzeitverhalten der einzelnen Anweisungsblöcke separat betrachtet. Da bei der O-Notation lediglich die größtmögliche Laufzeit betrachtet werden soll, kann einfach mit dem ungünstigsten Fall weitergerechnet werden. Besitzt zum Beispiel Anweisungsblock eins die Charakteristik $O(f(n))$ und Anweisungsblock zwei die Charakteristik $O(g(n))$, ergibt sich als Gesamtcharakteristik $O(\max\{f(n), g(n)\})$.

Die Theorie hinter der O-Notation ist relativ aufwendig. In der Praxis lassen sich jedoch einfache Regeln für die Anwendung aufstellen:

- $c = O(1)$

Eine konstante Anzahl an einfachen Anweisungen hat eine konstante, aber plattformabhängige Laufzeit und kann deshalb vernachlässigt werden.

- $c * O(f(n)) = O(f(n))$

Auch hier werden konstante Faktoren vernachlässigt. Diese Regel folgt direkt aus der Multiplikationsregel.

- $O(f(n)) + O(g(n)) = O(f(n))$

Das sequentielle oder parallele Ausführen zweier Code-Abschnitte mit derselben Komplexität hat besagte Komplexität.

- $O(f(n) + g(n)) = \max\{O(f(n)), O(g(n))\}$

Das sequentielle oder parallele Ausführen zweier Code-Abschnitte mit unterschiedlicher Komplexität hat die Komplexität des Aufwändigeren.

- $O(f(n)) * O(g(n)) = O(f(n) * g(n))$

Die Gesamt-Komplexität eines Code-Abschnitts in einem anderen ist das Produkt der beiden.

- $O(\log_b(n)) = O(\log(n))$

Besitzt ein Algorithmus logarithmische Komplexität, ist die Basis des Logarithmus irrelevant.

- $O(a_0 + a_1n + \dots + a_m n^m) = O(n^m)$

Die Komplexität eines Polynoms ist die der höchsten vorkommenden Potenz.

- $O(O(f(n))) = O(f(n))$

Die O-Notation betrachtet den Fall einer maximalen Laufzeit – diese wird durch doppelte Untersuchung nicht größer oder kleiner.

Im Folgenden wird die Laufzeitkomplexität an Hand eines einfachen Bubblesort-Algorithmus ermittelt.

```
prozedur bubbleSort( A : Liste zu sortierender Elemente )
  n := Länge( A )
  wiederhole
    vertauscht := falsch
    für jedes i von 1 bis n - 1 wiederhole:
      falls A[ i ] > A[ i + 1 ] dann
        vertausche( A[ i ], A[ i + 1 ] )
        vertauscht := wahr
      falls_ende
    wiederhole_ende
  n := n - 1
  solange vertauscht und n >= 1
prozedur_ende
```

Dieser Algorithmus im Pseudo-Code besteht neben einfachen Anweisungen der Komplexität $O(1)$ nur noch aus zwei Schleifen und einer Abfrage. Der Inhalt der Abfrage hat ebenfalls die Komplexität $O(1)$ und kann deshalb vernachlässigt werden. Die innere Schleife wird bei jedem Aufruf $(n-1)$ mal durchlaufen, die äußere im ungünstigsten Fall (umgekehrt sortierte Liste) n mal. Die Anzahl der einfachen Anweisungen beträgt somit maximal $\frac{n(n-1)}{2}$, was bei Vernachlässigung der Konstanten einer Laufzeitkomplexität von $O(n^2)$ entspricht.

Auf die weitere Mathematik hinter der Landau-Notation und besonders der Komplexitätstheorie soll in dieser Arbeit nicht weiter eingegangen werden. Der interessierte Leser findet eine ausführliche Betrachtung in (I.Wegener, 2003) oder einen kompakteren Artikel unter <http://www.wikipedia.de>.

C. Literaturverzeichnis

- Abd Elrazeq, O. G. (2014). *Development and Implementation of Transformation and Point Cloud Matching Algorithms for a Vision Based Tracking Algorithm*. Siegen.
- Adam, A. (2009). Einfluss der Referenzgebergeometrie und der Markereigenschaften auf die statische Genauigkeit optischer Navigationssysteme (Studienarbeit). Universität Siegen.
- Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS spring joint computer conference*. IBM Sunnyvale, California.
- AXIOS 3D® Services GmbH. (2012). *CamBar B2 Typ 1.2 - Benutzerhandbuch für medizinische Anwendung*. AXIOS 3D® Services GmbH, Marienstraße 16, D-26121 Oldenburg, Deutschland.
- AXIOS 3D® Services GmbH. (2012). *Einlernen von Lokatoren in Demo CamBar*.
- Bader, M. (12.. 03. 2014). *ROS Wiki: v4r_ellipses*. Abgerufen am 16.. Januar 2015 von http://wiki.ros.org/v4r_ellipses
- Bellekens, B., Spruyt, V., Berkvens, R., & Weyn, M. (2014). A Survey of Rigid 3D Pointcloud Registration Algorithms. *The Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, (S. 8-13).
- Blackmagic Design. (June 2011). SDK Decklink - Software Developers Kit.
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. (O. & Associates, Hrsg.)
- Brown, D. C. (1971). Close-range camera calibration. In *Photogrammetric Engineering* 37 (S. 855-866).
- Bundesministerium für Gesundheit. (07. 08 2015). *Bedeutung der Gesundheitswirtschaft*. Abgerufen am 27. 12 2015 von <http://www.bmg.bund.de/themen/gesundheitsystem/gesundheitswirtschaft/bedeutung-der-gesundheitswirtschaft.html>
- Canny, J. (Nov. 1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*.
- CCIR Comité Consultatif International des Radiocom. (1994). ITU-R BT 601.
- Chen, Q., Wu, H., & Wada, T. (2004). Camera Calibration with Two Arbitrary Coplanar Circles. *Computer Vision - ECCV 2004. Lecture Notes in Computer Science Volume 3023*, S. 521-532. Berlin Heidelberg: Springer.

- Christoph Stiller, A. B. (2009). *Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. (S. H. Hermann Winner, Hrsg.) Vieweg+Teubner.
- Debevec, P., & Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. *Proceedings of the 24th annual conference on Computer graphics and (S. 369-378)*. ACM Press/Addison-Wesley Publishing Co.
- Devlin, K., Chalmers, A., Wilkie, A., & Purgathofer, W. (9 2002). Tone reproduction and physically based spectral rendering. (T. E. Association, Hrsg.) *State of the Art Reports, Eurographics 2002*, S. 101-123.
- Dubrofsky, E. (March 2009). Homography Estimation. University of British Columbia.
- Elrazeq, O. G. (2015). *Calibration of a Vision Sensor Using an Industrial Manipulator*. Siegen: Universität Siegen, Master Thesis.
- Faugeras, O. (1995). *Three Dimensional Computer Vision: A Geometric Viewpoint*. Massachusetts Institute of Technology, Boston, Massachusetts: MIT Press.
- Faugeras, O. D., Luong, Q., & Maybank, S. (1992). Camera self-calibration: Theory and experiments. *Proceedings of European Conference on Computer Vision*.
- Fischler, M. A., & Bolles, R. C. (June 1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. (J.D.Foley, Hrsg.) *Graphics and Image Processing*.
- Fitzpatrick, J. M., West, J. B., & Maurer, C. R. (1998). Predicting Error in Rigid-Body Point-Based Registration. *IEEE Transactions on Medical Imaging*, 17, S. 164-702.
- Foote, T. (12. 06 2013). *ROS Übersicht: Einführung*. Abgerufen am 11. 08 2014 von <http://wiki.ros.org/de/ROS/Introduction>
- Foote, T. (23. 12 2013). *Ros Wiki - Concepts*. Abgerufen am 10. 9 2014 von <http://wiki.ros.org/de/ROS/Concepts>
- Foote, T. (2013). tf:The Transform Library. *2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA). Open Source Software Workshop*, S. 1-6. IEEE.
- Gauss, C. F. (1821). *Theoria combinationis observationum erroribus minimis obnoxiae*. Göttinger Digitalisierungszentrum.
- Gegler, A. (2005). *Bildsegmentierung mit Snakes und aktiven Konturen*.
- GvdHoorn. (11. März 2015). *ROS Industrial*. Abgerufen am 2015. März 24 von <http://wiki.ros.org/Industrial>
- Hartley, R. u. (2008). *Multiple View*. Cambridge University Press.

- Hauck, G. (1883). Neue Constructionen der Perspective und Photogrammetrie. *Journal für reine und angewandte Mathematik*, 95, S. 1-35.
- Heidrich, M. (06.. 02. 2009). Erstellen eines Programms zur Visualisierung von Zahlentabellen in Parallelen Koordinaten. Siegen: Universität Siegen.
- I.Wegener. (2003). *Komplexitätstheorie: Grenzen der Effizienz von Algorithmen*. Springer Verlag.
- Inselberg, A. (2009). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer Verlag.
- KUKA Robot Group. (2007). *CREAD/CWRITE: Programming CREAD/CWRITE and related statements* (Version 1.3 Ausg.).
- KUKA Roboter GmbH. (2007). *Communication*. Germany: KUKA Roboter GmbH.
- KUKA Roboter GmbH. (2010). *KR C2 edition2005*. Augsburg.
- KUKA Roboter GmbH. (kein Datum). *KUKA.LOAD*. Abgerufen am 19. Dezember 2014 von <http://www.kuka-robotics.com/germany/de/products/software/planning/>
- Lechky, O. (1985). World's first Surgical Robot in B.C. *The Medical Post*, 93.
- Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature* 293, S. 133-135.
- Mannschreck, M. (14.. April 2015). Gesundheitsausgaben im Jahr 2013 bei 314,9 Milliarden Euro. (S. Bundesamt, Hrsg.) 132/15.
- Miyazaki, D., Tan, R., Hara, K., & Ikeuchi, K. (2003). Polarization-based Inverse Rendering from a Single View. *Proceeding of International Conference on Computer Vision*, (S. 982-987).
- Pieck, S. (2007). *Leistungsanalyse eines optischen 3D/6D Lokalisiersystems und dessen Integration in ein chirurgisches Assistenzsystem*. ZESS Zentrum für Sensorsysteme. Shaker Verlag.
- Point Cloud Library*. (kein Datum). Abgerufen am 24. 4 2015 von http://docs.pointclouds.org/trunk/classpcl_1_1_iterative_closest_point.html
- point cloud library documentation*. (kein Datum). Abgerufen am 2. 5 2015 von http://pointclouds.org/documentation/tutorials/fpfh_estimation.php
- pointclouds*. (kein Datum). Abgerufen am 2. 3 2015 von http://pointclouds.org/documentation/tutorials/template_alignment.php
- Poynton, C. (2003). *Digital Video and HDTV Algorithms and Interfaces*. San Francisco: Morgan Kaufmann.
- Radu Bogdan Rusu, C. (9-13 May 2011). Robotics and Automation (ICRA), 2011 IEEE International Conference on. *3D is here: Point Cloud Library (PCL)* (S. 1-4). Shanghai: IEEE.

- Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. München.
- Shafer, S. (1984). Using Color to Separate Reflection Components. Rochester, NY: Computer Science Department, University of Rochester.
- Shen, H., & Cai, Q. (2009). Simple and efficient method for specular removal in an image. *Applied Optics*, Vol. 48(Issue 14), S. 2711-2719.
- Shuqair, M. (2014). *Specularity Removal for Detecting a Flexible Manipulator for Surgical Operations with an Endoscopic Camera*. Siegen, Germany: Master Thesis, University of Siegen.
- Suzuki, S., & Abe, K. (1985). Topological Structural Analysis of Digitized Binary. (A. P. Inc, Hrsg.) *Computer Vision, Graphics and Image Processing* 30, S. 32-46.
- Tönnies, K. (März 2009). *Vorlesungsskripte "Grundlagen der Computer Vision"*. Abgerufen am 24. November 2014 von AG Bildverarbeitung und Bildverstehen: http://www.isg.cs.uni-magdeburg.de/bv/index.php?article_id=23
- Tsai, R. (1986). An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (S. 364-374). Miami Beach, Florida: IEEE.
- Tsai, R., & Huang, T. (1984). The perspective view of three points. *IEEE T-PAMI*, 6, S. 13-27.
- Vapnik, W., & Chervonenkis, A. (1974). *Theory of Pattern Recognition*.
- West, J. B., & Maurer, C. R. (kein Datum). Designing Optically Tracked Instruments for Image-Guided Surgery. *IEEE Transactions on Medical Imaging*, 23, S. 533-545.
- Wiles, A. D., Thompson, D. G., & Frantz, D. D. (2004). Accuracy assessment and interpretation for optical tracking Systems. *Medical Imaging*.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (S. 1330-1334). IEEE Transactions.

D. Abbildungsverzeichnis

Abbildung 1: Da Vinci® System der Firma Intuitive Surgical	11
Abbildung 2: Active Cannula der Vanderbilt University	12
Abbildung 3: CardioARM der Carnegie Mellon University	12
Abbildung 4: Früher Prototyp des Stiff-Flop Armes	13
Abbildung 5: Beispiel für eine Segmentierung auf der Basis des GrabCut-Algorithmus	15
Abbildung 6: Stiff-Flop Arm mit 3 Segmenten	16
Abbildung 7: Vorgehensweise nach dem V-Modell	19
Abbildung 8: KUKA KR16 Industrieroboter	20
Abbildung 9: Richard Wolf EndoCam mit Kaltlichtquelle	21
Abbildung 10: CamBar B2 (M) mit Abmessungen des erreichbaren Messvolumens	23
Abbildung 11: Koordinatensysteme eines mobilen Roboters	26
Abbildung 12: Das Vision-System mit Armerkennung und Vermessung	29
Abbildung 13: Programmablaufplan der Intensity-Node	32
Abbildung 14: CbCr-Fläche des YCbCr-Farbmodells im RGB-Farbraum	34
Abbildung 15: Chroma Subsampling	35
Abbildung 16: Organisation der Bilddaten im Speicher	35
Abbildung 17: Funktionsweise einer Lochkamera	37
Abbildung 18: Aufnahme der Pixelmatrix eines CCD-Chips unter dem Mikroskop	39
Abbildung 19: Geometrische Verzeichnung	41
Abbildung 20: Tangentiale Linsenverzeichnung	42
Abbildung 21: Tangentiale Verzerrung	42
Abbildung 22: Kamerakalibrierung mit OpenCV	44
Abbildung 23: Abfolge der einzelnen Bildverarbeitungsschritte	45
Abbildung 24: SORT-P Targets der Firma Atesos medical AG	46
Abbildung 25: Begriffe der Epipolarometrie	48
Abbildung 26: Ausgangsbilder mit Epipolarlinien und Bildpaar nach der Rektifizierung	53
Abbildung 27: FLE (Fiducial Localization Error)	55
Abbildung 28: FRE (Fiducial Registration Error)	55
Abbildung 29: TRE (Target Registration Error)	56
Abbildung 30: Anisotroper Verlauf des dynamischen Fehlers	57

Abbildung 31: Lokator zur Feststellung der Kamera-Pose	58
Abbildung 32: Lokator zur Feststellung der Manipulatorpose	59
Abbildung 33: Lastanalyse des Roboterwerkzeugs mit Kuka Load 5.0	60
Abbildung 34: Lokator zur Definition des Basiskoordinatensystems des Industrieroboters	61
Abbildung 35: Programmablaufplan der CamBar-Node	62
Abbildung 36: Aufbau der ROS Datenbank-Node	63
Abbildung 37: Darstellung des ROS-Industrial Schichtenmodells	65
Abbildung 38: Aufbau der Robotersteuerung KR C2	67
Abbildung 39: Programmablaufplan der KUKA-node	69
Abbildung 40: Ergebnis des HDR Algorithmus	77
Abbildung 41: Definition Photometrischer Winkel	79
Abbildung 42: Ergebnisse der Glanzlichtentfernung	84
Abbildung 43: Teilschritte der Markererkennung	86
Abbildung 44: Konfigurationsoberfläche DetectTubeConnector	87
Abbildung 45: Programmablaufplan der DetectTubeConnector-Node	92
Abbildung 46: Beispiel für die Anwendung von Snakes auf Ultraschallbilder	93
Abbildung 47: Hyperebene zur Separierung der Objektklassen einer SVM	94
Abbildung 48: Objektvermessung	96
Abbildung 49: Geometrischer Aufbau der Endoskopkamera	97
Abbildung 50: Geometrische Entfernungsberechnung eines Zylinders zur Bildebene	98
Abbildung 51: Darstellung der Parameter zur vereinfachten Abstandsberechnung	99
Abbildung 52: Ergebnisse des Vermessungsalgorithmus	100
Abbildung 53: Ergebnisse des Vermessungsalgorithmus in der X-Z-Ebene	101
Abbildung 54: Programmablaufplan der Objektvermessungnode TubeDetect	102
Abbildung 55: Aufbau des Kalibrierungssystems mit verwendeten Koordinatensystemen	103
Abbildung 56: ICP Ausrichtung basierend auf einem Punkt-zu-Punkt-Ansatz	105
Abbildung 57: Radiussuche nach Nachbarpunkten zu P_q in P^k innerhalb des Radius r	106
Abbildung 58: Vollständiges Netz der im Radius r gefundenen Punkte	107
Abbildung 59: Transformationsvorschrift der Normalenvektoren	107
Abbildung 60: Beispiele für FPH zweier Bildpunkte einer Szene	108
Abbildung 61: Amdahls Gesetz	111
Abbildung 62: Visualisierung ausgewählter Parameter	113

Abbildung 63: Visualisierung ausgewählter Parameter	113
Abbildung 64: Beispiel für eine Histogrammauswertung	114
Abbildung 65: Integration der Systemkomponenten	114
Abbildung 66: Darstellung des Transformationsbaumes und des erkannten Stiff-Flop Volumens	115
Abbildung 67: Aufbau des kompletten Versuchssystems.	116
Abbildung 68: Der Arm wird manuell aus seiner Ruhelage bewegt.	116
Abbildung 69: 5-dimensionaler Punkt P in $\ \ $ -Koordinaten	A
Abbildung 70: Transformation eines Punktes in $\ \ $ -Koordinaten	B
Abbildung 71: Transformation einer Geraden in $\ \ $ -Koordinaten	B
Abbildung 72: Darstellung einer Geraden, Variante 1	C
Abbildung 73: Darstellung einer Geraden, Variante 2	D
Abbildung 74: Konstruktion eines fünfdimensionalen Hyperwürfels	E
Abbildung 75: Euklidischer Abstand in $\ \ $ -Koordinaten	F

E. Verwendete Abkürzungen

FLE	Fiducial Localization Error (Marker – Lokalisierungsfehler)
FRE	Fiducial Registration Error (Marker – Registrierungsfehler)
HD	High Definition Television (Videoauflösung 1920 x 1080 Pixel, 16:9)
HDMI	High Definition Multimedia Interface, eine Schnittstelle für die digitale Bild- und Ton-Übertragung
HDR	High Dynamic Range, Bildinformationen mit hohem Dynamikumfang
ICP	Iterative Closest Point, Algorithmus zum Matching von Punktwolken
KCP	KUKA Control Panel
Medical HD	Medical High Definition (Videoauflösung 1280 x 1024 Pixel, 5:4)
OpenCV	Open Computer Vision, freie Bildverarbeitungsbibliothek (http://www.opencv.org)
PCL	Point Cloud Library, eine frei verfügbare Bibliothek zur Verarbeitung von Punktwolken
PFH	Point Feature Histogram (PFH)
ROS-I	ROS Industrial
SD	Standard Definition (Videoauflösung, PAL: 720 x 576 Pixel)
SDK	Software Development Kit (Entwicklungsumgebung für eigene Programme)
SVM	Support Vector Machine, Klassifikator für beliebige Arten von Objekten
TRE	Target Registration Error (Ziel – Registrierungsfehler)
Ultra HD	Ultra High Definition (Videoauflösung 2160 x 3840 Pixel, 16:9)

F. Schlüsselwörter

8-Punkt-Algorithmus	44	RAVEN	8
Aktive Konturen	83	ReMeDi	7
Aligning object templates to a point cloud	96	ROBODOC	3
CamBar B2	17	Sensei	7
Chroma Subsampling	29	Sicherheit	5
Dynamikbereich	66	SPORT	8
EndoCam® HD 5509R2	16	Stiff-Flop	1, 10, 13
Entzerrung	39	Telediagnostik	7
Epipolargeometrie	40	Miyazaki	74
Farbunterabtastung	29	Neighborhood Concept	97
Fiducial Localization Error	48	OPC	58
Fiducial Registration Error	48	OpenCV	20
Fundamentalmatrix	43	Parallelen Koordinaten	1
Gesundheitswirtschaft	1	Passive Marker	39
High Dynamic Range	67	PCM	94
Hyperebene	86	PFH	97, XX
ICP	94	Plumb Bob	35
Iterative Closest Point	94	Point Cloud Library	20
ITU-R BT.601	28	Point Cloud Matching	94
Kalibrierung ² , 15, 23, 24, 25, 37, 38, 43, 47, 52, 76, 94		Point Feature Histogram	97, XX
Kamerakalibrierung	36	RANSAC	45
Kameramatrix	32	Referenzgeometrie	49
Kernel	86	Rekonstruktionsproblem	40
Korrespondenzproblem	40	Rektifizierung	46
KUKA KR 16-2	16	Rigid Body	40
KUKA.LOAD	52	Robotersteuerung	59
Lochkamera	31	ROS Industrial	57
Lokator	40	Segmentierungsverfahren	
MariaDB	22	Kantenorientiert	11
Medizinroboter		Modellbasiert	12
Acrobot	6	Pixelorientiert	11
Active Cannula	9	Regionenorientiert	11
Arbeitsraum	5	Texturorientiert	12
Arthrobot	3	Shen & Cai	70
Ausdauer	5	Snakes	83
CardioARM	9	Support Vector Machine	85
Caspar	6	SVM	85
CorPath	7	Target Registration Error	49
CyberKnife	7	TF	21
da Vinci	8	Überblick	
Einsatzfelder	4	Medizinrobotik	3
Fingerfertigkeit	5	Verzeichnung	
Genauigkeit	4	Radial	34
Innomotion	7	Tangential	34
iSYS 1	7	V-Modell	15
Online-Konsultation	6	VxWorks	60
		YCbCr	28