

**Oberflächenrekonstruktion im Rahmen von
Augmented Reality in der Leberchirurgie**

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Computervisualistik

vorgelegt von

Benjamin Durth

Betreuer: Dipl.-Ing. Christian Hansen (Fraunhofer MEVIS)

Erstgutachter: Prof. Dr. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Dr. Darko Ojdanić (Fraunhofer MEVIS)

Koblenz, im Juli 2010

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Bremen, 6. Juli 2010)

(Unterschrift)

Danksagung

```
/*
```

Diese Arbeit markiert den Abschluss meines Studiums, sie steht jedoch für viel mehr als nur das. Ich danke an dieser Stelle allen, ohne die ich dieses Ziel nie erreicht hätte. Für Eure entgegengebrachte Freundschaft und Liebe, für Eure Geduld und Euer Verständnis werde ich immer dankbar sein.

Meiner lieben Familie, die mich in jeder Lebenslage und nicht zuletzt auch finanziell bedingungslos unterstützt hat, bin ich zu tiefst verbunden, danke Marlies, Albert und Patrick!

Ein großes Danke geht, für den Support aus der Heimat, ganz besonders an Eva Schaefer, Stefan Bröhl, Andreas Ens, Sascha Ludwig, Heiko Michels, Christian Schneider, Andreas von Arb und Prof. Dr. Müller.

Die Arbeit entstand am Fraunhofer-Institut für Bildgestützte Medizin MEVIS in Bremen, mein Betreuer Christian Hansen hat mich in dieser Zeit stets sicher begleitet und immer wieder aufs Neue motiviert.

Den vielen Mitarbeitern und Studenten, die ich in meiner Zeit hier kennenlernen durfte, will ich ebenfalls für ihre Unterstützung und Freundschaft danken. Ganz besonders gilt dies für Jan Wieferich und Kunal Masse, wegen denen ich mich im hohen Norden gleich heimisch gefühlt habe.

Daniel Demedts danke ich ganz speziell für seine ruhige, offene Art, seine stets kompetente Hilfe bei Problemen und fürs gründliche Korrekturlesen.

Jan Strehlow hat mir bei Testdurchläufen hilfsbereit unter die Arme gegriffen und immer wieder gute Denkanstöße gegeben - und nicht zuletzt ein zweites Zuhause, zusammen mit Anna-Lena. ;)

```
*/
```

Inhaltsverzeichnis

1 Einleitung	1
2 Motivation und Zielsetzung	4
2.1 Potentieller Nutzen für die Leberchirurgie	4
2.2 Übergeordneter Kontext und Zieldefinition	6
3 Grundlagen	10
3.1 Die menschliche Leber	10
3.2 Computergestützte Leberchirurgie	12
3.2.1 Registrierung	12
3.2.2 Navigation	14
3.2.3 Tracking	15
3.2.4 Visualisierung	16
3.3 Problemstellung und Nutzen einer Kalibrierung	17
3.4 Methoden zur Oberflächenrekonstruktion	18
4 Kalibrierung des Projektor-Kamera-Systems	20
4.1 Ermittlung der Kameraintrinsik	21
4.2 Intrinsische Kalibrierung des Projektors	22
4.3 Extrinsische Kalibrierung	25
5 Oberflächenrekonstruktion mittels Structured-Light	27
5.1 Geometrische Triangulierung und das Gray-Code-Verfahren	28
5.2 Das Sinusphasen-Verfahren	31
6 Konzeption des Systems zur Oberflächenrekonstruktion	32
6.1 Wahl einzusetzender Methoden	32
6.2 Softwaretechnischer Entwurf in MeVisLab	34
6.3 Konzeption einer Testumgebung	37
7 Realisierung	41
7.1 Eingesetzte Softwareumgebung	41

7.2	Implementierung benötigter Module	42
7.2.1	Kameraansteuerung	43
7.2.2	Kalibrierungsmodul	45
7.2.3	Rekonstruktionsmodul	49
8	Validierung	55
8.1	Hypothesen	55
8.2	Auswertungsmethodik	56
8.3	Versuchsreihen	57
9	Ergebnisse	61
9.1	Kalibrierung des Projektor-Kamera-Systems	61
9.2	Oberflächenrekonstruktion	65
10	Resümee und Ausblick	75
Anlagen		77
Literaturverzeichnis		80

Abbildungsverzeichnis

1.1	Schema der adaptiven Lichtprojektion	1
1.2	Schema der Oberflächenrekonstruktion	2
2.1	Traditionelle Navigationsunterstützung per Ultraschalluntersuchung der Leber.	5
2.2	Navigationsunterstützung per Structured-Light-Verfahren.	6
2.3	Visualisierung im Rahmen klassischer Navigation und per Lichtprojektion im intraoperativen Einsatz.	7
3.1	Vergleich der menschliche Leber als schematische Abbildung und ihre Darstellung im operativen Eingriff	11
3.2	Registrierung per Abtastung von Landmarken.	13
3.3	Physikalische Simulation der Leber mittels Feder-Masse-Modell.	14
3.4	Markerbasiertes Tracking in der comptergestützten Leberchirurgie.	16
3.5	Texturiertes Gefäßbaummodell und Einsatz im Rahmen direkter Lichtprojektion.	17
3.6	Das allgemeine Lochkameramodell.	18
3.7	Vergleich kontaktbasierter und kontaktloser Verfahren zur Oberflächenrekonstruktion.	19
4.1	Schematischer Aufbau zur Projektor-Kamera-Kalibrierung	21
4.2	Kamerakalibriermuster und Eckpunkt detektion.	22
4.3	Schematischer Geräteraufbau zur Verdeutlichung der extrinsischen Kalibrierung	26
5.1	Problemstellung geometrischer Triangulierung	28
5.2	Identifikation der Postition projizierter Pixel im Kamerabild.	30
5.3	Schematische Kodierung von Projektorzeilen	30
5.4	Arbeitsschritte des Sinusphasen-Verfahrens	31
6.1	Modulbausteine im schematischen Datenfluss der Oberflächenrekonstruktion . .	34
6.2	Sequenzdiagramm des Gesamtverfahrens zur Oberflächenrekonstruktion	36
6.3	Schematischer Geräteaufbau	38
6.4	Zu Testzwecken verwendete Phantommodelle	39
6.5	Verwendung einer Schweineleber als Testobjekt	40

7.1	Beispielnetzwerk in der Software MeVisLab	42
7.2	Klassenbeziehungen der Kameraanbindung	43
7.3	Schema der Kanalwandlung bei der Kameraansteuerung	44
7.4	Benutzungsoberfläche des Moduls <i>VideoCapture</i>	45
7.5	Schematische Datenflussbeschreibung des Moduls <i>VideoDeviceCalibration</i>	46
7.6	Grafische Benutzungsoberfläche des Moduls <i>VideoDeviceCalibration</i>	47
7.7	Manueller Zuschnitt, Eckpunktkorrektur und Gesamtvisualisierung von Kalibrermustern durch das Modul <i>VideoDeviceCalibration</i>	48
7.8	Visualisierung des Projektor-Kamera Aufbaus	49
7.9	Grafische Benutzungsoberfläche und Beispieldaten des Moduls <i>StructuredLightPatternGenerator</i>	50
7.10	Schematischer Datenfluss des Moduls <i>StructuredLightMacro</i>	51
7.11	Grafische Benutzungsoberfläche des Moduls <i>StructuredLight</i>	52
8.1	Für Versuche eingesetzter Geräteaufbau.	59
8.2	Ablauf eines Versuchs.	60
9.1	Zur Ermittlung der Kameraintrinsik eingesetzte, beispielhafte Aufnahmen.	62
9.2	Zur Ermittlung der Projektiorintrinsik eingesetzte, exemplarische Aufnahmen.	64
9.3	Initiale Rekonstruktionsergebnisse (Phantommodell Kugel).	66
9.4	Rekonstruktionsergebnisse des Sinusphasen-Verfahrens (Referenzdatensätze)	66
9.5	Rekonstruktionsergebnisse (Schweineleber)	67
9.6	Projektion von Gray-Code-Mustern auf das Phantommodell <i>L_A</i>	68
9.7	Visualisierung der Korrespondenzen zu Projektorzeilen bzw. Spalten sowie rekonstruierte Tiefenwerte des Phantommodells <i>L_A</i>	68
9.8	Verschiedene Ansichten der rekonstruierten, texturierten Oberfläche des Phantommodells <i>L_A</i>	69
9.9	Streifenbildung als Rekonstruktionsartefakt	70
9.10	An rekonstruierten Oberflächendaten durchgeführte Nachverarbeitungsschritte.	72
9.11	Rekonstruktionsergebnisse (Phantommodell <i>L_B</i>)	73
9.12	Rekonstruktionsergebnisse (Anatomiemodell)	74

Tabellenverzeichnis

8.1	Hardwarekomponenten der eingesetzten Testumgebung	57
9.1	Exemplarische Zeitmessung intrinsischer Kamerakalibrierung	62
9.2	Exemplarische Zeitmessung intrinsischer Projektorkalibrierung.	64
9.3	Vergleich der rekonstruierten Oberfläche mit dem Referenzdatensatz (Leberphantom L_A)	70
9.4	Vergleich der rekonstruierten Oberfläche mit dem Referenzdatensatz (Leberphantom L_B)	71

Listings

1	Umschreiben der Bildkanäle von BGR in RGB	78
2	Generierung eines Sinusphasenbildes	78
3	Aufnahme einer Projektionsfolge im Sinusphasen-Verfahren durch die Python-Funktion sinusPhaseCaptureSeries	79

1 Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Konzeption und Umsetzung eines Verfahrens zur Gewinnung dreidimensionaler Oberflächendaten. Dieses Verfahren kann beispielsweise im Zuge von computergestützten Leberoperationen dazu benutzt werden die Leberoberfläche als dreidimensionales Modell zu rekonstruieren. Gewonnene Daten können zur Navigationsunterstützung genutzt werden und präziseres Arbeiten der Chirurgen an den Strukturen der Leber ermöglichen. Die Oberflächenrekonstruktion stellt zudem den grundlegenden Bestandteil eines *Augmented-Reality-Systems* dar. Jenes am Fraunhofer Institut für bildgestützte Medizin (MEVIS) geplante und auf Lichtprojektion basierende Verfahren, soll in Zukunft dazu dienen eine anatomisch korrekte, adaptive Projektion von inneren Strukturen der Leber auf dessen Oberfläche durchzuführen, siehe Abbildung 1.1. Diese Arbeit stellt somit einen Teil der dazu notwendigen Grundlagenarbeit dar.

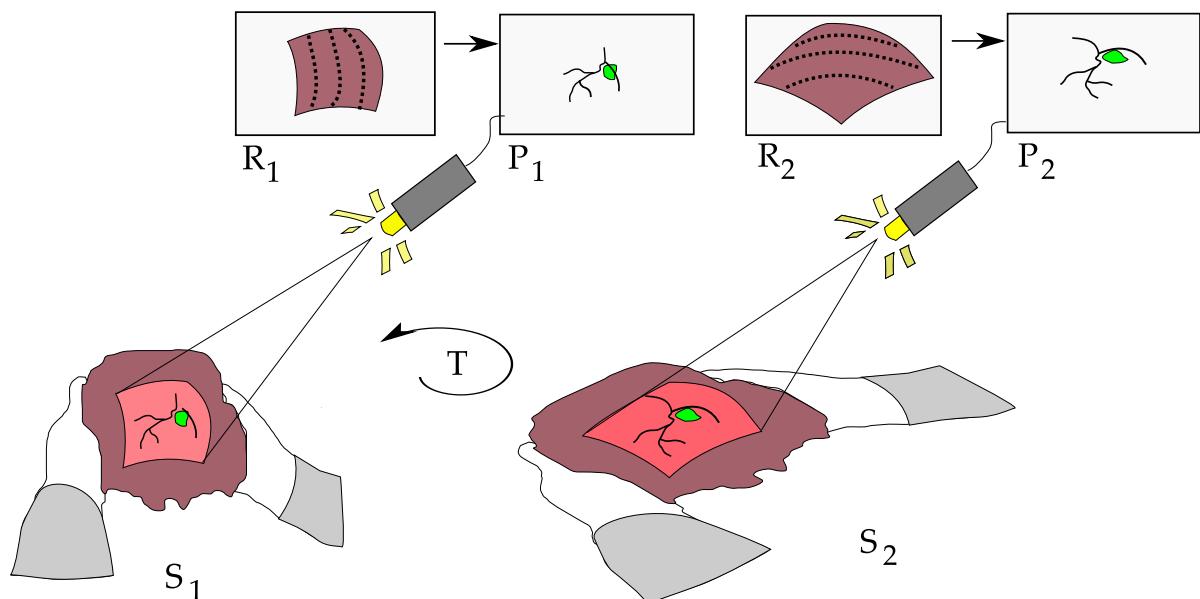


Abbildung 1.1: Schema der adaptiven Lichtprojektion. Die Transformation T zwischen den Szenen S_1 und S_2 wird durch Vergleich der Oberflächenrekonstruktionen R_1 und R_2 ermittelt. Die Anwendung von T auf das Modell P_1 des Gefäßbaums ergibt das Modell P_2 , welches als 2D-Bild durch den Projektor P in die Szene projiziert wird.

Zur Gewinnung der Oberflächendaten kommt in der vorliegenden Arbeit ein *Projektor-Kamera-System* zum Einsatz, welches auf der Verwendung eines *Structured-Light-Ansatzes* basiert. Hierbei werden Tiefenpunkte beispielsweise mittels optischer Triangulierung zwischen Kamera, Projektor und Oberflächenpunkten bestimmt. Das allgemeine Funktionsprinzip der Rekonstruktion ist in Abbildung 1.2 dargestellt.

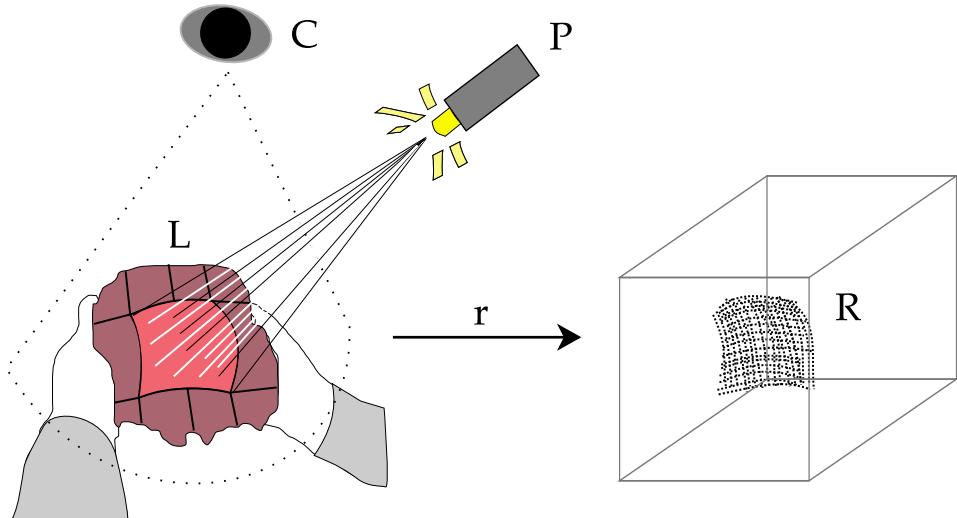


Abbildung 1.2: Schema der Oberflächenrekonstruktion durch ein Structured-Light-Verfahren. Die Kamera C beobachtet die Szene inklusive Leber L , auf welche der Projektor P aufprojiziert. Das Ergebnis der Rekonstruktion mittels Triangulierung r ist die in Weltkoordinaten vorliegende Punktwolke R . Eine detaillierter Darstellung zu der bei der Triangulierung zu lösenden Problematik wird in Abbildung 5.1 geboten.

Aufbau der Arbeit Nachdem nun ins Thema der Arbeit eingeführt wurde, folgt in Kapitel 2 die *Motivation* des Verfahrens aus medizinischer und praktischer Sicht. Daraufhin wird eine Zielsetzung erarbeitet, die sich aus, für das Oberflächenrekonstruktionsverfahren relevanten Aspekten aus Medizin und Technik zusammensetzt.

Allgemeine *Grundlagen*, die zum weiteren Verständnis der Arbeit nötig sind, werden in Kapitel 3 vorgestellt. Ein kurzer Einblick in den aktuellen Stand bildgestützter Chirurgie sowie signifikante Eigenschaften der Leberanatomie werden in diesem Zuge geboten. Die allgemeine Problemstellung der Kalibrierung und Oberflächenrekonstruktion wird zudem vorgestellt.

Kapitel 4 widmet sich dem Teilproblem der *Kalibrierung*. Hier folgt die Beschreibung der gewählten Methoden, sowie eine eingehende Darstellung der Schritte zur Durchführung der Kalibrierung des Projektor-Kamera-Systems.

Die Theorie zur *Oberflächenrekonstruktion* mittels Structured-Light wird in Kapitel 5 geboten. Hierbei werden mit dem Gray-Code- und Sinusphasen-Verfahren zwei Hauptvertreter des Forschungsgebiets vorgestellt, wobei nur das Gray-Code-Verfahren auf die Kalibrierung des Projektor-Kamera-Systems direkt angewiesen ist.

In Kapitel 6 folgt der softwaretechnische Entwurf des Systems zur Oberflächenrekonstruktion sowie die *Konzeption* des Versuchsaufbaus und der damit durchzuführenden Testläufe.

Die softwaretechnische *Realisierung* und die Einbettung der Methoden im Rahmen der verwendeten Softwareumgebung *MeVisLab* wird in Kapitel 7 beschrieben.

Es folgt in Kapitel 8 die *Validierung* der Ergebnisse, die aus Konzeption und Realisierung hervorgegangen sind. Anhand von Labormodellen wird untersucht inwieweit die Anforderungen an das Verfahren eingehalten wurden.

Die *Ergebnisse* von Kalibrierung und Rekonstruktion werden in Kapitel 9 vorgestellt und besprochen.

Die Arbeit schließt in Kapitel 10 mit dem *Resümee* des Erreichten und zeigt Einschränkungen sowie möglichen Lösungsansätze auf. Weitere Entwicklungen, die einen Einsatz im klinischen Gebrauch ermöglichen können, werden in Form eines *Ausblicks* angesprochen.

2 Motivation und Zielsetzung

Operationen im Bereich der Weichteilchirurgie und speziell der Leberchirurgie haben durch exakt und schnell arbeitende Technologien wie computergestützte Verfahren und Sensormesungen, allgemein Computergestützte Intervention (CI), eine stetige Verbesserung erfahren. Vorteile, die hierbei entstehen, sind vor allem die durch hochwertige Bildgebungsverfahren wie Computertomographie (CT), oder Magnetresonanztomographie (MRT), meist präzisere Diagnostik und die somit gesteigerte Planungsgenauigkeit von Eingriffen. Ferner entsteht kein Informationsverlust wie beispielsweise beim Röntgen, da nicht nur 2D- sondern auch 3D-Daten des Körpers erfasst werden können. Die Gewinnung exakter Bilddaten ermöglicht und verbessert gleichermaßen die Diagnose, Analyse sowie Behandlungsplanung einer Vielzahl von Krankheiten. Einerseits werden dabei Risikofaktoren einer Operation besser abschätzbar, andererseits werden minimal invasive Eingriffe ermöglicht, welche die Genesungs dauer des Patienten stark begünstigen können.

2.1 Potentieller Nutzen für die Leberchirurgie

Bei Tumorbefall der Leber und Behandlung mittels *Resektion*, kann die CI bereits eine wichtige Rolle spielen. Weil das Organ an sich stark regenerationsfähig ist, können in der Regel bis zu drei Viertel seines Gewebes gefahrlos entnommen werden [Pawlak u. a., 2008]. Je nach gesundheitlichem Zustand des Patienten ist nach einer Rekonvaleszenz von sechs bis acht Wochen die präoperative Größe der Leber wieder erreicht. [J u. a., 2006]. Der guten Regenerationsfähigkeit der Leber steht jedoch ein beträchtlicher Risikofaktor gegenüber. Bei ihrer Resektion besteht die Gefahr der Durchtrennung wichtiger Blutgefäße, was zu Komplikationen bis hin zu einem Stillstand der Leberdurchblutung führen kann. Daraufhin kann partieller Gewebeverlust oder im schlimmsten Fall vollständiges Organversagen die Folge sein. Es ist daher wichtig eine sehr exakte Planung von Schnittflächen zur Entfernung des Tumors, den sogenannten Resektionsflächen, vorzunehmen. Idealerweise zudem so, dass wichtige Gefäße geschont und mehr als ausreichend Restlebervolumen erhalten bleibt.

Um nicht vom geplanten Schnittweg abzuweichen, sind während eines Eingriffs exakte Positionsdaten der chirurgischen Instrumente in Relation zum behandelten Organ nötig. Die Leber ist ein farblich stark uniformes Organ und mit wenigen, anatomisch markanten Merkmalen

versehen, so dass dem Chirurg kaum Anhaltspunkte zur genauen Orientierung bleiben. Die nach einer präoperativen Untersuchung der Leber erstellten 3D-Planungsdaten werden deshalb während der Intervention für einen exakten Abgleich mit dem vorgefundenen Organzustand verwendet.

Die Navigation wird weiterhin durch die Tatsache erschwert, dass die Leber meist freigelegt und somit in einem beweglichen Zustand behandelt wird. Positionsdaten der Leber sind dabei grundsätzlich nur solange exakt, wie Lage und Orientierung während einer Operation mit Planungsdaten in Deckung gebracht werden können. Ein gängiges Verfahren der CI ist es zu Beginn der Operation ein dreidimensionales Planungsmodell der Leber gegen ein intraoperativ erstelltes Ultraschall-Bild zu registrieren. Da aber intraoperative Transformationen wie Translation oder auch Deformation der Leber stattfinden, wird es im weiteren Verlauf notwendig eine erneute Registrierung des aktuellen Organzustands durchzuführen. Errechnete Positionen von Operationswerkzeugen können auf diese Weise valide gehalten werden, siehe dazu auch Abschnitt 2.2.1.

Die zentrale Motivation der Arbeit ist es nun, durch eine automatische Oberflächenrekonstruktion dazu beizutragen, die Zahl der manuellen Registrierungen zu vermindern. Im Gegensatz zu einer, in Abbildung 2.1 gezeigten, manuellen Ultraschallabtastung kann eine Oberflächenrekonstruktion zudem automatisch, kontaktlos und innerhalb eines kleinen Zeitrahmens ablaufen. Dies wird mittels dem in Abbildung 2.2 gezeigten Structured-Light-Verfahren ermöglicht.

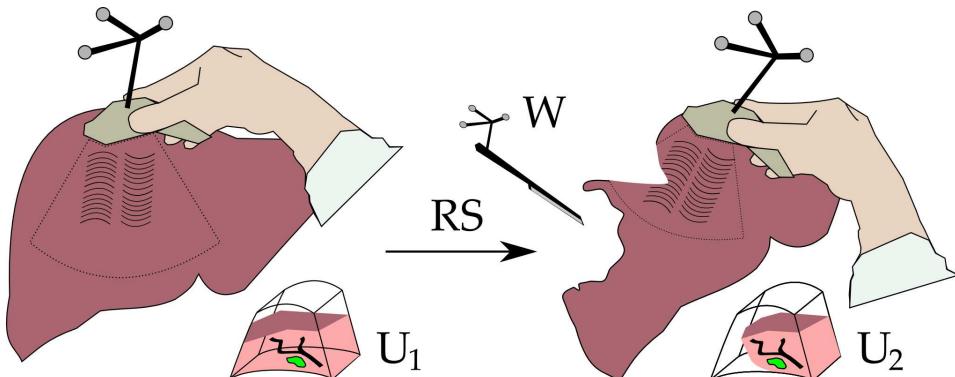


Abbildung 2.1: Traditionelle Navigationsunterstützung per Ultraschalluntersuchung der Leber. Zu Anfang wird die Ultraschalluntersuchung U_1 vorgenommen, nach der Resektion RS per getracktem Werkzeug W wird der aktuelle Zustand innerer Organstrukturen durch eine weitere Ultraschalluntersuchung U_2 ermittelt.

Eine weitere Motivation zur Oberflächenrekonstruktion und somit für die Verwendung eines Lichtprojektionsverfahrens an Stelle bisheriger Methoden, ist die Entlastung des allgemeinen Operationsablaufs. Die Visualisierung des Navigationsprozesses beruht bisher meist auf der Nutzung eines Monitors, welcher Modell und Werkzeuge in einer gemeinsamen virtuellen Sze-

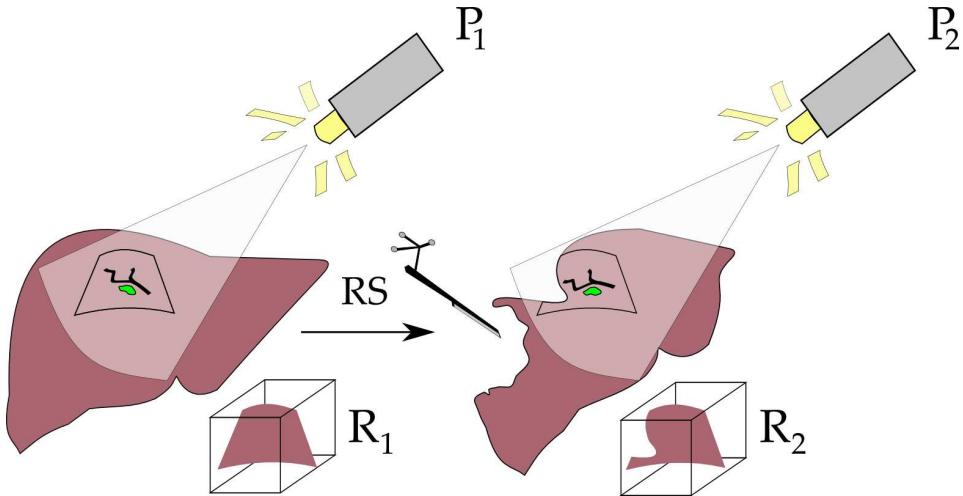


Abbildung 2.2: Navigationsunterstützung per Structured-Light-Verfahren. Anfangs wird die Leberoberfläche R_1 rekonstruiert. Nach Transformation im Zuge der Resektion RS wird die Leberoberfläche erneut rekonstruiert. Anhand der gewonnenen Leberoberfläche R_2 kann die Projektion P_2 erstellt werden und auf der Leberoberfläche als Navigationshilfe dienen.

ne darstellt, siehe Abbildung 2.3 (a). Zur Überprüfung seiner Arbeitsschritte muss der Chirurg hierbei laufend Sichtwechsel von der Operationsstelle zum Monitor und zurück durchführen. Hierbei wird es nötig die durch die Visualisierung gesammelten Informationen auf die tatsächliche Operationsstelle abzubilden. Der dazu notwendige kognitive Aufwand kann den Chirurgen, der stets hochkonzentriert arbeiten muss, zusätzlich belasten. Es gilt allgemein jedoch die Zielsetzung, jedwede Ablenkung vom eigentlichen Operationsgeschehen, sowie unnötige Aufwandssteigerungen zu vermeiden. Eine automatische Oberflächenrekonstruktion kann helfen dies zu gewährleisten, indem es dem in Abbildung 2.2 gezeigten Lichtprojektionssystem benötigte Oberflächendaten zur Verfügung stellt. Durch die mittels lichtstarkem Projektor durchgeführte, direkte Aufprojektion von internen Organstrukturen auf die Leberoberfläche, könnte die fehleranfällige Navigationsunterstützung per Monitor somit hinfällig werden. Subjekt und Visualisierung sind auf diese Weise stets im Blickfeld des Chirurgen fusioniert.

2.2 Übergeordneter Kontext und Zieldefinition

Das übergeordnete Ziel dieser Arbeit besteht darin, ein Gesamtsystem zur Navigationsunterstützung, durch Verfolgung (Tracking) von elastischen Organtransformationen, im Rahmen weiterer Forschung zu entwickeln. Dieses Gesamtsystem wird sich unter anderem auf die Verwendung eines Lichtprojektionsverfahren stützen. Es soll neben der detaillierten Messung der Organoberfläche zudem eine Verfolgung der Position und Orientierung des gesamten Organs bieten. Die Aufprojektion von Planungsdaten auf die Oberfläche soll als abschließender Schritt



(a) Nachteilige Aufmerksamkeitsteilung der klassischen CI mit getrackten chirurgischen Instrumenten (unten) sowie Kontrollmonitor (oben), Bildquelle: Medical Computing Lab, School of Computing, Queen's University

(b) Fusionierung von Organ und Gefäßvisualisierung mittels direkter Lichtprojektion. Die Durchführung erfolgte am Robert-Bosch Krankenhaus Stuttgart im Rahmen eines Kooperationsprojektes zwischen MeVis und der Universität Karlsruhe, Bildquelle: Fraunhofer MEVIS

Abbildung 2.3: Visualisierung im Rahmen klassischer Navigation und per Lichtprojektion im intraoperativen Einsatz.

so adaptiert, d.h. das zu projizierende Bild so vorverarbeitet werden, dass es sich an die rekonstruierte Oberfläche bestmöglich angleicht. Somit wäre eine anatomisch korrekte Visualisierung von Planungsdaten oder auch segmentierter Organstrukturen im Rahmen von intraoperativen Eingriffen möglich. Ziel dieser Arbeit ist es nun, als einen ersten Schritt, das dazu benötigte Verfahren zur Oberflächenrekonstruktion umzusetzen.

Nachdem im vorangegangenen Abschnitt in die allgemeine Problemstellung der Leberchirurgie und die Motivation der Oberflächenrekonstruktion eingeführt wurde, folgt nun die Formulierung einer darauf basierenden, konkreten Zielsetzung. Dabei werden Anforderungen an das Rekonstruktionsverfahren aufgezeigt, wobei es mit dem medizinischen und technischen Bereich maßgeblich zwei Felder sind in denen diese erhoben werden können. Festzuhalten bleibt, dass die zu entwickelnde Oberflächenrekonstruktion in erster Linie im Rahmen von Laboruntersuchungen verifiziert werden wird. Auf lange Sicht und im Rahmen weiterer Forschung zudem im klinischen Gebrauch. Anforderungen, welche bereits maßgeblich den praxisbezogenen, klinischen Gebrauch berücksichtigen, werden schon in dieser Arbeit mit in die Betrachtung einbezogen. Somit sollen konzeptionelle Grundlagen für weitere Entwicklungen geschaffen werden.

Systemanforderungen Zur Oberflächenrekonstruktion soll ein Verfahren auf Basis von Structured-Light zum Einsatz kommen. Das Ziel wird es also sein, ein dazu benötigtes Projektor-Kamera-System einzurichten. Auf Seiten zu verwendender Hardware soll das Verfahren auf handelsüblichen Geräten, d.h. Computern, Projektoren und Kameras, lauffähig sein und weiter

keine speziellen Anforderungen an die verwendete Peripherie stellen. Auf softwaretechnischer Seite soll es in Form von Modulen in der Software MeVisLab umgesetzt werden. Module zur Ansteuerung und Kalibrierung einer Kamera und zur Oberflächenrekonstruktion sind dabei Gegenstand dieser Arbeit. Module, die ein Markertracking sowie eine Projektionskorrektur umsetzen sind ebenfalls als eigenständige Einheiten identifizierbar und in Zukunft zu implementieren.

Medizinische Anforderungen Von medizinischer Seite aus gilt die bestmögliche Integration der Oberflächenrekonstruktion in den chirurgischen Arbeitsablauf als Leitziel. Um Einsatzpunkte des Verfahrens zu ermitteln bietet es sich im Folgenden an den tatsächlichen Ablauf einer Leberresektion näher zu betrachten. Dieser gliedert sich in vorbereitende Maßnahmen zur Operation und eine Reihe von Schritten, die während der chirurgischen Behandlung der Leber durchgeführt werden.

Im Rahmen der Vorbereitung des Eingriffs sind Schritte wie Aufbau und Inbetriebnahme des Projektor-Kamera-System notwendig. Hierbei ist es von vorrangiger Bedeutung, dass innerhalb einer kurzen Zeitspanne und ohne großen Aufwand eine akkurate Kalibrierung durchgeführt werden kann. Die Kalibrierung sollte idealerweise semi- oder sogar vollautomatisch ablaufen und ein Minimum an Interaktion erfordern. Während des Eingriffs selbst soll nach Möglichkeit keine manuelle Korrektur am Aufbau vorgenommen werden müssen. Solange die erstellte Oberflächenrekonstruktion als valide angesehen wird, soll zudem keine weitere Interaktion mit dem System notwendig sein.

Vom chirurgischen Arbeitsablauf her steht am Beginn einer Leberoperation die Sichtung des Organs durch ihre Freilegung und damit die Trennung von angrenzenden Gewebestrukturen wie z.B. dem Zwerchfell. Es folgt meist eine Ultraschallaufnahme, die dazu dient den aktuellen Zustand des Organs zu erfassen und die, per Markertracking verfolgten, Instrumentenpositionen in räumliche Relation zur Leber zu bringen. Ab diesem Punkt ist eine Navigationsunterstützung geboten, die solange das Organ nur minimal transformiert wird, als valide angesehen werden kann.

Um eine Oberflächenrekonstruktion durchführen zu können ist die Gewährleistung einer freien Sicht auf die Operationsstelle Voraussetzung. Ist dies nicht der Fall, so ist die durchgeführte Oberflächenrekonstruktion invalide, was dem behandelnden Chirurgen mittels visueller Rückmeldung des Systems angezeigt werden sollte. Etwaige auftretende Abbildungsfehler sollen ebenfalls zur Invalidierung der Rekonstruktion führen. Ein maßgeblicher Faktor, der sich ebenfalls negativ auf die Rekonstruktion auswirken kann ist das Auftreten von Reflexionen auf der Leberoberfläche. Diese sind durch die Viskosität des Organs und die Verwendung von Operationslampen bedingt. Eine Anforderung ist es also die entstehenden Reflexionen minimal zu halten oder so gut wie möglich zu kompensieren.

Immer dann, wenn im Verlauf des Eingriffs umfangreicher an der eigentlichen Resektionsfläche gearbeitet wird, kann es notwendig sein, eine erneute Rekonstruktion durchzuführen. Da sich die Leber nach einem größeren Arbeitsschritt jedoch in Form und Größe oft stark verändert hat, wird es wichtig, die Differenz zur vorherigen Rekonstruktion zu berechnen. Daraus können entweder Schlüsse zur nötigen Anpassung der Projektion gezogen werden oder es wird bei zu starker Veränderung des Organs von einer Rekonstruktion abgesehen. Ist das Organ bereits zu großen Teilen resiziert und somit stark deformiert, so würde die Validität der Projektion des Gefäßbaums stark beeinträchtigt werden und ein Informationsgewinn durch ihre Durchführung tendenziell unwahrscheinlich. Die Rekonstruktion wäre hier ein Faktor, der den Workflow durch die Lieferung schlechter Ergebnisse verlangsamen könnte. Um das Vorgehen der Chirurgen allgemein nicht zu verzögern sollte zudem der zeitliche Aufwand für die Rekonstruktion so gering wie möglich gehalten werden. Eine echtzeitfähige Rekonstruktion ist von technischer Seite her zwar machbar, siehe hierzu [Zhang u. Yau, 2007], in dieser Arbeit jedoch kein Ziel. Nach Abschluss der Resektion und somit gegen Ende des Eingriffs sind keine weiteren Rekonstruktionen notwendig.

Es bleibt festzuhalten, dass während des Eingriffs die Oberflächenrekonstruktion nicht zu jedem Zeitpunkt zwingend erforderlich ist. Im fortgeschrittenen Stadium der eigentlichen Leberresektion ist sie sogar nicht uneingeschränkt sinnvoll, beziehungsweise erwünscht. Die zentralen Einsatzmöglichkeiten des hier entwickelten Verfahrens zur Oberflächenrekonstruktion sind eher im Anfangsstadium der durchzuführenden Leber-OP anzusiedeln.

Zusammenfassung Die Zielsetzung aus technischer Sicht ist in erster Linie eine Oberflächenrekonstruktion unter Benutzung eines Structured-Light-Ansatzes und des dafür nötigen Projektor-Kamera-Systems zu realisieren. Softwaretechnisch gilt es die Module zur Ansteuerung und Kalibrierung von Projektor und Kamera sowie das Modul zur Oberflächenrekonstruktion in der Software MeVisLab zu implementieren. Aus medizinischer Sicht besteht das Ziel darin, dass die nötigen Systemaufbauten innerhalb kurzer Zeit durchführbar sind und die Kalibrierung dabei unkompliziert ist, bzw. im besten Fall automatisch abläuft. Ebenso soll die Rekonstruktion performant sein und dabei gleichzeitig akkurate Ergebnisse liefern. Des Weiteren sollte ein in der Praxis verwendetes System Rückmeldung über Abbildungsfehler liefern und bei Bedarf eine Rekonstruktion invalidieren.

3 Grundlagen

Nachdem medizinische Hintergründe der Oberflächenrekonstruktion aufgezeigt und die Zielsetzung erarbeitet wurde, folgt nun die Darstellung der für das tiefere Verständnis der Problemstellung nötigen Grundlagen. Um einerseits die medizinischen Rahmenbedingungen zu verstehen, in die sich die Oberflächenrekonstruktion einfügen muss, werden relevante Eigenschaften der Leber sowie des Zielgebiets der computergestützten Leberchirurgie aufgezeigt und im Hinblick auf die Zielsetzung besprochen. Die in diesem Feld verwendete Registrierung, Navigation sowie das Tracking, dienen als Grundlage des, auf die Oberflächenrekonstruktion aufbauenden Verfahrens der adaptiven Lichtprojektion. Diese Techniken werden folglich kurz angesprochen.

Zur essentiell erforderlichen Kalibrierung des Projektor-Kamera-Systems wird zunächst deren allgemeine Theorie vorgestellt. Daraufhin wird die generelle Funktionsweise von Verfahren zur Oberflächenrekonstruktion erläutert. Detailliertere Ausführungen zur Arbeitsweise der zur Problemlösung gewählten Kalibrierung und Oberflächenrekonstruktion folgen in den Kapiteln 4 sowie 5

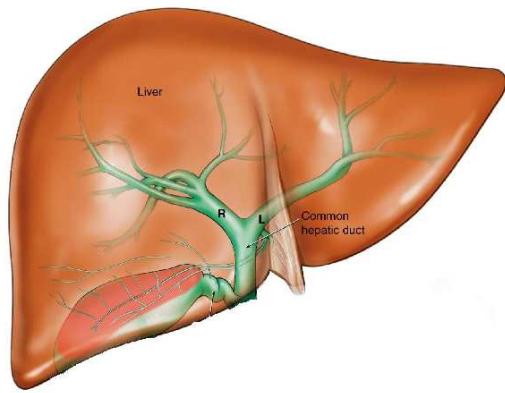
3.1 Die menschliche Leber

Die Leber übernimmt als zentrales Stoffwechselorgan lebenswichtige Aufgaben im Organismus des Menschen, so u.A. in der Eiweißproduktion und dem Abbau von Giftstoffen. Mit einem Gewicht von bis zu 3000 Gramm ist sie eines der größten Organe [J u. a., 2006]. Der Fakt, dass die Leber an sich sehr viskos und ihre Oberfläche farblich homogen strukturiert ist, ergibt für das in Abbildung 2.3 (b) dargestellte Lichtprojektionsverfahren sowie die Oberflächenrekonstruktion per Structured-Light zwei wichtige Implikationen.

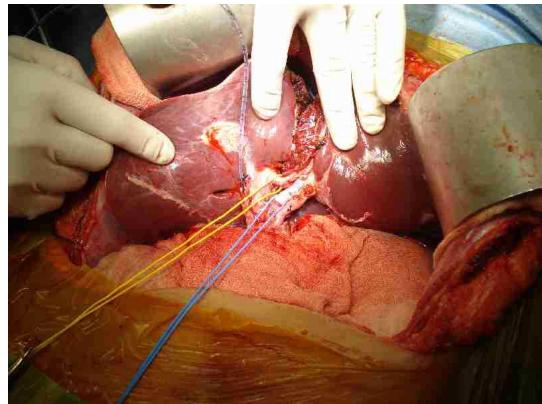
Die hohe Viskosität der Leber erschwert in einer Operation einerseits ihre Immobilisierung, so dass eine Verschiebung nicht gänzlich verhindert werden kann. Ein Navigationsverfahren muss wegen auftretender Transformation deshalb stets dafür Sorge tragen, den aktuellen Zustand der Leber zu erfassen und gegen Planungsdaten abzugleichen. Für ein Lichtprojektionsverfahren hieße dies, dass immer dann eine Oberflächenrekonstruktion durchgeführt werden muss, wenn die Transformation der Leber eine erneute Navigationsunterstützung erforderlich macht.

Eine Alternative zum Tracking auf Basis manueller Registrierung per Ultraschall können optische Trackingverfahren darstellen. Diese sind dazu ausgelegt, zum Beispiel durch Analyse von Videomaterial, markante Merkmale im Bild zu detektieren. Ein solches Verfahren könnte die Verfolgung der Leber prinzipiell ermöglichen. Hier gilt es jedoch zu beachten, dass die Leber wegen der Homogenität ihrer Oberfläche nicht reich an natürlichen, anatomischen Merkmalen ist, an denen sich ein optisches Trackingverfahren orientieren könnte.

Zwar wäre eine potentielle Landmarke der Leber das sogenannte „Ligamentum falciforme“, eine in Abbildung 3.1 (a) dargestellte, quer über die Oberfläche verlaufende Gewebefurche. Da diese in der intraoperativen Bildgebung jedoch oft nicht deutlich genug zum Vorschein kommt, entfällt die Option, sie als Anhaltspunkt zu nutzen. Die Verwendung eines optischen Trackings, das auf anatomischen Markern basiert ist somit also tendenziell auszuschließen.



(a) Die menschliche Leber als schematische Abbildung, mit der mittig eingezeichneten Gewebefurche, dem „Ligamentum falciforme“. Bildquelle: Arizona Transplant Associates, PC



(b) Die menschliche Leber im Rahmen eines operativen Eingriffs mit dem Auftreten von Glanzlichtern. Bildquelle: Dr Vinay Kumaran, vkumaran.net

Abbildung 3.1: Vergleich der menschliche Leber als schematische Abbildung und ihre Darstellung im operativen Eingriff

Ein weiterer Punkt, den es bei der Verwendung eines Structured-Light-Verfahrens zu beachten gilt, ist die Tatsache, dass bei direkter Lichtprojektion, die Viskosität der Leber zum vermehrten Auftreten von Glanzlichtern beitragen kann, siehe Abbildung 3.1 (b). Dieser Effekt kann allgemein durch im OP verwendete, lichtstarke Lampen hervorgerufen werden und die Rekonstruktionsergebnisse negativ beeinflussen. Ein beobachteter Bildpunkt, für den kein Hell-Dunkel-Wechsel registriert werden kann, weil die betreffende Stelle wegen eines Glanzlichts immer hell erscheint, macht die genaue Rekonstruktion des dazugehörigen Oberflächenpunktes unmöglich. Hier muss deshalb darauf geachtet werden, etwaige Störquellen zu eliminieren, oder aber mögliche Störfaktoren zu minimieren. Während das Lichtprojektionsverfahren also aktiv arbeitet sollten alle Lampen so weit wie möglich abgedimmt werden.

3.2 Computergestützte Leberchirurgie

Die computergestützte Leberchirurgie (CL) ist eine Variante der CI und stellt ein chirurgisches Konzept bzw. eine Menge von Methoden dar, die auf der Verwendung von Computertechnologien basieren. Dieses Konzept beinhaltet in erster Linie die computerbasierte Gewinnung von Bilddaten, die Planung von Eingriffen mittels Simulationsverfahren und die intraoperative Navigation auf Basis von Registrierung und Tracking. Der Ablauf eines Eingriffs in der CL lässt sich dabei grob in präoperative, intraoperative sowie postoperative Maßnahmen gliedern.

Im Zuge präoperativer Maßnahmen findet die Gewinnung von meist dreidimensionalem Bildmaterial, etwa mittels CT bzw. MRT statt. In der Planungsphase erfolgt eine detaillierte Auswertung des Bildmaterials durch medizinisches Personal. In diesem Schritt wird unmittelbar nach einer Erstbefundung, wie etwa der Diagnose eines Lebertumors, der Operationsbereich festgelegt. Optimale Schnittpfade werden computergestützt berechnet, was seinerseits wiederum durch vorberechnete Risikomodelle abgesichert werden kann.

Zu Beginn der intraoperativen Phase steht meist eine Ultraschalluntersuchung. Die so gewonnenen Daten werden zur Registrierung des aktuellen Organzustands mit präoperativen Ultraschalldaten zur Deckung gebracht. Hierauf beginnt die Navigationsphase, in welcher Instrumente beispielsweise mechanisch oder aber optisch verfolgt werden. Die ermöglicht die Echtzeitvisualisierung der exakten Instrumentenposition in Bezug zu relevanten Organstrukturen. Im Folgenden werden die einzelnen Kernbereiche der CL näher beschrieben.

3.2.1 Registrierung

Registrierung ist der Prozess, bei dem ein 2D- bzw. 3D-Bild zur Deckung mir einem Referenzbild gebracht wird, oder aber zwei unabhängige Koordinatensysteme in ein Weltkoordinatensystem transformiert werden. Operationsinstrumente können durch die Abtastung von vorher festgelegten Landmarken, welche Raumpositionen innerhalb des für die Leber aufgespannten Koordinatensystems entsprechen, registriert werden. Hiermit wird erreicht, dass die Koordinatensysteme von Instrument und Organ einander entsprechen, siehe Abbildung 3.2. Somit ist die jeweilige Lage der Instrumente in Relation zur Leber, wie sie sich zum Zeitpunkt der letzten intraoperativen Ultraschalluntersuchung dargeboten hat, bekannt. Für rigide Objekte, welche sich im Raum ausschließlich verschieben, rotieren bzw. skalieren lassen, ist eine Abbildung ausreichend, welche sechs bzw. sieben Freiheitsgrade abdeckt. Liegt eine Skalierung vor, so werden mindestens drei Korrespondenzpunkte benötigt, für alle übrigen Fälle reichen hingegen zwei aus. Im Gegensatz zur rigen Registrierung muss bei einem deformierbaren Objekt wie der Leber eine nicht-affine Registrierung zum Einsatz kommen. Die Registrierung der Leber

kann in erster Linie also deshalb komplex werden, da das Organ einer ständigen Deformation unterliegt. Diese können intraoperativ aus den folgenden Gründen auftreten:

- Behandlung durch die Hand des Chirurgen.
- Atmung und Herzschlag des Patienten.
- Höhenverstellung des Operationstisches.
- Verlagerung des Patienten durch OP-Personal.
- Anatomische Veränderung der Leber seit der präoperativen CT/MRT Aufnahme.
- Unterschiedliche Füllstände des Magens oder Verlagerung umliegender Organe.

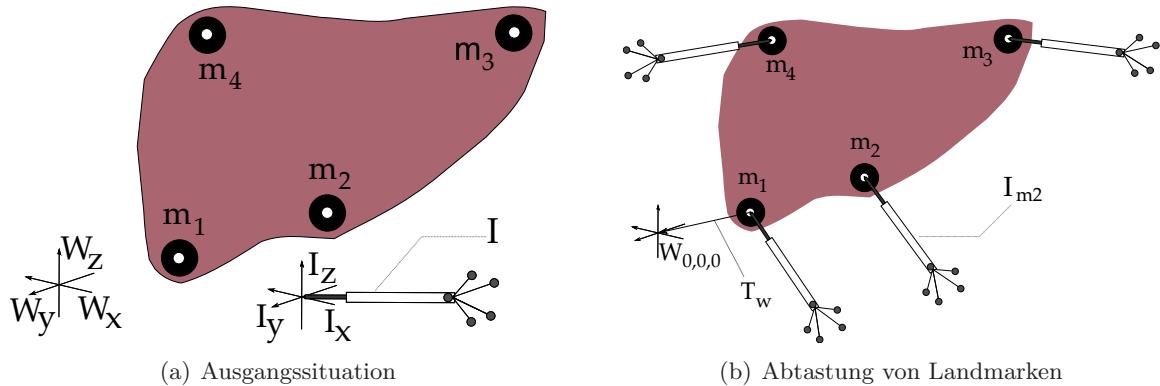


Abbildung 3.2: Registrierung per Abtastung von Landmarken. Aus den im Koordinatensystem $I(x, y, z)$ des getrackten Instruments I abgetasteten Landmarken $m_1 \dots m_4$ kann beispielsweise mit Versatz durch eine frei gewählte Transformation T_w der Ursprung des gemeinsamen Weltkoordinatensystems $W(0, 0, 0)$ festgelegt werden.

Eine rigide Transformation ist für das Tracking der Leberoberfläche somit nicht ausreichend, da die Abbildung einer nicht-affinen Transformation mittels Translation, Rotation oder Skalierung nicht möglich ist. Dieser Fakt wird aus Gründen der Vereinfachung und der Performanz, in der Praxis jedoch meist nicht in die Betrachtung einbezogen. Es ist in aller Regel zudem kein Ziel, das gesamte Organ inklusive dessen Verformung zu verfolgen, sondern vielmehr nur Teilbereiche, die für die Operation tatsächlich von Belang sind. Verschiedene Ansätze der nicht-affinen Verfolgung von Organoberflächen bestehen bereits.

Diese Aufgabenstellung wird von [Zhu u. a., 2008] beispielsweise als reines Optimierungsproblem angesehen, welches bei schlecht texturierten Oberflächen wie der Leber jedoch an seine Grenzen stößt. Die Lösung des Problems, per Tracking von fest auf der Leberoberfläche angebrachten Markern, wie sie hingegen [Weber u. a., 2008, Weber] vorschlägt, ist eine invasive Methode und wegen ethischer Gründe eventuell nicht durchsetzungsfähig. Die Verfolgung von Verformungen des gesamten Organs stellt damit weiterhin eine große Herausforderung in der CL dar.

3.2.2 Navigation

Die Navigation dient im Rahmen der CL dazu, die laufende Positionsbestimmung von Instrumenten in Relation zu Organstrukturen zu ermöglichen. Zur Navigationsunterstützung in der Leberchirurgie sind bereits vielfältige Ansätze verfolgt worden. Darunter in erster Linie das Ultraschallverfahren in Kombination mit manueller oder automatischer Registrierung von Markern [Markert u. a., 2007]. Aber beispielsweise auch physikalisch basierte Deformationsmodelle [Cash u. a., 2005] sowie die direkte Aufprojektion, wie sie [Krempien u. a., 2008] vorschlägt.

Navigation auf Basis von Ultraschall und anatomischen Markerpunkten setzt die in Abbildung 3.2 verdeutlichte Registrierung ein. Mittels intraoperativer Ultraschalluntersuchung an der freigelegten Leber werden im Vorfeld am Lebermodell vereinbarte Landmarken ausgemacht und mittels mechanisch oder optisch getracktem Operationswerkzeug entsprechend aufeinander registriert.

Ein anderer Ansatz zur Navigation wird mit deformierbaren Modellen verfolgt. Hierbei wird das Deformationsverhalten der Leber rechnerisch angenähert. Die dabei verwendete physikalische Simulation wird beispielsweise durch ein Feder-Masse-System ermöglicht. Mehrere Ansätze versuchen hierbei die aktuelle, durch Instrumente verursachte Deformation des Organs zu schätzen und daraufhin eine elastische Registrierung mit dem präoperativen Datensatz durchzuführen. Auf der algorithmischen Seite sind für deformierbare Modelle vor allem Finite-Element-Methoden (FEM), wie sie beispielsweise [Cash u. a., 2005] verwendet sehr exakt, jedoch in gleichem Maße rechenintensiv. In der Arbeit von [Suzuki u. a., 2004] wird ein Kollisionsmodell erstellt, das auf äußere Kräfte reagiert und im Innern des Modells befindliche Massenpunkte verschiebt, siehe hierzu Abbildung 3.3. Im Rahmen der Navigation kann dieses Verfahren dazu dienen, die Auswirkung von Instrumenten auf innere Strukturen der Leber zu simulieren.

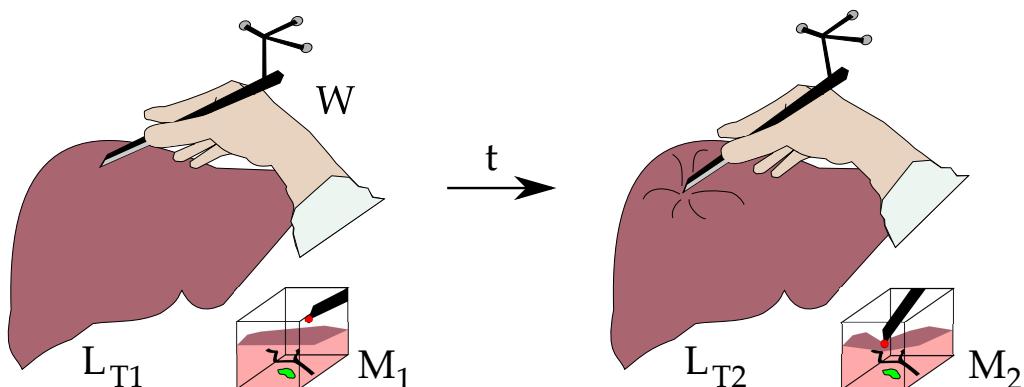


Abbildung 3.3: Physikalische Simulation der Leber mittels Feder-Masse Modell. Das getrackte Werkzeug W transformiert die Leber L_{T1} nach einem Zeitschritt t durch Druck auf dessen Oberfläche in den Zustand L_{T2} . Die Auswirkungen auf das physikalische Modell der inneren Leberstrukturen sind in M_1 und M_2 dargestellt.

Ein weiterer Navigationsansatz ist die Verwendung eines Atlas-Modells, siehe [Clements u. a., 2007]. Hierbei werden im Vorfeld eine Menge möglicher Verformungen des Modells simuliert und in einer Datenbank abgespeichert. Intraoperativ gewonnene Oberflächendaten werden mit diesen Datensätzen verglichen, wonach eine auf die aktuelle Organdeformation passende Modelldeformation ausgewählt wird. Nachteile bestehen hier einerseits im Aufwand beim Anlegen der Datenbank für die jeweilige Leber, zudem kann es vorkommen, dass nicht für jede Deformation eine geeignete Entsprechung vorliegt. Die Verwendung einer falschen Deformation kann jedoch die Validität der Navigation gefährden.

3.2.3 Tracking

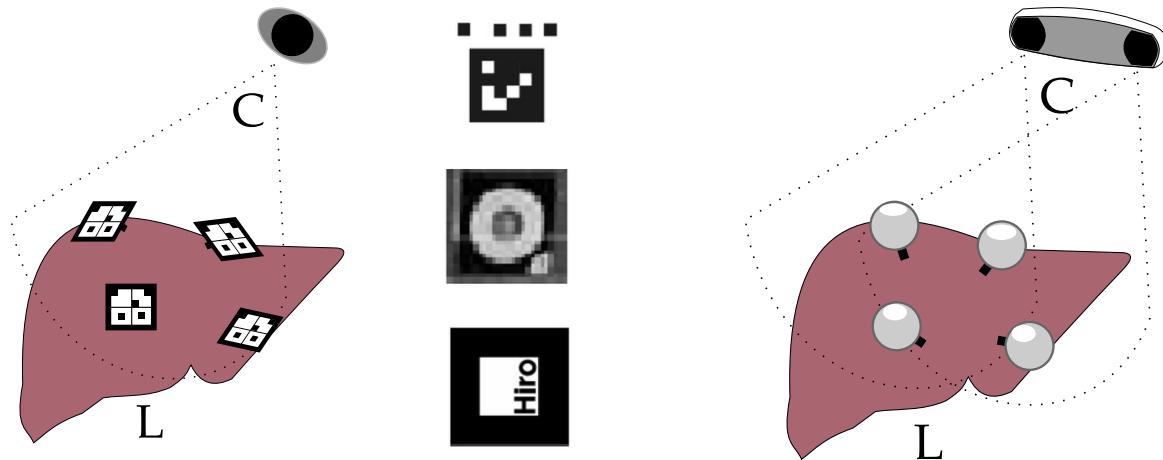
Ein vordergründiges Ziel des Trackings im Rahmen der CL besteht in der Bestimmung affiner Transformationen der Leber, also ihrer Position und Ausrichtung im Raum. Beim visuellen Tracking von Markern, d. h. markanten anatomischen oder aber manuell erzeugten Merkmalen, gilt es diese in einer Bildaufnahme zu detektieren und auf dieser Grundlage die benötigten, räumlichen Objektinformationen zu errechnen.

Planares Markertracking Eine Möglichkeit zur Erstellung von Markern ist der Druck von Mustern, die aus möglichst vielen Blickrichtungen gut erkennbar sind. Beispiele für Marker, die in aktuellen Softwarepaketen breite Verwendung finden, sind in Abbildung 3.4 (a) aufgeführt. Um eine gute Erkennung in einer Bildaufnahme zu gewährleisten, sollte der Kontrast zwischen einem Marker und dem umliegenden Raum möglichst groß sein. Ein Algorithmus muss den Marker zudem eindeutig identifizieren können. Hierbei sind üblicherweise mindestens vier eindeutige Punkte innerhalb des Markers zu detektieren und gegen eine Datenbank, die perspektivisch transformierte Versionen des Markers enthält, zu testen.

Durch Aufbringung dieser oder ähnlicher Marker auf die Oberfläche der Leber würde eine nicht-affine Verfolgung ihrer Oberfläche ermöglicht werden. Vorteile dieser Trackingmethode sind einerseits die geringen Kosten. Nachteile entstehen hingegen, wenn Teile der Oberfläche von Markern verdeckt sind, oder aber eine zu große Schräglage vorliegt. In beiden Fällen führt dies zu einer verschlechterten Erkennungsrate von planaren Markern. Eine zu schnelle Bewegung des Markers und ein stark unstrukturierter Hintergrund können sich zudem negativ auf die Erkennung auswirken.

Infrarottracking Eine Alternative zum Tracken von planaren Mustern besteht im Tracken von unter Infrarotlicht stark reflektierenden Kugeln. Hierbei wird in der Regel mit zwei Infrarotkameras gearbeitet, welche die Position von detektierten Markern per Triangulierung errechnen.

Bei der Verwendung von mehr als zwei reflektierenden Kugeln, wie in Abbildung 3.4 (b) dargestellt, können Position und Orientierung des Organs sehr genau bestimmt werden.



(a) Schematische Verwendung von planaren Markern und einer monokularen Kamera C in der CL (links), gebräuchliche planare Marker (rechts)

(b) Schematische Verwendung von Tracking per Stereo-Infrarotkamera C in der CL

Abbildung 3.4: Markerbasiertes Tracking in der comptergestützten Leberchirurgie.

3.2.4 Visualisierung

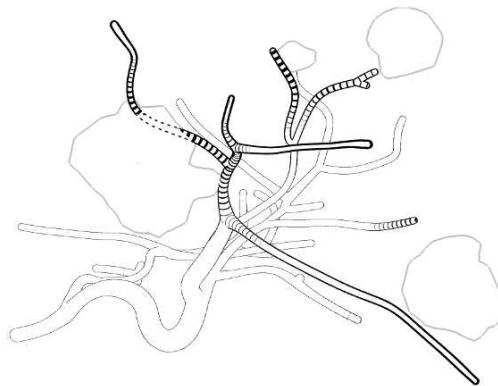
In der CL können Visualisierungen in jeder Phase der Behandlung zum Einsatz kommen. Neben klassischer Visualisierung per Monitor werden vermehrt Methoden aus dem Bereich der Virtual-Reality sowie Augmented-Reality erforscht.

Präoperativ werden Visualisierungsmethoden beispielsweise im Rahmen der Eingriffsplanung verwendet. Hierbei werden segmentierte Strukturen der Leber visuell voneinander abgehoben oder Operationsverläufe simuliert. Intraoperativ dient eine Visualisierung in erster Linie dazu, die Ansicht von Instrumenten und Leberdatensätzen zu fusionieren. Ein großer Vorteil ist hierbei, dass trotz einer vorliegenden Sichtverdeckung des Instruments, anhand der Visualisierung eine Navigation ermöglicht werden kann. Je nach Anforderung eines Eingriffs kann eine Visualisierung durch Methoden der Bildverarbeitung und Computergrafik angepasst werden. Je nach aktueller Anforderung werden Strukturen im Lebermodell visuell gekennzeichnet oder aber irrelevante Bereiche ausgeblendet.

Virtual Reality Die Visualisierung im Rahmen von Virtual-Reality-Anwendungen bietet die Wahrnehmung der Wirklichkeit und deren physikalische Eigenschaften in einer computergenerierten Umgebung. Entscheidende Faktoren sind hierbei die Echtzeitfähigkeit und die

Interaktionsmöglichkeiten innerhalb dieser künstlichen Realität. Für die Weichteilchirurgie wurden bereits zahlreiche derartige Systeme entworfen und getestet, siehe beispielsweise [Suzuki u. Hattori, 2008]. In erster Linie werden sie zum Operationstraining eingesetzt, im Rahmen der CL dienen sie beispielsweise dazu, eine vorher geplante Resektion unter realitätsnahen Bedingungen und unter Benutzung getrackter Werkzeuge zu simulieren.

Augmented Reality Die bereits vorgestellten, projektiven Methoden, verfolgen einen Ansatz, bei dem durch direkte Lichtprojektion wichtige Organstrukturen auf die Leberoberfläche projiziert werden. Bei der Planung und im frühen Stadium eines Eingriffs kann somit eine Navigationsunterstützung geboten werden. Um Organstrukturen unterhalb der Leberoberfläche lokalisierbar zu machen, benutzt [Kahrs u. a., 2006] eine Projektion, die abhängig von der Betrachterposition transformiert wird. Im Zweifelsfall kann eine projektive Methode auch ohne Instrumentenregistrierung auskommen, da sich der behandelnde Chirurg stets an der gebotenen Visualisierung orientieren kann. Erste Forschung im Bereich der Augmented Reality wurden in Form einer Aufprojektion von Gefäßbäumen am Fraunhofer MEVIS bereits betrieben. Mittels der von [Hansen, 2006] vorgestellten Streifenkodierung wird dem Chirurg ein erster Anhaltspunkt zu Lage und Tiefe von Gefäßen und anderen relevanten Organstrukturen vermittelt.



(a) Modell des Leber-Gefäßbaums mit Verwendung von Textur zur Tiefenkodierung, Bildquelle: Fraunhofer MEVIS



(b) Direkte Lichtprojektion des links gezeigten Gefäßbaum-Modells auf eine Schweineleber, Bildquelle: Fraunhofer MEVIS

Abbildung 3.5: Texturiertes Gefäßbaummodell und Einsatz im Rahmen direkter Lichtprojektion.

3.3 Problemstellung und Nutzen einer Kalibrierung

Die Ermittlung von intrinsischen Parametern, welche die Projektion bestimmen, sowie der extrinsischen Parameter, also Standort und Blickrichtung des bildgebenden bzw. projizierenden Gerätes, heißt Kalibrierung.

Zur Modellierung der Bildaufnahme bzw. Projektion wird dazu meist ein Lochkameramodell, siehe Abbildung 3.6, welches eine Vereinfachung der physikalischen Wirklichkeit darstellt, angenommen. Anstatt eines realen Objektives wird ein einzelner Punkt als Projektionszentrum modelliert. Zu den betrachteten intrinsischen Parametern einer Kamera gehört ihre Brennweite, also der Abstand des Projektionszentrums von der Bildebene, der Punkt des optischen Kamerazentrums und die Distorsion sowie Abmessung eines Kamerapixels.

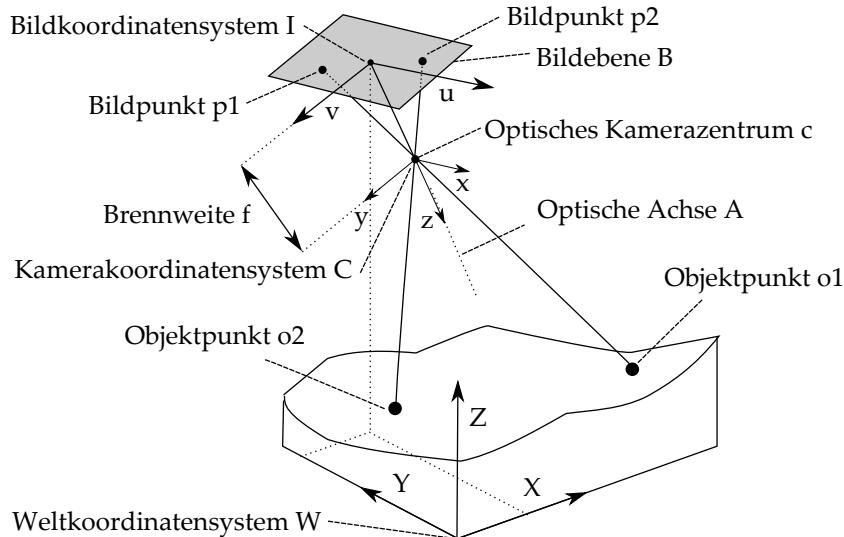


Abbildung 3.6: Das allgemeine Lochkameramodell. Licht wird von Objektpunkten durch das Kamerazentrum auf den dahinter liegenden Bildebener abgebildet.

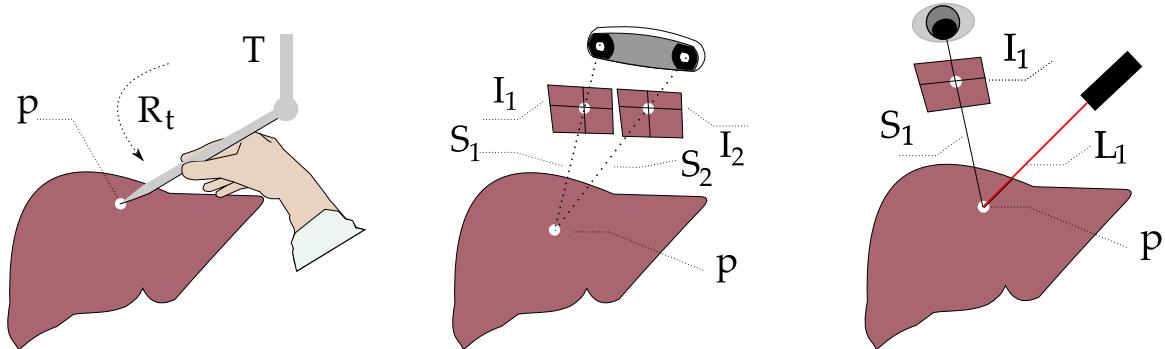
Im Fall eines Lochkameramodells reduziert sich die Kalibrierung auf die Bestimmung einer 3D–2D Transformation von Objektpunkten im Weltkoordinatensystem auf Bildpunkte der zweidimensionalen Bildebene. Ist die Intrinsik einer Kamera bekannt, so können aufgenommene Bilder dermaßen transformiert werden, dass sie denen einer idealen Kamera entsprechen, d. h. sie werden geometrisch entzerrt. Dies beugt in erster Linie der Verfälschung von Messdaten vor und ist für die zu entwickelnde Oberflächenrekonstruktion demnach essentiell. Die Bestimmung externer Geräteparameter mittels Kalibrierung ist zudem die Hauptgrundlage der durchgeführten Triangulierung. Position und Blickrichtung von Kamera und Projektor werden dabei benötigt, um die Raumposition der Schnittpunkte von Sehstrahlen zu errechnen. Ist die Kalibrierung ungenau, so wirkt sich dies negativ auf die Rekonstruktion aus.

3.4 Methoden zur Oberflächenrekonstruktion

Zur Oberflächen- oder allgemein dreidimensionalen Rekonstruktion sind bereits eine Reihe von Verfahren entwickelt worden. Hierzu zählen zum Beispiel kontaktbasierte Verfahren wie die in

Abbildung 3.7 (a) dargestellte mechanische Sondierung. Die Raumposition des Werkzeugkopfes ist hierbei stets bekannt, weswegen Oberflächenpositionen manuell abgetastet und gespeichert werden können. Diese Methode ist zwar sehr akkurat, aber auch zeitintensiv. Weil Berührungen der Leberoberfläche dabei erforderlich sind, diese jedoch dadurch verformt werden kann, eignet sich diese Form der Oberflächenrekonstruktion nicht für den Einsatz in der CL.

Der Hauptanteil der Verfahren zur dreidimensionalen Rekonstruktion ist optisch gestützt und kontaktlos. Bei optischen Verfahren wird zwischen aktiven und passiven Ansätzen unterschieden. Passive Verfahren, wie beispielsweise das in Abbildung 3.7 (b) gezeigte Stereosehen, arbeiten auf Basis einer statischen Szene und wirken sich in keiner Weise physisch auf diese aus. Aktive Verfahren hingegen alternieren die Szene optisch, beispielsweise durch Laser- oder Lichtprojektion, um hiermit eine Triangulierung von Tiefenpunkten zu ermöglichen.



(a) Mechanische Sondierung. Die affine Transformation R_t des Werkzeugs T mechanisch gemessen, womit es möglich wird Objektpunkte p aufzuzeichnen.

(b) Kontakloses Stereosehen. Durch Korrespondenzfindung ermittelte Bildpunkte I_1 und I_2 (in weiß) ermöglichen die Berechnung des Oberflächenpunktes p als Schnittpunkte der Sehstrahlen S_1 und S_2 .

(c) Oberflächenrekonstruktion mittels Laserprojektion. Der im Bild I_1 detektierte Laserpunkt ermöglicht die Berechnung des Oberflächenpunktes p als Schnittpunkt des Sehstrahls S_1 mit dem Laserstrahl L_1 .

Abbildung 3.7: Vergleich kontaktbasierter und kontaktloser Verfahren zur Oberflächenrekonstruktion.

Bei der Rekonstruktion mittels Laser ist der ausgesendete Strahl als Richtungsvektor beschreibbar, falls der Laser kalibriert ist, also Standpunkt und Blickrichtung bekannt sind. Ist die Kamera kalibriert, so lässt sich wiederum ein Sehstrahl vom Kamerazentrum bis auf den Oberflächenpunkt, der vom Laser beleuchtet wird, ermitteln. Dazu muss vorher jedoch die Korrespondenz des auf dem Objekt befindlichen Laserpunktes auf der Bildebene der Kamera gefunden werden.

4 Kalibrierung des Projektor-Kamera-Systems

In diesem Kapitel wird der Ablauf, der in den Grundlagen vorgestellten Kalibrierung mathematisch veranschaulicht. Zur Lösung des Kalibrierungsproblems wird die von [Tsai, 1986] entwickelte Methode eingesetzt. Hiermit werden Intrinsik sowie Extrinsik beider Geräte ermittelt.

Problemstellung Die Schwierigkeit der Kalibrierung eines Projektor-Kamera-Systems liegt einerseits in der Tatsache begründet, dass die Projektion nur mittels einer Kamera beobachtet werden kann und beleuchtete Raumpunkte nicht vom Projektor selbst gemessen werden können. Intrinsische Parameter des Projektors müssen folglich durch eine Kamera ermittelt werden. Aus der Beobachtung der Projektion und Schätzung der für diese Projektion verantwortlichen intrinsischen Parameter können wiederum extrinsische Parameter wie Standpunkt und Orientierung des Projektors zurück gerechnet werden, was als inverse Projektion bezeichnet wird.

Ein zentraler Aspekt bei der Kalibrierung des Projektor-Kamera-Systems ist die konzeptionelle Betrachtung beider Geräte als Kamera. Ein Projektor wird als inverse Kamera angesehen, da Bilder nicht aufgenommen sondern projiziert werden. Während eine Kamera 3D-Weltpunkte auf einen photosensitiven 2D-Bildträger abbildet, beleuchtet ein Projektor mittels 2D-Bildträger eine Menge von Punkten im dreidimensionalen Raum. So beleuchtete Punkte können im 2D-Kamerabild beobachtet und auf die vom Projektor ausgehenden Lichtstrahlen im dreidimensionalen Raum zurück geführt werden. Dieser Umstand ermöglicht es wiederum, die Kalibrierung des Projektors mit der einer Kamera gleich zu setzen.

In Abbildung 4.1 ist der schematische Geräteaufbau zur Kalibrierung des Projektor-Kamera-Systems dargestellt. Neben den beiden Geräten wird eine gemeinsame Kalibrierebene benötigt, die mit einem festen Kalibriermuster versehen ist und ausreichend Platz für die Aufprojektion eines zweiten Musters bietet. Im Folgenden wird beschrieben, wie die Intrinsik der Geräte sowie die Extrinsik des Projektors ermittelt werden.

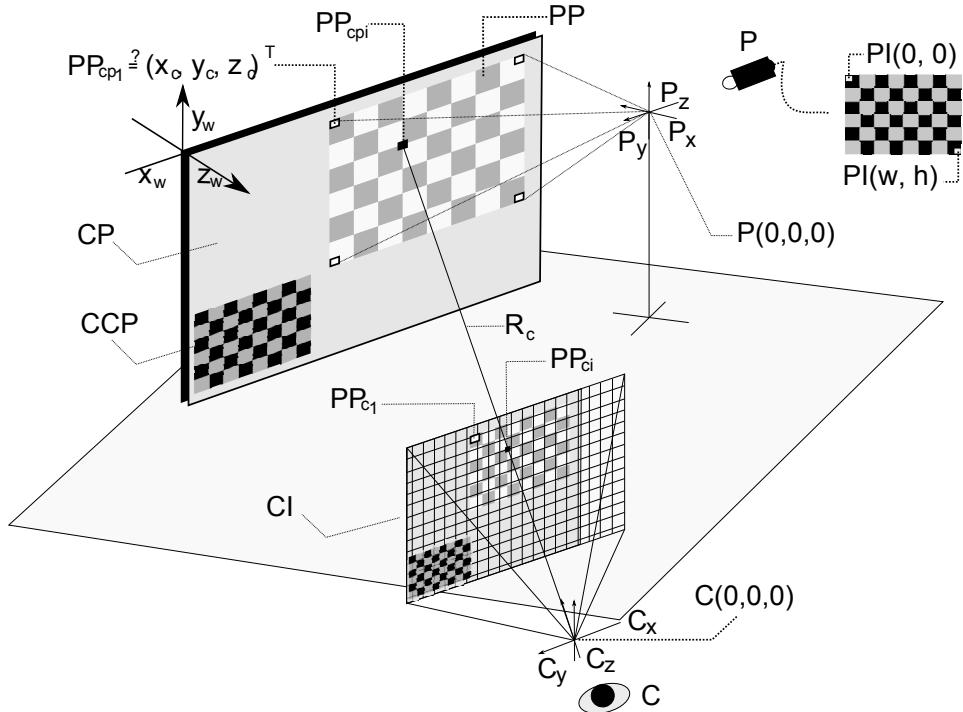
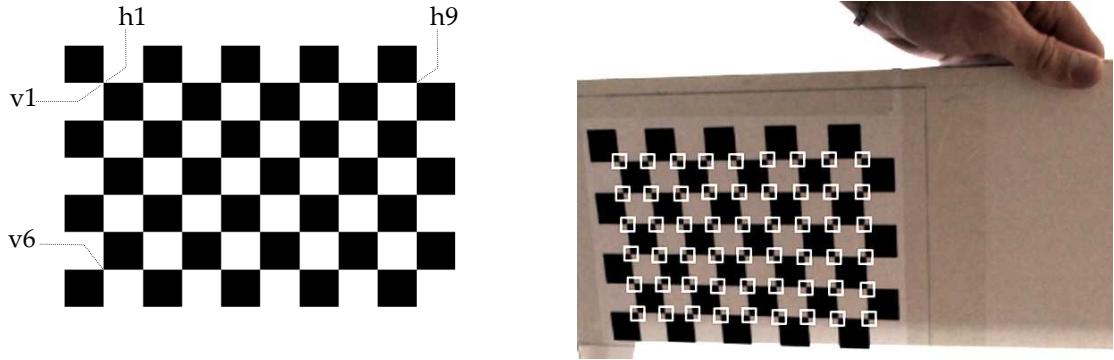


Abbildung 4.1: Schematischer Aufbau zur Projektor-Kamera-Kalibrierung. Kamera und Projektor besitzen jeweils ihre eigenen Koordinatensysteme (C_x, C_y, C_z) , und (P_x, P_y, P_z) . Die gemeinsame Kalibrierebene CP beinhaltet ein fest aufgebrachtes Kamerakalibermuster CCP . Mittels Projektor wird ein zweites Muster als Bild PI auf CP projiziert. Der Abbildung dieses Musters auf der Kalibrierebene CP entspricht PP . Um die Kalibrierung des Projektors vorzunehmen gilt es u.A. die räumliche Entsprechung PP_{cp1} des Bildpunktes $PI(0, 0)$ im Weltkoordinatensystem $W = (X_w, Y_w, Z_w)$, zu bestimmen.

4.1 Ermittlung der Kameraintrinsik

Die im weiteren Verlauf verwendeten Bezeichner beziehen sich auf Abbildung 4.1. Vor Beginn der Kalibrierung sollte das zur Kamerakalibrierung benötigte Muster CCP , vgl. auch Abbildung 4.2 (a), auf einer planaren Fläche CP , wie einer Spanplatte oder Leinwand befestigt worden sein. Auf der Fläche muss zudem genügend Platz für die Aufprojektion eines weiteren Musters PP durch den fokussierten Projektor P geboten werden. Der Ursprung des Weltkoordinatensystems sollte zu diesem Zeitpunkt festgelegt werden, [Falcao u. a., 2008] schlagen vor, die linke obere Ecke der Kalibrierebene als Weltursprung zu betrachten. Somit liegen X- und Y-Achse innerhalb der Ebene CP und die Kamera blickt in Richtung der negativen Z-Achse.

Mittels OpenCV wird die Kamera wird nun nach der von [Tsai, 1986] eingeführten und [Zhang, 2000] weitergeführten Methode intrinsisch kalibriert, was durch die formatfüllende Aufnahme mehrerer Ansichten des planaren Kalibermusters CCP durchgeführt wird. Die extrinsischen Parameter der Kamera sind im Verlauf der intrinsischen Kalibrierung mittels Tsais' Methode dabei stets unverändert. Unter Benutzung einer Kantendetektion werden nun die Eckpunkte



(a) Bildvorlage des Kamerakalibrierungsmusters mit horizontalen Eckpunkten $h_1 \dots h_9$ und vertikalen Eckpunkten $v_1 \dots v_6$.

(b) In Weiß eingefasste, detektierte Eckpunkte des Kalibriermusters in einer Aufnahme der Kamera.

Abbildung 4.2: Kamerakalibermuster und Eckpunkt detektion.

des Kalibermusters im Kamerabild ermittelt (4.2 (b)), Durch Ermittlung einer projektiven Transformation von den statischen Eckpunkten des Kamerakalibermusters auf diese Eckpunkte, lässt sich die Extrinsik der Kalibrierebene pro Aufnahme darstellen. Die intrinsischen Parameter der Kamera (Kc_{int}) werden durch eine geschlossene Lösung analytisch errechnet, zusätzlich wird die radiale Bildverzerrung vierten Grades per Kleinste-Quadrate-Methode ermittelt. Der entstandene Reprojektionsfehler wird mit Hilfe der Levenberg-Marquardt-Methode nichtlinear minimiert, was zur weiteren Annäherung an die tatsächlichen Kameraparameter beiträgt. Zu den mathematischen Details siehe [Tsai, 1986] und [Bradski u. Kaehler, 2008].

4.2 Intrinsische Kalibrierung des Projektors

Die Kamera ist nun intrinsisch kalibriert. Im nächsten Schritt gilt es die Intrinisk des Projektors zu ermitteln. Hierzu wird neben das fest aufgebrachte Muster zur Kamerakalibrierung, das Muster PI projiziert, worauf das Kameraaufnahmen beide Muster, CCP und PP enthalten.

Bestimmung von Ebenenparametern Die Parameter der im Kamerabild sichtbaren Ebene CP wurden während der extrinsischen Kamerakalibrierung ermittelt. Aus der so rekonstruierten Rotation r und Translation t lässt sich eine kombinierte 4×4 Transformationsmatrix der Extrinsik von CP aufstellen.

$$CP_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Mittels CP_{ext} wird es möglich, die 3D-Raumpositionen der Eckpunkte des projizierten Musters PP anhand geometrischem Strahl-Ebene-Schnitt zu berechnen. Vom Ursprung des Kamerakoordinatensystems, also dem optischen Kamerazentrum c aus, werden nun Sehstrahlen durch jeden dieser Eckpunkte im Kamerabild berechnet und mit der Kalibrierebene CP geschnitten, vgl. R_c in Abbildung 4.1.

Ebenennotation Um Ebenenschnittpunkte zu berechnen wird CP in der Normalenform dargestellt, wofür ein Ebenenaufpunkt p sowie eine Normale n hinreichend sind. Als Ebenenaufpunkt wird der Translationsvektor t der kombinierten Matrix CP_{ext} verwendet, die Z-Achse der Fläche CP_{ext} entspricht ihrer Normale n und ist somit die dritte Spalte ihrer Rotation r .

$$p = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \quad (4.2)$$

$$n = \begin{bmatrix} r_{13} & r_{23} & r_{33} \end{bmatrix}^T \quad (4.3)$$

Die Ebene CP besteht aus einer Menge von darin liegenden Punkten r , und lässt sich nun mittels Normale n und Ebenenpunkt p in der Hesse-Normalenform notieren. Liegt ein Punkt p in der Ebene CP , so liegt ebenfalls der Positionsvektor $\overrightarrow{(r - p)}$ innerhalb der Ebene und Orthogonalität von $\overrightarrow{(r - p)}$ zur Flächennormale n liegt vor, sprich es gilt folgendes Skalarprodukt:

$$n \cdot \overrightarrow{(r - p)} = 0 \quad (4.4)$$

Es wird weiter definiert:

$$\hat{x} = (1, 0, 0)^T \quad (4.5)$$

$$\hat{y} = (0, 1, 0)^T \quad (4.6)$$

$$\hat{z} = (0, 0, 1)^T \quad (4.7)$$

$$n = a\hat{x} + b\hat{y} + c\hat{z} \quad (4.8)$$

$$d = -n \circ p \quad (4.9)$$

$$r = x\hat{x} + y\hat{y} + z\hat{z} \quad (4.10)$$

Hierbei sind \hat{x} , \hat{y} und \hat{z} die kartesischen Einheitsvektoren, $n = (a, b, c)^T$ die Flächennormale, d der Abstand der Ebene zum Ursprung und r der Punkt, der auf seine Lage in der Ebene zu

testen ist. Die Ebene selbst ist somit durch die in 4.11 aufgestellte Bedingung beschreibbar:

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \quad (4.11)$$

$$a, b, c \in \mathbb{R} \quad (4.12)$$

$$a, b, c \neq 0 \quad (4.13)$$

Sehstrahl-Ebenenschnitt Nun folgt die Aufgabe Sehstrahlen zu berechnen, die durch die im 2D-Bild detektierten Eckpunkte, z.B. PP_{ci} des projizierten Kalibermusters verlaufen und die in Abb.4.1 dargestellten Eckpunkte, z.B. PP_{cpi} , auf der Ebene CP schneiden, wie es beispielsweise der Sehstrahl R_c tut. Durch die anfangs ermittelten Kameraparameter $Kc = Kc_{int} \cdot Kc_{ext}$, mit $Kc \in \mathbb{R}^{3 \times 3}$ kann eine projektive Transformation durchgeführt werden, die für einen Eckpunkt PP_{ci} einen vom Kamerazentrum ausgehenden Sehstrahl R liefert. Den Skalierungsfaktor s gilt es dabei so zu bestimmen, dass der damit skalierte Sehstrahl R' die Ebene CP berührt.

$$\begin{bmatrix} R'_x \\ R'_y \\ R'_z \end{bmatrix} = \begin{bmatrix} s \cdot R_x \\ s \cdot R_y \\ s \cdot R_z \end{bmatrix} \quad (4.14)$$

$$\begin{bmatrix} R'_x \\ R'_y \\ R'_z \end{bmatrix} = [Kc_{int} \cdot Kc_{ext}]^{-1} \cdot \begin{bmatrix} CI_x \\ CI_y \\ 1 \end{bmatrix} \quad (4.15)$$

In einem letzten Schritt werden nun die 3D-Positionen I der Schnittpunkte aller soeben erzeugten Sehstrahlen R mit der Kalibrierebene CP ermittelt. Der Skalierungsfaktor s eines Sehstrahls R muss dazu derart gefunden werden, dass die Ebenengleichung aus 4.11 erfüllt ist. Substitution jener Gleichung ergibt:

$$a(s \cdot R_x) + b(s \cdot R_y) + c(s \cdot R_z) + d = 0 \quad (4.16)$$

$$s = \frac{-d}{a \cdot R_x + b \cdot R_y + c \cdot R_z} \quad (4.17)$$

Die Parameter a, b und c bezeichnen die Flächennormale n von CP und d den Abstand der Fläche vom Ursprung $C(0, 0, 0)$. Der Ebenenschnittpunkt I eines Sehstrahls R mit der Ebene CP berechnet sich also mittels Skalierungsfaktor s als:

$$I_{(R,CP)} = \begin{bmatrix} s \cdot R_x & s \cdot R_y & s \cdot R_z \end{bmatrix}^T = \begin{bmatrix} R'_x & R'_y & R'_z \end{bmatrix}^T \quad (4.18)$$

Für die Eckpunkte aller Aufnahmen des projizierten Kalibriermusters werden nun die Schnittpunkte I bestimmt, wodurch eine große Menge an Punkten entsteht, welche in verschiedenen, pro Bild spezifischen Ebenen des 3D-Raums liegen. Die Bestimmung der Projektiorintrinsik ist nun analog zur Kameraintrinsik und wird durch die Eingabe aller Schnittpunkte I in den dazu verwendeten Algorithmus nach [Tsai, 1986] vorgenommen.

4.3 Extrinsische Kalibrierung

Alle folgenden Bezeichner beziehen sich auf Abbildung 4.3. Ein Projektor wird, wie bereits erwähnt, als inverse Kamera angesehen, die einen zweidimensionalen Bildpunkt in die dreidimensionale Szene abbildet, sprich einen Szenenpunkt beleuchtet. Für eine Kamera gilt die umgekehrte Abbildungsrichtung vom beobachteten Raumpunkt auf das zweidimensionale Kamerabild, vergleiche hierzu R_c und R_p . Ist die Kameraintrinsik bekannt, so lässt sich für einen gegebenen Kamerabildpunkt ein Sehstrahl R (Gleichung 4.15) in die beobachtete Szene rekonstruieren.

Für die Abbildung des Ebenenschnittpunktes I (Gleichung 4.18) auf die Bildebene des Projektors gilt nun allgemein:

$$\begin{bmatrix} PI_u \\ PI_v \\ 1 \end{bmatrix} = [Kp_{int} \cdot Kp_{ext}] \cdot I_{(R,PP)} \quad (4.19)$$

Bekannt ist, dass der Projektor stets ein reguläres Gitter von Eckpunkten projiziert, weswegen die Sollpositionen aller Eckpunkte in der Bildebene PI des Projektors bekannt sind. Da zudem die intrinsische Projektormatrix Kp_{int} ermittelt wurde, muss schlussendlich eine extrinsische Transformationsmatrix Kp_{ext} gefunden werden, die Gleichung 4.19 für alle Eckpunkte in PI und korrespondierende Schnittpunkte I in PP erfüllt.

Hierzu kann die Homographieabbildung zwischen der gemeinsamen Kalibrierebene CCP , sowie der Ebene PP , welche vom projizierten Muster aufgespannt wird, ermittelt werden. Konzeptuell wird der Projektor also als eine zweite Kamera angesehen, welche jedoch CCP aus einem von der Kamera verschiedenen Blickwinkel aus beobachtet hat. Die Homographieabbildung von CCP zu PP liefert die Rotation R_p des Projektors P :

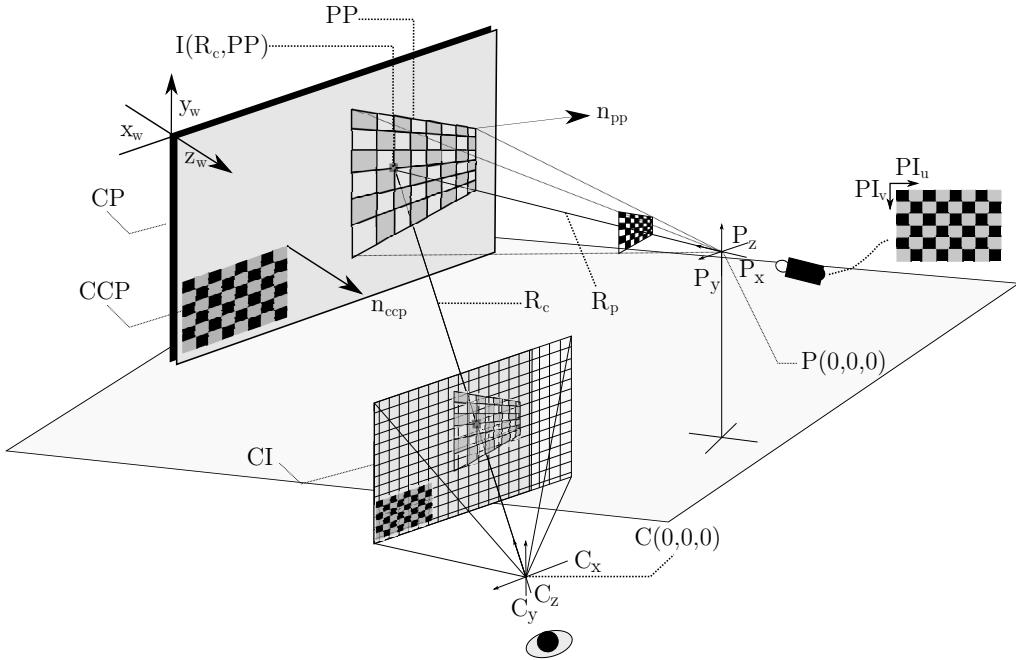


Abbildung 4.3: Schematischer Geräteraufbau zur Verdeutlichung der extrinsischen Kalibrierung. Gemeinsame Kalibrierebene CP , Kamerakalibermuster CCP mit dazugehöriger Normale n_{ccp} , sowie Projektionspattern-Ebene PP mit Normale n_{pp} und beispielhafter Ebenenschnittpunkt $I(R_c, PP)$ des Kamera-Sehstrahls R_c .

$$R_p = K p_{int} \cdot I_{(R_c, CP)} \quad (4.20)$$

Die Translation des Projektors ist gefunden, wenn der Schnittpunkt der inversen, ins Projektorkoordinatensystem überführten, Sehstrahlen R_{pi} gefunden ist. Alternativ lässt sich die Translation des Projektors auch folgendermaßen berechnen. Kamera sowie Projektor besitzen voneinander getrennte Koordinatensysteme. Die Vektoren zum gemeinsam beobachteten Weltpunkt $I_{(R_c, PP)}$ sind R_c für die Kamera und R_p für den Projektor, welche sich in ihren jeweiligen Koordinatensystemen bestimmen lassen. Nun gilt es die Projektorposition P_{pos} , welche in Projektorkoordinaten vorliegt, in die Projektorposition P_{pos}^c in Kamerakoordinaten zu transformieren.

Sei H die Homographie von CCP zu PP so gilt:

$$P_{pos}^c = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}^c = R_c - H \cdot R_p \quad (4.21)$$

5 Oberflächenrekonstruktion mittels Structured-Light

Alle Structured-Light-Verfahren arbeiten kontaktlos und bildbasiert, um durch aktive Lichtprojektion eine Rekonstruktion von dreidimensionalen Informationen durchzuführen. Auf in einer Szene enthaltene Objekte wird ein meist reguläres, wie der Name des Verfahrens andeutet, strukturiertes Bildmuster bzw. eine Abfolge von Mustern projiziert. Dies dient dem Ziel durch Beobachtung des Intensitätswechsels an einer Position im Kamerabild auf Tiefenwerte der Szene zu schließen. Eine Methode ist es beispielsweise, miteinander korrespondierende Sehstrahlen von Kamera und Projektor zu bestimmen und miteinander zu schneiden.

Die Vorteile von Structured-Light-Verfahren liegen vor allem in der Geschwindigkeit der Rekonstruktion. Die gesamte sichtbare Szene wird in einem automatisch ablaufenden Schritt rekonstruiert, zudem ist nach [Akca u. a., 2007] die Genauigkeit bereits mit moderater Hardwareausstattung sehr zufriedenstellend. Je nach verwendetem Verfahren schwankt die Menge der zur Codierung aller Projektorpixel notwendigen Bilder. Im Vergleich zum Sinusphasen-Verfahren, welches lediglich drei Projektionen benötigt, wird beim Gray-Code-Verfahren, je nach Projektorauflösung, ein Vielfaches benötigt. Um dem Nachteil der langsamen Rekonstruktion beizukommen, beobachten [Narasimhan u. a., 2008] die temporären Lichtschwankungen moderner DLP-Projektoren mit Hilfe einer Hochgeschwindigkeitskamera. Hierdurch wird die Anzahl der pro Sekunde rekonstruierbaren Einzelbilder erheblich gesteigert. Des Weiteren sind in der Forschung Methoden entwickelt worden, die auf nicht sichtbarem Infrarotlicht basieren, bzw. ultrahochfrequente Projektionstechnik verwenden. In der Arbeit von [Lee u. a., 2005] wird eine Infrarotkamera eingesetzt um die Beeinflussung des vom Menschen sichtbaren Lichtspektrums zu vermeiden.

Es werden mit dem Gray-Code- sowie Sinusphasen-Verfahren nun zwei unterschiedliche Structured-Light-Ansätze näher betrachtet. Der zentrale Unterschied beider Verfahren besteht darin, dass die Kalibrierung des Projektor-Kamera-Systems lediglich zur Rekonstruktion mittels Gray-Codes notwendig ist, da hierbei eine geometrische Triangulierung durchgeführt wird.

5.1 Geometrische Triangulierung und das Gray-Code-Verfahren

Die Rekonstruktion von Tiefenwerten einer Szene wird im Rahmen von Structured-Light-Verfahren u.A. durch die in Abbildung 5.1 veranschaulichte Triangulierung verwirklicht. Zu deren Ermöglichung muss hierbei jedoch zuerst die Lösung eines allgemeinen Korrespondenzproblems erfolgen.

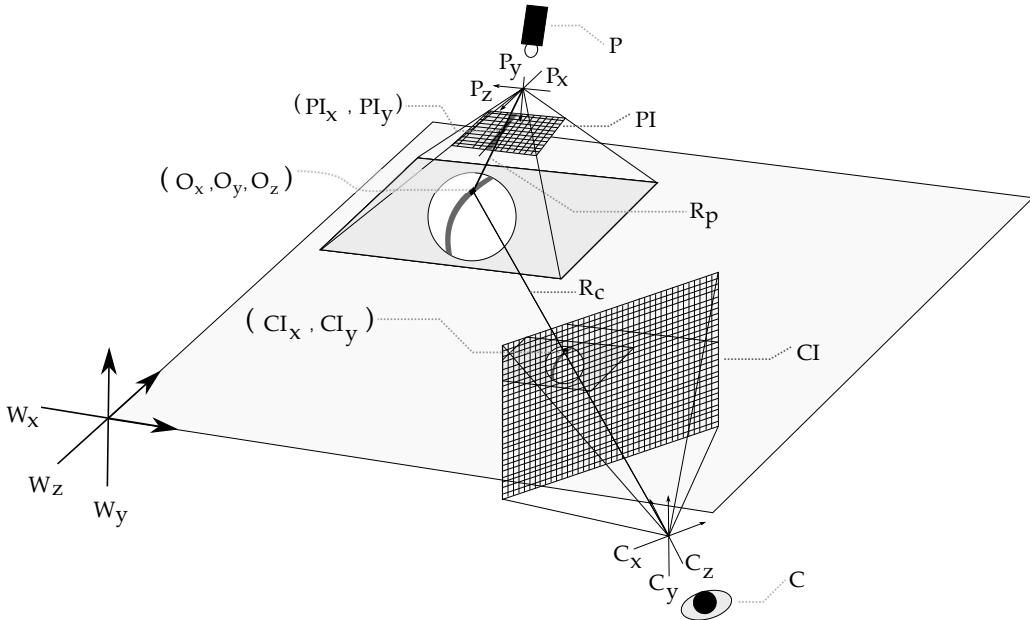


Abbildung 5.1: Problemstellung geometrischer Triangulierung. Bekannt ist der im Kamerabild CI beobachtete Pixel (CI_x, CI_y) , gesucht ist der Objektpunkt (O_x, O_y, O_z) . Dieses Korrespondenzproblem zu lösen bedeutet die Unbekannte (PI_x, PI_y) als den für die Projektion verantwortlichen Pixel des Projektionsmusters PI zu ermitteln. Geometrische Triangulierung zwischen Kamerastrahl R_c und Projektorstrahl R_p liefert daraufhin den gesuchten Objektpunkt O .

Das Rekonstruktionsproblem reduziert sich also auf die Herstellung einer Korrespondenz zwischen beobachtetem Bildpixel mit einem Pixel im projizierten Muster. Beim dieser Problemstellung geht es im Kern um die Frage, was einerseits die Kamera an einer bestimmten Pixelposition beobachtet hat, z.B. einen bestimmten Binärcode durch wechselnde Beleuchtung und Abdunklung eines Raumpunktes. Andererseits gilt es zu ermitteln welche Pixelposition des projizierten Musters auf diesen Raumpunkt projiziert wurde. Sind Kamera und Projektor, wie in Kapitel 4 beschrieben, kalibriert worden, so kann mit der Rekonstruktion des zu einem Kamerabildpunkt gehörigen Raumpunktes begonnen werden. Die Kalibrierung ist hierfür grundlegend, da sie durch die Ermittlung von intrinsischen Parametern die Herleitung eines Strahls pro Bildpunkt ermöglicht. Zugleich sind durch die extrinsische Kalibrierung von Kamera und Projektor, Position und Orientierung ihrer Koordinatensysteme innerhalb eines

gemeinsamen Weltkoordinatensystems bekannt.

Die Lösung des Korrespondenzproblems und der damit ermöglichten Triangulierung im 3D-Raum, wird durch die Projektion von geeignet kodierten Lichtmustern erzielt. Würde es dagegen gelten, mittels einer Folge von unkodierten Bildern das Korrespondenzproblem zu lösen, so wäre eine sehr große Zahl von Projektionen notwendig, wie das folgende Beispiel verdeutlicht. Es wird hierbei von einem 1024 Pixel breiten und 768 Pixel hohen Projektormuster ausgegangen. Entweder kann jede Pixelposition einzeln eingefärbt und die Projektion im Bild gesucht werden, was mit 786432 nötigen Projektionen jedoch überaus zeitintensiv wäre. Es könnten hingegen auch Zeilen und Spalten des projizierten Bildes nacheinander projiziert werden. Wir dazu nacheinander jede Projektorspalte und Zeile projiziert und wird dabei eine bestimmter Kamerapixel beobachtet, so erscheint dieser zwei mal abgedunkelt. Das erste Mal von einer Projektorzeile und das zweite Mal von einer Projektorspalte. Somit ist zwar die Korrespondenz aller Kamerapixel zu Projektorpixeln gefunden, der Zeitaufwand ist aber auch hierbei nicht vertretbar, da immer noch 1792 aufeinander folgende Projektionen benutzt werden müssten. Der Einsatz einer effizienteren Kodierung wird also zwingend notwendig.

[Salvi u. a., 2004] unterscheiden hierbei zwischen zwei Arten der Kodierung. Wird lediglich ein einzelnes Bild projiziert, so spricht man von rein spatialer Kodierung. Hierbei muss dafür Sorge getragen werden, dass jeder projizierte Bildpunkt eindeutig identifiziert werden kann. Durch die laufende Projektion eines statischen Musters, wie beispielsweise eines Farbverlaufs, wären dabei dynamische Szenen rekonstruierbar. Die Unterscheidung von sehr ähnlichen Farbtönen durch eine Kamera, ist technisch gesehen jedoch schwierig und führt zur Bildung von Rekonstruktionsartefakten. Robuster gegen Artefakte sind hingegen temporale Kodierungen, welche durch die Verwendung einer Abfolge von Mustern, Redundanz bieten und die Rekonstruktionsergebnisse allgemein verbessern. Mit den Gray-Codes wird nun eine solche temporale Kodierung vorgestellt.

Das Gray-Code-Verfahren Die Kodierung von Structured-Light-Sequenzen mittels Gray-Codes ist eine Weiterentwicklung der durch [Posdamer u. Altschuler, 1982] eingeführten, temporalen Kodierung mittels Binärcodes. Im Gegensatz zu rein spatialer Kodierung mittels eines einzigen Musters, sind temporale Kodierungen robuster gegen Bildrauschen oder aber Oberflächentexturierung, da jedes Bildpixel mit einem n-bit langen Wort kodiert werden kann. Die Einführung von Gray-Codes fußt auf der Kommunikationstheorie, in der eine Informationskodierung gesucht wird, die möglichst robust gegenüber dem Rauschen des Übertragungskanals ist. Der Übertragungskanal verläuft hierbei zwischen Projektor und Kamera, wobei das Licht als Trägermedium genutzt wird. Bei Projektion sowie Aufnahme dieses Lichtstroms können diverse Störfaktoren, wie unstete Lichtverhältnisse oder die Komprimierung des aufgenommenen Bildes die Übertragung des ursprünglichen Codewortes verfälschen. Gray-Codes wurden

erstmals von [Inokuchi u. a., 1984] für den Einsatz in Structured-Light-Systemen vorgeschlagen. Ihre Benutzung hat im Gegensatz zu Binärcodes den in Abbildung 5.3 dargestellten Vorteil, dass benachbarte Codewörter sich nur um ein Bit unterscheiden, die Hamming-Distanz also eins beträgt. Der maximale Fehler beim Lesen des Codewortes durch den Empfänger, hier die beobachtende Kamera, kann als nur in einem Bit bestehen, was eine leichte Fehlerkorrektur ermöglicht. Ein Nachteil von Gray-Codes liegt in der Geschwindigkeit des Verfahrens. Für ein 1024 Pixel breites Projektionsmuster müssen bereits $\log_2(1024) \cdot 2 = 20$ unterschiedliche Muster projiziert werden um jede Projektorposition eindeutig zu kodieren. Wird auf ein Objekt im Raum beispielsweise die Bildfolge $p1 \dots p6$ aus Abbildung 5.2 projiziert und von einer Kamera an einer Pixelposition $c(x, y)$ als sechsstelliger Binärkode beobachtet, so ist die korrespondierenden Pixelposition $p(x, y)$ des Projektors mittels einer Binärbaumsuche eindeutig zu bestimmen. Jedes Bit des Codeworts wird durch das Vorhandensein oder aber die Abwesenheit der Beleuchtung zu jedem der sechs Zeitschritte gesetzt. In Abbildung 5.3 ist die Kodierung von Projektorzeilen des Projektorbildes mittels binärem Suchbaum schematisch dargestellt.

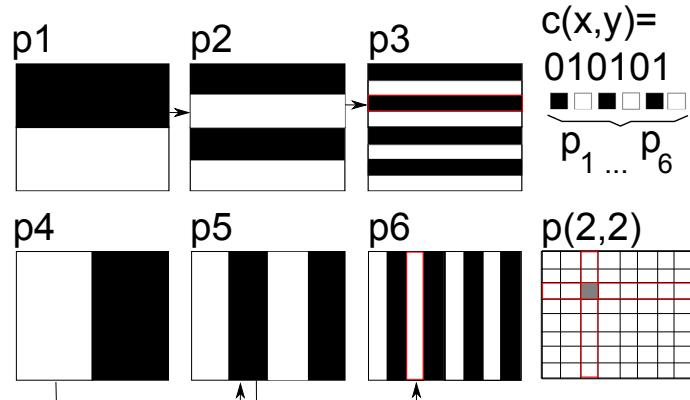


Abbildung 5.2: Identifikation der Position projizierter Pixel im Kamerabild mittels horizontalen ($p1 \dots p3$) und vertikalen ($p4 \dots p6$) Binärkode-Projektionsmustern. Wird für ein Kamerapixel $c(x, y)$ der Binärkode $c(x, y, t) = 010101$ beobachtet so identifiziert dieser den dazugehörigen Projektorpixel $p(2, 2)$ als Belichtungsquelle.

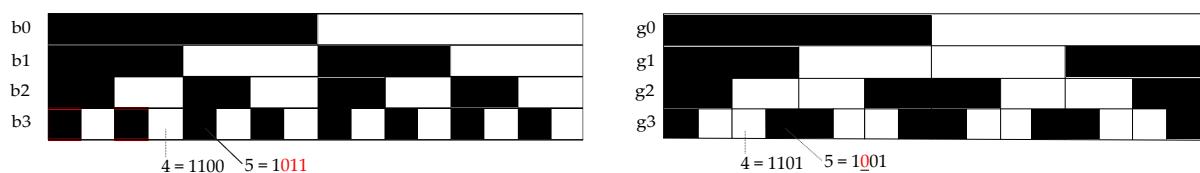


Abbildung 5.3: Schematische Kodierung von Projektorzeilen im Fall eines Binärcodes (links) und Gray-Codes (rechts). Ein 4-bit breites Codewort $b_0 b_1 b_2 b_3$ ermöglicht die Unterscheidung von 16 Bildzeilen. Wird für einen Kamerapixel die Folge $b_0 b_1 b_2 b_3 = 1100$ beobachtet, so liegt dieser Pixel in der vierten Projektorzeile. Die Gray-Codes für Zeile 4 und 5 unterscheiden sich lediglich in einem Bit (rot markiert), im Fall der Binärcodes bestehen bereits 3 Unterschiede.

5.2 Das Sinusphasen-Verfahren

Durch die Projektion von drei phasenversetzten Bildern einer Sinusschwingung, kann eine Tiefenrekonstruktion ohne Kalibrierung erfolgen, siehe Abbildung 5.4. Für jede Pixelposition im Bild wird der Wert aller drei Aufnahmen betrachtet und jeweils eine Phase als Grauwert in den Wertebereich $[0 \dots 1]$ gemappt. Die Phasensumme und der Phasenbereich werden ebenfalls für jede Pixelposition berechnet. Falls die Phasen sehr ähnlich sind werden sie ggf. maskiert, da es sich wahrscheinlich um Rauschen handelt. Die Gesamtphase p wird nun berechnet als:

$$p = \frac{\arctan(\sqrt{3} \cdot p1 - p3, 2 \cdot p2 - p1 - p3)}{2 \cdot \pi} \quad (5.1)$$

Es folgt das sogenannte Aufrollen des nun vorliegenden Phasenbildes, das Wiederholungen von 2π -unstetigen Verläufen enthält, in ein stetiges Phasenbild. Hierzu wird im Rahmen eines Flood-Filling-Algorithmus mit dem Pixel im Bildmittelpunkt begonnen und rekursiv die Nachbarpixel betrachtet, um die Phaseninformation nach außen hin zu propagieren. Der gesuchte Tiefenwert wird nun rekonstruiert als:

$$z = (p - (0.5 - (y - \frac{\frac{imageheight}{2.0}}{zskew}))) \cdot zscale \quad (5.2)$$

Hierbei ist p der aktuelle Pixelwert der aktuellen Bildzeile y . Weiter bezeichnet $skew$ die vom Nutzer zu wählende Schiefstellung der Oberflächenpunkte und $zscale$ die Tiefenskalierung. Für weitere mathematische Details siehe [Zhang u. Huang, 2006].

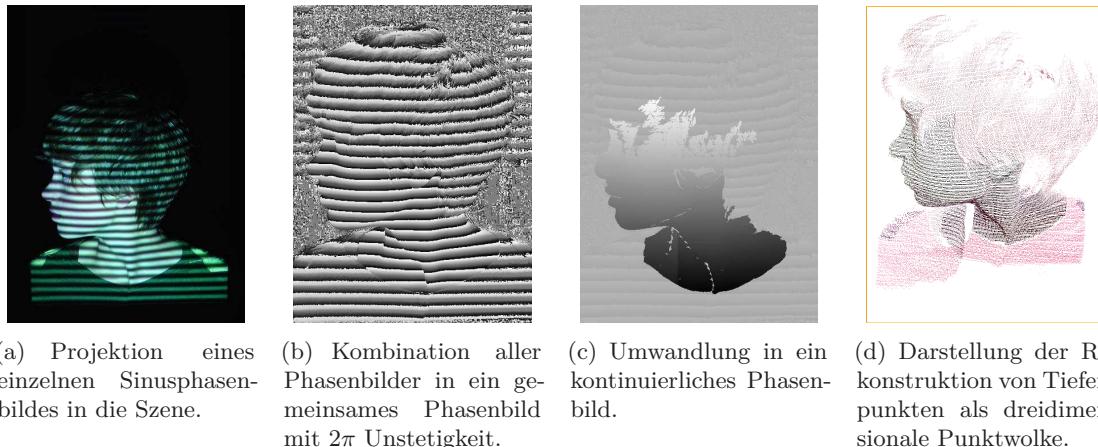


Abbildung 5.4: Arbeitsschritte des Sinusphasen-Verfahrens, Bildmaterial [McDonald, 2009]

6 Konzeption des Systems zur Oberflächenrekonstruktion

Nach Vorstellung der Projektor-Kamera-Kalibrierung und von zwei Verfahren zur Oberflächenrekonstruktion mittels Structured-Light, wird nun die Konzeption des zu realisierenden Systems vorgenommen. Hierzu wird zuerst die Wahl der einzusetzenden Methoden getroffen. Maßgeblich entscheidend für diese Wahl sind dabei die in Kapitel 2 genannten Anforderungen an eine Oberflächenrekonstruktion. Die für ein aufbauendes System zur Verfolgung von Organtransformationen zu wählenden Methoden werden zudem angesprochen. Im Anschluss wird die Oberflächenrekonstruktion selbst softwaretechnisch entworfen. Hierbei gilt es eine Modularisierung vorzunehmen und den Datenfluss zwischen den dabei beteiligten Modulen zu beschreiben. Ein Versuchsaufbau zur praktischen Validierung und softwareseitige Methoden zur qualitativen Validierung der Genauigkeit des Verfahrens werden zudem konzipiert.

6.1 Wahl einzusetzender Methoden

Das Gray-Code- sowie Sinusphasen-Verfahren finden im Bereich Structured-Light eine breite Verwendung. Für den Anwendungsbereich der CL sind sie insofern potentiell einsetzbar, da sie deren Hauptanforderungen erfüllen. Beide Verfahren sind gleichermaßen dazu geeignet vor dem Beginn der Intervention und innerhalb eines kleinen Zeitrahmens präpariert zu werden. Das Sinusphasen-Verfahren kommt ohne eine Kalibrierungsprozedur aus und ist im Anschluss an den Systemaufbau daher umgehend einsatzbereit. Das Gray-Code-Verfahren hingegen benötigt als weiteren Schritt eine Kalibrierung des Projektor-Kamera-Systems. Dies sollte jedoch den zeitlichen Workflow nicht maßgeblich beeinträchtigen, weil die Kalibrierung halbautomatisch erfolgen kann. Die Rekonstruktionsgeschwindigkeit beider Verfahren wird, nach Studium der Fachliteratur, als vollkommen ausreichend angesehen. Die Rekonstruktion einer einzelnen Ansicht ist hierbei stets von der verwendeten Hardware abhängig und vom algorithmischen Aufwand her moderat. Ein Flaschenhals für die Geschwindigkeit ist hierbei vielmehr die Gewinnung von Bilddaten. Diese hängt in erster Linie von der Projektions- und Bildaufnahmegeschwindigkeit ab.

Im Rahmen dieser Arbeit werden nun Anforderungen erfüllt, die sich in erster Linie auf die Kalibrierung und Oberflächenrekonstruktion beziehen. Wichtig ist hierbei als erster Schritt die Implementierung einer robusten, semi-automatischen Kalibrierungsmethode. Weiterhin soll eine Oberflächenrekonstruktion entwickelt werden, der die Verwendung einer monokularen Kamera genügt, die performant ist und den Workflow in zeitlicher Hinsicht unbelastet lässt.

Hieraus ergibt sich als einzusetzende Methode ein Structured-Light-Verfahren, welches sich aus einem Projektor-Kamera System zusammen setzt und alle an ein Verfahren zur Oberflächenrekonstruktion gestellten Anforderungen erfüllt. Um zu ermitteln welches Verfahren sich potentiell besser dazu eignet, wird das Gray-Code- sowie Sinusphasen-Verfahren umgesetzt. Weiterhin ist die nahtlose Integration in das verwendete Framework ein Hauptanspruch. Hierbei sind Modularisierung und Performanz des Datenworkflows besonders wichtig. An das Verfahren gestellte Anforderungen und zur Lösung gewählte Methoden sind somit die Folgenden:

- Umsetzung eines Systems zur Oberflächenrekonstruktion mittels minimalem Structured-Light-Aufbau.
- Monokulare Rekonstruktion der Leberoberfläche mittels Projektor-Kamera-System.
- Halbautomatische Kalibrierung von Projektor und Kamera mittels Tsai-Methode.
- Schnelle Rekonstruktion mittels kalibrierungsgestütztem Verfahren (Gray-Codes).
- Schnelle Rekonstruktion ohne Durchführung einer Kalibrierung (Sinusphasen-Verfahren).
- Integration beider Verfahren in die Software MeVisLab.

Zukünftige Entwicklungen Im Rahmen weiterer Entwicklung, hin zu einem Gesamtsystem zur Navigationsunterstützung durch Verfolgung von elastischen Organtransformationen, wird es notwendig werden eine Trackingmethode einzusetzen. Zur Verfolgung rigider Transformationen wurden in Abschnitt 3.2.3 Methoden des visuellen Trackings, sowohl planares Markertracking- sowie Infrarottracking vorgestellt. Rigit Transformationen können beide Verfahren mittels Anbringung von Markern an der Organoberfläche tracken. Wie beschrieben ist das Verfolgen von planaren Markern jedoch fehleranfällig. Falls sich diese an einer starken Oberflächenwölbung befinden ist die Schräglage unter Umständen zu groß und eine Detektion nicht mehr möglich. Da die Adaption an gewölbte Oberflächen jedoch eine Anforderung ist, sollte hier das Infrarottracking dem Tracken von planaren Markern vorgezogen werden.

Anforderungen an eine adaptive Projektion sind die Anpassung an Farbe und Wölbung der Organoberfläche. Sowohl die Anpassung an stark gewölbte Oberflächen, sowie den Farb- und Intensitätsausgleich können die, im Ausblick der Arbeit vorgestellten, Verfahren bieten. Die Minimalisierung des Abbildungsfehlers, bei der Projektion auf ein rigides Objekt, ist ebenfalls eine

Anforderung die es dabei umzusetzen gilt. Erst nach Kenntnis des aktuellen Abbildungsfehlers, wird die Bereitstellung eines visuellen Feedbacks, das über den Grad des Abbildungsfehlers informiert, bzw. den Abbruch der Rekonstruktion veranlasst, ermöglicht. Ein geeigneter Algorithmus zur Feststellung der Projektionsvalidität und ihrer Verbesserung muss dazu also zukünftig entwickelt werden.

6.2 Softwaretechnischer Entwurf in MeVisLab

Die Oberflächenrekonstruktion wird in ihrer Gesamtheit im Framework MeVisLab integriert, welches in Abschnitt 7.1 noch detaillierter vorgestellt werden wird. Da die Software komponentenorientiert ist, werden vorerst Bausteine identifiziert, die in Kombination miteinander einen schematischen Programmablauf ergeben. In Abbildung 6.1 sind die Schritte beim Ablauf der Oberflächenrekonstruktion aufgezeigt. Jeder Schritt entspricht dabei einem abgeschlossenen, abstrakten Modul, welches für sich die für die Funktion jeweils nötigen Verfahren zusammenfasst.

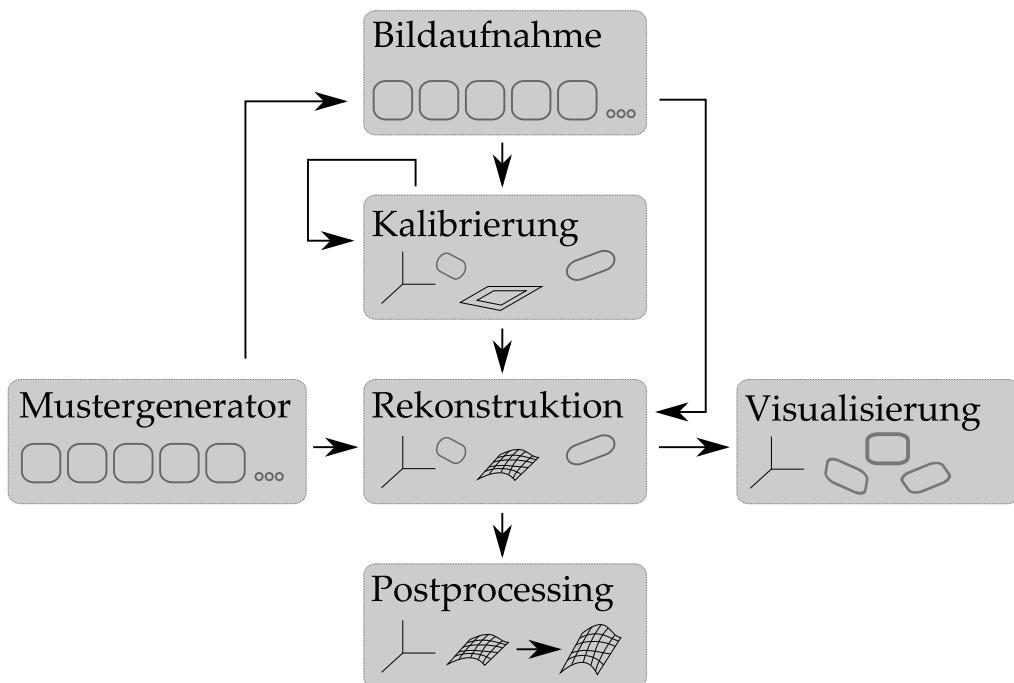


Abbildung 6.1: Modulbausteine im schematischen Datenflusses der Oberflächenrekonstruktion. Ein Modul zur Bildaufnahme liefert Szenenbilder an ein Kalibrierungs-Modul. Ist die Kalibrierung erfolgt, so werden entzerrte Bilddaten direkt ans Rekonstruktions-Modul weitergereicht. Gewonnene Oberflächendaten werden zur weiteren Verarbeitung an ein Postprocessing-Modul gegeben. Dieses kann beispielsweise zur Aufbereitung der Daten oder zur Berechnung von Abbildungsfehlern dienen.

Modulkommunikation Um den vorgestellten Datenfluss mit beteiligten Modulen in MeVisLab umzusetzen, gilt es nun die tatsächlich einzusetzenden Module herauszustellen. Der Datenfluss selbst wird als eine Reihe von Datenverarbeitungsprozessen modelliert, welche zwischen diesen Modulen ablaufen. In Abbildung 6.2 ist der Ablauf der Oberflächenrekonstruktion in Form eines Sequenzdiagramms dargestellt. Dieses behandelt in erster Linie die Hauptbestandteile des Verfahrens: Bildaufnahme, Kalibrierung und Rekonstruktion. Bei einer kontinuierlichen Rekonstruktion mehrerer Ansichten eines Objekts ist vor allem der Datenfluss in Richtung Rekonstruktionsmodul von Belang.

Im Rahmen der Bildaufnahme dient das Modul *VideoCapture* dazu, in einer festlegbaren Wiederholrate, Kamerabilder anzufordern. Das Visualisierungsmodul *View2D* liest das so erzeugte Bild aus und dient zu dessen Visualisierung in Echtzeit. Das Modul *VideoDeviceCalibration* kann an *VideoCapture* die Anfrage *requestFrame* richten um den letzten aktuellen Frame übergeben zu bekommen. Die Detektion von Kalibriermustern kann im laufenden Videobild erfolgen und eine erfolgreiche Detektierung als visuelles Feedback zurück an *View2D* gereicht werden. Hauptaufgabe des Moduls *VideoDeviceCalibration* ist es, Kamerabilder auf denen die benötigten Kalibriermuster erkannt werden, zu speichern und auf dieser Grundlage eine vollständige Kalibrierung von Kamera und Projektor durchzuführen. Die automatische Detektion wird durch eine manuelle Auswahl von Mustereckpunkten ergänzt, die wiederum auf statische, im Dateisystem gespeicherten Bildern operiert.

Das Visualisierungsmodul *ViewCalibration* erhält als Eingabe die ermittelten extrinsischen Gerätewerte, um diese zum Zweck der Kontrolle als dreidimensionale Szene darzustellen. Das Modul *StructuredLight* erhält als Eingabe hingegen die vollständige Kalibrierung beider Geräte. Durch das Modul *PatternGenerator* wird nach Angabe von Parametern zum projizierten Gray-Code bzw. zum Sinusphasenbild, ein Muster auf ein für den Projektor vorgesehenes Visualisierungsfenster *ProjectorPattern* weiter gereicht. Die erfolgreiche Anzeige eines Musters wird an *StructuredLight* gemeldet, worauf mittels *VideoCapture* ein Bild der aktuellen Szene abgespeichert wird. Nach erfolgreicher Speicherung, wird das nächste zu projizierende Muster an *ProjectorPattern* gereicht, bis die Sequenz vollständig ist.

Die Hauptfunktion des *StructuedLight*-Moduls ist *reconstructSurface*, welches für die aktuell aufgenommene Sequenz eine dreidimensionale Rekonstruktion durchführt. Diese Daten werden als Punktfolge ausgegeben und können wiederum visualisiert oder aber weiter verarbeitet werden. Das Modul *PostProcessor* führt Nachverarbeitungsschritte wie Entrauschen und Zuschnitt der Punktfolge auf ein kleineres Volumen durch. Die Differenz zwischen Referenzdatensatz und der Oberflächenrekonstruktion soll durch das Modul *DiffViewer* visualisiert werden.

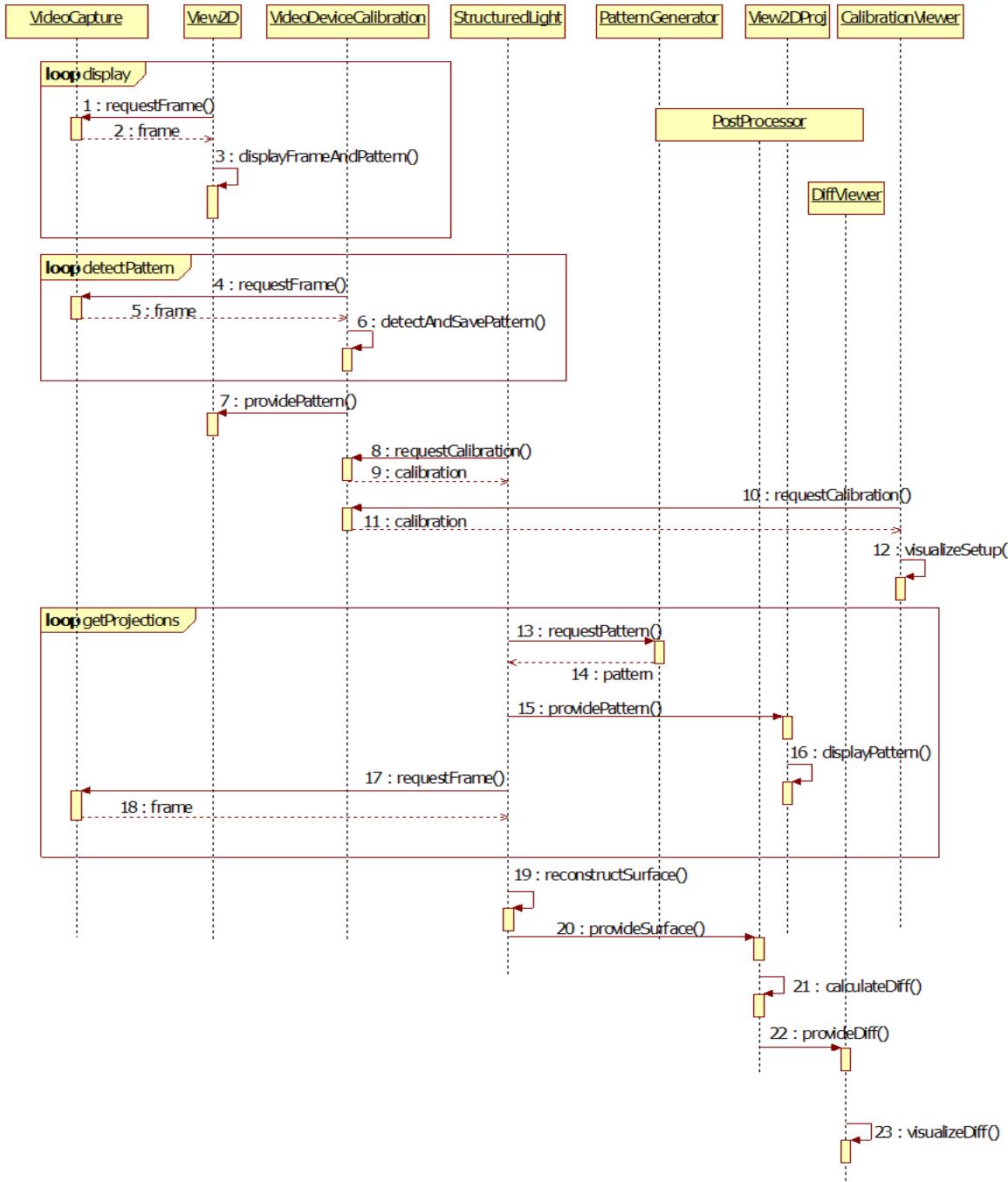


Abbildung 6.2: Sequenzdiagramm des Gesamtverfahrens zur Oberflächenrekonstruktion. Der in Abbildung 6.1 beschriebene Datenfluss abstrakter Module ist hier als Austausch von Nachrichten zwischen tatsächlichen MeVisLab-Modulen konzipiert. Zuerst wird der von der Kamera eingehende Bildstrom visualisiert und Kalibriermuster darin detektiert. Nach der Kalibrierung des Projektor-Kamera-Systems wird eine kontinuierliche Rekonstruktion als Schleifere *reconstructSurface* gestartet.

6.3 Konzeption einer Testumgebung

Neben dem Entwurf der Softwaremodule in MeVisLab gilt es praktische Tests zu konzipieren, die zur Validierung des Verfahrens eingesetzt werden können. Dazu wird in erster Linie eine geeignete Testumgebung benötigt. Hauptziel beim Aufbau dieser ist es, eine möglichst optimale Konfiguration von Aufbauten und Umgebungsvariablen zu ermitteln. Für die Validierung des Verfahrens ist die Ermittlung dieser Parameter essentiell. In wie fern sich die Variation von Parametern auf die Rekonstruktion auswirkt wird in Kapitel 8 aufgezeigt. Weiterhin werden rigide Testobjekte, sogenannte Phantome benötigt. Diese stellen Referenzdatensätzen mit bekannter Geometrie dar und ermöglichen somit den Vergleich mit Rekonstruktionsergebnissen.

Konzeption des Geräteaufbaus

Alle folgenden Bezeichner beziehen sich auf Abbildung 6.3, in welcher der schematische Aufbau der benötigten Gerätschaften dargestellt ist.

Der Winkel a von Kamera zu Projektor wird hierbei nicht zu spitz oder stumpf gewählt, wobei sich beide Optiken auf der in etwa gleichen Höhe d befinden. Dies kann theoretisch die Präzision der Kalibrierung und somit der Triangulierung unterstützen, da Extremwerte beim Schnitt von Sehstrahlen von vorn herein vermieden werden. Die Verwendung von Stativen ist zudem unerlässlich, um unabsichtliche Veränderungen im Aufbau ausschließen zu können und somit die Validität der Ergebnisse zu gewährleisten.

Die Blickpunkte von Kamera und Projektor sollten im Idealfall in einem gemeinsamen Punkt COI übereinstimmen, wobei das zu rekonstruierende Objekt O im Schärfebereich FB liegen sollte, welcher von beiden Geräten inkludiert wird. Alle Automatismen der Geräte, wie Auto-Fokus oder automatischer Beleuchtungsabgleich müssen deaktiviert werden, um die Veränderung der Kalibrierung während der Aufnahme einer Bildserie auszuschließen und eingesetzte Testparameter nachvollziehbar zu halten.

Das projizierte Muster sollte das gesamte Objekt bedecken, um alle Oberflächenpunkte zu kodieren. Auftretenden Verschattungen gilt es ggf. durch leichte Korrekturen der Projektionsposition zu vermindern. Das Objekt sollte im Kamerabild formatfüllend zu sehen sein, um auch Details erfassen zu können.

Auch auf die Parameter der Umgebung selbst muss geachtet werden. Je nach Lichtstärke eines Projektors ist die Abdunklung der Umgebung sinnvoll, um somit einen stärkeren Kontrast von projizierten Mustern herzustellen. Hierbei muss jedoch auch darauf geachtet werden, dass ein zu hoher Kontrast zur Übersteuerung der Kamera führen kann oder aber Glanzlichter verstärkt. Je nach eingesetzten Gerätschaften sind hier also manuelle Feineinstellungen vorzunehmen.

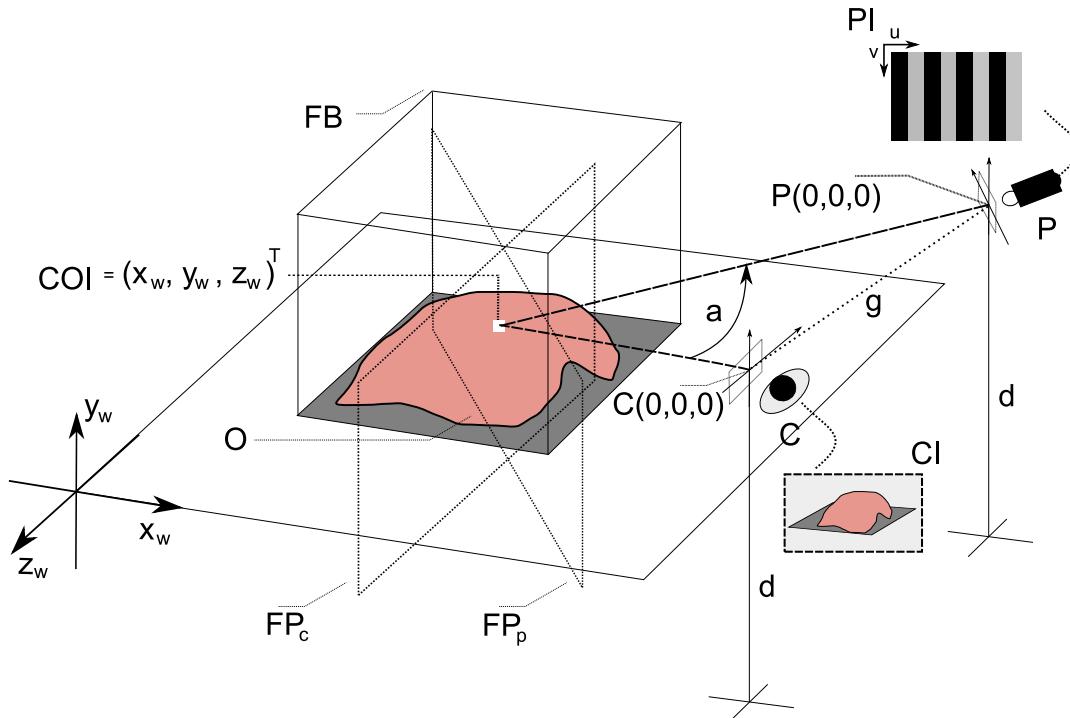


Abbildung 6.3: Schematischer Geräteaufbau mit Kamera C , Projektor P und Objekt O . Die Höhe der optischen Zentren beträgt d , der Abstand zwischen Kamera und Projektor beträgt g , der gemeinsame Blickpunkt ist COI . Die Schärfeebene des projizierten Bildes PI ist als FP_p , die Raumebene die im Kamerabild CI am schärfsten abgebildet wird als FP_c angegeben. Das Objekt sollte im besten Falle innerhalb des Fokalebereichs FB liegen, den sowohl Projektor als auch Kamera abdecken.

Eingesetzte Phantommodelle

Um die Rekonstruktion hinreichend genau testen zu können, wurden einerseits Phantommodelle der menschlichen Leber, sowie einfache geometrische Objekte verwendet. Zu den mittels 3D-Druck-Verfahren erstellten Modellen der Leber existieren dreidimensionale Referenzdatensätze, wodurch ein Vergleich der rekonstruierten Oberfläche mit diesen ermöglicht wird. Einfache geometrische Objekte sind exakt ausgemessene Phantome wie eine Kugel oder eine planare, rechteckige Fläche. Hiermit können gewonnene Rekonstruktionen ihrerseits gegen computergenerierte Referenzdatensätze verglichen werden.

Die in Abbildung 6.4 (a) und (b) gezeigten Leberphantome L_A und L_B besitzen eine rau, matte Oberfläche. Diese entsprechen, bis auf ihre Färbung, insofern nicht einer tatsächlichen Leberoberfläche, da sie keine viskose, feuchte Konsistenz aufweisen. Das verwendete Anatomiemodell der Leber ermöglicht es hingegen die Reflektionseigenschaften des realen Organs tendenziell nachzubilden. Da Glanzlichter eine der Hauptfehlerquellen bei der Oberflächenrekonstruktion sind, wurden Modelle wie das in Abbildung 6.4 (c) dargestellte in die Betrachtung mit einbezogen. Inwiefern das Rekonstruktionsverfahren in der Lage ist an diesen Stellen den-

noch Daten zu gewinnen, bzw. welche Konfiguration der Testumgebung vorteilhaft für die Minderung von Glanzlichtern ist, wird anhand dieses Modells untersucht werden.

In Abbildung 6.4 (d)-(f) sind einige weitere, markante Merkmale, der im Rahmen von MRT-Untersuchungen gewonnenen Daten eines Leberphantoms dargestellt. Selbstverdeckungen und starke Ein- bzw. Auswölbungen sind zentrale Herausforderungen für das Verfahren zur Oberflächenrekonstruktion. Eine über die gesamte Projektionsfolge, vollkommen verschattete Region ist nicht zu rekonstruieren. Im Fall von starker Wölbung können im Zweifelsfall jedoch, durch Einwirkung von Streustrahlung, teilweise verschattete Regionen aufgehellt und rekonstruiert werden.

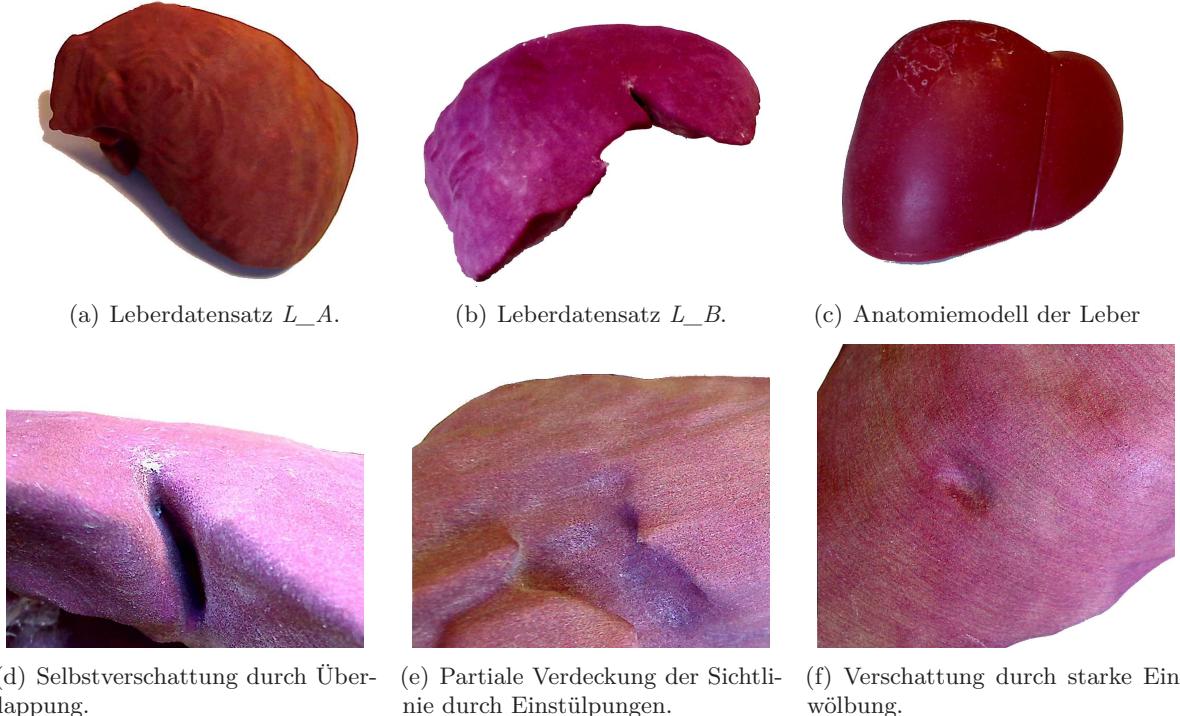
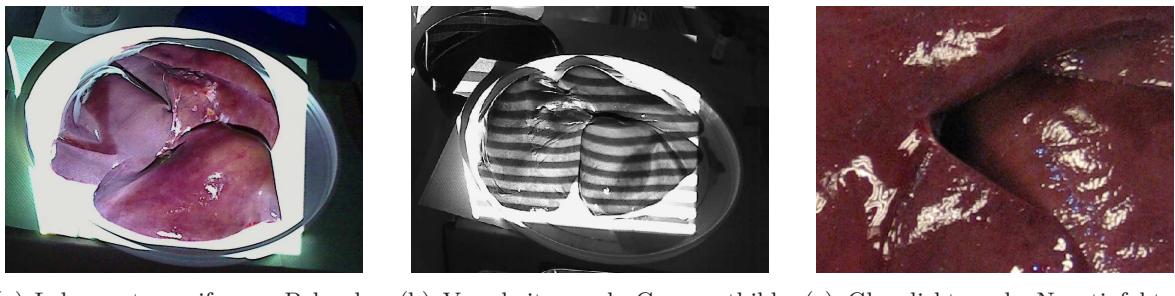


Abbildung 6.4: Zu Testzwecken verwendete Phantommodelle. Bei den in Abbildungen (a) und (b) dargestellten Phantomen L_A sowie L_B handelt es sich um, mittels 3D-Druck gewonnene Hohlkörper aus Kunsstoff, welche aus segmentierten MRT-Datensätzen stammen. Das in (c) dargestellte Phantom ist ein Festkörper und besteht aus einer elastischen Gummierung. Für die Oberflächenrekonstruktion problematische Eigenschaften der Leberoberfläche, sind in (d)-(f) aufgezeigt.

Tests mittels Schweineleber

Neben den verwendeten, rigiden Phantomen, wurden zur praxisbezogenen Validierung des Verfahrens, Schweinelebern verwendet. Vordergründige Unterschiede sind hier in erster Linie die Viskosität des realen Organs, die je nach Lagerung differenzierte Selbstverdeckungen zur Folge hat. Glanzlichter sind zudem sehr viel zahlreicher und ausgeprägter als bei allen verwendeten Phantomen, siehe dazu Abbildung 6.5 (c). Zur vergleichenden Durchführung von mehreren Rekonstruktion des aktuellen Organzustands eignet sich eine Schweineleber im Laborversuch, da keine Selbsttransformation stattfindet. Im operativen Einsatz ist dies durch die im Grundlagenkapitel aufgeführten Einwirkungen auf das Organ nur bedingt gewährleistet.

Im Hinblick auf den Einsatz von Structured-Light-Verfahren in der CL wird das Ziel darin bestehen das Ausmaß der Problematik zu untersuchen, die bei der Rekonstruktion einer realen Leberoberfläche auftritt. Im Anschluss an das nun folgenden Realisierungskapitel wird aufgezeigt in welchem Maße diese Problematik besteht und wie ihr begegnet werden kann.



(a) Leber unter uniformer Beleuchtung durch den Projektor. (b) Verarbeitung als Grauwertbild während der Projektionsfolge. (c) Glanzlichter als Negativfaktor über Teile der Organoberfläche.

Abbildung 6.5: Verwendung einer Schweineleber als Testobjekt

Zusammenfassung In diesem Kapitel wurde der Einsatz eines minimalen Structured-Light-Systemaufbaus zur Durchführung der Oberflächenrekonstruktion beschlossen, wobei keine besonderen Anforderungen an eingesetzte Gerätschaften gestellt werden. Mit der Tsai-Kalibrierung wird ein Standardverfahren eingesetzt, das bereits eine breite Anwendung im Bereich der Computer-Vision gefunden hat. Um die Rekonstruktion durchzuführen, sollen zwei Vertreter des Structured-Light-Ansatzes zur Anwendung kommen und verglichen werden. Die Identifikation von Modulen, die in MeVisLab entwickelt werden sollen, wurde im Anschluss durchgeführt. Zur Validierung dieser Realisierung wurde eine minimale Testumgebung konzipiert, in der einerseits die Oberflächen rigider Leberphantome, als auch einer realen Schweineleber untersucht werden sollen.

7 Realisierung

In diesem Kapitel werden die zur Realisierung der Oberflächenrekonstruktion verwendeten Softwarepakete und deren Kernpunkte vorgestellt. Es folgt die Beschreibung der durchgeführten Implementierung, welche die in der Konzeption ermittelten Teilmodule umsetzt. Dies umfasst Module zur Kameraansteuerung, Kalibrierung des Projektor-Kamera-Systems, zur Oberflächenrekonstruktion und dem Postprocessing.

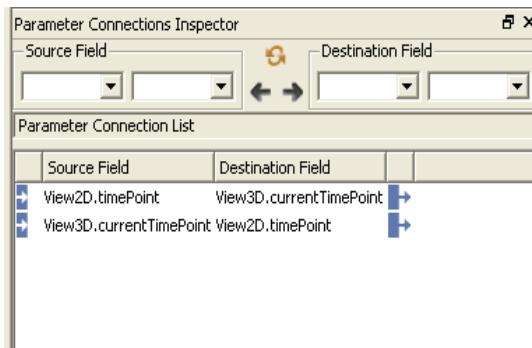
7.1 Eingesetzte Softwareumgebung

Zielplattform der Entwicklung aller Module ist das Softwareframework MeVisLab. Für Teile der Modulentwicklung zur Kameraansteuerung und Kalibrierung wurden Funktionen der Softwarebibliothek OpenCV eingesetzt. Beider Systeme werden nun kurz besprochen.

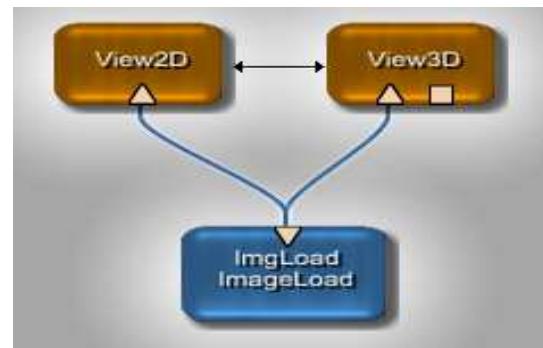
MevisLab Die grafische Entwicklungsumgebung MeVisLab dient zur Bildverarbeitung und Visualisierung von in erster Linie medizinischen, multidimensionalen Datensätzen. Aktuell wird das Softwarepaket von der *Mevis Medical Solutions AG* (MMS) entwickelt und stellt eine wichtige Arbeitsgrundlage für die hausinterne Forschung am Fraunhofer MEVIS dar. Das übergeordnete Konzept der Software basiert auf dem Paradigma der grafischen Programmierung. Hierbei werden Module, welche Grundfunktionalitäten zur Datenverarbeitung, Visualisierung oder Interaktion bereitstellen, in Programmnetzwerken miteinander kombiniert. Netzwerke lassen sich zu Makromodulen zusammenfassen, was dazu dient die Übersichtlichkeit größerer Netzwerkstrukturen zu erhöhen, oder aber schlicht um Funktionalitäten zu kapseln. Per Datenaustausch über gemeinsame Ein- und Ausgabeschnittstellen wird der Programmfluss zwischen einzelnen Modulen umgesetzt. Die Entwicklung stützt sich auf C++ als Programmiersprache, während das Scripting der durch die plattformübergreifende Bibliothek QT [Blanchette u. Summerfield, 2008] entwickelten Benutzungsoberfläche durch die objektorientierte Skriptsprache Python von Statten geht. Als Entwicklungswerkzeug für die Programmierung kam *Microsoft Visual C++ 2008* zum Einsatz, wobei die getestete Zielplattform Windows XP war.

In Abbildung 7.1 ist ein minimales Beispielnetzwerk in MeVisLab dargestellt. In 7.1 (a) ist die bidirektionale Verknüpfung (SourceField \longleftrightarrow DestinationField) von Modulparametern gezeigt.

Der vom Modul *ImageLoad* (7.1 (b)) gelieferten Bilddatensatz wird gleichzeitig in den Visualisierungsmodulen *View2D* sowie *View3D* dargestellt, was über die vom Modul ausgehenden Verbindungslien symbolisiert wird. Über die als bidirektionalen Pfeil dargestellte Verbindung zwischen diesen beiden Modulen wird eine Synchronisation des darzustellenden Zeitschritts des durch *ImageLoad* geladenen Datensatzes geregelt. Die Änderung dieses Parameters in einem der beiden Module veranlasst die sofortige Angleichung im jeweils anderen Modul.



(a) Bidirektionale Verknüpfung von Parametern der Module *View2D* und *View3D*.



(b) Visualisierung des Netzwerkes. Es besteht Bottom-Up und Left-To-Right Datenfluss vom Modul *ImageLoad* an gesehen.

Abbildung 7.1: Beispielnetzwerk in der Software MeVisLab

OpenCV Da zu Beginn der vorliegenden Arbeit, in der Software MeVisLab keine plattformübergreifende Anbindung von Kameras zur Verfügung stand, wurde die Software OpenCV [Bradski u. Kaehler, 2008] eingesetzt, um diese zu schaffen. OpenCV ist eine initial von der Intel Corporation erstellte Programmzbibliothek, die sich schwerpunktmaßig der Bildverarbeitung und dem maschinellen Sehen widmet. In C und C++ verfasst, ist die Software zudem stark leistungsoptimiert und mittlerweile quelloffen verfügbar gemacht worden. Funktionen zur Kameraansteuerung und Kalibrierung sind in OpenCV plattformübergreifend integriert, was das entscheidende Argument für die letztendliche Nutzung darstellte.

7.2 Implementierung benötigter Module

Ein erstes Problem, das es im Rahmen der Implementierung zu lösen galt, bestand in der Umsetzung einer auf OpenCV aufbauenden Kameraansteuerung. Da in OpenCV zur Kalibrierung nötige Verfahren größtenteils bereitstanden, galt es diese in die eigene Implementierung an gegebener Stelle zu integrieren. Der Hauptanteil der Implementierungsarbeit bestand schließlich in der Umsetzung der Structured-Light-Verfahren, namentlich des Sinusphasen- und Gray-Code-Verfahrens, sowie deren Integration in MeVisLab.

7.2.1 Kameraansteuerung

Die Einbindung der OpenCV-Bibliothek erfolgt über das automatische Buildsystem von MeVisLab, wobei benötigte Header-Dateien nach Angabe der dazugehörigen Pakete *OpenCV_gui* und *MLVideoCaptureInterface* in der Projektdatei des zu entwickelnden Moduls automatisch hinzugefügt wurden. Das zur Kameraansteuerung erstellte Modul *VideoCapture*, bindet für je ein Betriebssystem eine Headerdatei ein, die ihrerseits das abstrakte, plattformunabhängige Interface *VideoCaptureInterface* implementiert. Für die Verwendung auf Windows wurden alle abstrakten Interfacefunktionen mittels OpenCV in Form der Klasse *OpenCVVideoCapture* implementiert. Das Schema der Implementierung ist in Abbildung 7.2 dargestellt.

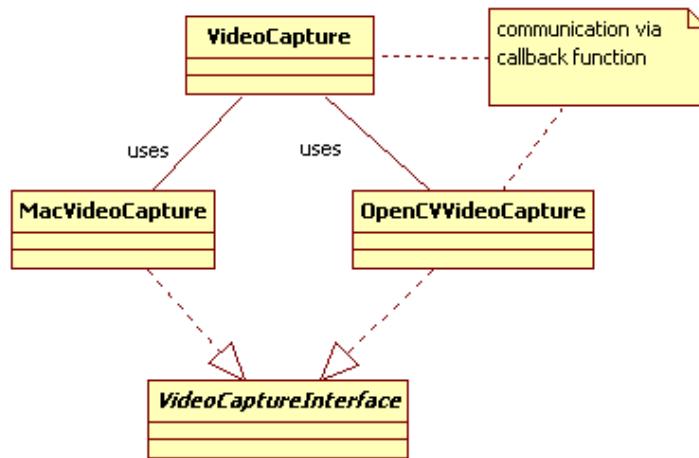


Abbildung 7.2: Klassenbeziehungen der Kameraanbindung. Das abstrakte Interface *VideoCaptureInterface* wird durch die Backends *OpenCVVideoCapture* und *MacVideoCapture* implementiert. Die Kommunikation der Backends mit dem Hauptmodul *VideoCapture* ist jeweils durch eine Callback-Funktion umgesetzt.

Aus Gründen der Performance wurde bei der Entwicklung des *VideoCapture*-Moduls ein nebenläufiger Thread mittels *QT* erstellt, der zum kontinuierlichen Abfragen des aktuellen Kamerabildes diente. Das Modul nutzt eine Callback-Funktion um mit dem aktuell verwendeten Backend zu kommunizieren. Sobald ein neues Kamerabild im Backend bereit steht, wird dies im Hauptmodul gemeldet, worauf dieses eine Speicherstelle an das Backend zurückgibt, an welche die Daten des aktuellen Bildes geschrieben werden sollen. Das Schreiben der Daten ist hierbei Aufgabe des Backends, welches mittels der Funktion *getVideoFrameAsMLImage* die Lieferung des aktuellen Videoframes im MeVisLab-Bildformat übernimmt.

Hierbei galt es zu beachten, dass von OpenCV ein von MeVisLab abweichendes Bildformat verwendet wird. Die Reihenfolge der Farbinformation wurde von BGR in RGB gewandelt und für MeVisLab in die Kanäle eines standardmäßig verwendeten, 6D Datensatz geschrieben,

siehe Abbildung 7.3. Nach Wandlung des aktuellen Videobildes, wird dieses an den Ausgang des *VideoCapture* Moduls weiter gereicht und steht somit als Eingabequelle für die darauf aufbauenden Module zur Kamerakalibrierung und Oberflächenrekonstruktion zur Verfügung.

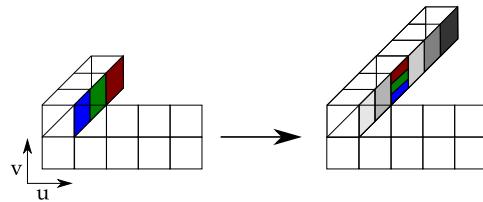
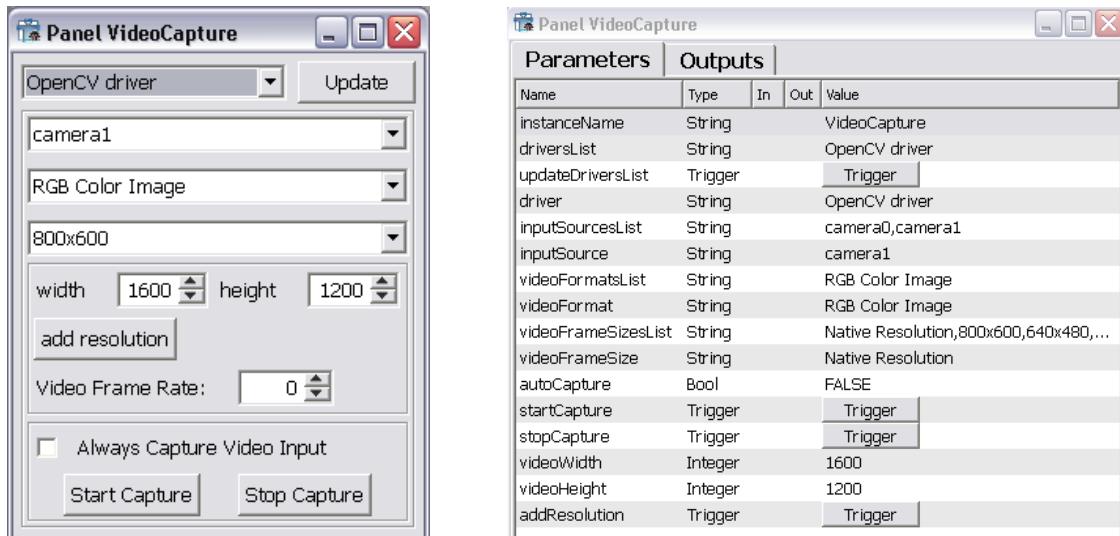


Abbildung 7.3: Schema der Kanalwandlung bei der Kameraansteuerung. Das OpenCV-Bildformat (links) wird in das von MeVisLab verwendete Format (rechts) gewandelt. Von einem 2D-Bild mit drei Farbkomponenten in der Sortierung BGR wird in diesem Beispiel in den dritten Kanal eines 2D-Bildes konvertiert, der die Farbwerte in der Reihenfolge RGB übernimmt.

Einsatz von OpenCV Wichtige OpenCV-Funktionen die zur Kameraansteuerung eingesetzt wurden sind *cvCaptureFromCAM*, womit eine angebundene Kamera automatisch erkannt wird. Die Funktion *cvSetCaptureProperty* dient dazu Parameter für die Bildaufnahme zu setzen, wonach mittels Aufruf von *cvQueryFrame* das aktuelle Kamerabild zurück gegeben wird. Zur Wandlung des OpenCV-Bildformats in das für MeVisLab benötigte, wurde die in Listing 1 aufgeführte Funktion *getCurrentFrameAsMLImage* implementiert. In Zeilen 18 bis 21 wird, per Dereferenzierung eines Zeigers, auf die aktuelle Position im Zielbild der Wert des Kamerabildes an der Stelle *srcPos* geschrieben.

Benutzeroberfläche In Abbildung 7.4 ist die, für das *VideoCapture*-Modul erstellte Benutzeroberfläche dargestellt. Neben der Auswahl des zu benutzenden Backends, wurde die Wahl der anzusteuerten Kamera, der Farbraum in dem das Bild ausgegeben werden soll und die Wahl der zu verwendenden Auflösung implementiert. Da OpenCV wegen der zu wahren Plattformunabhängigkeit lediglich eine rudimentäre Erkennung von Kameras unterstützt und nicht alle vom Gerät direkt unterstützten Auflösungen ermittelt, wird zusätzlich eine manuelle Angabe der gewünschten Auflösung unterstützt. Während der Videostream aktiviert ist können die genannten Parameter beliebig verändert werden. Je nach gewählter Auflösung kann die Wandlung des Videobildes in das von MeVisLab benötigte Format allerdings rechenintensiv werden. Die Framerate sollte in solchen Fällen limitiert werden.



(a) Grafische Benutzeroberfläche der Kameraansteuerung. Wahl des zu verwendenden Backends, der Kamera, des Farbraums und der Auflösung. Die Bildraten sowie die Aktivität des Videoeingangs sind zudem manuell kontrollierbar.

(b) Ansicht der Parameter des Moduls *VideoCapture*. Alle Parameter sind mittels Scripting und aus dem C++-Modulcode heraus per get- bzw. set-Methoden, im Fall von Parametern von Typ „Trigger“ durch eine touch-Methode kontrollierbar.

Abbildung 7.4: Benutzeroberfläche des Moduls *VideoCapture* und dessen Parameter.

7.2.2 Kalibrierungsmodul

Nach Erhalt des aktuellen Videobildes im MeVisLab-Format, bestand die Aufgabe darin, dieses zur Kalibrierung des Projektor-Kamera-Systems zu nutzen. Ein Anspruch, der sich bereits in der Motivation ergab, war die leichte Durchführbarkeit, sowie hohe Genauigkeit des Kalibrierungsvorgangs. Um dieser Anforderung zu entsprechen wurde mit dem von [Zhang, 2000] vorgestellten Ansatz, ein einerseits minimalistisches, aber auch breit angewendetes und robustes Verfahren, welches auf Kalibrierung mittels planarem Schachbrettmuster basiert, gewählt.

Sind die 2D-Koordinaten der zu einem Muster gehörigen Eckpunkte im Kamerabild ermittelt, so kann die, in der Softwarebibliothek OpenCV [Bradski u. Kaehler, 2008] ebenfalls verwendete Kalibrierungsmethode nach Zhang, die auf die Ergebnisse von Tsai aufbaut, angewendet werden. Dazu galt es die Bildpositionen der Kalibermuster-Eckpunkte von Kamera und Projektor in aufgenommenen Kamerabildern durch die Methode *cvFindChessboardCorners* zu detektieren und zu sammeln. Die Bestimmung der Raumpositionen dieser Eckpunkte wurde durch die Implementierung des in Abschnitt 4.2 vorgestellten Strahl-Ebenenschnitts vorgenommen. Die Ermittlung von Geräteintrinsiken wurde daraufhin mittels der OpenCV-Methode *cvCalibrateCamera2* vorgenommen. Die Extrinsik des Projektors wurde unter Einsatz der Methode *cvFindExtrinsicCameraParams2* ermittelt.

Datenfluss Innerhalb von MeVisLab kann unmittelbar nach dem manuellen Starten des Videostreams, über einen Kontrollmonitor wie etwa dem Modul *View2D*, das von der Kamera gelieferte Bild angezeigt werden. In Abbildung 7.5 ist ein Ausschnitt des zum Gesamtverfahren gehörigen Modulnetzwerks schematisch dargestellt, wobei der Videostream vom Modul *Video-Capture* in das zur Kalibrierung genutzte Markomodul *VideoDeviceCalibrationMacro* durchgereicht wird. Aus diesem Makromodul werden wiederum diverse Daten wie die per *OpenCV* detektierte und manuell verfeinerte Eckpunktmenge, sowie das Videosignal selbst als Ausgaben bereitgestellt. So erzeugte Ausgabewerte und ermittelte Kalibrierungsparameter dienen sodann als Grundlage der Visualisierung des Gesamtaufbaus und im nächsten Schritt zur Oberflächenrekonstruktion.

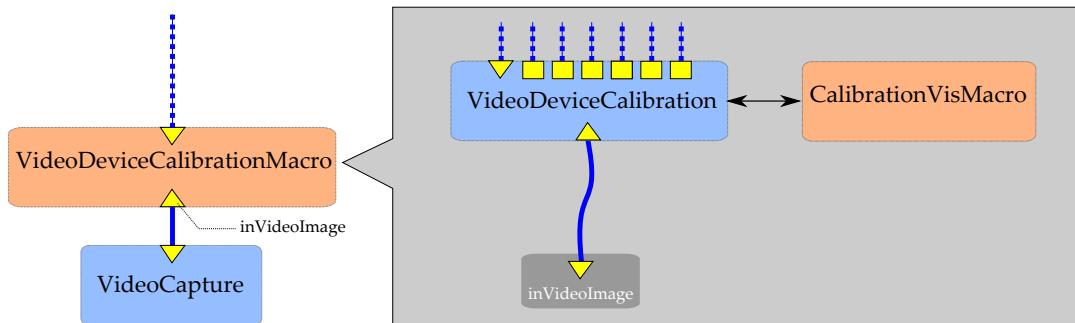
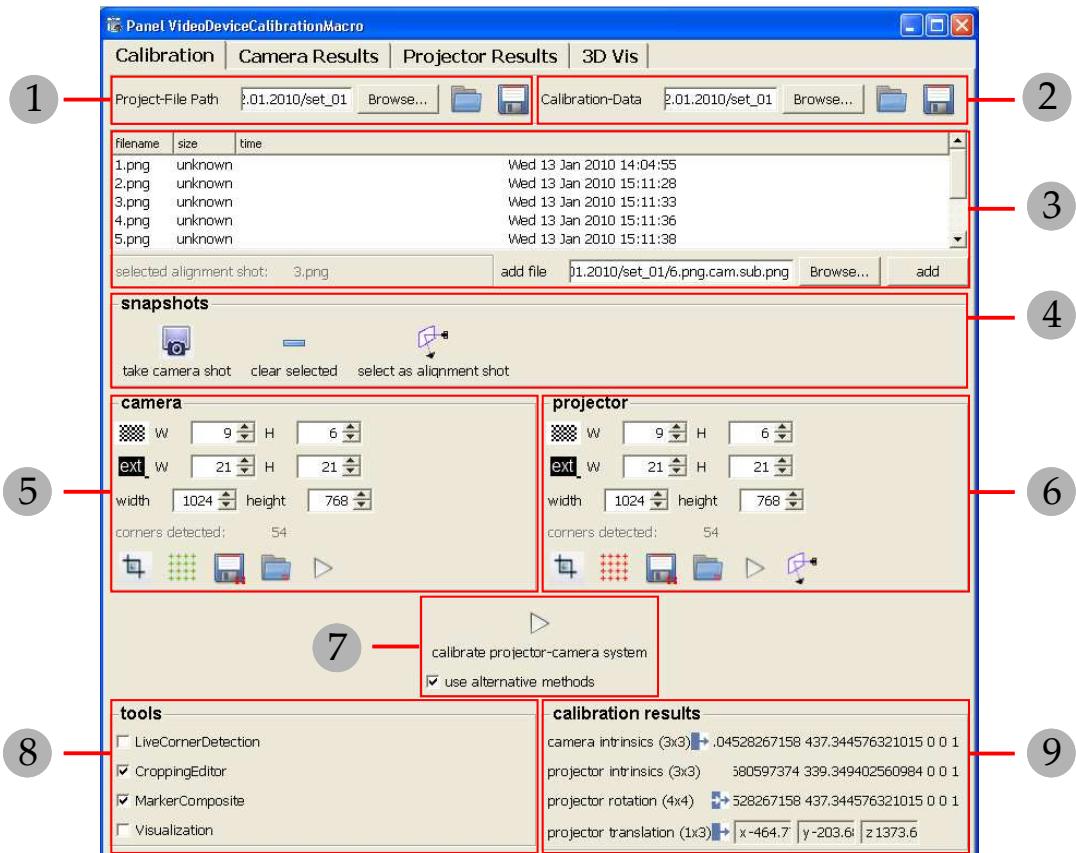


Abbildung 7.5: Schematische Datenflussbeschreibung des Moduls *VideoDeviceCalibration*. Die Eingabe des Videostroms vom Modul *VideoCapture* in das Makromodul *VideoDeviceCalibrationMacro* aus Netzwerksicht (links). Durchreichung des Videostroms an das im Makromodul befindliche Modul *VideoDeviceCalibration* (rechts). Kalibrierungsergebnisse werden mittels Verbindung von Modulparametern mit dem Makromodul *CalibrationVisMacro* visualisiert.

Benutzung Die in Abbildung 7.6 gezeigte Benutzeroberfläche zur Kalibrierung bietet alle nötigen Elemente um eine in- wie extrinsische Kalibrierung des Projektor-Kamera-Systems halbautomatisch durchzuführen. Zu referenzierten Elementen wird jeweils der Bereich (X) genannt, in dem sie sich befinden.

Ziel der Kalibrierung ist es, Bilder zu akquirieren auf denen ein oder aber beide spezifizierten Kalibriermuster, zum Einen das auf eine rigide Ebene angebrachte, sowie das auf diese Ebene projizierte Muster erkannt werden. Ist dies der Fall, so kann das betreffende Bild mittels der Schaltfläche *take camera shot* (4) abgespeichert werden. Ein hinzugefügtes Bild kann nach Belieben wieder aus der Liste der für die Kalibrierung verwendeten Bilder (3) entfernt oder hinzugefügt werden, alle Bilder werden dabei im Dateisystem abgespeichert. Des Weiteren wird die Verwaltung des Projekts mit Angabe des Verzeichnisses *Project-File-Path* (1) für zu ladende oder speichernde Projektdateien und Aufnahmen und die Verwaltung von Kalibrierungsergebnissen *Calibration-Data* (2) geboten. In der Sektion *tools* (8) sind für die Kalibrie-


 Abbildung 7.6: Grafische Benutzeroberfläche des Moduls *VideoDeviceCalibration*.

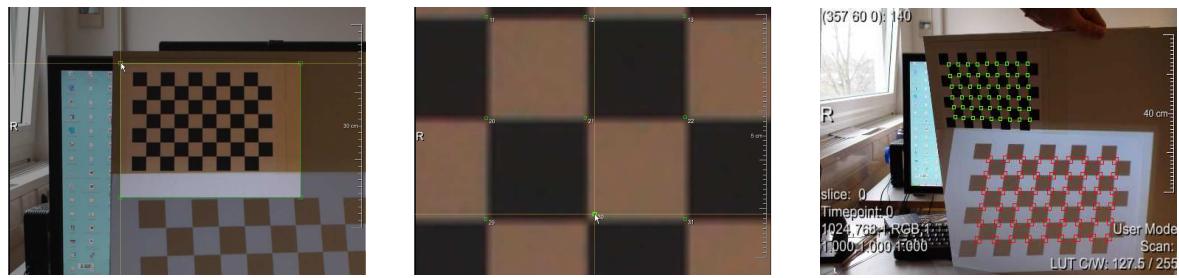
zung zuschaltbare Hilfsmittel aufgeführt. Die Detektion von Kalibriermuster zur Laufzeit, wird durch Anwählen des Eintrags *LiveCornerDetection* (8) aktiviert, was sich je nach verwendeter Auflösung jedoch nachteilig auf die Performance des Systems auswirken kann. Die Anwahl des Eintrags *CroppingEditor* (8) ruft zwei Editorenfenster auf, die für die in der Dateiliste aktuell gewählte Aufnahme das Zuschneiden des Ausgangsbild auf das enthaltene Projektor- bzw. Kamerakalibermuster bieten, siehe Abbildung 7.7 (a). Durch das Fenster *MarkerComposite* (8) können detektierte Eckpunkte beider Muster zu Kontrollzwecken als Overlay auf dem Ausgangsbild visualisiert werden, siehe Abbildung 7.7 (c).

Die Einstellung von Parametern der zu detektierenden Kalibermuster, wie Anzahl von Eckpunkten in Höhe und Tiefe, sowie deren Ausdehnung als Millimeterangabe, kann zudem vorgenommen werden (5). Die im unteren Bereich dargestellten Symbole bewirken (v.l.n.r) den Zuschnitt im erwähnten *CroppingEditor* (8), die Re-Detektion von auf dem Bild befindlichen Kalibermuster, das manuelle Speichern und laden von Kalibermuster-Eckpunkten als XML-Dateien im durch *Project-File-Path* (1) festgelegten Verzeichnis und die Kalibrierung der Kamera. Analog dazu können die in (6) dargestellten Symbole für den Projektor benutzt werden, wobei zusätzlich die extrinsische Kalibrierung des Projektor-Kamera-Systems angeboten

wird.

Die intrinsische Kalibrierung der Geräte kann einzeln ausgeführt werden, während die extrinsische Kalibrierung des Projektor-Kamera-Systems nach Wahl einer hierzu zu verwendenden, als *alignment shot* (3) bezeichneten Aufnahme erfolgt. Nach intrinsischer Kalibrierung beider Geräte kann das Kalibrierungsmodul dazu genutzt werden lediglich eine extrinsische Kalibrierung des Gesamtaufbaus anhand dieser Aufnahme durchzuführen. Die automatische Kalibrierung des gesamten Systems wird durch die Schaltfläche *calibrate projector-camera system* (7) angestoßen, worauf die ermittelten Intrinsiken und die Extrinsik in der Sektion *calibration results* (9) angezeigt werden.

Weil es nicht auszuschließen ist, dass beide verwendeten Kalibermuster die gleichen Abmessungen haben, ist es von Nöten gewesen eine Trennung der betreffenden Bereiche im Kamerabild zu ermöglichen und die Muster somit eindeutig zum jeweiligen Gerät zuordnen zu können. Um die Kalibermuster komfortabel trennbar zu machen wurde eine Funktion zum Ausschneiden von Bildregionen, siehe hierzu Abbildung 7.7 (a) implementiert.



(a) Manuelles Zuschneiden akquirierter Bilder per frei aufziehbarer und verschiebbarerem Rechteck. Ausblendung von Bildbereichen die außerhalb der als Maske fungierenden Rechtecks liegen.

(b) Detailansicht der manuellen Korrektur von den im Bild automatisch detektierten Kalibermuster-Eckpunkten.

(c) Gesamtvisualisierung von detektierten Kalibermuster-Eckpunkten des oberen Kamera- und im unteren Bereich, des auf die gemeinsame Kaliberebene projizierten Projektor-Kalibermusters.

Abbildung 7.7: Manueller Zuschnitt, Eckpunktkorrektur und Gesamtvisualisierung von Kalibermustern durch das Modul VideoDeviceCalibration.

Hierbei werden markierte Regionen als separate Bilddatei abgespeichert. Auf diesen Bildausschnitten detektierte Eckpunkte werden als XML-Liste von zweidimensionalen Punkten abgespeichert. Führt man die Eckpunktterkennung auf separat gespeicherten Bildausschnitten aus, so liegen die Eckpunktkoordinaten relativ zu diesen Bilddimensionen vor. Hier wurde es nötig eine Umrechnung auf die Positionen im Ursprungsbild durchzuführen. Zur Kontrolle der für Bildausschnitte gespeicherten Eckpunkte wurde eine weitere Kontrollvisualisierung eingeführt, welche beide Eckpunktmenge auf einem Gesamtbild visualisiert, siehe hierzu Abbildung 7.7 (c).

Die eigentliche Kalibrierung unterteilt sich in die intrinsische Kalibrierung von Kamera und Projektor, sowie die extrinsische Kalibrierung des Projektor-Kamera-Systems. Die intrinsische Kalibrierung des Projektors setzt eine intrinsische Kalibrierung der Kamera voraus. Ermittelte Kalibrierwerte werden als Parameter des Moduls *VideoDeviceCalibration*, sowie als separate XML-Dateien gespeichert, die bei Bedarf geladen werden können.

Nach erfolgreicher Kalibrierung werden die ermittelten extrinsischen Werte, für eine erste, visuelle Kontrolle dieser Parameter, in Form einer virtuellen Aufbaus dargestellt, siehe Abbildung 7.8.

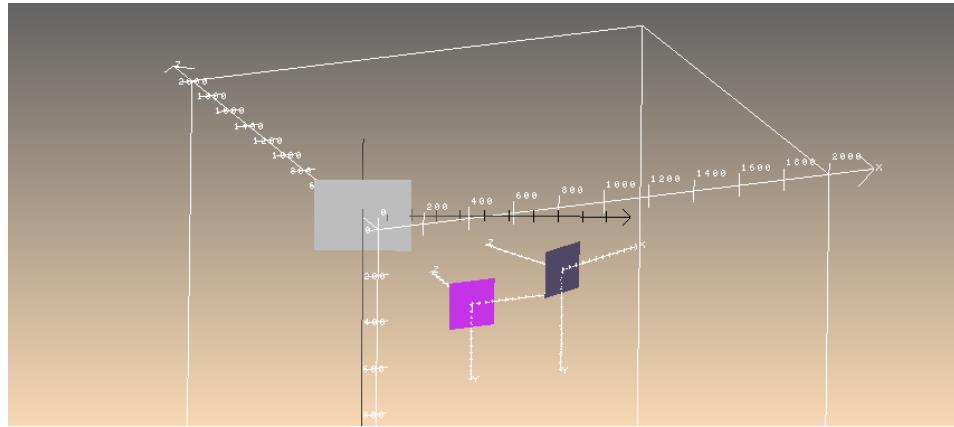


Abbildung 7.8: Visualisierung des Projektor-Kamera Aufbaus unter Benutzung der ermittelten extrinsischen Kalibrierung. Dargestellt sind die gemeinsame Kalibrierebene in grau, die Kamera in violettt (mittig) und der Projektor in lila, (rechts).

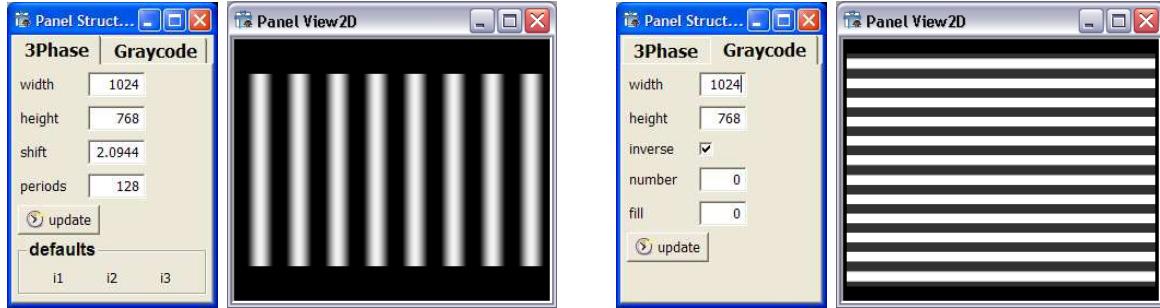
Hierbei wurden ermittelte extrinsische Werte des Moduls *VideoDeviceCalibration* zudem mit Parametern des zur Visualisierung erstellten Modulnetzwerkes verknüpft, sodass bei erneuter Kalibrierung automatisch ein visuelles Feedback erfolgte. Im praktischen Einsatz diente dies einer ersten, subjektiven Verifikation der Kalibrierung.

7.2.3 Rekonstruktionsmodul

Zur dreidimensionalen Oberflächenrekonstruktion wurde das Modul *StructuredLight* umgesetzt, welches einen Gray-Code- sowie Sinusphasen-Algorithmus implementiert und sich dabei an den von [Lanman u. Taubin, 2009] und [McDonald, 2009] vorgeschlagenen Verfahren orientiert. Im Folgenden wird zunächst das Modul *StrcturedLightPatternGenerator* beschreiben, welches zur Erzeugung von Projektionsmustern beider Verfahren dient. Daraufhin wird auf den Datenfluss und die Handhabung des Hauptmoduls *StructuredLight* eingegangen.

StructuredLightPatternGenerator Zur Generierung Das Modul *StructuredLightPatternGenerator* dient. Hierzu kann ein erzeugtes Muster über den Modulausgang beispielsweise auf

ein *View2D* Modul weitergeleitet werden und dem Projektor als Bildquelle dienen, oder aber zur späteren Verwendung lediglich abgespeichert werden. Für die Erzeugung von Sinusphasenbildern kann neben Bildhöhe und Breite, die zu verwendende Phasenverschiebung sowie Anzahl von Perioden festgelegt werden, siehe hierzu auch die Implementierung in Listing 2.



(a) Grafische Benutzeroberfläche zum Erzeugen von Projektionsmustern des Sinusphasen-Verfahrens (links). Vertikales Sinusphasenbild als Modulausgabe (rechts).

(b) Grafische Benutzeroberfläche zum Erzeugen von Gray-Code-Projektionsmustern (links). Horizontales Gray-Code-Muster als Modulausgabe (rechts).

Abbildung 7.9: Grafische Benutzeroberfläche und Beispieldaten des Moduls *StructuredLightPatternGenerator*.

Standardwerte für die Phase sind -120° , 0° und 120° , welche von der implementierten Sinusphasen-Rekonstruktion erwartet werden. Bei der Erzeugung von Gray-Code-Mustern bestimmt die gewählte Bildbreite *width* die Anzahl n erzeugbarer Muster mit $n = \log_2(\text{width}) \cdot 2$. Wird der in Abbildung 7.9 (b) gezeigte Parameter *number* im Bereich $0 \dots \log_2(\text{width})$ gewählt, so wird ein vertikales Gray-Code-Muster erzeugt, während im Bereich $\log_2(\text{width}) \dots \log_2(\text{width}) \cdot 2$ ein horizontales Muster erzeugt wird. Die Anwahl des Parameters *inverse* veranlasst eine Bildinvertierung des ausgegebenen Musters. Wird dem Parameter *fill* ein Wert ungleich -1 zugewiesen, so wird das Bild mit dem per *fill* angegebenen Grauwert gefüllt, der im Bereich $0 \dots 255$ liegen kann. Die uniforme Füllung des projizierten Bildes mit einem Grauwert ist für die Aufnahme einer globalen Illumination vor und für die Aufnahme einer Abschattung nach Projektion der Gray-Code-Sequenz vorgesehen.

StructuredLight Das Modul *StructuredLight* stellt die Implementierung der Rekonstruktion mittels Sinusphasen- und Gray-Code-Verfahren bereit. Wie in Abbildung 7.10 gezeigt, wird es im Rahmen des Makromoduls *StructuredLightMacro* eingesetzt und über seine Bildeingänge einerseits mit dem Videostream der Kamera und andererseits mit zu projizierenden Mustern versorgt. Die von den jeweiligen Verfahren rekonstruierte Oberfläche wird in Form von Punktwolken an die Ausgänge des *Structured-Light*-Moduls zur Nachbearbeitung durchgereicht.

Datenfluss Das Modul *Structured-Light* erwartet als Eingabe den von der Kamera kommenden Videostream, sowie das zu projizierende Muster. In Abbildung 7.10 ist das Durchreichen des Videostreams ins Makro-Modul *StructuredLightMacro* dargestellt. Innerhalb des Structured-Light-Moduls werden Aufnahmen, die zur Rekonstruktion verwendet werden von diesem Eingang abgenommen und im Dateisystem abgespeichert. Der zweite dargestellte Bildeingang wird vom Modul *StructuredLightPatternGenerator*, mit den für die Rekonstruktion benötigten, zu projizierenden Bildmustern bedient.

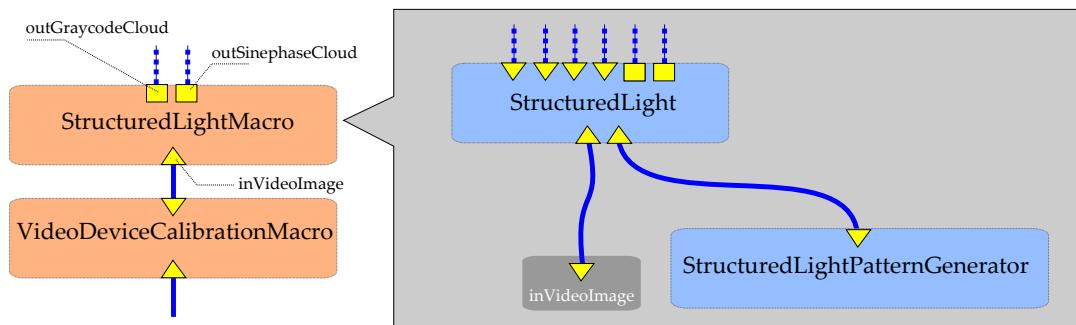
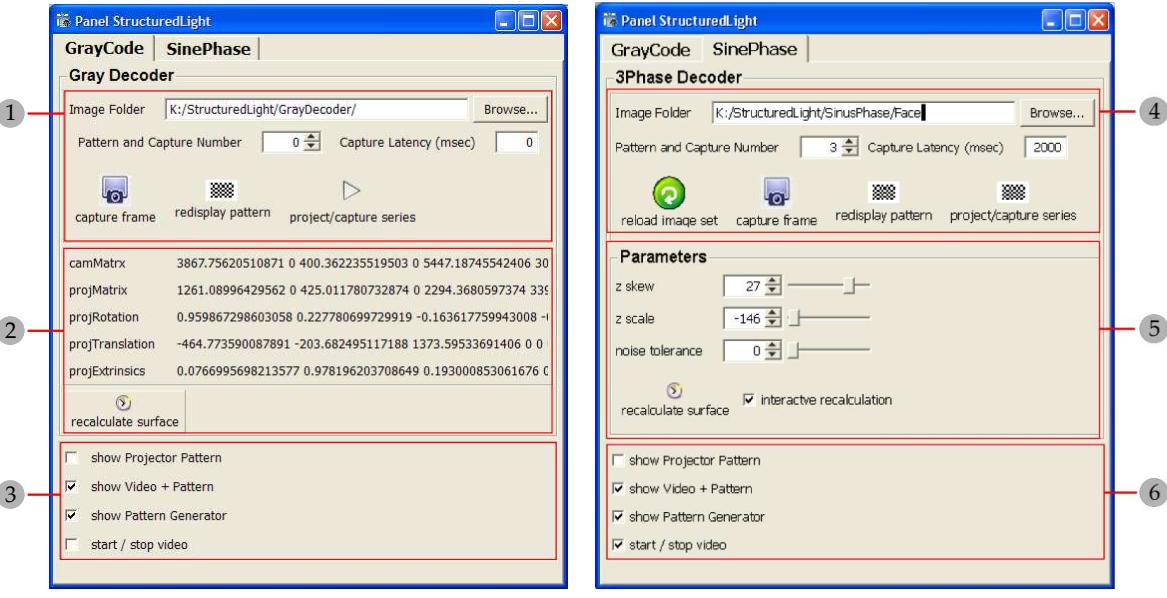


Abbildung 7.10: Verknüpfung von Ausgang und Eingang der Makromodule *VideoDeviceCalibration* und *StructuredLight* zur Durchreichung des Videostreams der Kamera. Eingabe des Videostreams und von dynamisch erzeugten Kalibriermuster in zwei Bildeingänge des Moduls *StructuredLight*. Ausgaben des Moduls sind das fürs Sinusphasen-Verfahren erzeugte Phasenbild, die durchgereichten Eingangsdaten und rekonstruierte Oberflächendaten in Form einer Punktwolke für das jeweilige Verfahren.

Gray-Code-Benutzung Der Prozess der Rekonstruktion ist im Fall des Gray-Code-Verfahrens nach erfolgter Kalibrierung vollautomatisch. Über die bereitgestellte Benutzungsoberfläche kann aber auch, wie im Folgenden gezeigt wird, eine manuelle Steuerung durchgeführt werden. Die in Abbildung 7.11 (a) dargestellte Benutzungsoberfläche des Gray-Code-Verfahrens gliedert sich in drei Hauptbereiche. Im oberen Drittel wird der Ordnerpfad zur Speicherung von zur Rekonstruktion verwendeten Aufnahmen im Dateisystem gesetzt (1).

Aufnahmen werden hierbei nach einem festen Schema als $gc[n].png$, mit n als laufende Nummer, welche dem aktuell gesetzten Parameter *Pattern and Capture Number* entspricht, abgespeichert. Der Parameter *Capture Latency* (1) ermöglicht eine Angabe von Millisekunden, die zwischen Aufnahmen gewartet werden soll, womit bei Bedarf eine Synchronisierung von Projektor und Kamera erreicht wird. Bevor das nächste Muster projiziert werden kann, besteht im Regelfall in MeVisLab nicht die Notwendigkeit, durch das Setzen einer Wartezeit auf die Abspeicherung des aktuellen Bildes zu warten. In Listing 3 wird durch die Verwendung der Befehle *processEvents* und *processInventorQueue* die Abarbeitung aller momentan in der MeVisLab-internen Event-Schleife enthaltenen Befehle veranlasst. Ohne Verwendung dieser Anweisungen ist es nicht



(a) Benutzungsoberfläche zum Gray-Code-Verfahren (b) Benutzungsoberfläche zum Sinusphasen-Verfahren

Abbildung 7.11: Grafische Benutzungsoberfläche des innerhalb des Moduls *StructuredLight* implementierten Gray-Code- und Sinusphasen-Verfahren.

garantiert, dass auf die Beendigung des Speicherprozesses gewartet wird bevor die in Zeile 6 begonnene Schleife erneut betreten und ein nächstes Muster zur Projektion gebracht wird.

Die Schaltfläche *capture frame* (1) dient zur manuellen Speicherung des aktuellen Kamerabildes, während *redisplay pattern* zur Erzeugung des durch *Pattern and Capture Number* (1) festgelegten Gray-Code Musters verwendet wird. Die vollautomatische Rekonstruktion wird über die Schaltfläche *project/capture series* (1) angestoßen. Nach Projektion und Aufnahme aller benötigten Aufnahmen wird hierbei die Hauptfunktion des Gray-Code-Verfahrens ausgeführt.

Im zweiten Drittel der Benutzungsoberfläche (2) sind die als Grundlage der Rekonstruktion dienenden Kalibrierungswerte aufgelistet. Diese Parameter werden durch direkte Parameterverbindungen vom Modul *VideoDeviceCalibration* bezogen. Um die Auswirkung von Feinjustierungen dieser Kalibrierwerte auf die Rekonstruktion testen zu können, ist eine manuelle Veränderung dieser Werte und die neue Rekonstruktion durch die Schaltfläche *recalculate surface* möglich.

Im unteren Drittel (3) sind weitere Werkzeuge aufgeführt. Zur Anzeige des aktuellen Projektions-Patterns dient die Schaltfläche *show Projector Pattern*. Dieses Fenster ist dazu gedacht in den Anzeigebereich des Projektors gezogen zu werden. Zur visuellen Kontrolle kann über die Schaltfläche *show Video + Pattern* ein *SynchroView2*-Modul zur Simultaaanzeige des Kamerabildes und Projektionsmusters aufgerufen werden. Die Anzeige der *StructuredLightPat-*

ternGenerator-Benutzungsoberfläche wird durch die Schaltfläche *show Pattern Generator*, das Starten sowie Anhalten des Videostreams der Kamera über *start/stop video* ausgelöst.

Gray-Code-Rekonstruktion Kernbestandteile der Implementierung des Gray-Code-Verfahrens sind die Funktionen *decodeGrayCodes* zur Zuordnung eines Kamerapixels zum entsprechenden Projektormpixel, sowie *reconstructScene* zur Rekonstruktion mittels Triangulierung. Die Umsetzung dieser beiden Funktionen wird im Folgenden kurz dargestellt.

Da die Projektionsreihenfolge festgelegt ist, kann zwischen Gray-Codes zur Identifikation der Projektorzeile- wie Spalte eindeutig unterschieden werden. Durch Beobachtung aller Kamerabildpixel werden somit beide benötigten Gray-Codes ermittelt. Um verschattete Pixel von der Rekonstruktion auszunehmen wird eine Maske angelegt die konstante, also nicht von Beleuchtung und Abschattungswechsel der projizierten Sequenz betroffen Pixel aufnimmt. Die anfängliche Aufnahme eines weißen bzw. schwarzen projizierten Bildes ermöglicht es gleichermaßen, verschattete Pixel zu detektieren, als auch ein Schwellwert für die Illumination eines Pixels fest zu legen. Um durch indirekte Beleuchtung hervorgerufenen Fehler in der Rekonstruktion zu vermeiden kann anstatt eines globalen Schwellwertes auch ein lokaler Schwellwert benutzt werden. Durch die zusätzliche Projektion der Inversen eines jeden Projektionsmusters wird die Rekonstruktion robuster gegenüber Streulicht. Ein vormals wegen Streulichts als beleuchtet angesehener Pixel wird im inversen Bild als ebenfalls beleuchtet erkannt, was folgerichtig dazu führt, dass er als ursprünglich unbeleuchtet angesehen wird.

Für alle Bildpixel werden nun Sehstrahlen von Kamera und Projektor errechnet und miteinander geschnitten. Da sich diese Sehstrahlen in einigen Fällen nicht schneiden, wurde zusätzlich der Schnitt von Kamerasehtstrahlen mit der Ebene einer Projektorspalte- bzw. Zeile im Raum durchgeführt.

Sinusphasen Benutzung Die in Abbildung 7.11 (b) dargestellte Benutzungsoberfläche des Sinusphasen-Verfahrens gliedert sich in drei Bereiche. Im ersten Drittel kann das Verzeichnis *Image Folder* (4) zur Zwischenspeicherung der drei nötigen Aufnahmen angegeben werden. Zum erneuten Laden der dort gespeicherten Bilder kann der Button *reload image set* benutzt werden. Über das Feld *Pattern and Capture Number* ist das zu projizierende Muster wählbar, der Button *redisplay pattern* erzeugt dieses und stellt es auf dem Modulausgang bereit. Über den Button *capture frame* kann eine manuelle Aufnahme des aktuellen Kamerabildes angefertigt werden, gespeichert wird diese als *sp/[Pattern and Capture Number].png*. Der Button *project/capture series* löst die automatische Projektion und Aufnahme aller benötigten Bilder aus. Bei Bedarf kann hierbei eine Latenz per *Capture Latency* eingestellt werden, um die Synchronisation von Projektor und Kamera manuell durchzuführen. Im zweiten Drittel (5) können

die Parameter der Rekonstruktion eingestellt werden, wobei die Drehung, Tiefenskalierung sowie Rauschtoleranz der gesamten Punktwolke gesteuert werden kann. Der Button *recalculate surface* veranlasst eine Neuberechnung der Oberflächenrekonstruktion, während diese auch interaktiv ablaufen kann, was durch Markieren des Feldes *interactive recalculation* geschieht. Im letzten Drittel (6) sind die zum Gray-Code-Verfahren analogen *tools* (3) aufgeführt.

Zusammenfassung Das System zur Oberflächenrekonstruktion mittels Structured-Light wurde als stufenweises Netzwerk, unter Benutzung voneinander unabhängigen Modulen umgesetzt. Der Datenfluss erfolgt Bottom-Up, was bedeutet, dass in einem ersten Schritt Videobilder vom *VideoCapture* Modul geliefert werden. Daraufhin folgt die Kalibrierung unter Benutzung dieser Videobilder durch das Modulnetzwerk *VideoDeviceCalibrationMacro*. Hierbei werden die zum manuellen Zuschneiden von Videobildern entwickelten Makromodule *camPatternCrop* und *projPatternCrop* verwendet. In einem letzten Schritt wird das Modulnetzwerk *StructuredLightMacro* mit der ermittelten Kalibrierung und dem laufenden Kamerabild versorgt. Am Ende der Verarbeitungskette stehen die Ausgänge des Moduls *StructuredLight*, welche durch das Sinusphasen- und Gray-Code-Verfahren rekonstruierte Punktwolken bereit stellen.

8 Validierung

In diesem Kapitel wird die Validierung der erstellten Oberflächenrekonstruktionsverfahren durchgeführt. Als erster Schritt werden Hypothesen aufgestellt, deren Wahrheitsgehalt es im weiteren Verlauf zu überprüfen gilt. Die dazu nötige Methodik und der konzipierte Versuchsablauf werden in einem nächsten Schritt eingeführt. Im anschließende Ergebniskapitel wird die ausführliche Präsentation und Besprechung der gewonnenen Daten vorgenommen.

8.1 Hypothesen

Die in Kapitel 2 ermittelten Kernpunkte können im weitesten Sinne als zu evaluierende Hypothesen verstanden werden. Zu überprüfen ist also, ob das Verfahren eine semi-automatische bis automatische Kalibrierung bietet, deren Zeitaufwand unter einer Minute bleibt. Daneben muss auch untersucht werden, ob die Kalibrierung akkurat ist. Weiterhin soll belegt werden, dass die rekonstruierte Oberfläche einen minimalen Abbildungsfehler im Vergleich zum Referenzobjekt aufweist.

Hypothese 1

Der Zeitaufwand für die Kalibrierung des Projektor-Kamera-Systems bewegt sich unterhalb der Minutengrenze.

Hypothese 2

Das Verfahren zur Kalibrierung des Projektor-Kamera-Systems liefert akkurate Ergebnisse.

Hypothese 3

Der bei der Rekonstruktion entstehende Abbildungsfehler ist in ausgewählten Testszenarien minimal.

8.2 Auswertungsmethodik

Um die soeben aufgestellten Hypothesen auswerten zu können, muss jeweils eine Methodik eingeführt werden, die sich dazu eignet diese qualitativ zu validieren.

Validierung der Kalibrierung Die Kalibrierungsprozedur besteht einerseits aus der Ermittlung der Intrinsik beider Geräte und andererseits aus der Bestimmung der Extrinsik. Durch mehrmalige Zeitmessung der praktischen Prozedur kann ein Mittelwert zum Zeitaufwand für die Gerätekalibrierung ermittelt werden. Im Fall der Intrinsik kann gesagt werden, dass bei deren Bestimmung der tendenziell größte Aufwand betrieben werden muss. Zentraler Punkt ist hierbei jedoch auch die Tatsache, dass die intrinsische Kalibrierung nur einmalig durchgeführt werden muss und erst dann wieder nötig wird, wenn intrinsische Parameter der Geräte modifiziert werden. Die Methode zur Aufwandsbestimmung ist hierbei die Messung und Mittlung der benötigten Zeitspanne. Die Extrinsik kann dagegen sehr schnell durchgeführt werden, da hierzu nur eine Aufnahme benötigt wird, auf der beide Kalibriermuster erkennbar sind. Auch hier wird der zeitliche Aufwand der praktischen Prozedur bestimmt und gemittelt.

Bewertung der Kalibrierungsgüte Die qualitative Auswertung der Kalibrierung wird durch Vergleich mit ermittelten Werten einer Referenzimplementierung erfolgen. Hierbei wird das von [Falcao u. a., 2008] vorgestellte und auf Arbeiten von [Bouquet, 2010, Bouquet] basierenden Framework zur Projektor-Kamera-Kalibrierung genutzt. Da die Güte der ermittelten Kalibrierwerte maßgeblichen Einfluss auf die Rekonstruktionsergebnisse hat, lassen Rekonstruktionsergebnisse, falls diese geometrisch stark verzerrt sind, auf einen Kalibrierungsfehler schließen.

Bewertung der Oberflächenrekonstruktion Um zu ermitteln wie die Bewertung der Rekonstruktion durchgeführt werden kann, müssen Parameter gefunden werden, die wesentlichen Einfluss auf die Ergebnisse haben. Neben der Art des Versuchsaufbaus sind dies vor allem die Oberflächeneigenschaften zu rekonstruierender Objekte und Umgebungsvariablen wie die Szenenbeleuchtung. Diese Parameter, wie auch die Stärke der Projektorluminanz und Kamerabelichtung müssen bei der Bewertung der Rekonstruktion zusätzlich beachtet werden, um Aussagen über Fehlerquellen zu ermöglichen. Die Güte einer rekonstruierten Oberfläche wird anhand des Abstandsmaßes bewertet, das im Vergleich zum dazugehörigen Referenzdatensatz gemessen wird.

8.3 Versuchsreihen

Es folgt die Beschreibung der Erfahrungswerte, die bei der Durchführung von Versuchen gesammelt und angewandt wurden. Um dabei eine hohe Aussagekraft zu gewährleisten, war es wichtig, dass ihr Ablauf aus überprüfbaren und wiederholbaren Schritten besteht. Die verwendete Parametrisierung wurde pro Versuch festgehalten. Hierbei wurde die ermittelte Kalibrierung, Nachbelichtungsdauer der Kamera und Helligkeit des Projektors, sowie Eigenschaften der Oberfläche notiert.

Wahl geeigneter Hardware Die Auswahl einer geeigneten Kombination von Projektor und Kamera war grundlegend für die Gewinnung akkurater Ergebnisse, sowohl im Hinblick auf die Kalibrierung als auch auf die Oberflächenrekonstruktion. Wird ein DLP-Projektor und eine zu kleine Belichtungszeit verwendet, so kann es vorkommen, dass ein Zwischenbild aufgenommen wird, aus dem der Projektor mittels Farbrad das eigentliche Bild kombiniert. Es empfiehlt sich hier einen DLP-Projektor ohne Farbrad zu benutzen, der die Synchronisation zwischen Projektor und Kamera vereinfacht, da nur voller Bilder projiziert werden. Um dieses Problem zu umgehen kann jedoch auch die Belichtungszeit der Kamera erhöht werden, beispielsweise für einen mit 60hz arbeitenden Projektor auf 1/60 Sekunde. Je nachdem wie weit entfernt sich die Oberfläche vom Projektor befindet, wird ein sehr lichtstarkes Gerät notwendig. Ein Kontrast von 1000:1 und eine Leistung von 2000 Ansi-Lumen erwiesen sich in einem gut abgedunkelten Raum allerdings als ausreichend. Optimal ist zudem ein Projektor mit minimaler Verzerrung, oder aber eine per Trapezkorrektur-Funktion einstellbare Projektion. Die Auflösung der Kamera muss mindestens so groß sein wie die des Projektors, um eine Beschränkung der Rekonstruktionsauflösung auf die der Kamera zu vermeiden. Der Detailreichtum der rekonstruierten Oberfläche wurde mit höherer Auflösung der Kamera zudem gesteigert. Eine Kamera, die Daten im RAW-Modus liefert ist zudem vorteilhaft, da Pixel nicht durch die in der Kamera ablaufende Kompression und somit entstehende Artefakte verfälscht werden. Die Bildwiederholrate der Kamera sollte zudem der des Projektors, oder eines Vielfachen entsprechen, um eine korrekte Synchronisation zu ermöglichen. Kamera und Projektor sollten stets mittels Stativ zum Einsatz kommen, um der versehentlichen Veränderung ihrer extrinsischen Parameter vorzubeugen. Eingesetzte Hardwarekomponenten sind in Tabelle 8.1 aufgeführt.

Gerät	Modell	Technische Werte
Projektor	Toshiba x200 LCD	max. 1600x1200, 3000 lm, 600:1
Kamera	Logitech QuickCam Pro 9000	RAW, max. 3264x2448
Computer	Dell XPS 420	Intel Core 2 Duo 3.16Ghz, 2GB RAM

Tabelle 8.1: Hardwarekomponenten der eingesetzten Testumgebung

Vorbereitung und Aufbau Im Rahmen der Vorbereitung eines Versuchs war die Einstellung des Umgebungslichts ein wichtiger Punkt. Für die Oberflächenrekonstruktion per strukturiertem Licht ist es je nach Leistung des Projektors von Vorteil, jegliches Umgebungslicht zu vermeiden und somit den Kontrast des projizierten Musters auf der Objektoberfläche zu steigern. Für die Rekonstruktion mittels Sinusphasen-Verfahren ist es wichtig, dass das projizierte Graustufenspektrum nicht vom Projektor beschnitten wird, da alle Helligkeitsabstufungen unterscheidbar sein müssen. Mittels Projektion eines Graustufen-Testbildes sollte verifiziert werden, dass dies nicht der Fall ist und bei Bedarf muss die Helligkeit, oder aber der Kontrast manuell geregelt werden. Im Kamerabild muss überprüft werden, ob Weißabstufungen auch als solche erkennbar sind, wozu die Farbbebalance oder Belichtungszeit eingestellt werden. Ist dies erfolglos, so kann ein Neutraldichte- bzw. Graufilter auf die Kameralinse montiert werden, um eine gleichmäßige Abdunklung im Bild zu erzielen. Um Unterschiede zwischen Weißtönen besser sichtbar zu machen, reicht es auch aus, sich das Lichtabfallgesetz zu Nutzen zu machen und die Szene weiter in den Hintergrund zu rücken. Um unbeabsichtigte Änderungen an einer vorgenommenen Konfiguration zu verhindern, wurden alle Automatismen von Projektor und Kamera zudem deaktiviert.

Beim daraufhin erfolgenden Aufbau von Projektor und Kamera war zu beachten, dass eine Rekonstruktion erleichtert wird, wenn die Blickrichtung der Geräte nicht koplanar ist, sondern ein spitzer Winkel zueinander besteht. Die Kamera sollte, um Streifen des projizierten Musters besser voneinander unterscheidbar zu machen, im besten Fall ihr volles Bild mit der Projektion ausfüllen. Die zu rekonstruierende Oberfläche sollte das Kamerabild zu einem Großteil bedecken und sich zudem im Schärfebereich des Projektors befinden. Verdeckungen, Selbstverschattungen sowie Glanzlichter machen eine Tiefenrekonstruktion für betroffene Bildpixel unmöglich, da keine Binärcodes aufgezeichnet werden können. Nach Möglichkeit galt es die Objektplatzierung solchermaßen vorzunehmen, dass diese Störfaktoren minimal gehalten wurden. In Abbildung 8.1 sind nochmals die für die Oberflächenrekonstruktion als entscheidend anzusehenden Aspekte der Vorbereitungsphase aufgezeigt. Um die Rekonstruktion vieler Oberflächendetails zu gewährleisten, war vor allem die Überdeckung des Objekts mit dem projizierten Muster und eine kurze Distanz zur Kamera vorteilhaft.

Durchführung In Abbildung 8.2 sind die Hauptschritte eines typischen Testablaufs festgehalten. Zusätzlich zur Objektposition und Orientierung wurden alle für den Testdurchgang verwendeten Parameter manuell notiert, die Kalibrierwerte wurden zusätzlich automatisch abgespeichert. Dies ermöglichte einen nachträglichen Vergleich von verschiedenen Konfigurationen des Systems in Hinblick auf erzielte Ergebnisse. Die Intrinsik der Kamera wurde in einem ersten Schritt anhand von Aufnahmen des Kalibermusters bestimmt, wobei das Muster jeweils formatfüllend sein sollte. Zur Bestimmung der Intrinsik des Projektors sind Aufnahmen beider

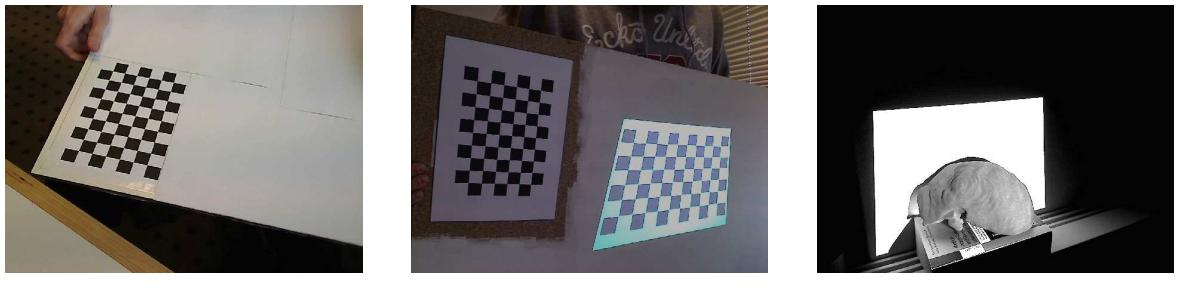


(a) Versuchsaufbau mit Projektor-Kamera-System.
 (b) Durchgeführte Aufbauten mit spitzem Winkel zwischen Kamera und Projektor.
 (c) Zu rekonstruierende Oberfläche im Fokus der Kamera.

Abbildung 8.1: Für Versuche eingesetzter Geräteaufbau.

Kalibermuster nötig, die Eckpunkte des Projektor-Kalibermusters sollten die gesamte projizierte Bildfläche ausfüllen, um auch die Randbereiche der Linse intrinsisch annähern zu können. Bei der Benutzung von zwei getrennten Kalibermustern, wie in Abschnitt 4.2 beschrieben, musste weiterhin Folgendes bedacht werden. Eine an den Geräten eingestellte Intrinsik, darf zwischen Kalibrierung und Oberflächenrekonstruktion nicht verändert werden. Um jedoch die Projektintrinsik gut kalibrieren zu können musste ein gewisser Abstand des Projektors von der Kalibrierebene geschaffen werden. Die Kamera musste das projizierte Muster dabei groß genug aufnehmen um eine automatische Detektion zu gewährleisten. Die Ebene zur intrinsischen Kalibrierung musste also eine ausreichende Größe haben, während für die Messung der Extrinsik ein kleinere Version zum Einsatz kommen konnte. Die Schärfeebenen von Kamera und Projektor waren somit festgelegt. Sollte nun jedoch ein Objekt in Nähe der Kamera rekonstruiert werden, so galt es zu beachten, dass die Schärfeebene womöglich hinter dem Objekt lag. Der Einsatz der in [Lanman u. Taubin, 2009] verwendeten, gemeinsamen Kalibrierebene, sollte sich für die Rekonstruktion von nah an der Kamera befindliche Objekte somit besser eignen, was jedoch nicht untersucht wurde.

Die während des Versuchs verwendete Gerätepositionierung wurde durch die Aufnahme bei der Kalibermuster auf der gemeinsamen Kalibrierebene errechnet, siehe Abbildung 8.2 (b). Um das Kameramuster robust detektieren zu können musste die Ebene im Schärfebereich der Kamera liegen, der Projektor musste einen ähnlichen Schärfebereich besitzen, damit das projizierte Muster ebenfalls erkannt wurde. Daraufhin wurde das zu rekonstruierende Objekt in wiederum diesen Schärfebereich positioniert und zwar so, dass es möglichst viel Platz im Kamerabild einnahm und die feinsten projizierten Gray-Code Muster voneinander unterschieden werden konnten. Während der Projektion von Mustern mussten Beleuchtung, die Positionen des Objekts, der Kamera und des Projektors, sowie die Geräteintrinsiken unverändert bleiben. Der nach Verarbeitung der aufgenommenen Bilder rekonstruierte Oberflächendatensatz wurde im Anschluss visualisiert oder weiter verarbeitet.



(a) Erstellung der zur intrinsischen Kalibrierung benötigten Kameraaufnahmen und Zeitmessung.
 (b) Erstellung von Aufnahmen zur extrinsischen Kalibrierung und Zeitmessung.
 (c) Rekonstruktion unter wechselnder Parametrisierung des Versuchsaufbaus.

Abbildung 8.2: Ablauf eines Versuchs.

Da das Sinusphasen-Verfahren ohne Kalibrierung arbeitet, mussten dessen Parameter in einem iterativen Prozess angeglichen werden, um korrekte Oberflächenrekonstruktionen zu ermöglichen. Die im Realisierungskapitel vorgestellten Parameter zur Schiefstellung und Tiefenskalierung, sind von der extrinsischen Konfiguration des Projektor-Kamera-Systems, d.h. von Abstand und Winkel der Geräte zueinander, abhängig. Um durch unzureichende Beleuchtung der Szene durch den Projektor oder durch Umgebungslicht hervorgerufene Artefakte zu kompensieren musste zudem ein Wert für die Rauschschwelle gefunden werden. Diese Schwelle kann angehoben werden um zu dunkel Bereiche auszuschließen. Daraufhin wurde mit der Tiefenskalierung experimentiert, bis die Oberflächenpunkte eine gewisse Tiefe erreicht hatten. Die Ermittlung des Parameters zur Schiefstellung erfolgte zum Schluss. Hierbei wurde der Winkel der Geräte zueinander manuell aneinander angenähert und subjektiv bestimmt.

Analyse Die gewonnenen, intrinsischen Kalibrierwerte wurden gegen gemessene Referenzwerte verglichen, die extrinsischen Werte wurde zusätzlich durch eine Visualisierung subjektiv geprüft. Vor allem die Qualität gewonnener Rekonstruktionen war ausschlaggebend für die Durchführung einer erneuten Kalibrierung. Neben der reinen Sichtprüfung von rekonstruierten Oberflächen wurde ein quantitativer Vergleich auf Basis von Referenzdaten durchgeführt. Im Fall der im Konzeptkapitel auf Seite 39 vorgestellten Lebermodelle wurde die rekonstruierte Punktfolge gegen den jeweiligen Referenzdatensatz registriert und ein Distanzmaß errechnet. Hierzu wurde zuerst das MeVisLab-Modul *PointCloudFilter* eingesetzt, um Rauschen aus dem Datensatz zu eliminieren. Dabei wurde pro rekonstruiertem Voxel eine Nachbarschaftsanalyse durchgeführt und Voxel, welche wenige Nachbarn in der näheren Umgebung besaßen eliminiert, siehe 9.10 (c). Bei der Abstandsmessung zu Referenzdatensätzen der Phantommodellen wurde die rekonstruierte Oberfläche er manuell registriert, d.h. auf die Referenzoberfläche gelegt und anschließend durch ein Iterative-Closest-Point-Verfahren (ICP) präziser aufeinander registriert. Daraufhin wurde eine automatische Messung des mittleren Abstands beider Punktfolgen durchgeführt.

9 Ergebnisse

In diesem Kapitel werden erreichte Ergebnisse der Kalibrierung und Oberflächenrekonstruktion exemplarisch präsentiert und analysiert. Im anschließenden Diskussionskapitel wird das Verfahren bewertet und ein Ausblick auf mögliche Weiterentwicklungen gegeben.

9.1 Kalibrierung des Projektor-Kamera-Systems

Die Güte der Kalibrierung wurde im Rahmen eines Back-To-Back-Tests mit den ermittelten Werten des von [Falcao u. a., 2008] vorgestellten Kalibrierungsframework verglichen. Es folgen die Beobachtungen und Ergebnisse zur intrinsischen und extrinsischen Kalibrierung.

Intrinsische Kalibrierung der Kamera Bei der Kalibrierung der Kameraintrinsik hat es sich als sinnvoll erwiesen einige Ansichten des Musters formatfüllend aufzunehmen. Somit kann beispielsweise die Linsenverzerrung über das gesamte Kamerabild besser angenähert werden. Eine sorgfältig durchgeführte Kalibrierung macht eine gewisse Zeitaufwendung nötig. Ein Vorteil der dies wieder ausgleicht, ist aber die Tatsache, dass die intrinsische Kalibrierung nur bei Änderung der Kameraoptik durchgeführt werden muss. Dies ist im Laufe einer Versuchsreihe meist nicht notwendig gewesen, bzw. wurde nicht weiter evaluiert.

Die Zeitmessung der intrinsischen Kamerakalibrierung wurde auf zwei unterschiedliche Arten durchgeführt. Um die Aufnahmen der notwendigen Muster durchzuführen wurde eine manuelle sowie automatische Methode prototypisch getestet. Bei beiden Methoden erhält der Benutzer visuelles Feedback über das im Videobild jeweils detektierte Muster. Im Fall der manuellen Methode bleibt es dem Nutzer überlassen eine Aufnahme durch Interaktion mit der Benutzungsoberfläche anzufertigen, während bei der automatischen Variante eine festgelegte Anzahl von Aufnahmen in festen Zeitabständen durchgeführt werden. Es hat sich in Tests erwiesen, dass für die Kamerakalibrierung in der Regel 8 bis 12 Aufnahmen ausreichend sind und sich die intrinsische Matrix nur noch marginal ändert. Bei manueller Kontrolle entstehen in der Regel zudem keine „Fehlaufnahmen“ - also Aufnahmen, auf denen das Muster zwar detektiert wird, aber womöglich zu stark verzerrt ist und sich negativ auf die Kalibrierung auswirkt. Dies kann hingegen beim automatischen Kalibriervorgang eintreten, durch den manuellen Ausschluss aus

der Liste der zur Kalibrierung zu nutzenden Aufnahmen kann dies jedoch korrigiert werden. Bei der Kalibrierung musste darauf geachtet werden, dass das Kalibriermuster keiner direkten Beleuchtung oder Punktlichtquellen ausgesetzt war. Dieser Störfaktor verhinderte in einer Reihe erster Versuche die Detektion des Kalibriermusters wegen dem Auftreten von Glanzlichtern. In Abbildung 9.1 sind die zur intrinsischen Kamerakalibrierung verwendeten Aufnahmen dargestellt. In Tabelle 9.1 ist die Zeitmessung der manuellen und automatischen Kalibrierung der Kameraintrinsik aufgeführt.

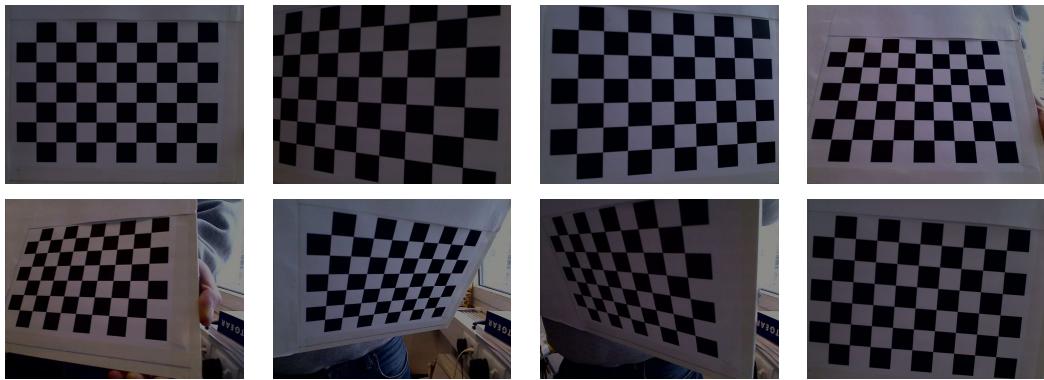


Abbildung 9.1: Zur Ermittlung der Kameraintrinsik eingesetzte, beispielhafte Aufnahmen.

Methode	Aufnahmen	Zeitaufwand	Mittlerer Zeitaufwand
Automatisch	8	8 s	14,6 s
	12	12 s	
	24	24 s	
Manuell	8	13 s	24 s
	12	25 s	
	24	34 s	

Tabelle 9.1: Exemplarische Zeitmessung intrinsischer Kamerakalibrierung

Der Zeitbedarf der Kalibrierung ist beim manuellen Verfahren erwartungsgemäß höher als bei der automatischen Methode, da der Anwender eigenständig über die Aufnahme von Bildern entscheiden muss. Auch wenn der geringe Zeitbedarf im ersten Moment vorteilhaft erscheint, so kann die gegebenenfalls nötige, manuelle Aussortierung von nicht verwendbaren Aufnahmen diesen Vorteil wieder aufheben. Zudem wurden bei der manuellen Durchführung der Kalibrierung kleinere Varianzen der Kameramatrix festgestellt, was auf eine größere Nähe der Werte zur tatsächlichen Intrinsik hindeutet. Die gewonnene Kameramatrix wurde nun mit den vom Referenzsystem ermittelten Werten verglichen, wobei die identischen Bilder als Grundlage dienten. Hierbei ergab sich keine wesentliche Abweichung der eigens ermittelten Kameramatrix von den Referenzwerten. In den Listings 9.1 und 9.2 sind beide Kameramatrizen aufgeführt.

```
Focal Length:   fc = [ 1300.10229    1302.54017 ] +- [ 8.81516    8.65624 ]
Principal point: cc = [ 785.53361    564.13498 ] +- [ 6.53438    5.55149 ]
Distortion:      kc = [ 0.06125    -0.14821    -0.00077    0.00102    0.00000 ]
                  +- [ 0.01394    0.03566    0.00131    0.00171    0.00000 ]
Pixel error:     err = [ 1.05030    0.82950 ]
```

Listing 9.1: Intrinsische Kameramatrix (Referenzsystem)

```
1321.642 0.000    745.800      | [ fx 0 cx ]  Fokallänge (fx,fy),
m: 0.000    1337.871 568.697    | [ 0 fy cy ]  Prinzipalpunkt (cx,cy)
0.000    0.000    1.000        | [ 0 0 1 ]
d: 0.024 -0.084 0.008 0.006 0.000 | Distorsionskoeffizient
```

Listing 9.2: Intrinsische Kameramatrix und Distorsion (per OpenCV)

Intrinsische Kalibrierung des Projektors Die Intrinsik des Projektors kann, wie bereits in Kapitel 4 erläutert, nur ermittelt werden, wenn beide Kalibriermuster im Kamerabild erkenn- und detektierbar sind. Die Bestimmung einer validen Projektorintrinsik hat auf praktischer Seite der durchgeführten Versuche, die weitaus größten Umstände bereitet. Weil hierfür anfangs eine mit $1m^2$ relativ unhandliche Spanplatte als Kalibrierebene zum Einsatz kam, wurden zwei Personen zur Durchführung unabdingbar. Durch die Größe der Kalibrierebene wurde zusätzlich ein größerer Abstand vom Projektor-Kamera-Aufbau nötig, als es bei der intrinsischen Kamera-kalibrierung der Fall war. Zudem musste die Beleuchtung des Kamerakalibriermusters, mittels Ausleuchtung durch einen Lampenschirm unterstützt werden, um das Muster überhaupt detektierbar zu machen. Da dem Ziel der Arbeit jedoch die Rekonstruktion von Nahaufnahmen der Leberoberfläche dienlicher sind, wurde schließlich eine kleinere, in Abbildung 9.2 dargestellte Kalibrierebene benutzt, was die intrinsische Kalibrierung des Projektors erleichterte.

Der Zeitaufwand der intrinsischen Projektorkalibrierung ist in der Regel höher als die der Kamera. Zur Trennung von Kamera- und Projektormuster in aufgenommenen Bildern wurde der in Abschnitt 7.2.2 beschriebene, manuelle Zuschnitt der Aufnahmen durchgeführt. Bei einer kleinen Anzahl von Aufnahmen bewegt sich diese manuelle Aktion zeitlich noch im Rahmen der Zielvorgaben, die darin besteht für die Kalibrierung des Projektor-Kamera-Systems die Minutengrenze nicht zu überschreiten. Eine automatische Kalibrierung anstatt der manuellen Trennung der Kalibriermuster wäre auch denkbar, jedoch nur solange die Kalibriermuster verschiedene Dimensionen besitzen, da sie eindeutig voneinander unterscheidbar sein müssen. Dieses Vorgehen sollte die Kalibrierungsgeschwindigkeit theoretisch auf die der Kameraintrinsik angleichen.

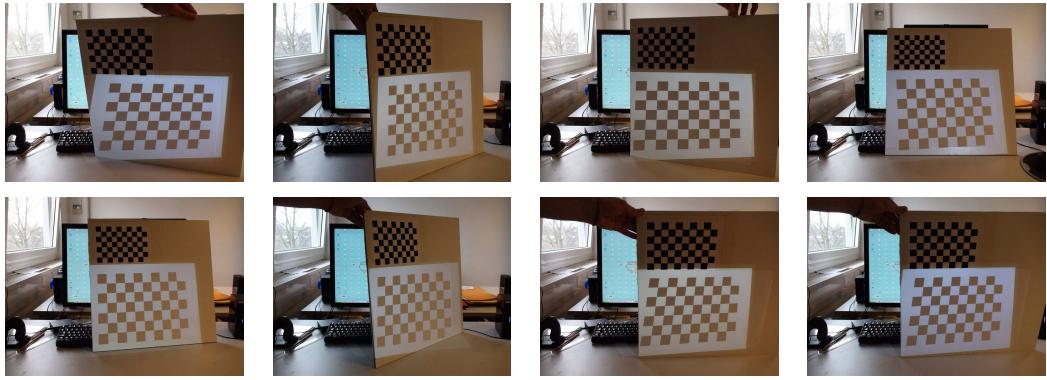


Abbildung 9.2: Zur Ermittlung der Projektorintrinsik eingesetzte, exemplarische Aufnahmen.

Methode	Aufnahmen	Zeitaufwand	Mittlerer Zeitaufwand
Manueller Zuschnitt der Kalibriermuster (Kamera und Projektor)	8 12 24	31 s 45 s 90 s	55,3 s

Tabelle 9.2: Exemplarische Zeitmessung intrinsischer Projektorkalibrierung.

Die gemessene, intrinsische Projektormatrix wurde wiederum mit den Werten der Referenzimplementierung verglichen. Hierbei ergab sich ebenfalls eine nur unwesentliche Abweichung. In einigen anderen Kalibrierungsversuchen und unter Verwendung ähnlicher Aufnahmen wurden jedoch erhebliche Abweichungen beobachtet. Die Verwendung von optimalen Aufnahmen, in denen das Projektormuster nicht zu großer Verzerrung durch starke Schiefstellung der Kalibrierbene unterliegt, scheint von zentraler Bedeutung für den in OpenCV verwendeten Algorithmus zur Kantenerkennung und Kalibrierung zu sein.

```
Focal Length:      fc = [ 1961.73940   1937.45620 ] +- [ 16.54801   17.43452 ]
Principal point: cc = [ 706.96945   780.38488 ] +- [ 41.40994   31.59107 ]
Distortion:       kc = [ -0.13596   0.22474   -0.00067   0.00541   0.00000 ]
                  +- [ 0.06954   0.23816   0.00775   0.00489   0.00000 ]
Pixel error:      err = [ 2.95561   1.42785 ]
```

Listing 9.3: Intrinsische Projektormatrix (Referenzsystem).

```
1984.598 0.000    760.778      | [ fx 0 cx ]  Fokallänge (fx,fy),
m: 0.000     1976.615 715.679      | [ 0 fy cy ]  Prinzipalpunkt (cx,cy)
0.000     0.000     1.000      | [ 0 0 1 ]
d: -0.157 0.327 -0.003 0.008 0.000 | Distorsionskoeffizient
```

Listing 9.4: Intrinsische Projektormatrix und Distorsion.

Extrinsische Kalibrierung Die Bestimmung der Geräteextrinsik konnte in der Praxis sehr schnell durchgeführt werden, da hierzu nur eine Aufnahme benötigt wurde, auf der beide Kalibriermuster erkennbar sind. Im Zuge der intrinsischen Projektorkalibrierung kann hierzu ein beliebige Aufnahme festgelegt werden. Die Durchführung einer Neuaufnahme und Kalibrierung per Zuschnitt nimmt dabei weniger als 20 Sekunden in Anspruch, was vertretbar ist. Die Extrinsik des Projektors wurde ebenfalls mit Referenzwerten verglichen und erwies sich als hinreichend genau. In Listing 9.5 sind extrinsische Daten des Projektors vergleichend aufgeführt.

0.9950	-0.0069	0.0995		1.2410	-0.0145	0.1053	
R:	0.0109	0.9991	-0.0397		0.0113	1.143	-0.0587
	-0.0991	0.0406	0.9942		-0.1171	0.060	1.1378
t:	-386.071	199.661	1381.07		-410.103	207.988	1401.00

Listing 9.5: Referenzkalibrierung (links) und per OpenCV ermittelte Extrinsik des Projektors (rechts)

Es kann abschließend festgehalten werden, dass die Genauigkeit der intrinsischen wie extrinsischen Kalibrierung des Projektor-Kamera-Systems im direkten Vergleich zum Referenzsystem gut abschneiden konnte. Die Kalibrierungsprozedur kann zudem bei der Verwendung weniger Aufnahmen durchaus unterhalb, der zum Ziel gesetzten Minutengrenze bleiben, was in einer Reihe von Versuchen auch gelang. Da eine sorgfältige Kalibrierung jedoch maßgebend für die Rekonstruktionsqualität ist, kann die Überschreitung dieses Grenzwertes als gerechtfertigt angesehen werden.

9.2 Oberflächenrekonstruktion

Die Güte der Rekonstruktion per Gray-Code-Verfahren wurde qualitativ an den beiden, in Abschnitt 6.3 vorgestellten Leberphantomen L_A und L_B , einem Kugelphantom, sowie anhand einer Schweineleber validiert (Abbildung 6.3). Das Sinusphasen-Verfahren hingegen wurde experimentell an Referenzdatensätzen getestet. Gewonnene Ergebnisse werden nun exemplarisch vorgestellt und diskutiert.

Initiale Tests Zu Anfang der Versuche, wurden initiale Tests durchgeführt, um die grundätzliche Funktionalität des Gray-Code- sowie Sinusphasenverfahrens sicher zu stellen . In Abbildungen 9.3 und 9.4 ist die Rekonstruktion eines vermessenen Kugelphantoms und das Ergebnis des Sinusphasenverfahrens auf Basis von Referenzdaten dargestellt. Das Sinusphasenverfahren wurde auf Referenzdatensätzen oder aber synthetischen Daten erfolgreich getestet, siehe Abbildung 9.4.

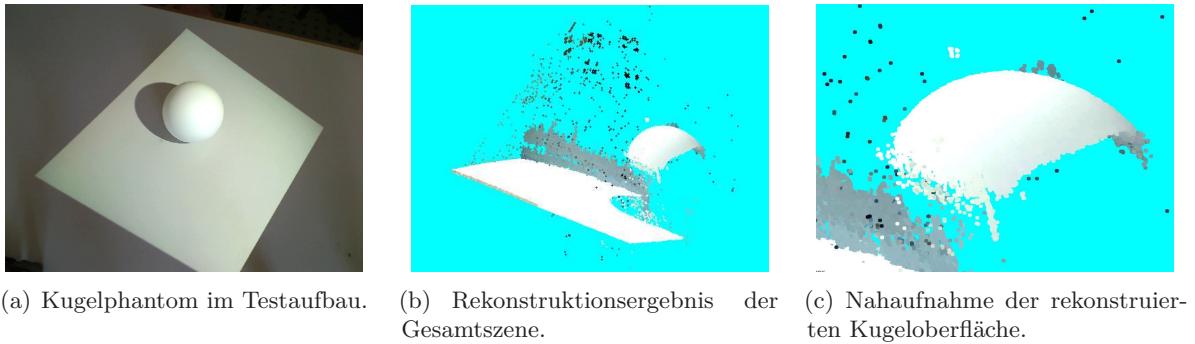


Abbildung 9.3: Initiale Rekonstruktionsergebnisse (Phantommodell Kugel).

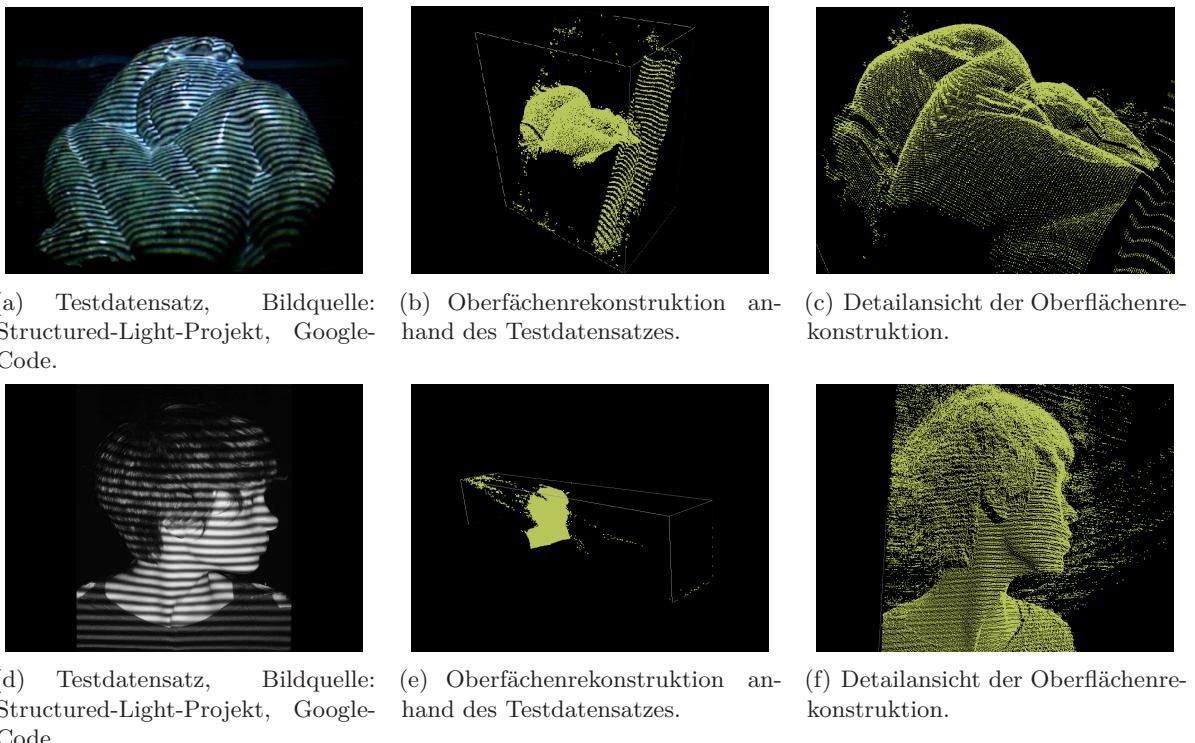


Abbildung 9.4: Rekonstruktionsergebnisse des Sinusphasen-Verfahrens (Referenzdatensätze)

In Abbildung 9.5 ist die Oberflächenrekonstruktion einer Schweineleber mit unbekannter Geometrie dargestellt. Hierbei galt es unter realistischen Bedingungen und mit einer, der menschlichen Leber sehr ähnlichen Organoberfläche, eine erste Aussage zur potentiellen Eignung des Verfahrens für die CL zu machen. Das Umgebungslicht wurde stark gedimmt um die Deaktivierung von OP-Lampen zu simulieren.

Bei diesem initialen Test des Gray-Code-Verfahrens, liegt einerseits eine unvorteilhafte Kameraspositionierung vor, was in einer fast vertikal projizierten, die Oberfläche unzureichend charakterisierenden Streifenprojektion resultiert, siehe Abbildung 9.5 (b). Das rekonstruierte

Tiefenbild enthält folglich eine sehr kleine Varianz. Zusätzlich sind bei diesem speziellen Testlauf die Kontrastverstärkung der Kamera sowie die Lichtstärke des Projektors stark übersteuert. Die Rekonstruktion leidet, wie es zu erwarten war aber vielmehr an den von der viskosen Oberfläche erzeugten Glanzlichtern.

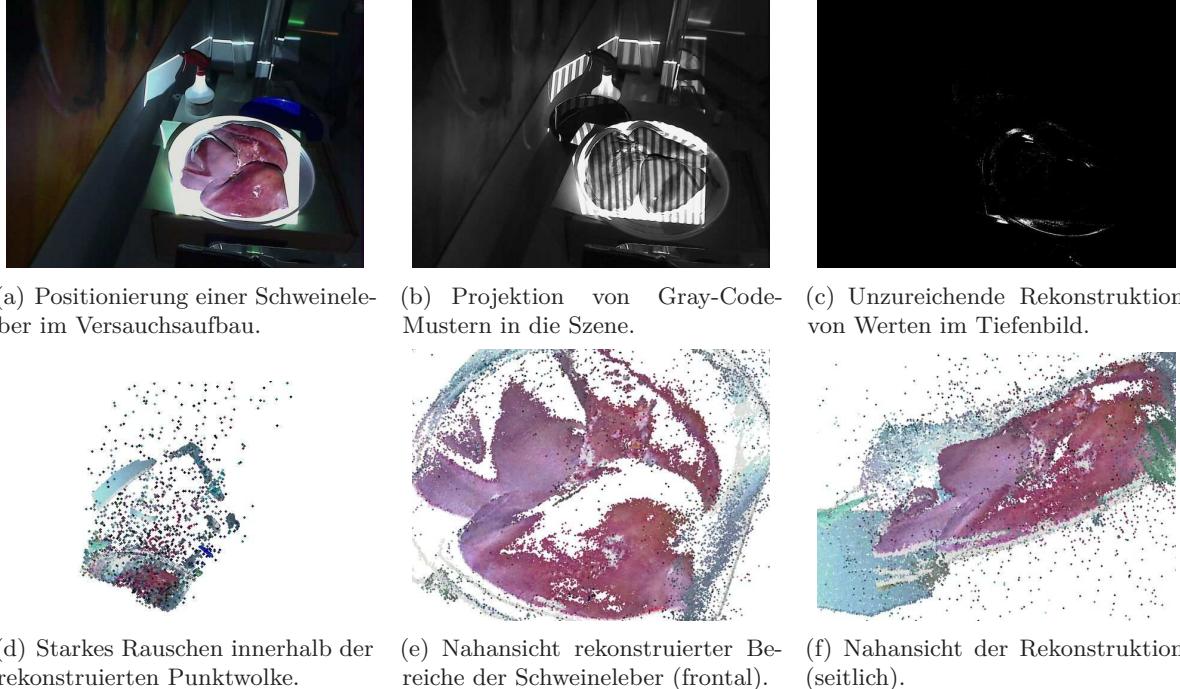


Abbildung 9.5: Rekonstruktionsergebnisse (Schweineleber)

Qualitative Tests Nach initialen Tests der Verfahren, wurde das Gray-Code-Verfahren anhand von Phantommodellen eingehender untersucht. In Abbildung 9.6 ist die Platzierung des Phantommodells L_A , sowie ein Schritt der horizontalen und vertikalen Streifenprojektion dargestellt. Da der eingesetzte Projektor sehr lichtstark war, konnte in diesem Fall mit einem relativ hohen Umgebungslicht gearbeitet werden. Entscheidend war bei diesem Versuch vielmehr die durchgeführte Objektplatzierung. Projizierte Streifenmuster, welche die Objektwölbungen widerspiegeln stellten sich als vorteilhaft für die Rekonstruktionsgüte heraus. Traf diese Bedingung nur unzureichend zu, so wiesen rekonstruierte Tiefenwerte keine hohe Varianz auf, was sich in einer relativ flachen, ebenenförmigen Rekonstruktion äußerte.

Die zu den Bildpixelen der Kamera gehörigen Zeilen und Spalten des Projektors, sowie das nach der Triangulierung entstehende Tiefenbild ist in Abbildung 9.7 dargestellt. Hierbei ist bereits erkennbar, dass im Tiefenbild ein breites Spektrum an Grauwerten vorhanden ist. Da die tatsächlich rekonstruierten Raumpunkte eine Fließkommagenaugigkeit besitzen, ist die Qualität der Oberflächenrekonstruktion jedoch erst anhand einer dreidimensionalen Visualisierung genau genug zu beurteilen.



Abbildung 9.6: Projektion von Gray-Code-Mustern auf das Phantommodell L_A .



Abbildung 9.7: Visualisierung der Korrespondenzen zu Projektorzeilen bzw. Spalten sowie rekonstruierte Tiefenwerte des Phantommodells L_A .

In Abbildung 9.8 ist die gewonnene, ungefilterte Oberfläche aus verschiedenen Blickwinkeln dargestellt. Die rekonstruierte Szene wurde von einer vorher erstellten Rekonstruktion der leeren Szene abgezogen, um somit den Objekthintergrund weitestgehend zu eliminieren. Zudem wurde die Texturierung eines Oberflächenpunktes mit zum entsprechenden Kamera-Sehstrahl gehörigen Kamerabildpunkt durchgeführt. Bestehende Artefakte wie Bereiche der Hintergrundwand und der Objektauflage, die in Abbildung 9.8 (a) zu sehen sind, röhren von einer minimalen Verschiebung dieser beweglichen Teile zwischen der Rekonstruktion der leeren Szene und der eigentlichen Oberflächenrekonstruktion her.

In Abbildung 9.8 ist die generelle Plastizität der Oberfläche bereits erkennbar, verschattete Bereiche im unteren Teil sind durch einfallendes Streulicht dabei relativ gut rekonstruiert worden. Gleichzeitig werden jedoch auch die in Abschnitt 6.3 angesprochenen Grenzen der Rekonstruktion von Einwölbungen deutlich. Da die Zeilen und Spalten des Projektors mit Sehstrahlen der Kamera geschnitten werden, kommt es zu leichten Versätzen der Schnittpunkte, was sich in Form einer streifenartigen Punkthäufung am Ebenenschnitt niederschlägt.

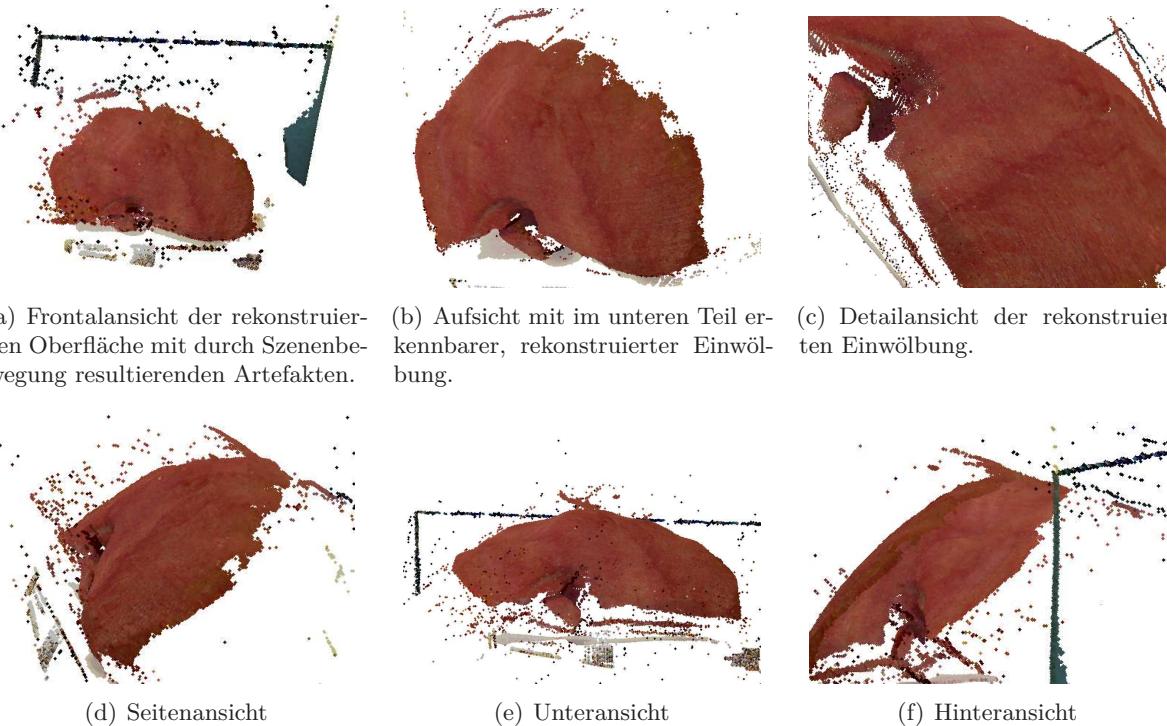


Abbildung 9.8: Verschiedene Ansichten der rekonstruierten, texturierten Oberfläche des Phantommodells L_A . Vorhandenes Rauschen röhrt von einer minimalen Verschiebung beweglicher Szenenteile zwischen zwei voneinander subtrahierten Rekonstruktionen her.

Wie in Abbildung 9.10 gezeigt, wurde die nun rekonstruierte Oberfläche im Rahmen der Nachverarbeitung zugeschnitten und von Rauscheffekten befreit. Diese Oberfläche wurde daraufhin gegen die in Abbildung 9.10 (f) dargestellte Referenzoberfläche verglichen. Hierbei wurden beide Datensätze erst manuell zur Deckung gebracht. Daraufhin wurde das Iterative-Closest-Point-Verfahren (ICP) auf beide Oberflächen angewendet. Dieses Verfahren nimmt solange rigide Transformationen der rekonstruierten Oberfläche vor, bis diese bestmöglich zur Deckung mit der Referenzoberfläche gebracht worden ist. In Abbildung 9.10 (d)-(f) sind die initialen Schritte des Verfahrens verdeutlicht. Hierbei wurde die rekonstruierte, wie auch die Referenzoberfläche sehr niedrig aufgelöst um das Verfahren zu testen. Nach einigen Iterationsschritten konvergiert das ICP-Verfahren, wonach die dadurch ermittelte, affine Transformation auf die rekonstruierte Oberfläche angewendet werden kann. Hierbei ist anzumerken, dass der Referenzdatensatz einem regulären Sampling unterzogen wurde und keine lokalen Häufungen wie an den Schnittkanten der rekonstruierten Oberfläche besitzt. Für die Verwendung des ICP-Verfahrens war dies nicht optimal, da dieses teilweise bereits vor Erreichen einer exakten Deckung konvergierte. Nach Durchführung des ICP-Verfahrens wurde die Distanz zwischen beiden Oberflächen ermittelt. Die Messung der euklidischen Distanz vor und nach Anwendung der Registrierung sind in Tabelle 9.3 aufgeführt.

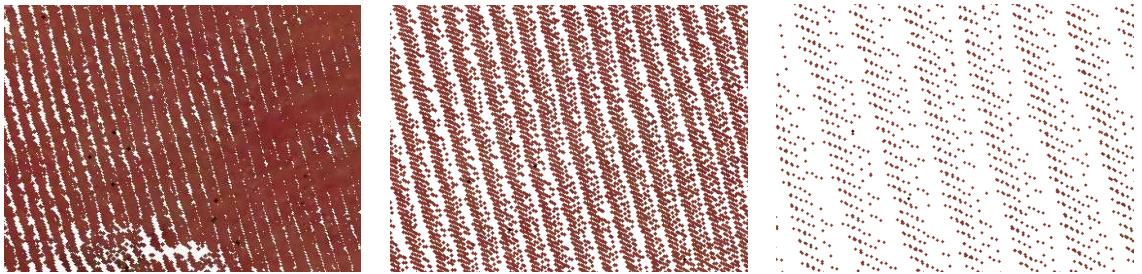


Abbildung 9.9: Streifenbildung bei dem während der Rekonstruktion per Gray-Code-Verfahren benutzten Schnitt von Kamera-Sehstrahlen mit Projektorzeilen- und Spalten als Ebenen im 3D-Raum.

Vergleich der rekonstruierten Oberfläche mit dem Referenzdatensatz (Leberphantom *L_A*)

Oberfläche	P	S	Euklidische Distanz
Referenz	20756	-365,-192,-524	(vor ICP) 35.0565
Rekonstruktion	87977	-414,-185,-535	(nach ICP) 8.90559

Tabelle 9.3: Messung zum Vergleich der rekonstruierten Oberfläche *L_A* mit der Referenzoberfläche. Punktmenge P, Schwerpunkt S und euklidische Distanz ED.

Die Ergebnisse der Rekonstruktion des Leberphantoms *L_A* sind durchaus zufriedenstellend. Der Abbildungsfehler von 8.9 Voxelpunkten, entspricht einem durchschnittlichen Abstand der Oberflächen von 9mm in der realen Welt. Hierbei wurden jedoch auch komplett Oberflächen miteinander verglichen, die Zielsetzung im Hinblick auf die Anwendung in der CL besteht jedoch in der korrekten Abbildung nur kleiner, planungsrelevanter Teile der Leberoberfläche. Der entstandene Fehler beim Vergleich beider Gesamtoberflächen konnte experimentell auf 3mm reduziert werden. Nach Anwendung des ICP-Verfahrens (Abbildung 9.10 (i)) wurden die überlagerten Oberflächen dazu zu einem 50x50mm großen Bereich zugeschnitten, erneut registriert und der Abstand gemessen.

Die am Phantommodell *L_B* durchgeführten Rekonstruktionen sind in Abbildung 9.11 aufgeführt. Dabei auftretende Artefakte resultieren aus einer zu hohen Entfernung der Kamera von der Szene. Hierdurch kann ein Gray-Code unter Umständen nicht vollständig bzw. fehlerfrei rekonstruiert werden, was wiederum zu falschen Ergebnissen beim Schnittpunkttest führt. Interessant bei diesem Modell sind die zahlreich vorhandenen Einstülpungen, die wegen der vorhandenen Verschattungen schwer rekonstruierbar waren. Ein Vergleich der rekonstruierten Oberfläche gegen den Referenzdatensatz ergab die in Tabelle 9.4 aufgeführten Werte. Hierbei wurden beide Oberflächen auf dieselbe Menge an Punkten reduziert.

Oberflächenvergleich (Leberphantom L_B)			
Oberfläche	P	S	Euklidische Distanz
Referenz	5300	-584, -162, -676	(vor ICP) 62.3612
Rekonstruktion	5300	-572, -183, -659	(nach ICP) 12.83951

Tabelle 9.4: Messung zum Vergleich der rekonstruierten Oberfläche L_B mit der Referenzoberfläche. Punktmenge P, Schwerpunkt S und euklidischer Distanz ED.

In Abbildung 9.12 ist als weiteres Beispiel die Oberflächenrekonstruktion eines Leber-Anatomiemodells dargestellt. An diesem Beispiel kann die zentrale Rolle einer geeigneten Umgebungsbeleuchtung verdeutlicht werden. Die Geometrie der Oberfläche an sich ist trotz der Entfernung der Kamera von der Szene und dem hellen Umgebungslicht (Abbildung 9.12 (a)) gut rekonstruiert worden (Abbildung 9.12 (b)). Neben den, durch die nicht optimale Parametrisierung bedingten Rauschartefakten, trägt die Oberflächenbeschaffenheit selbst einen großen Teil zur Verschlechterung der Ergebnisse bei. Durch ihre stark reflektierende Oberfläche treten Glanzlichter vermehrt auf. Die Kamerasensorik reicht in diesem Fall nicht mehr aus, um für diese Positionen einen Gray-Code zu detektieren, was am Fehlen von Objektpunkten an den betreffenden Stellen erkennbar wird (Abbildung 9.12 (f)).

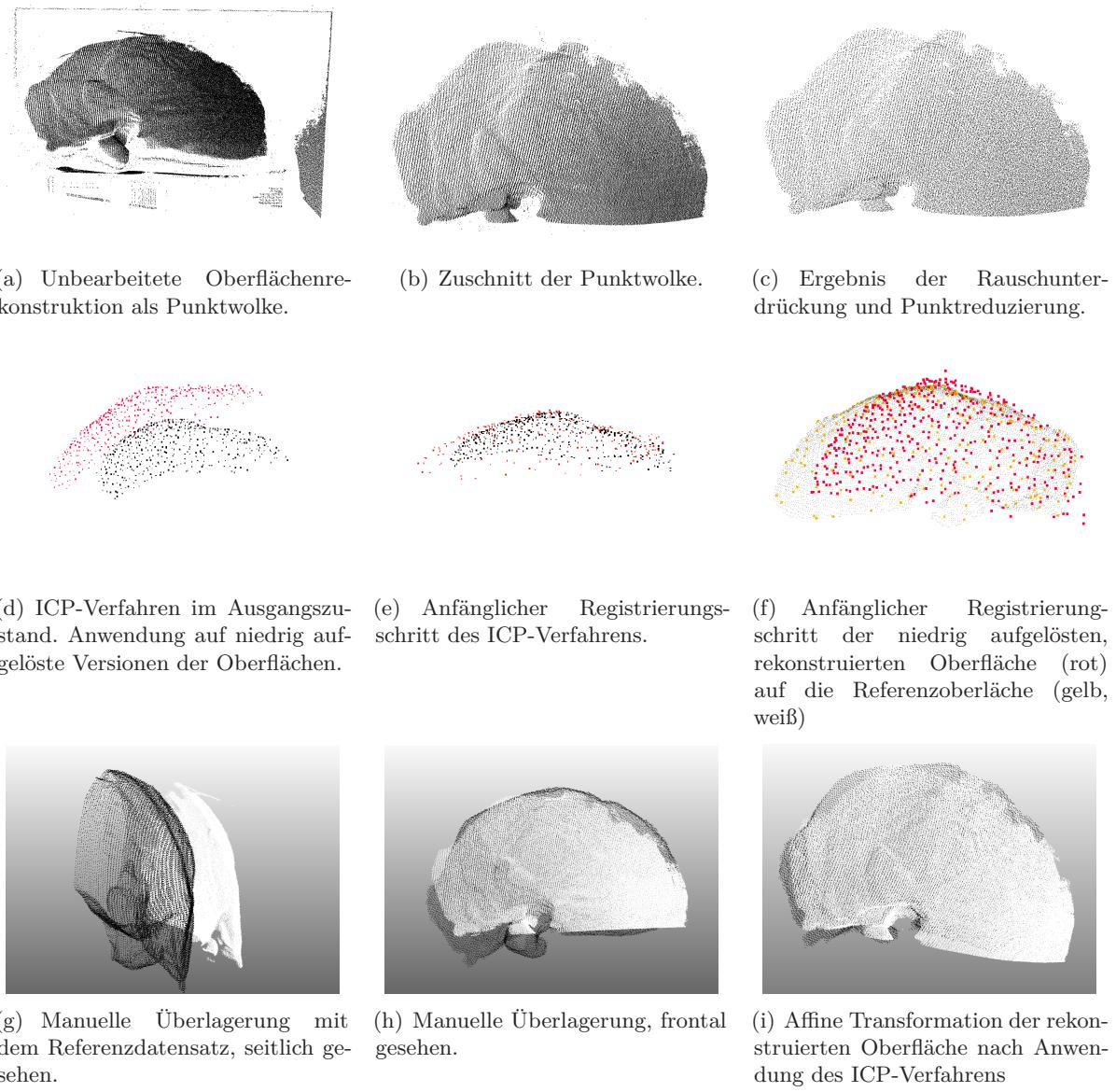


Abbildung 9.10: An rekonstuierten Oberflächendaten durchgeführte Nachverarbeitungsschritte.

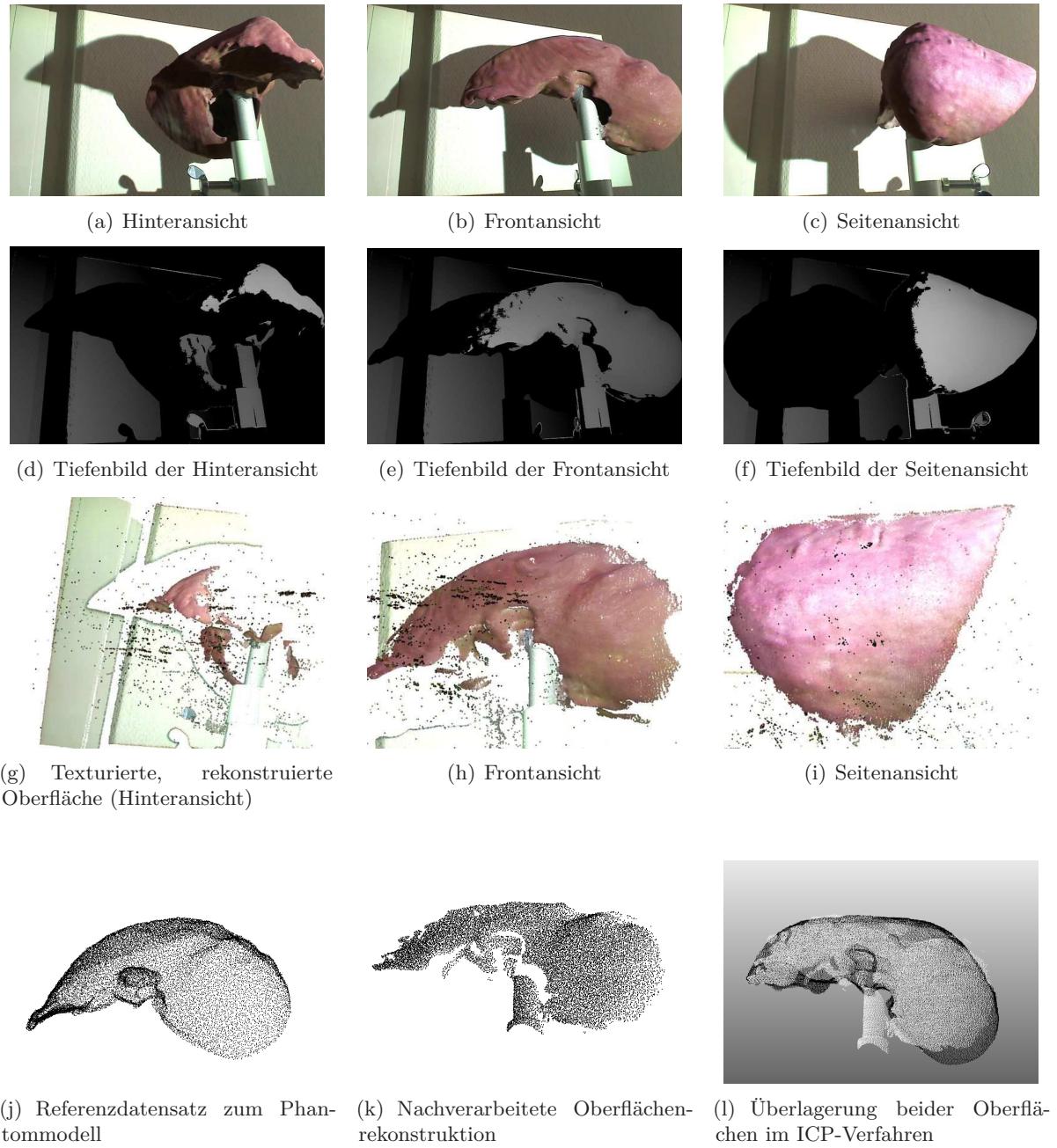


Abbildung 9.11: Rekonstruktionsergebnisse (Phantommodell L_B)

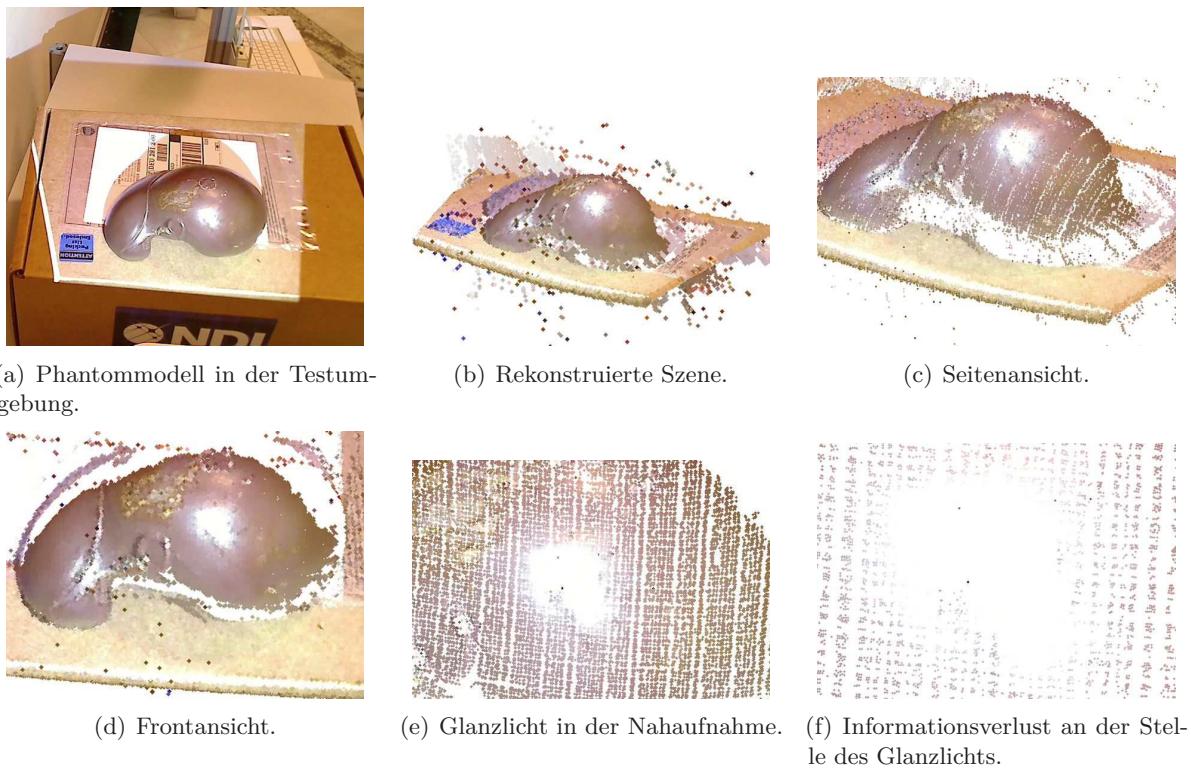


Abbildung 9.12: Rekonstruktionsergebnisse (Anatomiemodell). Durch Glanzlichter werden Tiefeninformationen verworfen.

10 Resümee und Ausblick

Ziel dieser Diplomarbeit war es, ein Verfahren zur Gewinnung dreidimensionaler Oberflächen-daten in der Software MeVisLab zu entwickeln. Unter Benutzung eines Kamera-Projektor-Aufbaus und dem auf Structured-Light basierenden Gray-Code-Verfahren wurde dieses Ziel erreicht. Als Abschluss wird die Arbeit resümiert und ein Ausblick auf die Lösung bestehender Problematiken im Rahmen einer Weiterentwicklung gegeben.

Zusammenfassung der Arbeit Die Implementierung einer Kameraanbindung, einer Projektor-Kamera-Kalibrierung und von Verfahren zur dreidimensionalen Rekonstruktion waren die hauptsächlichen Bestandteile der vorliegenden Arbeit. Neben der Implementierung von C++-Modulen wurde das in MeVisLab verwendete Konzept der grafischen Programmierung erfolgreich dazu eingesetzt, Netzwerke zur Visualisierung und Nachbearbeitung gewonnener Daten zu erstellen. Durch die Arbeit wurde in MeVisLab dabei ein relativ neuer Zweig der Entwicklung betreten. Da die Software tendenziell auf statischen Daten arbeitet, war die Verarbeitung von großen Datenströmen, wie sie eine Kamera liefert, anfangs problematisch. Die Kameraanbindung wurde daher basierend auf OpenCV umgesetzt und auf die Systemstruktur von MeVisLab mittels Threading angepasst.

Um das Projektor-Kamera-System komfortabel kalibrieren zu können, wurde auf eine gute Handhabung und Kontrollierbarkeit durch den Benutzer geachtet. Neben der freien Verwendbarkeit, sowie manuellen Kontrolle von Aufnahmen und detektierten Eckpunkten in Kalibriermustern, wurde dem Nutzer ein Feedback in Form einer Visualisierung des Aufbaus geboten. Der Vergleich, der von der Kalibrierung ermittelten Werte mit denen eines Referenzsystems, war sehr zufrieden stellend. Der Zeitaufwand der Kalibrierung bewegte sich zudem innerhalb der zum Ziel gesetzten Grenzen. Die Kalibrierung, welche eine der größten Hürden der Arbeit darstellte, kann somit als gut bewertet werden. Bei der Oberflächenrekonstruktion wurde zur korrekten Aufnahme einer Projektionsfolge eine Synchronisation zwischen Projektor und Kamera nötig, was ebenfalls gelöst werden konnte. Die Qualität der Oberflächenrekonstruktion selbst war zufrieden stellend. Abgesehen von, je nach Parametrisierung und verwendeter Hardware, auftretenden Artefakten, wie Rauschen, oder die Szenenverzerrung durch eine nicht optimal durchgeführte Kalibrierung, waren die Ergebnisse exakt. Im Rahmen weiterer Untersuchungen sollte die erfolgte Validierung des Verfahrens zur Evaluierung ausgeweitet werden um diese

ersten, positiven Ergebnisse zu untermauern. Das Gray-Code-Verfahren wurde qualitativen und praxisnahen Tests unterzogen. Hierbei wurde deutlich, dass unter optimaler Parametrisierung und Kalibrierung des Versuchsaufbaus die Gewinnung exakter Ergebnisse möglich ist. Die Nachteile des Verfahrens, wie die unzulängliche Rekonstruktion in Bereichen von Glanzlichtern, oder die Rauschanfälligkeit bei zu hoher Szenendistanz der Kamera, waren durch manuelle Korrekturen bis zu einem bestimmten Grad kompensierbar, aber nicht komplett lösbar.

Weiterentwicklung zum Einsatz in der Navigationsunterstützung Das Verfahren zur Oberflächenrekonstruktion bildet die Grundlage eines geplanten Gesamtsystems zur Navigationsunterstützung durch Verfolgung von elastischen Organtransformationen. Um den praktischen Einsatz in der CL zu gewährleisten, müssen jedoch zuerst Lösungen für bestehende Probleme gefunden werden. Das Problem, das es vorrangig zu lösen gilt, ist dabei die Anfälligkeit gegenüber Glanzlichtern. Wegen der Viskosität der Leber lässt sich dieses Problem nicht allein durch die Regelung der Belichtungsstärke des Projektors lösen. Hier kann jedoch Abhilfe durch den Einsatz einer Infrarotlichtprojektion geschaffen werden, welche unanfälliger gegen Glanzlichter ist [Lee u. a., 2005]. Die Oberflächenrekonstruktion sollte zum Einsatz in der CL vor allem durch den Einsatz weiter entwickelter Hardwarekomponenten beschleunigt werden. Eine Echtzeitrekonstruktion dynamischer Szenen ist zudem durch den Einsatz des Sinusphasenverfahrens denkbar. Dies müsste in weiteren Versuchen evaluiert werden. Da beim Sinusphasenverfahren exakte Tiefenwerte jedoch nur durch manuelle Parametrisierung erlangt werden können, müsste der Einsatz in der CL sorgfältig abgewägt werden.

Bei der Weiterentwicklung hin zum Navigationssystem werden weitere Probleme zu lösen sein. Da die Oberflächenrekonstruktion hierbei den grundlegenden Schritt darstellt, können in Zukunft entwickelte Verfahren darauf aufbauen. Die Verfolgung von elastischen Organtransformationen kann mittels Tracking von Infrarotmarkern umgesetzt werden. Nach somit gegebener Verortung des Organs im Raum, kann eine Oberflächenrekonstruktion durchgeführt werden, um Details des für die Operation relevanten Oberflächenbereichs ins Planungsmodell aufzunehmen. Um eine Navigation zu bieten, muss im nächsten Schritt die Aufprojektion des Planungsmodells erfolgen. Bei der Projektion auf Oberflächen müssen mehrere Faktoren beachtet werden, welche allgemein ihre Validität beeinflussen können. Wird ein Bild auf eine unebene Fläche projiziert, so wird es geometrisch verzerrt. Da zudem der Standort des Projektors während der Projektion stationär ist, ist die Projektion nicht für wechselnde Betrachterstandpunkte valide, sondern nur aus Sicht der Kamera selbst. Ein weiteres Problem kann die Veränderung der Projektion durch die Oberflächenfarbe sein, bei der Projektion kann es zudem in Randbereichen zu einem Helligkeitsabfall kommen. In der Arbeit von [Park u. a., 2006] wird ein Framework geboten, das all diese Probleme adressiert und dabei ebenfalls ein Projektor-Kamera-System behandelt.

Anlagen

Listings

```
bool OpenCVVideoCaptureThread::getCurrentFrameAsMLImage(ml::TSubImg<MLuint8> *  
    subImg){  
// ...  
    ml::Vector posRed (0,0,0, 0, 0,0);  
    ml::Vector posGreen(0,0,0, 1, 0,0);  
    ml::Vector posBlue (0,0,0, 2, 0,0);  
  
    MLuint8 *rPos = subImg->getImgPos(posRed );  
    MLuint8 *gPos = subImg->getImgPos(posGreen);  
    MLuint8 *bPos = subImg->getImgPos(posBlue );  
    int x, y;  
    MLuint8 *srcPos;  
  
    for (y = startY; y <= endY; ++y) {  
        srcPos = pixels + y * width + startX;  
        for (x = startX; x <= endX; ++x) {  
            *rPos++ = srcPos[2];  
            *gPos++ = srcPos[1];  
            *bPos++ = srcPos[0];  
            srcPos += 3;  
        } }  
        success = true;  
    }}}}  
frameMutex.unlock();  
return success;  
}
```

Listing 1: Umschreiben der Bildkanäle von BGR in RGB

```
for ( MLuint i = 0; i < width*height; i++ ){  
    val = (MLfloat) sin( (MLfloat) periods * (( (MLfloat) part *  
        (MLfloat)(i%height)) + (MLfloat) shift));  
    val = (val + 1.0) * 0.5 * 255.0;  
    *imgPos = (MLuint) val;  
    imgPos++;  
}
```

Listing 2: Generierung eines Sinusphasenbildes

```
def sinusPhaseCaptureSeries():
    network = ctx.parent()
    waitSecs = ctx.field("threePhaseCaptureLatency").intValue() / 1000

    # project pattern 1..3
    for i in range(1,4):
        ctx.field("threePhaseCaptureNumber").value = i
        if ( i == 1 ):
            network.field("StructuredLightPatternGenerator.3pimage1").touch()
        elif ( i == 2 ):
            network.field("StructuredLightPatternGenerator.3pimage2").touch()
        elif ( i == 3 ):
            network.field("StructuredLightPatternGenerator.3pimage3").touch()

    # capture current projection
    threePhaseCaptureImage()
    # wait
    MLAB.sleep( waitSecs )
    # process all events
    MLAB.processEvents()
    MLAB.processInventorQueue()
```

Listing 3: Aufnahme einer Projektionsfolge im Sinusphasen-Verfahren durch die Python-Funktion
sinusPhaseCaptureSeries

Literaturverzeichnis

- [Akca u. a. 2007] AKCA, Devrim ; REMONDINO, Fabio ; NOVÁK, David ; HANUSCH, Thomas ; SCHROTTNER, Gerhard ; GRUEN, Armin: Performance evaluation of a coded structured light system for cultural heritage applications. In: *Videometrics IX*, 2007, S. 29–30
- [Blanchette u. Summerfield 2008] BLANCHETTE, Jasmin ; SUMMERFIELD, Mark: *C++ GUI Programming with Qt 4*. Upper Saddle River, NJ, USA : Prentice Hall PTR, 2008
- [Bouguet 2010] BOUGUET, Jean-Yves: *Camera Calibration Toolbox for MATLAB*. http://www.vision.caltech.edu/bouguetj/calib_doc/. Version: 2010
- [Bradski u. Kaehler 2008] BRADSKI, Gary ; KAEHLER, Adrian: *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA : O'Reilly, 2008
- [Cash u. a. 2005] CASH, D.M. ; MIGA, M.I. ; SINHA, T.K. ; GALLOWAY, R.L. ; CHAPMAN, W.C.: Compensating for Intraoperative Soft-Tissue Deformations Using Incomplete Surface Data and Finite Elements. In: *MedImg 24* (2005), Nr. 11, S. 1479–1491
- [Clements u. a. 2007] CLEMENTS, L. W. ; DUMPURI, P. ; CHAPMAN, W. C. ; MIGA, M. I.: Atlas-based method for model updating in image-guided liver surgery. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* Bd. 6509, 2007 (Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference)
- [Falcao u. a. 2008] FALCAO, Gabriel ; HURTOS, Natalia ; MASSICHG, Joan: Plane-based calibration of a projector-camera system, VIBOT Master 2008. In: *VIBOT Master 2008* (2008)
- [Hansen 2006] HANSEN, Christian: *Verwendung von Textur in der Gefäßvisualisierung*, Universität Magdeburg, Diplomarbeit, 2006
- [Inokuchi u. a. 1984] INOKUCHI, S. ; SATO, K. ; MATSUDA, F.: Range Imaging System for 3-D Object Recognition. In: *ICPR84*, 1984, S. 806–808
- [J u. a. 2006] J, K. ; OLDHAFER ; BOURQUAIN, H. ; STAVROU, G. A.: Die Leberresektion (Teil I) - Anatomische Grundlagen, Indikationen und präoperative Planung. In: *CME Weiterbildung* (2006). – <http://www.chirurgiecelle.com/content/download/751/4862/file/LeberresektionI.pdf>

- [Kahrs u. a. 2006] KAHRS, Lüder ; HOPPE, Harald ; RIECHMANN, Matthias ; RACZKOWSKY, Jörg ; WORN, Heinz: Preclinical experiments for locating surgical structures below organ surfaces using projection-based augmented reality. In: *Computer Assisted Radiology and Surgery (CARS 2006)* (2006)
- [Krempien u. a. 2008] KREMPIEN, R. ; HOPPE, H. ; KAHRS, L. ; DAEUBER, S. ; SCHORR, O. ; BISCHOF, M. ; MUNTER, M. ; DEBUS, J. ; HARMS, W.: Projector-Based Augmented Reality for Intuitive Intraoperative Guidance in Image-Guided 3D Interstitial Brachytherapy. In: *International Journal of Radiation OncologyBiologyPhysics* 70 (2008), Nr. 3, S. 944–952
- [Lanman u. Taubin 2009] LANMAN, Douglas ; TAUBIN, Gabriel: Build Your Own 3D Scanner: 3D Photography for Beginners. In: *SIGGRAPH '09: ACM SIGGRAPH 2009 courses*. New York, NY, USA : ACM, 2009, S. 1–87
- [Lee u. a. 2005] LEE, Sukhan ; CHOI, Jongmoo ; BAEK, Seungmin ; JUNG, Byungchan ; CHOI, Changsik ; KIM, Hunmo ; OH, Jeongtaek ; OH, Seungsuk ; KIM, Daesik ; NA, Jaekeun: A 3D IR Camera with Variable Structured Light for Home Service Robots. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, S. 1859–1864
- [Markert u. a. 2007] MARKERT, Mathias ; WEBER, Stefan ; LUETH, Tim C.: Manual registration of ultrasound with CT/planning data for hepatic surgery. In: *Studies in health technology and informatics* 125 (2007), S. 319–321
- [McDonald 2009] McDONALD, Kyle: *Structured Light 3D Scanning*. 2009. – <http://www.instructables.com/id/Structured-Light-3D-Scanning/>
- [Narasimhan u. a. 2008] NARASIMHAN, Srinivasa G. ; KOPPAL, Sanjeev J. ; YAMAZAKI, Shuntaro: Temporal Dithering of Illumination for Fast Active Vision. In: *Proc. European Conference on Computer Vision*, 2008, S. 830–844
- [Park u. a. 2006] PARK, Hanhoon ; LEE, Moon-Hyun ; KIM, Sang-Jun ; PARK, Jong-Il: Surface-Independent Direct-Projected Augmented Reality. In: *Computer Vision - ACCV 2006*. 2006 (Lecture Notes in Computer Science), Kapitel 89, S. 892–901
- [Pawlik u. a. 2008] PAWLICK, Timothy M. ; SCHULICK, Richard D. ; CHOTI, Michael A.: Expanding Criteria for Resectability of Colorectal Liver Metastases. In: *Oncologist* 13 (2008), January, Nr. 1, S. 51–64
- [Posdamer u. Altschuler 1982] POSDAMER, J.L. ; ALTSCHULER, M.D.: Surface Measurement by Space-Encoded Projected Beam System. In: *CGIP* 18 (1982), January, Nr. 1, S. 1–17
- [Salvi u. a. 2004] SALVI, Joaquim ; PAGES, Jordi ; BATLLE, Joan: Pattern codification strategies in structured light systems. In: *Pattern Recognition* 37 (2004), April, Nr. 4, S. 827–849

- [Suzuki u. Hattori 2008] SUZUKI, Naoki ; HATTORI, Asaki: The road to surgical simulation and surgical navigation. In: *Virtual Reality* (2008)
- [Suzuki u. a. 2004] SUZUKI, S. ; SUZUKI, N. ; HATTORI, A. ; UCHIYAMA, A. ; KOBAYASHI, S.: Sphere-filled organ model for virtual surgery system. In: *Medical Imaging, IEEE Transactions on* 23 (2004), June, Nr. 6, S. 714–722
- [Tsai 1986] TSAI, R.: An efficient and accurate camera calibration technique for 3D machine vision. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 1986, S. 364–374
- [Weber u. a. 2008] WEBER, S. ; MARKERT, M. ; LUETH, T. C.: Surface tracking of organs for registration insoft tissue surgery. In: CATTIN, P. (Hrsg.) ; FEUERSTEIN, M. (Hrsg.) ; GROHER, M. (Hrsg.) ; WALSUM (Hrsg.) ; WEBER, S. (Hrsg.) ; ZIDOWITZ, S. (Hrsg.): *The MICCAI workshop on Image Guidance and Computer Assistance for Soft-Tissue Interventions (IGSTI)*, 2008, S. 65–74
- [Zhang u. Huang 2006] ZHANG, Song ; HUANG, Peisen S.: High-resolution, real-time three-dimensional shape measurement. In: *Optical Engineering* 45 (2006), Nr. 12, S. 123601+
- [Zhang u. Yau 2007] ZHANG, Song ; YAU, Shing T.: High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. In: *Optical Engineering* 46 (2007), Nr. 11, S. 113603+
- [Zhang 2000] ZHANG, Zhengyou: A Flexible New Technique for Camera Calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 11, S. 1330–1334
- [Zhu u. a. 2008] ZHU, Jianke ; HOI, Steven C. ; XU, Zenglin ; LYU, Michael R.: An Effective Approach to 3D Deformable Surface Tracking. In: *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*. Berlin, Heidelberg : Springer-Verlag, 2008, 766–779