

Punktbasiertes 3D-Tracking starrer und dynamischer Modelle mit einem Stereokamerasystem für Mixed Reality



Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Bernd Schwald
(Dipl.-Math.)

Referenten: Prof. Dr. J. Encarnação, Technische Universität Darmstadt
Prof. Dr. Gabriel Zachmann, Technische Universität Clausthal

Tag der Einreichung: 02.08.2006
Tag der mündlichen Prüfung: 12.09.2006

Darmstadt 2006
Hochschulkennziffer: D17

Danksagung

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Zentrum für Graphische Datenverarbeitung e.V. (ZGDV) in der Abteilung Visual Computing.

Ich danke Herrn Prof. Dr.-Ing. Dr. h.c. mult., Dr. E.h., Hon. Prof. mult. José L. Encarnaçã, für die Möglichkeit an der TU Darmstadt zu promovieren und die Übernahme des Referats. Weiterhin danke ich Herrn Prof. Gabriel Zachmann für die Übernahme des Koreferats.

Ich möchte mich bei allen aktuellen und ehemaligen Kollegen und Studenten mit denen ich zusammengearbeitet habe bedanken. Mein besonderer Dank gilt Tim Gleue für das Korrekturlesen dieser Arbeit im anstrengenden Rohzustand und die konstruktiven und hilfreichen Anregungen.

Herzlicher Dank gilt auch meiner Freundin Silke Eschenbeck für das Korrekturlesen, die Unterstützung und das Verständnis bei der Durchführung dieser Arbeit. Ebenso danke ich meinen Eltern und meinem Bruder Steffen für den Rückhalt, den sie mir geben.

Bernd Schwald, Darmstadt, August 2006

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung und Zielsetzung	3
1.3	Organisation der Arbeit	4
1.4	Zusammenfassung der wichtigsten Ergebnisse	5
2	Optische Trackingverfahren und Stereokamerasysteme	8
2.1	Grundbegriffe	8
2.1.1	Kameraaufbau und Interaktionsbereich	9
2.1.2	Verwendung von Stereosystemen	11
2.2	Kameramodell und Geometrie mehrerer Kameras	11
2.3	Kalibrierung des Kamerasystems	14
2.4	Optische Trackingtechnologien	16
2.4.1	Infrarottracking	16
2.4.2	Pattertracking	20
2.4.3	Tracking auf Basis von 3D-Modellen	22
2.5	Zusammenfassung	22
3	Rekonstruktion der 3D-Koordinaten und Korrespondenzanalyse	24
3.1	Marker und Punkte	25
3.2	Berechnung der 3D-Koordinaten	26
3.2.1	Rekonstruktionsfehler	29
3.2.2	Alternative Berechnungen des angenäherten Schnittpunkts	31
3.2.3	Rekonstruktion mit mehr als zwei Kameras	34
3.3	3D-Punktmodelle	35
3.4	Korrespondenzanalyse zwischen den 2D-Punkten	36
3.4.1	Epipolarkriterium	36
3.4.2	Weitere Kriterien	38
3.4.3	Verfolgung und Prädiktion auf 2D-Ebene	40
3.4.4	Korrespondenzanalyse bei mehr als zwei Kameras	41
3.5	Distanztoleranz	41
3.6	Zusammenfassung	42

4	Starre Modelle	43
4.1	Identifikation einzelner Marker	44
4.1.1	Verfolgung eines einzelnen Markers	46
4.1.2	Verfolgung eines einzelnen Markers zusammen mit einem Modell	46
4.1.3	Identifikation mehrerer einzelner Marker	48
4.2	Zweipunktmodelle	54
4.2.1	Algorithmus und Definition für Zweipunktmodelle	56
4.2.2	Glättung aufeinander folgender Positionen	58
4.2.3	Reduktion der Liste der Punktepaare	59
4.2.4	Mehrdeutigkeiten	60
4.3	Multipunktmodelle	62
4.3.1	Berechnung der Modelltransformation aus drei beliebigen Markern	63
4.3.2	Berechnung der Modelltransformation aus mehreren beliebigen Markern	66
4.3.3	Modellerkennungsstrategien	67
4.3.4	Detektion mehrerer Modelle	77
4.3.5	Verwechselbare Distanzen	80
4.4	Zusammenfassung	84
5	Anlernen starrer Modelle	85
5.1	Lernalgorithmus für Multipunktmodelle	87
5.1.1	Distanzmessung	87
5.1.2	Modellerzeugung und -validierung	89
5.1.3	Hinzufügen neuer Punkte	90
5.2	Evaluation des Lernalgorithmus	90
5.2.1	Anlernen mit unbewegtem Objekt	92
5.2.2	Anlernen mit bewegtem Objekt	93
5.2.3	Anlernen mit bewegtem Objekt und teilweise verdeckten Markern	95
5.3	Definition lokaler Koordinatensysteme	97
5.3.1	Bestimmung eines lokalen Koordinatensystems durch Kalibrierung	99
5.3.2	Bestimmung eines lokalen Koordinatensystems durch freie Plat- zierung	100
5.4	Zusammenfassung	102
6	Dynamische Modelle	103
6.1	Dynamische Gruppenmodelle	105
6.1.1	Handgruppenmodelle	107
6.1.2	Interaktion durch Handgruppenmodelle	108
6.1.3	Interaktion mit zwei Händen	110
6.2	Dynamische Gelenkmodelle	112
6.2.1	Arm- und Fingergelenkmodelle	114
6.2.2	Körpermodelle	114
6.3	Dynamisches Flächenmodell	115
6.4	Kombination starrer und dynamischer Modelle	118

6.5	Zusammenfassung	121
7	Filterung und Prädiktion	122
7.1	Mittelung und Lineare Prädiktion	123
7.1.1	Mittelung von Rotationen	125
7.2	Kalman Filter	126
7.3	Nachführung der Transformation	127
7.4	Zusammenfassung	128
8	Anwendungen	129
8.1	MEDARPA - MEDical Augmented Reality for PATients	129
8.2	Korrektur von Magnetresonanz-Tomographie-Aufnahmen	135
8.3	Multiuser-Anwendung im Cybernarium	137
8.4	Zusammenfassung	138
9	Zusammenfassung und Ausblick	139
9.1	Zusammenfassung	139
9.2	Ausblick	140
A	Darstellung von Rotationen	141
B	3D-Punkt Residuum	144
C	Beispiel zur Identifikation mehrerer einzelner Marker	146
D	Beispiel zur Reduktion der Liste von Punktepaaren bei Zweipunktmodellen	148
	Abbildungsverzeichnis	150
	Tabellenverzeichnis	154
	Index und Abkürzungen	155
	Literaturverzeichnis	157
	Lebenslauf	167

Kapitel 1

Einleitung

1.1 Motivation

Der Umgang mit virtuellen Welten ist inzwischen für die meisten Anwender am Computer nicht mehr ungewöhnlich: durch virtuelle Welten werden Informationen vermittelt, durch bessere Veranschaulichung die Arbeit erleichtert oder sie dienen als Freizeitbeschäftigung. **Virtual Reality** (VR), im Deutschen auch als Virtuelle Realität bezeichnet, hat begonnen in den Benutzeralltag Einzug zu halten, und selbst der Begriff **Augmented Reality** (AR), im Deutschen auch als Erweiterte Realität bezeichnet, ist einigen nicht mehr unbekannt. Insbesondere bei Sportübertragungen im Fernsehen ist die erweiterte Sicht auf das aktuelle Sportgeschehen durch die korrekte Überblendung des Fernsehbilds mit einer virtuellen Bestmarke beim Skisprung oder der Abseitslinie beim Fußball ein fester Bestandteil geworden.

VR und AR sowie die verschiedenen Zwischenformen, die unter dem Begriff **Mixed Reality** (MR) zusammengefasst sind, haben vor allem eines gemeinsam: **Tracking**, das heißt die Verfolgung von Objekten oder Personen in Echtzeit, bildet einen wesentlichen technologischen Bestandteil zahlreicher MR-Anwendungen. Während im VR-Bereich Tracking ein wichtiges Instrument ist, um Anwendungen immersiver zu machen und eine natürlichere Interaktion mit den virtuellen Welten zu ermöglichen, ist Tracking für AR von noch elementarer Bedeutung. Der Blickpunkt des Betrachters und die Positionen von Objekten müssen bekannt sein und räumlich verfolgt werden. Bei kamerabasierten Systemen kann die Verfolgung auch nur auf der Bildebene, also im Zweidimensionalen, stattfinden. Durch das Tracking können virtuelle Informationen oder Objekte in der realen Sicht oder auf Bildern korrekt positioniert und ausgerichtet dargestellt werden. Während der Begriff Tracking in vielen anderen Zusammenhängen, wie der Erfassung von Konsumentenverhalten oder der Bewegungen von Internet-Nutzern, verwendet wird, steht er im Kontext dieser Arbeit ausschließlich für die Erfassung von Positions- und/oder Orientierungsdaten von Personen und/oder Objekten, um eine Echtzeit-Visualisierung in MR-Anwendungen zu unterstützen.

Tracking bildet die Grundlage für neue Interaktionsformen mit dem Computer, die jenseits der Verwendung von Maus und Tastatur liegen. Ein in der Computerspielindustrie bereits umgesetztes einfaches Beispiel für eine neue Interaktionsform stellt das millio-

nenfach verkaufte Sony Playstation 2 EyeToy dar, bei dem Computerspiele durch reine Bewegung der Hände oder des Körpers vor einer Kamera ohne weitere Zusatzgeräte gesteuert werden. Diese Form der Interaktion erfordert vom Benutzer ein erhöhtes Maß körperlicher Betätigung, allerdings erfolgt die Erfassung von Bewegungen nur auf der 2D-Bildebene, das heißt, insbesondere räumliche Bewegungen in die Tiefe (von der Kamera aus gesehen) können nicht wahrgenommen werden.

Um 3D-Tracking, also räumliches Verfolgen von Personen oder Objekten, zu ermöglichen, bedarf es anderer Technologien. Dabei gibt es unterschiedliche Ansätze wie beispielsweise elektromagnetisches, inertiales und optisches Tracking. Diese Technologien erlauben meist eine Verfolgung mit drei oder sechs Freiheitsgraden (DOF, degrees of freedom), was der Bestimmung einer Position beziehungsweise Orientierung - oder beidem gleichzeitig - entspricht. Optisches Tracking zeichnet sich dabei gegenüber anderen Technologien durch hohe Präzision, geringe Störanfälligkeit und Kabellosigkeit aus. Optische Trackingverfahren und -systeme stellen einen großen und zunehmenden Anteil der insgesamt verwendeten Trackingsysteme dar.

Die Grundlage dieser Arbeit bilden optische Stereokamerasysteme, das heißt Systeme, die häufig aus zwei, aber auch mehreren Kameras bestehen, und eine Bestimmung räumlicher Daten ermöglichen. Solche Systeme werden im Allgemeinen vor ihrer Verwendung kalibriert, sodass eine sehr präzise Rekonstruktion räumlicher (3D-)Punkte aus korrespondierenden 2D-Punkten der synchron aufgenommenen Kamerabilder erfolgen kann. Um korrespondierende Punkte in den unterschiedlichen Sichten der Kameras leichter zu finden gibt es die Möglichkeit Marker in der zu erfassenden Szene zu platzieren.

Solche Trackingsysteme finden ihre Anwendung in zunehmendem Maße im VR- und AR-Bereich zur Verfolgung von Interaktionsgeräten, zur Blickpunktbestimmung oder zur Erfassung menschlicher Bewegungsabläufe. Je nach Anwendungsgebiet werden bei der Verarbeitung der 3D-Daten, die aus Markern berechnet werden, unterschiedliche, spezialisierte Algorithmen benötigt, um die entsprechenden Transformationen zu bestimmen.

Während einige technologische Bestandteile eines Stereokamerasystems, wie die Bildverarbeitung, intrinsische und extrinsische Kamerakalibrierung, Korrespondenzanalyse zwischen Bildern, Rekonstruktion eines 3D-Punktes aus 2D-Punkten oder Filterung von Trackingergebnissen, schon vielseitig untersucht wurden, gibt es zur Verarbeitung der nach der 3D-Rekonstruktion vorliegenden Daten in Echtzeit wenig Untersuchungen.

Der Inhalt dieser Arbeit setzt sich daher mit verschiedenen neuen Trackingalgorithmen für Stereokamerasysteme auseinander, die auf der Verarbeitung rekonstruierter 3D-Punkte beruhen. Dies beinhaltet auch automatische und nicht automatische Methoden zum Anlernen von Eigenschaften verschiedener Punktkonstellationen als Grundlage für die Algorithmen. Die rekonstruierten 3D-Punkte können dabei im Fall eines Infrarottrackingsystems die Koordinaten eines Infrarotmarkers sein. Bei einem markerlosen Ansatz können dies beispielsweise die Koordinaten von Eckpunkten eines zu verfolgenden Objekts sein. Verschiedene Trackingalgorithmen werden dabei benötigt, um sowohl starre als auch dynamische Punktanordnungen erfassen und verfolgen zu können.

Ein wesentlicher Aspekt ist die Effizienz der Algorithmen, da eine Verarbeitung innerhalb weniger Millisekunden stattfinden muss, um eine Visualisierung in Echtzeit unterstützen zu können. Geht man dabei von einer Bildwiederholrate der für das Tracking

verwendeten Kameras von mindestens 20 Bildern pro Sekunde aus, dann muss die Erfassung und Verarbeitung der einzelnen Bilder in weniger als 50 ms erfolgen.

Der Einsatzbereich der aus den Algorithmen und Verfahren dieser Arbeit resultierenden Trackingsysteme umfasst unter anderem klassische VR-Anwendungen, bei denen beispielsweise eine Shutterbrille und ein Handgerät verfolgt werden müssen, Anwendungen, bei denen eine größere Anzahl von Benutzern genau verfolgt werden muss, beidhändige 3D-Interaktion für einen zukünftigen 3D-Arbeitsplatz, sowie die Verfolgung komplexer, fester, realer Objekte, wie sie typischerweise im AR-Bereich vorkommen können. Außerhalb des MR-Bereichs kann die Verfolgung fester Objekte beispielsweise genutzt werden, um auf einfache Weise für Roboterarme zurückzulegende räumliche Pfade festzulegen oder um mit Hilfe der dynamischen Trackingalgorithmen ein auf der Verfolgung der Hände basierendes System zur Erfassung von Zeichensprache zu realisieren.

1.2 Problemstellung und Zielsetzung

Das Ziel dieser Arbeit besteht darin, auf Basis der bereits durch ein Stereokamerasystem zur Laufzeit rekonstruierten Menge von 3D-Punkten, Algorithmen zu entwickeln, die in Echtzeit eine Verfolgung von Objekten und/oder Personen zulässt (Tracking). Die Algorithmen sollen auch für Systeme geeignet sein, die mit möglichst einfacher Hardware auskommen. Dazu zählen insbesondere Stereokamerasysteme mit nur zwei Kameras, die für den VR- und AR-Bereich besonders geeignet sind. Ein Schwerpunkt dieser Arbeit besteht in der Eignung der Algorithmen für Infrarottrackingsysteme, bei denen die rekonstruierten 3D-Punkte die Koordinaten einheitlicher kugelförmiger, also nicht unterscheidbarer, Marker sind.

Weitere Grundlagen für die gesamte Verarbeitungspipeline eines solchen Trackingsystems, also Kalibrierung der Kameras und Bildverarbeitung, Rekonstruktion der 3D-Punkte, sowie die Filterung der Trackingergebnisse sollen hinsichtlich der Zusammenhänge mit den zu entwickelnden Trackingalgorithmen kurz dargestellt werden.

Für die Trackingalgorithmen selbst soll eine Unterteilung in verschiedene Bereiche erfolgen. Dies sind unter anderem die Verfolgung einzelner Punkte, die Verfolgung einer beliebigen Anzahl relativ zueinander fest (starr) positionierter Punkte oder die Verfolgung einer sich verändernden (dynamischen) Anordnung von Punkten. Für die verschiedenen Bereiche sollen jeweils Modelle entwickelt werden, die eine effiziente und zuverlässige Verfolgung unterstützen. Neben dem eigentlichen Tracking soll dabei auch das Initialisieren, beziehungsweise das Anlernen der Modelle, behandelt werden.

Die hierzu notwendige Erweiterung eines 3D-Punktes durch ein Punktmodell, welches Zusatzinformationen trägt, wie zum Beispiel dessen letzte Position oder bestimmte Indizes, soll so formuliert werden, dass es für alle hier entwickelten Trackingalgorithmen verwendbar ist. Dies soll insbesondere eine Kombination oder alternative Nutzung verschiedener Trackingalgorithmen innerhalb eines Trackingframeworks erlauben.

Neben der Entwicklung der Theorie für die Trackingalgorithmen soll der direkte Bezug zu praktischen Anwendungen dargestellt werden. Insbesondere die durch die Anwendungen vorgegebenen Anforderungen an die Trackingalgorithmen sollen dabei berücksichtigt werden.

1.3 Organisation der Arbeit

Die vorliegende Arbeit unterteilt sich in drei übergeordnete Bereiche: die einleitenden Kapitel 2 und 3 zur Bereitstellung der Grundlagen für das Thema, die Kapitel 4, 5 und 6, welche die verschiedenen Trackingalgorithmen und damit den Schwerpunkt der eigenen Arbeit beschreiben, und die abschließenden Kapitel 7 und 8, die sich zum einen mit der Filterung von Trackingergebnissen und zum anderen mit verschiedenen praktischen Anwendungen dieser Arbeit beschäftigen. Dabei entspricht die Anordnung der einzelnen Kapitel und Bereiche der groben Struktur der aufeinanderfolgenden Schritte eines Trackingsystems. Im Einzelnen haben die Kapitel folgenden Inhalt:

Kapitel 2 beinhaltet eine kurze Einführung in die Themen optische Trackingverfahren und Stereokamerasysteme. Die für diese Arbeit relevanten Begriffe und technischen Merkmale werden hier eingeführt: Ein mathematisches Kameramodell, die Kalibrierung der Kameras sowie verschiedene Arten von Markern und deren Verarbeitung werden beschrieben.

Auf Basis dieser Grundlagen wird in Kapitel 3 beschrieben, wie die Rekonstruktion der 3D-Punkte aus korrespondierenden 2D-Punkten aus den einzelnen Kamerabildern erfolgt und mit welchen Methoden die Korrespondenzen bestimmt werden. Dabei wird das in den folgenden Kapiteln verwendete 3D-Punktmodell aufgestellt, welches neben den Punktkoordinaten weitere Eigenschaften mit einschließt.

Kapitel 4 beschreibt die Erkennung und Verfolgung starrer Modelle, also relativ zueinander fester Punkt- oder Markerkonstellationen. In diesem und den folgenden Kapiteln wird die Existenz rekonstruierter 3D-Punkte als gegeben vorausgesetzt. Algorithmen und Modellbeschreibungen für die Verfolgung einzelner Punkte und Zweipunktkonstellationen werden präsentiert. Den Hauptteil bildet die Modellbeschreibung und Verfolgung von Multipunktmodellen, also starrer Konstellationen mit mindestens drei Punkten.

Eine wesentliche Grundlage für die Erkennung von Multipunktmodellen ist die initiale Bestimmung der das Modell beschreibenden 3D-Punkte. Dieses Anlernen der Punkte wird in Kapitel 5 vertieft. Eine Methode für das automatische Anlernen von Modellen, die während der Lernphase bewegt werden können, um auch komplexe Objekte erfassen zu können, wird präsentiert und evaluiert.

Im Anschluss an die starren Modelle werden in Kapitel 6 dynamische Modelle entwickelt und diskutiert. Dies sind Konstellationen von Punkten, deren Positionen sich zwischen zwei Erfassungszeitpunkten relativ zueinander verändern können, die aber dennoch eine Einheit bilden. Dynamische Gruppenmodelle insbesondere für ein- oder beidhändige Interaktion werden genauer beschrieben, Gelenkmodelle und Flächenmodelle werden kurz dargestellt. Der Abschnitt 6.4 fasst abschließend zusammen, welche starren und dynamischen Modelltypen kombiniert werden können, um eine hybride Trackinglösung bezüglich der verschiedenen Typen von Modellen zu erhalten.

Das Kapitel 7 enthält verschiedene Methoden zur Filterung und Prädiktion der Transformationen, die aus den in den vorausgegangenen Kapiteln entwickelten Trackingalgorithmen resultieren.

Schließlich folgt zum Abschluss des Inhalts dieser Arbeit ein Kapitel, welches drei ausgewählte, praktisch umgesetzte Anwendungen mit den in dieser Arbeit beschriebenen und

implementierten Algorithmen vorstellt. Die Beschreibung der Anwendungen in Kapitel 8 umfasst eine allgemeine Darstellung der jeweiligen Applikation, sowie den besonderen Zusammenhang mit den Algorithmen.

Eine Zusammenfassung der Arbeit sowie ein Ausblick befindet sich in Kapitel 9. Die Anhänge A bis D enthalten unter anderem Ergänzungen zur Darstellung von Rotationen und verschiedene Beispiele.

1.4 Zusammenfassung der wichtigsten Ergebnisse

Im Rahmen dieser Arbeit wurden mehrere Verfahren für das Tracking starrer und dynamischer Objekte mit einem Stereokamerasystem entwickelt. Ausgangspunkt für die Beschreibung der Algorithmen sind durch ein Stereokamerasystem rekonstruierte 3D-Punkte, die in dem in dieser Arbeit entworfenen 3D-Punktmodell vorliegen. Passend zu den Algorithmen wurden verschiedene Arten von starren und dynamischen Modellen entwickelt, die die verschiedenen Eigenschaften repräsentieren und deren Verfolgung erlauben. Die beschriebenen Algorithmen und Modelle wurden in erster Linie für ein System bestehend aus zwei Kameras, die im Infrarotbereich Bilder aufnehmen, entworfen und implementiert. Dabei unterstützen die Algorithmen und Modelle eine skalierbare Hardware, erlauben also neben der Verwendung hochwertiger Kameras und Objektive auch eine Anpassung an einfachere Produkte, die weniger exakte Werte liefern, aber je nach Anwendungsbereich angemessen sind. Sie können aber auch als Erweiterung anderer ähnlicher Systeme oder für Systeme mit einer anderen zu Grunde liegenden Trackingtechnologie genutzt werden.

Ein besonderer Schwerpunkt liegt auf der Beschreibung des neu entwickelten und parametrisierbaren Algorithmus für Multipunktmodelle und der entsprechenden Modellbeschreibung selbst [Schwald 02a], [Schwald 05a]. Multipunktmodelle sind die abstrakte Beschreibung einer Konstellation von mindestens drei fest zueinander platzierten Punkten beziehungsweise Markern mit allen zusätzlich benötigten Parametern. Durch die Festlegung der verschiedenen Parameter des Modells und des Algorithmus kann dabei letzterer so modifiziert werden, dass er bestimmte erwünschte Eigenschaften, wie geringe Verarbeitungszeit oder Robustheit stärker oder schwächer berücksichtigt.

Ein weiterer Schwerpunkt liegt auf dem automatischen Anlernen [Schwald 04a] von Multipunktmodellen durch freie Bewegung des zu verfolgenden Objekts. Dieser neue, im Rahmen dieser Arbeit entwickelte Algorithmus erlaubt ein Anlernen komplexer Punkt-konstellationen mit einem Kamerasystem, welches auch aus wenigen Kameras bestehen kann, also insbesondere einem Zweikamerasystem.

Neben den Multipunktmodellen werden als starre Modelle die hier definierten Ein- und Zweimarkernmodelle mit den entsprechenden Algorithmen präsentiert. Die Zweipunktmodelle eignen sich dabei insbesondere für Anwendungen, in denen die Interaktion zahlreicher Benutzer erforderlich ist [Schwald 05c].

Während die in dieser Arbeit entwickelten starren Modelle vor allem für die Verfolgung von Objekten geeignet sind, bieten die im Rahmen dieser Arbeit entwickelten dynamischen Modelle die Möglichkeit einer direkteren Form der Interaktion mit dem Computer durch den menschlichen Körper. Die Modelle bestehen dabei aus keiner festen Anordnung

von Markern sondern einer Anordnung, die Veränderungen zwischen den Erfassungszeitpunkten zulässt. Der Schwerpunkt der Bearbeitung der dynamischen Modelle liegt auf dem hier entwickelten und als Gruppenmodell bezeichneten Ansatz, der für eine direkte Interaktion mit den Händen besonders geeignet ist [Schwald 05b].

Außer der Beschreibung der verschiedenen einzelnen hier entwickelten Modelle und Algorithmen ist es Ziel dieser Arbeit, möglichst umfassend die verschiedenen Arten von Trackingansätzen auf Basis rekonstruierter 3D-Punkte darzustellen. Abbildung 1.1 zeigt das Gesamtkonzept dieser Arbeit, wobei in horizontaler Ausrichtung aufeinanderfolgende Verarbeitungsschritte beschrieben sind und in vertikaler Richtung die Bandbreite der verschiedenen Ansätze. Außerdem bietet das hier dargestellte Konzept die Möglichkeit, verschiedene Arten von Modellen, also insbesondere starre und dynamische Modelle, innerhalb eines Systems zu kombinieren.

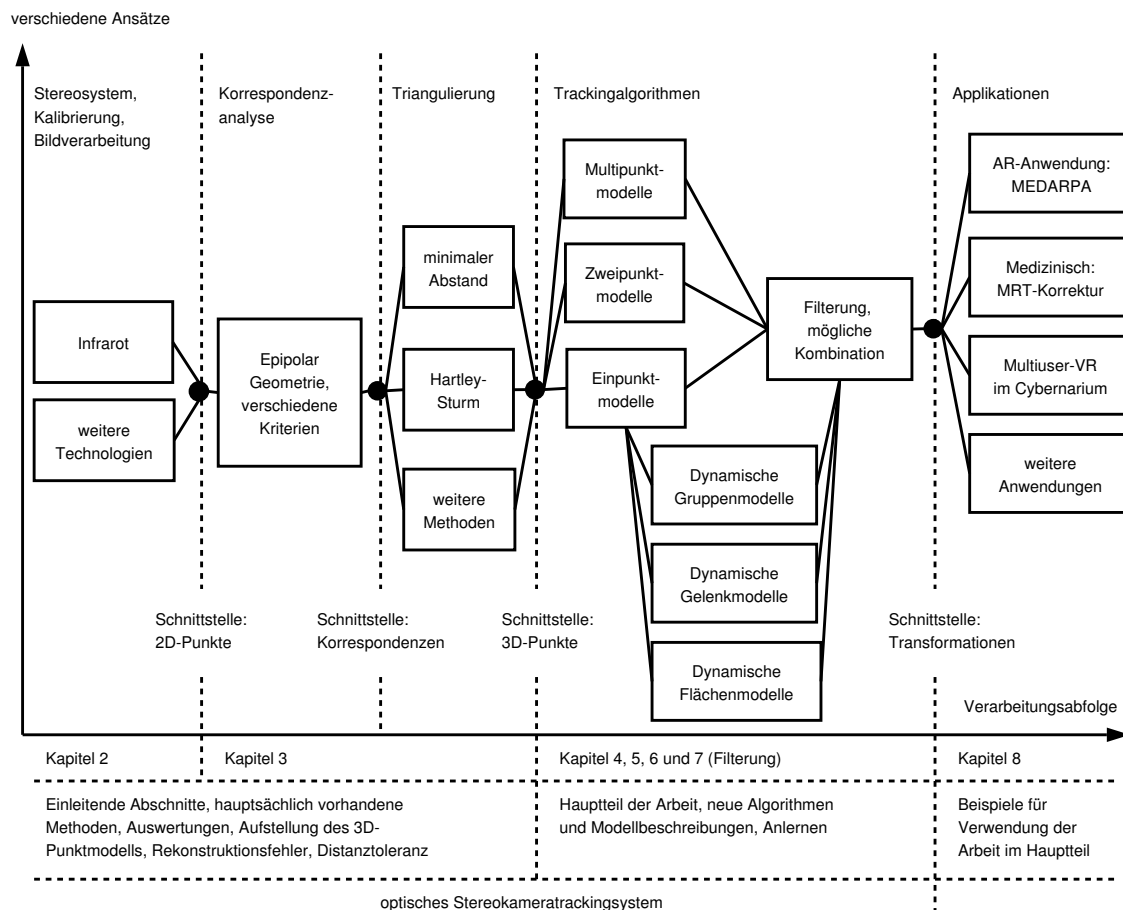


Abbildung 1.1. Gesamtkonzept dieser Arbeit. In horizontaler Ausrichtung sind aufeinanderfolgende Verarbeitungsschritte beschrieben und in vertikaler Richtung die Bandbreite der verschiedenen Ansätze.

Die Umsetzung und die Verwendbarkeit der in dieser Arbeit beschriebenen Verfahren in der Praxis wird anhand verschiedener umgesetzter Anwendungen aus dem Mixed Reality

Bereich [Schnaider 03], [Schwald 03a], [Schwald 03b], [Röddiger 04], [Schwald 04d], [Schwald 04e], [Wesarg 04], dem rein medizinischen Bereich [Dold 04], beziehungsweise zur Nutzung innerhalb eines hybriden Trackingsystems [Schwald 02b], [Schwald 04b], [Schwald 04c] dargestellt.

Kapitel 2

Optische Trackingverfahren und Stereokamerasysteme

Ein Kamerasystem bezeichnet in dieser Arbeit eine Kombination aus Hardware, also insbesondere ein Computer und eine oder mehrere Kameras, und einer Software, welche die aufgenommenen Bilder verarbeitet. Kamerasysteme bestehend aus mindestens zwei Kameras ermöglichen die Betrachtung einer Szene aus zwei oder mehreren verschiedenen Blickwinkeln und werden hier als **Stereokamerasysteme** bezeichnet. **Trackingsysteme** können Stereokamerasysteme, die für die Verfolgung von Objekten im Dreidimensionalen eingesetzt werden, sein. Allerdings gibt es auch verschiedene andere Technologien, wie optische Systeme mit nur einer Kamera oder einem Laser, elektromagnetische und inertielle Trackingsysteme. Im Rahmen dieser Arbeit spielen nur Stereokamerasysteme eine Rolle.

Um mit einem Stereokamerasystem eine präzise 3D-Rekonstruktion und eine stabile Verfolgung von Objekten im Dreidimensionalen mit einer möglichst geringen Latenz zu ermöglichen, bedarf es jedoch mehr, als der reinen Betrachtung aus unterschiedlichen Blickwinkeln. Verschiedene Eigenschaften, Überlegungen zum Kameraaufbau und zum Interaktionsbereich, sowie weitere Grundbegriffe werden in den folgenden Abschnitten vorgestellt und diskutiert.

2.1 Grundbegriffe

Soll bei der Verwendung eines Stereokamerasystems eine hohe Präzision der 3D-Rekonstruktion erreicht werden, müssen die Kameras **synchron** laufen, da ansonsten die verschiedenen Kameras - insbesondere bei bewegten Szenen - Bilder von ungleichen Zuständen aufnehmen. Verfolgt man beispielsweise bei einer Bildwiederholrate (**Framerate**) von 30 fps (frames per second) ein Objekt, welches sich mit $20 \frac{\text{km}}{\text{h}} \approx 5,6 \frac{\text{m}}{\text{s}}$ bewegt, so verschiebt sich das Objekt zwischen der Aufnahme zweier Bilder einer Kamera um knapp 19 cm. Insbesondere für bewegungsreiche Szenen ist also ein synchronisiertes Kamerasystem unverzichtbar, auch wenn Kameras mit höherer Framerate eingesetzt werden. Auch bei weniger bewegungsreichen Szenen empfiehlt es sich, die Kameras zu synchronisieren, da sich die Fehler, die durch unsynchronisierte Bilder entstehen, nur schwer zu

einem späteren Zeitpunkt in der Verarbeitung ausgleichen lassen. In [Rasmussen 06] ist ein Stereokamerasystem beschrieben, bei dem auf die Synchronisation der Kameras verzichtet wird, um möglichst einfache Hardware verwenden zu können.

Der Begriff **Frame** wird in dieser Arbeit als Bezeichnung für das zu einem Zeitpunkt aufgenommene n -Tupel von Vollbildern n synchronisierter Kameras verwendet. Dementsprechend bezieht sich die Framerate sowohl auf alle Kameras, als auch auf jede einzelne Kamera. Der Begriff **Updaterate** bezieht sich auf die Frequenz der Ausgabe eines Trackingsystems und wird insbesondere dann benutzt, wenn Vergleiche zu nicht optischen Systemen gezogen werden. Die Framerate und die Updaterate eines optischen Trackingsystems sind oft, aber nicht notwendigerweise immer, gleich.

Ein weiterer entscheidender Faktor für die Qualität der Messungen im Dreidimensionalen ist die Kalibrierung der Kameras. Zum einen können und sollten Kameras intrinsisch kalibriert werden, das heißt diejenigen Parameter einer Kamera werden bestimmt, die unabhängig vom Aufbau des gesamten Kamerasystems sind. Außerdem sollten die Kameras extrinsisch kalibriert, also deren Positionierung und Orientierung zueinander bestimmt, werden. Die Kalibrierung von Kameras gehört zu den wesentlichen Aufgaben bei der Entwicklung eines Trackingsystems und wird in Abschnitt 2.3 weiter ausgeführt.

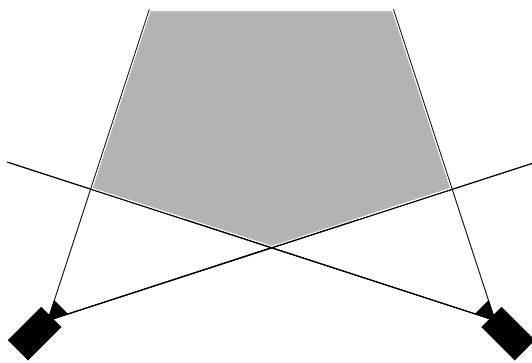
2.1.1 Kameraaufbau und Interaktionsbereich

Auch der Aufbau der Kameras spielt eine entscheidende Rolle, nicht nur für die Genauigkeit der 3D-Rekonstruktion, sondern auch für den **Interaktionsbereich** eines Trackingsystems. Unter dem Interaktionsbereich eines Trackingsystems wird im Folgenden der Bereich im dreidimensionalen Raum bezeichnet, der von mindestens zwei Kameras aufgenommen werden kann, also die Schnittmenge der Sichtvolumina der Kameras. Um den Interaktionsbereich möglichst groß zu gestalten, sind für die Platzierung von zwei Kameras folgende Überlegungen zu berücksichtigen:

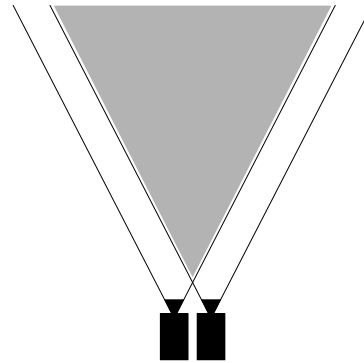
- Die optischen Achsen der Kameras sollten sich annähernd schneiden, sodass die Kameras ungefähr denselben Bereich im Raum, allerdings aus unterschiedlichen Perspektiven, erfassen.
- Ist die erste Forderung erfüllt, gilt: Je näher die Kameras zueinander positioniert stehen, desto größer ist das Interaktionsvolumen.

Die erste Forderung betrifft also eher die Anforderung an die Rotation der Kamera, die zweite Forderung betrifft die Translation. Wenn die Kameras nicht so orientiert sind, dass sich die optischen Achsen näherungsweise schneiden, kommt es im Extremfall dazu, dass überhaupt kein gemeinsamer Bereich der Sichtvolumina der Kameras existiert und somit kein Interaktionsvolumen definiert wird. Wie in Abbildung 2.1 grob skizziert, erhöht eine enge Stellung der Kameras (b) den Interaktionsbereich dadurch, dass bei zunehmender Entfernung von den Kameras der Interaktionsbereich größer wird, bei einer eher rechtwinkligen Anordnung der Kameras (a) hingegen kleiner wird.

Allerdings wird der Blickwinkel der Kameras durch die sehr enge Stellung und ähnliche Ausrichtung nahezu identisch. Eine solche Konstellation der Kameras macht eine



(a) Die Kameras stehen ungefähr im 90° Winkel zueinander. Der Interaktionsbereich ist nahe bei den Kameras breit und verjüngt sich mit zunehmender Entfernung von den Kameras.



(b) Die Kameras sind ungefähr parallel ausgerichtet. Der Interaktionsbereich wird in zunehmender Entfernung der Kameras breiter.

Abbildung 2.1. Unterschiedliche Platzierung und Ausrichtung zweier Kameras in der Aufsicht. Der Interaktionsbereich ist grau unterlegt.

präzise Rekonstruktion im Dreidimensionalen schwieriger, insbesondere gilt dies für die Genauigkeit der rekonstruierten Koordinaten in die Tiefe, also entlang der optischen Achsen. Da die Kameras in diesem Fall ähnlich orientiert sind, sind die zur Bildebene orthogonalen optischen Achsen nahezu parallel. Das bedeutet, dass die Verschiebungen eines Objekts annähernd parallel zu den beiden optischen Achsen nur eine dazu relativ geringe Veränderung auf den beiden Kamerabildern bewirkt und es somit schwierig ist, diese exakt zu bestimmen. Besser für die Präzision in Tiefe, Höhe und Breite gleichzeitig ist eine Ausrichtung der Kameras ungefähr im 90° Winkel, wie in Abbildung 2.1(a) dargestellt.

Für die in dieser Arbeit beschriebenen Algorithmen und Anwendungen ist es im Allgemeinen vorteilhaft, die Kameras aufgrund der obigen Erläuterungen nicht zu eng beieinander zu platzieren. Da dennoch ein möglichst großer Interaktionsbereich auch in die Tiefe erreicht werden soll, muss hier je nach Anwendung ein Kompromiss gefunden werden.

Das Ausrichten der Kameras ist also ein wichtiger Aspekt bei der Installation eines Stereokamerasystems. Um dem Benutzer die Installation zu vereinfachen, verwenden daher manche kommerziellen optischen Trackingsysteme [NDI 06] einen festen Aufbau, das heißt, dass die Kameras fest miteinander verbunden sind. Allerdings schränkt ein fester Aufbau der Kameras die Anpassung des Interaktionsbereichs des Stereokamerasystems auf die Anwendung ein.

Art und Qualität der Kameras, der Objektive und der Beleuchtungstechnik haben ebenfalls Einfluss auf die Qualität der rekonstruierten Ergebnisse. Insbesondere gute Objektive erleichtern die Kalibrierung der einzelnen Kameras und des Stereokamerasystems, siehe Abschnitt 2.3.

2.1.2 Verwendung von Stereosystemen

Neben zahlreichen Anwendungen wie zum Beispiel 3D-Scannern oder Systemen zur 3D-Rekonstruktion komplexer Objekte [Neugebauer 97], [Rusinkiewicz 02] gibt es auch zahlreiche verschiedene Ansätze von Trackingsystemen, die sich Stereokamerasysteme zu Nutze machen. Unter anderem sind dies Infrarottrackingsysteme wie unter [ART 06a], [NDI 06], [Vicon 06] beschrieben, auf die in Abschnitt 2.4.1 näher eingegangen wird. Je nach System werden diese sowohl für die Verfolgung von Objekten im VR-Bereich (zum Beispiel Shutterbrille oder Interaktionsgerät) als auch beim Motion Capturing, typischerweise für die Verfolgung menschlicher Bewegung, eingesetzt.

Weitere Systeme [Claron 06], [Orin 06] nutzen ebenfalls Marker, allerdings nicht nur Infrarotmarker. Des Weiteren gibt es Trackingsysteme, die ganz auf Marker verzichten, und beispielsweise versuchen die Position und die Zeigerichtung einer menschlichen Hand [Malerczyk 04] oder die Position des Kopfes [Park 06] zu erfassen.

Um eine Rekonstruktion aus verschiedenen Sichten mit nur einer Kamera möglich zu machen kann man außerdem Spiegel einsetzen [Fukushige 06]. Dabei wird die einzige Kamera so ausgerichtet, dass sie ein Objekt sowohl direkt als auch dessen Spiegelbild erfassen kann. Auf diese Art entstehen zwei verschiedene Ansichten. Die Geometrie zur Verarbeitung dieser beiden Ansichten ähnelt der Verarbeitung der Ansichten von verschiedenen Kameras. Dieser Ansatz eignet sich dann, wenn im Interaktionsbereich ein Spiegel platziert werden kann und dieser bei der Interaktion nicht verdeckt wird.

2.2 Kameramodell und Geometrie mehrerer Kameras

Die mathematische Beschreibung insbesondere von Mehrkamerasystemen wurde in den letzten Jahren immer weiter untersucht, sodass umfassende Literatur [Sonka 99], [Hartley 04] vorhanden ist. Als erster Schritt für die mathematische Beschreibung eines Stereokamerasystems bedarf es der mathematischen Modellierung einer einzelnen Kamera. Das hier verwendete mathematische **Kameramodell** basiert auf dem Prinzip einer **Lochkamera** und wird für präzise Anwendungen durch Parameter für die Linsenverzeichnung erweitert. In den folgenden Kapiteln steht der Begriff Kameramodell im Allgemeinen für das mathematische Modell, ebenso wie der Begriff Kamera dafür verkürzt verwendet wird. Nur in den Abschnitten, die sich mit der Hardware auseinandersetzen bezeichnen die Begriffe physikalische Kameras.

Die Parameter des Kameramodells, welches der Implementierung in dieser Arbeit zugrunde liegt, sind die folgenden extrinsischen Parameter:

$T \in \mathbb{R}^3$	der Brennpunkt und Ursprung des Kamerakoordinatensystems in Weltkoordinaten und
$R \in SO_3$	die Rotation der Kamera im Weltkoordinatensystem.

Intrinsische Parameter sind:

- $f \in \mathbb{R}$ die Brennweite der Kamera,
- $P = (u_0, v_0) \in \mathbb{R}^2$ der Hauptpunkt (**Principal Point**), also der Schnittpunkt zwischen optischer Achse und der dazu orthogonalen Bildebene,
- $r_u, r_v \in \mathbb{R}$ Skalierungsfaktoren in x- und y-Richtung auf der Bildebene,
- $s \in \mathbb{R}$ ein Verzerrungsfaktor (**skew**) und
- $\kappa \in \mathbb{R}$ ein Koeffizienten für die radiale Linsenverzeichnung.

Die intrinsischen Parameter werden häufig unterschiedlich benannt und es werden auch verschiedene Parameter benutzt. Beispielsweise wird für die Modellierung der radialen Linsenverzeichnung teilweise mehr als ein Koeffizient verwendet, um diese genauer angeben zu können. Da die Wirkung weiterer Linsenverzeichnungsparameter aber zunehmend geringer wird, wird aus Effizienzgründen oft auf diese verzichtet. Abbildung 2.2 zeigt ein Lochkameramodell, ohne Berücksichtigung der Linsenverzeichnung.

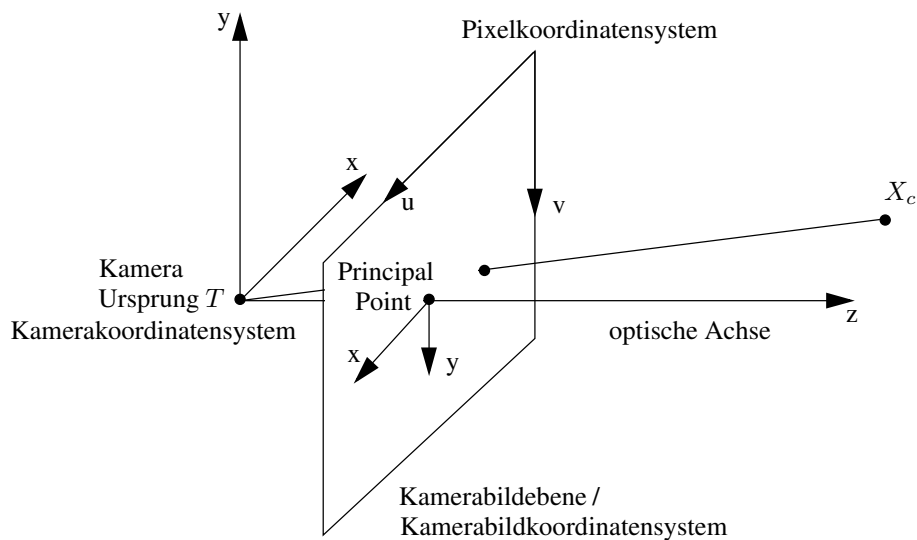


Abbildung 2.2. Das Lochkameramodell zeigt das Kamera-, Kamerabil-, und Pixelkoordinatensystem und einen auf die Kamerabildebene projizierten Punkt X_C .

Unter der Voraussetzung, dass alle Kameraparameter bekannt sind, lässt sich ein Punkt $X_W = (x_w, y_w, z_w) \in \mathbb{R}^3$ im Weltkoordinatensystem auf die Bildebene der Kamera projizieren und in Pixelkoordinaten angeben. Dazu wird der Punkt X_W zunächst in Kamerakoordinaten X_C transformiert:

$$X_C = R^{-1} \cdot X_W - R^{-1} \cdot T. \quad (2.1)$$

Der Punkt $X_C = (x_c, y_c, z_c) \in \mathbb{R}^3$ in Kamerakoordinaten wird daraufhin auf die Kamerabildebene, die im Abstand f zum Ursprung liegt, projiziert. Der projizierte Punkt ist dann mit Hilfe des Strahlensatzes einfach als $(\frac{-f \cdot x_c}{z_c}, \frac{-f \cdot y_c}{z_c})^T$ zu berechnen. In homogenen

Koordinaten ist dies der Punkt $(\frac{-f \cdot x_c}{z_c}, \frac{-f \cdot y_c}{z_c}, 1)^T$. Die Parameter u_0, v_0, r_u, r_v und s lassen sich in einer Matrix zusammenfassen, sodass die Transformation in Pixelkoordinaten durch

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \underbrace{\begin{pmatrix} r_u & s & u_0 \\ 0 & r_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_K \begin{pmatrix} \frac{-f \cdot x_c}{z_c} \\ \frac{-f \cdot y_c}{z_c} \\ 1 \end{pmatrix} = \begin{pmatrix} -f \cdot r_u & -f \cdot s & u_0 \\ 0 & -f \cdot r_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{pmatrix} \quad (2.2)$$

erfolgen kann. Diese Matrizen werden unter anderem als Finite Projektive Kamera beziehungsweise **Kamerakalibrierungsmatrix** bezeichnet. Der endgültige, projizierte Punkt in Pixelkoordinaten ist gegeben durch $(u, v)^T = (\frac{u'}{w'}, \frac{v'}{w'})^T \in \mathbb{R}^2$. Das Pixelkoordinatensystem ist nicht diskret, da hier ein 2D-Punkt in Pixelkoordinaten beispielsweise auch der Schwerpunkt einer segmentierten Region sein kann, der durch Summierung und Gewichtung im Allgemeinen nicht $\in \mathbb{N}_0^2$ ist.

Bei obiger Berechnung der Pixelkoordinaten wurde keine Linsenverzeichnung berücksichtigt. Um diese zu berücksichtigen, werden die Koordinaten $u_u = \frac{-f \cdot x_c}{z_c}$ und $v_u = \frac{-f \cdot y_c}{z_c}$ des projizierten Punktes mit

$$\begin{aligned} u_d &= \frac{2u_u}{\sqrt{1 - 4(u_u^2 + v_u^2)\kappa}} \\ v_d &= \frac{2v_u}{\sqrt{1 - 4(v_u^2 + v_u^2)\kappa}} \end{aligned} \quad (2.3)$$

transformiert [Lenz 87] und dann weiter auf Pixelkoordinaten transformiert. Das u im Index steht dabei für „undistorted“, beziehungsweise das d für „distorted“. Die Umkehrfunktionen der beiden obigen Gleichungen sind:

$$\begin{aligned} u_u &= \frac{u_d}{1 + (u_d^2 + v_d^2)\kappa} \\ v_u &= \frac{v_d}{1 + (v_d^2 + v_d^2)\kappa} \end{aligned} \quad (2.4)$$

Der radiale Linsenverzeichnungsparameter κ ist für die verwendeten Objektivs mit tonnenförmiger Verzeichnung negativ, sodass die Ausdrücke unter den Wurzeln in den Gleichungen (2.3) nicht negativ werden können. Ergibt sich für κ ein positiver Wert, müssen die entsprechenden Werte von u_u und v_u bei jeder Berechnung berücksichtigt werden. Für die Modellierung der Linsenverzeichnung gibt es verschiedene andere Ansätze [Tsai 87], [Presscott 97], die insbesondere bei Berücksichtigung einer dezentrierten radialen Verzeichnung [Jain 95] komplexer und aufwändiger in der Berechnung werden.

Für die Rekonstruktion von 3D-Punkten ist es notwendig, aus den Pixelkoordinaten $(u, v)^T$ eines Punktes, beispielsweise dem Schwerpunkt einer segmentierten Region, wieder 3D-Koordinaten zu gewinnen. Dabei bildet ein Punkt auf der Bildebene auf eine Halbgerade im Raum ab. Die Halbgerade beginnt im Kameraursprung und schneidet den Punkt auf der Bildebene. Dazu wird $(u, v, 1)^T$ mit K^{-1} multipliziert und danach um die

Linsenverzeichnung mit den oben aufgeführten Gleichungen (2.4) korrigiert, sodass er in Kamerabildkoordinaten

$$U = \left(\begin{array}{c} \frac{u_d}{1+(u_d^2+v_d^2)\kappa} \\ \frac{v_d}{1+(u_d^2+v_d^2)\kappa} \end{array} \right) \quad \text{mit} \quad \left(\begin{array}{c} u_d \\ v_d \end{array} \right) = \left(\begin{array}{c} \frac{1}{r_u}((u - u_0) - \frac{s}{r_v}(v - v_0)) \\ \frac{1}{r_v}(v - v_0) \end{array} \right) \quad (2.5)$$

vorliegt. Daraus ergibt sich $(U, f)^T \in \mathbb{R}^3$ im Kamerakoordinatensystem, als ein Punkt auf der Halbgeraden.

Für die Geometrie mit zwei Kameras sind die **Fundamentalmatrix** $F \in \mathbb{R}^{3 \times 3}$ und die **Essentialmatrix** $E \in \mathbb{R}^{3 \times 3}$ nützliche Hilfsmittel. Die Fundamentalmatrix drückt die Korrespondenz eines homogenen Punktes $(u_1, v_1, 1)^T$ der ersten Kamera mit einem homogenen Punkt $(u_2, v_2, 1)^T$ der zweiten Kamera durch

$$(u_2, v_2, 1)F(u_1, v_1, 1)^T = 0 \quad (2.6)$$

aus. Die Fundamentalmatrix hat Rang 2. Analoges gilt für die Essentialmatrix und Punkte $K^{-1}(u_1, v_1, 1)^T$ und $K^{-1}(u_2, v_2, 1)^T$. Eine detaillierte Beschreibung der Eigenschaften dieser Matrizen findet sich unter anderem in [Hartley 04]. Die Fundamentalmatrix kann natürlich keine Linsenverzeichnung berücksichtigen.

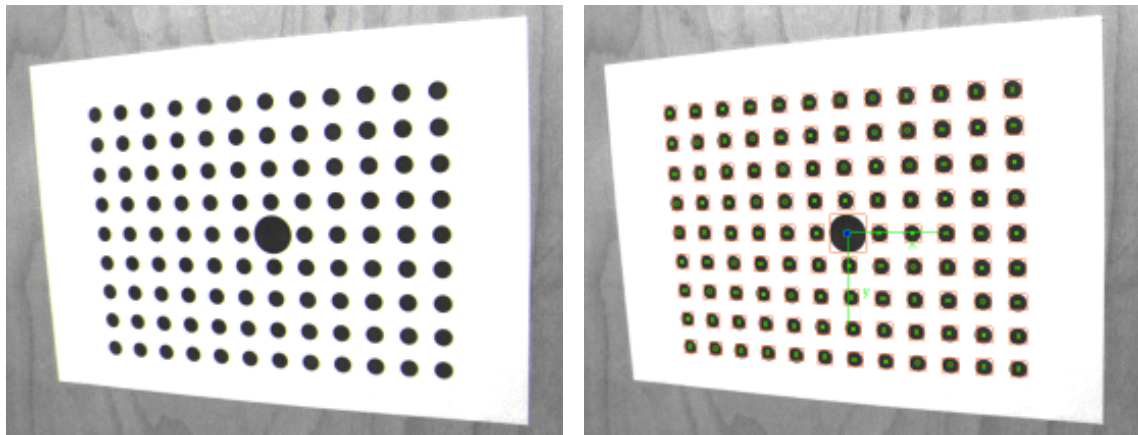
Wie diese und weitere Eigenschaften genutzt werden, um 3D-Punkte aus zwei Kamerabildern zu rekonstruieren, wird in Kapitel 3 weiter erläutert.

2.3 Kalibrierung des Kamerasystems

Eine elementare Voraussetzung für das Tracking mit einem Stereokamerasystem ist die Kalibrierung der Kameras, da diese erst eine effiziente und genaue 3D-Rekonstruktion ermöglicht. Dabei werden zunächst einmalig die **intrinsischen** Parameter jeder Kamera bestimmt und dann nach jedem Aufbau des Kamerasystems die **extrinsischen** Kameraparameter, also die Ausrichtung der Kameras zueinander.

Die intrinsische Kamerakalibrierung erfolgt meist durch die Aufnahme eines Musters (**Pattern**), typischerweise mit Kreisscheiben oder Rechtecken, sodass einzelne Punkte im Kamerabild als Schwerpunkte segmentierter Regionen bestimmt werden können. Abbildung 2.3 zeigt die Aufnahme solch eines Patterns. Zusammen mit dem Wissen über die realen Abstände dieser Punkte auf dem Pattern, lassen sich dann die intrinsischen Kameraparameter eines wie in Abschnitt 2.2 beschriebenen und zugrundegelegten Kameramodells berechnen. Solche Verfahren sind unter anderem in [Tsai 86] und [Lenz 87] beschrieben. In [Zhang 98], ist darüber hinaus beschrieben, wie mehrere Aufnahmen des Patterns, ohne festgelegte Positionen, in einem Optimierungsprozess genutzt werden können, um die intrinsischen Parameter zu berechnen.

Wird beispielsweise das Pattern aus Abbildung 2.3 verwendet, um eine Kalibrierung nach der Methode von [Zhang 98] durchzuführen, werden in jeder der n Aufnahmen des Patterns 108 Punkte erfasst, deren reale 3D-Koordinaten in einem Koordinatensystem, das den Ursprung auf dem Pattern hat, bekannt sind (dabei gilt $z = 0$ für alle Punkte) und deren Pixelkoordinaten aus dem Bild berechnet werden. Für die Optimierung liegen $108n$



(a) Kalibrierungspattern.

(b) Kalibrierungspattern mit Overlays.

Abbildung 2.3. Aufnahme eines Kalibrierungspatterns ohne Überlagerungen des Kamerabildes (Overlays) und mit Overlays nach der Segmentierung, als Visualisierung der Segmentierungsergebnisse.

Korrespondenzen zwischen 3D- und 2D-Punkten vor, aus denen mit Hilfe der nichtlinearen Optimierungsmethode nach Levenberg und Marquardt [Marquardt 63], [Press 02] die intrinsischen Kameraparameter berechnet werden können. Als Nebenprodukt wird die Position und Orientierung jedes Patterns berechnet.

Sind die intrinsischen Parameter jeder Kamera in einem Stereosystem bestimmt, kann die extrinsische Kamerakalibrierung folgen. Ein Verfahren für diese Kalibrierung ist der in [Azarbayejani 95] beschriebene halbautomatische Ansatz, der als Grundlage für diese Arbeit herangezogen wurde. Der Ansatz ist deshalb halbautomatisch, weil der Benutzer zunächst ein Gerät im Interaktionsvolumen bewegen muss, das einen eindeutig identifizierbaren Punkt in beiden Kameras erzeugt, sodass 2D-Punkt-Korrespondenzen (U_i^1, U_i^2) , $1 \leq i \leq m$, zum Beispiel mit $m = 300$, automatisch vorliegen. Als solches Gerät eignet sich zum Beispiel eine Leuchtdiode, insbesondere eine infrarote, da diese in Kameras (ohne Infrarot-Sperrfilter) klar zu erkennen ist.

Mit linearen Methoden, insbesondere der **Singulärwertzerlegung** (SVD) lässt sich nach [Hartley 92] eine Annäherung für die Transformation zwischen zwei Kameras unter Nutzung der Eigenschaften der Essentialmatrix berechnen. Das Ergebnis dieses Verfahrens wird als Startwert für eine nichtlineare Optimierung mit dem bereits oben erwähnten Levenberg-Marquardt Algorithmus verwendet. Das Prinzip der Kalibrierung beruht dabei darauf, dass ein 2D-Punkt U_i^1 mit Hilfe der aktuell bestimmten Transformation und den bereits bekannten intrinsischen Kameraparametern entsprechend der Formeln in Abschnitt 2.2 auf einen Punkt \tilde{U}_i^2 im anderen Kamerabild abgebildet wird. Das Residuum,

$$r_i = \|\tilde{U}_i^2 - U_i^2\|^2, \quad (2.7)$$

welches den Abstand zum originalen Punkt U_i^2 darstellt, wird für alle Punkte aufsummiert. Die Summe der Residuen ist die Kostenfunktion, die dann iterativ minimiert wird.

Werden wie hier beschrieben nur Ein-Punkt-Korrespondenzen verwendet, kann auf diese Weise zwar Positionierung und Orientierung der Kameras berechnet werden, es fehlt jedoch ein Skalierungsfaktor. Dieser kann nachträglich, ebenso wie ein spezielles Weltkoordinatensystem, bestimmt werden. Beispielsweise platziert man dafür im Interaktionsbereich ein vermessenes (Kalibrierungs-) Objekt mit mindestens drei nicht kollinearen, identifizierbaren und vermessenen Punkten, typischerweise in rechtwinkliger Anordnung. Da das Kamerasystem bereits kalibriert ist, lassen sich beliebig skalierte Distanzen zwischen den Punkten berechnen und mit den tatsächlich gemessenen Distanzen vergleichen, sodass das Trackingsystem auf die realen Distanzen geeicht wird. Das Weltkoordinatensystem wird relativ zum Kalibrierungsobjekt platziert.

Das hier verwendete Verfahren verlangt insbesondere keinen bestimmten Kameraaufbau, wie zum Beispiel gleiche Orientierung der Kameras, und ist weitestgehend flexibel. Soll ein System mit mehr als zwei Kameras kalibriert werden, gibt es die Möglichkeit je zwei Kameras als Paar zu kalibrieren, wobei natürlich aufgenommene Messpunkte mehrfach genutzt werden können, wenn sie in mehreren Kameras erfasst werden konnten. Des Weiteren gibt es auch für Systeme mit drei oder mehr Kameras spezielle Ansätze, die unter anderem in [Hartley 04] festgehalten sind.

2.4 Optische Trackingtechnologien

In den folgenden Abschnitten werden verschiedene kamerabasierte Verfahren beschrieben, die als technologische Basis für die in dieser Arbeit beschriebenen Algorithmen sinnvoll einsetzbar sind. Den für diese Arbeit wichtigsten Bereich optischer Trackingsysteme stellen Infrarottrackingsysteme mit retroreflektierenden Markern oder Dioden dar, die im folgenden Abschnitt beschrieben werden.

2.4.1 Infrarottracking

Im Infrarotbereich arbeitende Stereokamerasysteme für Trackingaufgaben (**Infrarot-trackingsysteme**) setzen sich als Trackingsysteme für Virtual und Augmented Reality Anwendungen im Innenbereich zunehmend durch. Vorteile gegenüber anderen Technologien sind insbesondere, dass zu verfolgende Personen oder Objekte nicht verkabelt werden müssen, sie ein großes Interaktionsvolumen und eine hohe Genauigkeit im Vergleich zu anderen Trackingtechnologien haben. Die zu verfolgenden Objekte sind häufig feste Körper, an denen **passive** oder **aktive** Marker angebracht werden. Passive Marker sind retroreflektierende Kugeln, die Infrarotlicht von neben den Kameras befestigten Scheinwerfern auf die Kamerabilder reflektieren. Alternativ dazu werden auch Infrarotdioden als aktive Marker verwendet, die ohne Scheinwerfer von mit Infrarotfiltern ausgestatteten Kameras deutlich wahrgenommen werden können.

Soweit sich die bei allen optischen Trackingverfahren existierende Verdeckungsproblematik einschränken lässt, das heißt eine Verdeckung des zu verfolgenden Objekts nicht dauerhaft stattfindet, sind Infrarotsysteme außerdem robust.

Markerbasierte Ansätze sind nicht für alle Anwendungen geeignet oder überhaupt verwendbar, da das Anbringen von Markern in manchen Situationen unpraktisch ist oder nicht durchgeführt werden kann. Für zahlreiche Anwendungen ist jedoch eine Verwendung von Markern unproblematisch und die dadurch im Allgemeinen entstehenden Vorteile, wie eine hohe Framerate und die Robustheit des Trackings, sind unverzichtbar.

Infrarottrackingsysteme existieren inzwischen sowohl als Prototypen im wissenschaftlichen Umfeld [Madritsch 96] [Dorfmueller 99b], [Ribo 01], [Magneau 02a], [Mulder 03], [Rasmussen 06] als auch als kommerzielle Systeme, zum Beispiel ARTtrack [ART 06a], Polaris [NDI 06] oder Vicon [Vicon 06]. Der Fokus in der Entwicklung dieser Systeme liegt meist auf der zu erreichenden Genauigkeit, welche insbesondere für Anwendungen im Bereich Augmented Reality oder speziell im medizinischen Umfeld ein wichtiges Kriterium für die Verwendung darstellt. Bessere Systeme geben eine statische Positionsgenauigkeit mit einem Fehler unter 1 mm unter vorgegebenen Bedingungen an.

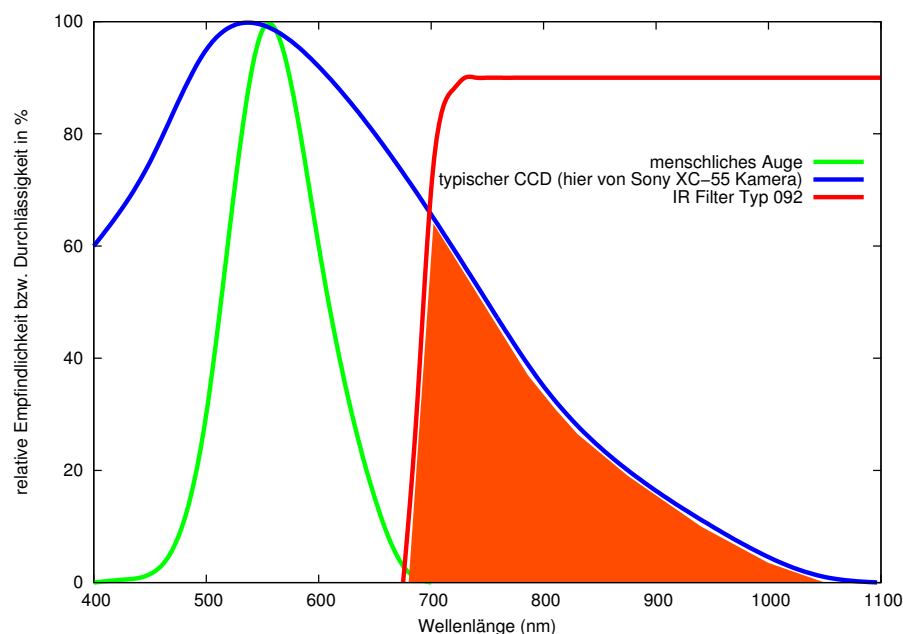


Abbildung 2.4. Relative spektrale Empfindlichkeiten beziehungsweise Durchlässigkeiten. Der ausgefüllte Bereich zwischen der CCD Kurve und der des Infrarotfilters wird von den Kameras erfasst.

Die in den folgenden Kapiteln beschriebenen Algorithmen wurden nicht ausschließlich, aber im Wesentlichen, als Teil des Infrarottrackingsystems **EOS** entwickelt [Schwald 02a], [Schwald 04b] und sind Bestandteil dieses Systems. Der Vorgänger dieses Systems geht auf Arbeiten von Klaus Dorfmueller-Ulhaas [Dorfmueller 98, Dorfmueller 99a] zurück, dessen System in Bezug auf die Grundlagen, wie der Kamerakalibrierung und der Rekonstruktion, dem EOS System ähnelt, aber sich ansonsten insbesondere bei den in den Kapiteln 4 und 5 beschriebenen Zwei- und Multipunktmodellen und der in Kapitel 6 beschriebenen Verarbeitung dynamischer Modelle unterscheidet.

Anhand von EOS wird im Folgenden das Prinzip und die generelle Vorgehensweise eines Infrarottrackingsystems dargestellt. Verwendet werden monochrome (s/w) Kameras, da Farbbilder für die Verwendung mit Infrarotlicht keinen Vorteil bringen und im Gegenteil häufig mit Infrarotsperrfiltern ausgestattet sind. Soweit also in Kameras keine Infrarotsperrfilter eingebaut sind, können diese im für Menschen nicht sichtbaren Infrarotbereich Bilder aufzeichnen. Abbildung 2.4 zeigt den Bereich des Lichtspektrums, den ein typischer CCD-Chip im Vergleich zum menschlichen Auge erfasst. Spezielle, aber auch teure Infrarotkameras sind mit Chips ausgestattet, die im Infrarotbereich noch deutlich empfindlicher sind, als ein CCD-Chip. Des Weiteren kann man die Kameras mit Infrarotfiltern ausstatten, die die gegensätzliche Wirkung von Infrarotsperrfiltern haben: nur Licht ab einer gewissen Wellenlänge kann die Filter durchdringen. Diese Filter werden eingesetzt, um den Hintergrund des Kamerabilds, der kein Infrarotlicht ausstrahlt oder reflektiert, möglichst schwarz erscheinen zu lassen.

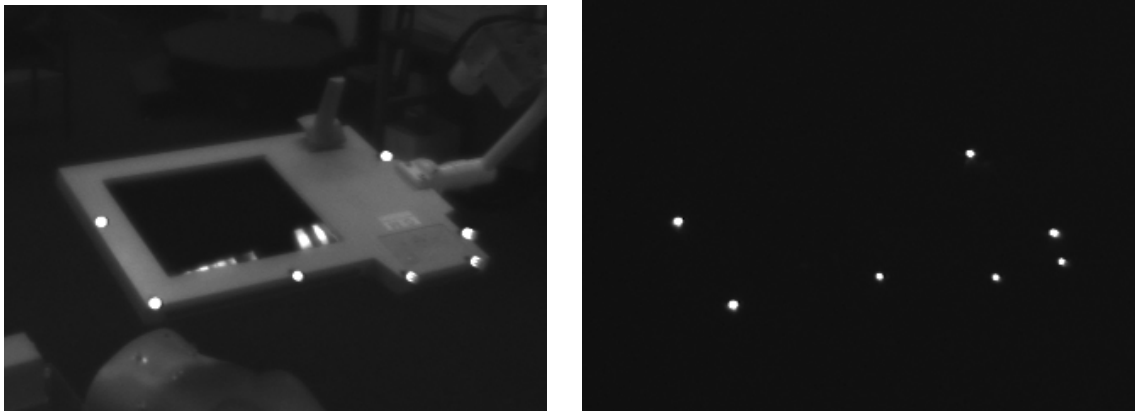
Die Empfindlichkeit der Chips gegenüber Infrarotlicht macht man sich bei einem Infrarottrackingsystem zu Nutze, indem man meist kugelförmige, mit einer retroreflektierenden Folie überzogene Objekte mit an oder neben den Kameras montierten Lichtquellen bestrahlt oder anblitzt. Durch die retroreflektierende Eigenschaft der Folie wird das auf diese Marker gestrahlte Licht direkt auf die Kameras zurückgeworfen, sodass die Marker als sehr helle Lichtpunkte im Kamerabild erscheinen und sich aufgrund der Infrarotfilter auf den Kameras vom Hintergrund deutlich abheben. Alternativ zu diesem sogenannten passiven Ansatz, können auch aktive Infrarotdioden statt der retroreflektierenden Marker verwendet werden.

Infrarotdioden haben gegenüber retroreflektierenden Kugeln den Vorteil, dass sie sich im Allgemeinen unauffälliger anbringen lassen. Dem steht allerdings zum einen der Nachteil gegenüber, dass die Dioden mit Strom versorgt werden müssen und zum anderen, dass der Abstrahlwinkel der Dioden beschränkt ist, also die Abstrahlung vor allem nicht kugelförmig ist. Durch den beschränkten Abstrahlwinkel erscheinen die Dioden im Kamerabild meist nicht kreisförmig, was die Verarbeitung erschwert, oder noch schlechter, die Dioden bei ungünstiger Orientierung, gar nicht zu erkennen sind. Kann man bei einer Anwendung davon ausgehen, dass eine ungünstige Ausrichtung der Dioden kaum vorkommt und stellt die Stromversorgung kein Problem dar, sind Infrarotdioden eine gute Alternative zum passiven Ansatz. Abbildung 2.5 zeigt ein Kamerabild, auf dem ein zu verfolgendes Objekt mit Infrarotdioden versehen worden ist.

Um die daran anschließende Bildverarbeitung noch einfacher zu machen, können auf die Objektiv der Kameras Infrarotfilter montiert werden, sodass der sichtbare Bereich des Lichts für die Chips abgeschnitten wird. Abbildung 2.4 zeigt den verbleibenden Bereich von Wellenlängen, den die Kameras erfassen und Abbildung 2.5 zeigt den Vergleich zwischen zwei Kamerabildern mit und ohne Infrarotfilter.

Der Ausgangspunkt für die Bildverarbeitung ist ein Grauwertbild (jedes Pixel erhält einen Wert zwischen 0 (schwarz) und 255 (weiß)), welches überwiegend nahezu schwarz ist, mit einzelnen nahezu weißen, kreisförmigen Bereichen für jeden sichtbaren Marker. Für die **Segmentierung** der Marker bieten sich vor allem zwei Methoden an:

- eine Erkennung der nahezu kreisförmigen Bereiche, zum Beispiel durch Hough-Transformation, oder



(a) Kameravision ohne Infrarotfilter.

(b) Kamerabild durch Infrarotfilter.

Abbildung 2.5. Kamerabild mit und ohne Infrarotfilter zeigt ein mit aktiven Markern versehenes Objekt.

- eine Schwerpunktbestimmung von Regionen, deren Pixel alle einen Schwellwert für Intensität (Grauwert) überschreiten.

Beim zweiten, intensitätsbasierten Ansatz wird das Kamerabild zeilenweise nach Pixeln untersucht, deren Wert über einem Segmentierungsgrenzwert l mit $0 \leq l \leq 255$ liegt. Bei idealen Lichtverhältnissen, also einheitlich, beinahe schwarzem Hintergrund, kann der Schwellwert sehr tief gewählt werden, zum Beispiel $l = 40$, bei ungünstigeren Lichtverhältnissen muss er nahe bei 255 gewählt werden. Aufeinanderfolgende Pixel über dem Schwellwert werden zu horizontalen Liniensegmenten zusammengefügt, die dann bei der Verarbeitung der folgenden Zeilen zu ganzen Regionen zusammengefasst werden können. Eine Skizzierung dieses Vorgehens ist in Abbildung 2.6 dargestellt. Dieses Vorgehen ist sehr effizient (wenige Millisekunden), sodass die Bildverarbeitung eine im Vergleich zur noch folgenden 3D-Rekonstruktion und Modellerkennung vernachlässigbare Rechenzeit einnimmt und damit auch die Latenz des Systems nicht wesentlich erhöht. Im Gegensatz dazu ist die oben erwähnte Hough-Transformation im Allgemeinen rechenintensiver.

Für die aus der Verarbeitung eines Bildes hervorgehenden Regionen wird der Schwerpunkt, der mit den Intensitäten der in der Region enthaltenen Pixel gewichtet wird, berechnet. Fasst man ein Pixel q als ein Tripel $q = (u_q, v_q, w_q)$ mit seiner Position (u_q, v_q) und seiner Intensität w_q auf, so berechnet sich der Schwerpunkt $U = (u_u, u_v)^T$ der Region R , wobei R die Menge der Bildkoordinaten seiner Pixel darstellt, durch

$$U = \begin{pmatrix} u_u \\ u_v \end{pmatrix} = \frac{\sum_{q \in R} w_q \begin{pmatrix} u_q \\ v_q \end{pmatrix}}{\sum_{q \in R} w_q}. \quad (2.8)$$

Die Menge $\mathcal{U} = \{U_1, \dots, U_p\}$ der einzelnen Schwerpunkte von p Regionen entspre-

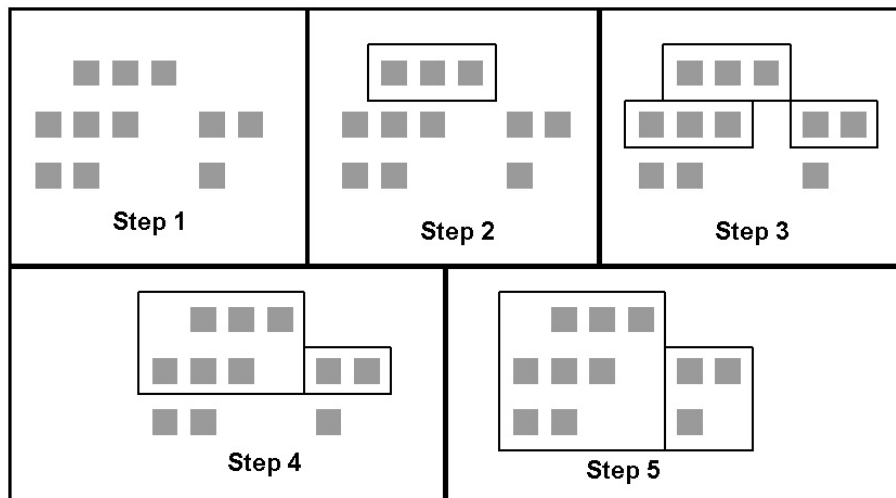


Abbildung 2.6. Zusammenfügung von Pixeln zu Liniensegmenten und dann zu Regionen bei einer intensitätsbasierten Segmentierung. Die Quadrate repräsentieren Pixel, deren Grauwert über dem Schwellwert liegen. Das Bild wurde aus [Figueiredo 02] entnommen.

chen den in Abschnitt 2.2 eingeführten 2D-Punkten in Pixelkoordinaten, die in Kapitel 3 weiter verarbeitet werden.

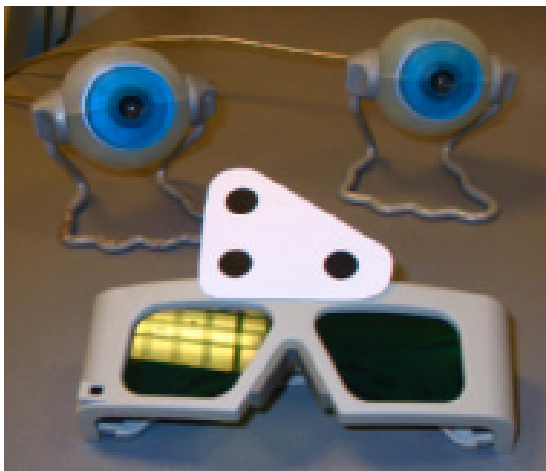
2.4.2 Pattertracking

Neben dem Infrarottracking mit passiven oder aktiven Markern, gibt es andere weitverbreitete markerbasierte optische Trackingtechnologien, deren Marker meist ebene Muster darstellen. Weit verbreitet im AR-Bereich ist ARToolKit [Kato 99], [Kato 06], welches sich quadratischer Marker bedient, die innerhalb eines schwarzen Rahmens identifizierbare und eindeutig orientierbare Muster haben sollten. Die Abbildung 3.2(a) zeigt ein Beispiel für einen solchen Marker.

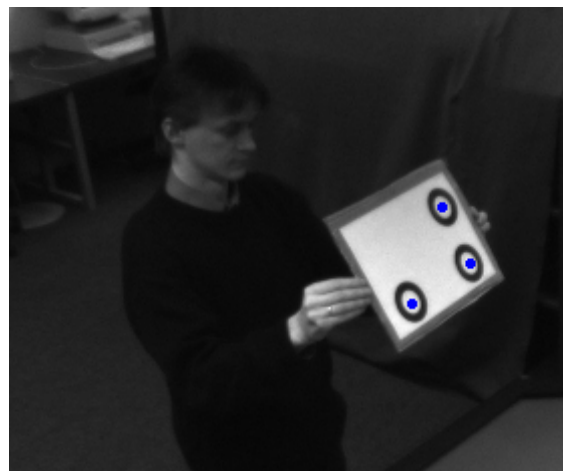
Der Vorteil solcher Trackingverfahren gegenüber dem oben beschriebenen Infrarottracking besteht vor allem darin, dass die Hardware weniger aufwändig ist und dass ein einzelner Marker sowie eine einzelne Kamera ausreicht, um Translation und Rotation des Markers relativ zur Kamera bestimmen zu können. Bei ARToolKit-artigen Markern liegt außerdem mehr Information, als bei nicht unterscheidbaren, symmetrischen Markern vor und sollte auch genutzt werden. Diese detailreicheren Marker haben auch Nachteile: Zum einen müssen sie so groß sein, beziehungsweise die Kamera so dicht am Marker sein, dass das Muster auf dem Marker erkannt werden kann, zum anderen ist die Abhängigkeit von den Lichtverhältnissen größer, als bei Tracking im Infrarotbereich. Für Anwendungen, die ein präzises Tracking benötigen, reicht die Qualität der berechneten Orientierung häufig nicht aus, sodass mehrere Marker verwendet werden müssen, die einen beträchtlichen Bereich des Kamerabildes einnehmen.

Andere Ansätze mit flachen Markern sind dem mathematischen Prinzip beim Infra-

rottracking ähnlicher und könnten von den Algorithmen in den Kapiteln 4 und 5 profitieren. Abbildung 2.7(a) zeigt ein Pattern, das aus drei schwarzen Punkten auf einer dreiecksförmigen weißen Fläche besteht, welches dann von einem Stereokamerasystem verfolgt wird [Mulder 03]. Die Erkennung der Punkte in den Kamerabildern basiert dabei im Wesentlichen auf einer Segmentierung mit Kantendetektion, die aufgrund der Nähe zu den Kameras, bei der angestrebten Verwendung als Desktoptrackingsystem, als unproblematisch beschrieben wird. Die Abbildung 2.7(b) zeigt die Verfolgung kreisförmiger Marker mit einem Stereosystem, das auf den Grundlagen der vorliegenden Arbeit basiert.



(a) Flaches Pattern bestehend aus drei Punkten [Mulder 03].



(b) Kreise in einer rechtwinkligen Anordnung als Marker.

Abbildung 2.7. Pattern, die aus punktförmigen Elementen bestehen.

Eine weitere Art von Markern wird in [van Rhijn 04] zur Verfolgung von Eingabegeräten für VR/AR-Umgebungen präsentiert. Die Marker sind balkenförmig und in einer bekannten Geometrie angeordnet, in diesem Fall auf einem Würfel, sodass sie in ihrer Gesamtheit, also auch wenn nur manche der Marker zu erkennen sind, identifizierbar sind und daraus Position und Orientierung berechnet werden können.

Es gibt eine Vielzahl weiterer Marker, die für die Verfolgung von Objekten eingesetzt werden. Das für diese Arbeit interessante Unterscheidungsmerkmal zwischen den verschiedenen Markern spaltet diese in zwei Gruppen:

- Marker, die aus nicht unterscheidbaren Markierungen bestehen, sodass deren Identifikation und die Bestimmung von Translation und Rotation über die geometrische Anordnung erfolgt. Beispiele wurden in Abbildung 2.7 gezeigt.
- Marker, die schon einzeln identifizierbar sind. Aus diesen können auch meist direkt Translation und Rotation bestimmt werden. Typisch für diese Gruppe sind ARTool-Kit Marker. Solche Marker werden auch häufig mit einer einzelnen Kamera und nicht mit einem Stereosystem verfolgt.

Die im ersten Punkt beschriebenen Marker sind im Ansatz den in Abschnitt 2.4.1 beschriebenen Markern der Infrarottrackingsysteme ähnlich und können mit den Algorithmen dieser Arbeit sinnvoll verwendet werden.

2.4.3 Tracking auf Basis von 3D-Modellen

Eine weitere Technik, um Objekte mit Stereokamerasystemen zu verfolgen, kann dann verwendet werden, wenn exakte geometrische Modelle, beispielsweise ein CAD-Modell dieser Objekte, vorliegen. Im Gegensatz zu den Modellen, wie sie in den folgenden Kapiteln dieser Arbeit verwendet werden, werden hier schon vorliegende detailreiche, geometrische Modelle, wie Stadtmodelle [Kretschmer 04] oder ein vermessenes technisches Gerät [Beier 03], [Kang 06] adressiert.

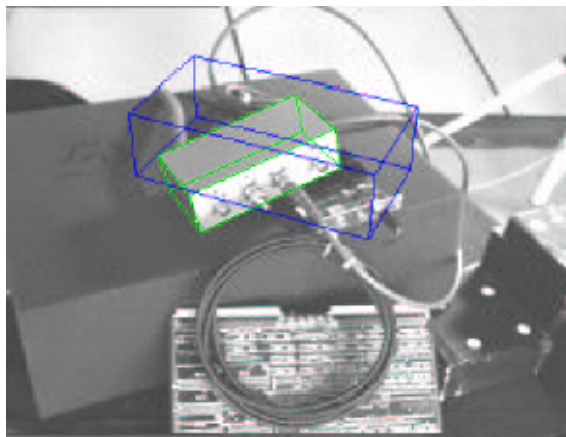


Abbildung 2.8. Monoskopisches modellbasiertes Tracking aus [Comport 04].

Liegt ein solches geometrisches Modell vor, muss man sich für das Tracking aufgrund der möglichst geringen Rechenzeit auf eine reduzierte Darstellung beschränken. Erfolgt eine solche Beschränkung in der Form, dass nur markante 3D-Punkte gesucht, erkannt und verfolgt werden sollen, könnte das in dieser Arbeit beschriebene Multipunktmodell zur Bestimmung von Position und Orientierung eingesetzt werden. Abbildung 2.8 zeigt einen modellbasierten Ansatz aus [Comport 04], bei dem eine Kamera verwendet wird. Hier werden Kanten aus dem Kamerabild detektiert und mit dem Modell verglichen.

2.5 Zusammenfassung

In diesem Kapitel wurden Grundlagen aus den Bereichen Stereokamerasystem und optisches Tracking eingeführt. Bei einem Stereokamerasystem gibt es dabei zahlreiche Faktoren beziehungsweise Voraussetzungen, die eine präzise Verfolgung im Raum bestimmen: die synchrone Erfassung von Bildern, eine gute intrinsische Kalibrierung der einzelnen Kameras, eine gute extrinsische Kalibrierung des gesamten Systems und eine der Trackingtechnologie angepasste Bildverarbeitung. Ein weiterer Faktor, der auch die Ausprägung des Interaktionsbereichs beeinflusst, ist der Kameraaufbau. Das für die

Kamerakalibrierung zugrunde gelegte mathematische Kameramodell entspricht dem der gängigen Computer Vision Standardliteratur.

Verschiedene vorgestellte Bildverarbeitungsansätze und Trackingtechnologien liefern in den einzelnen Kamerabildern 2D-Punkte, wobei in dieser Arbeit ein besonderer Schwerpunkt auf der Infrarottrackingtechnologie liegt. Das teilweise im Rahmen dieser Arbeit entwickelte Infrarottrackingsystem bildet die Testumgebung für die später aufgeführten Algorithmen. Diese kann sowohl mit retroreflektierenden kugelförmigen Markern (passiv), als auch mit Infrarotdioden (aktiv) verwendet werden.

Damit sind die Grundlagen für den Aufbau eines solchen Systems beschrieben, sowie die Gewinnung von 2D-Punkten aus den einzelnen Kamerabildern, als erster Schritt der Verarbeitung innerhalb eines Trackingsystems. Im nächsten Kapitel folgt die Verarbeitung der 2D-Punkte der einzelnen Kamerabildern zu 3D-Punkten.

Kapitel 3

Rekonstruktion der 3D-Koordinaten und Korrespondenzanalyse

Die **Rekonstruktion** der 3D-Koordinaten von in der Szene befindlichen Punkten aus den Stereokamerabildern, die mit einem wie in Abschnitt 2.3 beschriebenen System aufgenommen wurden, stellt sich in den meisten Fällen als problemlos dar. Dies ist insbesondere der Fall, wenn Korrespondenzen zwischen den 2D-Punkten verschiedener Kameras bekannt sind. Die Bestimmung dieser Korrespondenzen zwischen 2D-Punkten verschiedener Kameras wird auch als **Korrespondenzanalyse** bezeichnet. Die zugrundeliegende Theorie für die Rekonstruktion und vor allem die für die Korrespondenzanalyse sehr nützliche Epipolar Geometrie ist in der Literatur unter anderem in [Trucco 98], [Sonka 99], [Faugeras 01] und [Hartley 04] beschrieben.

Nach den Bildverarbeitungsschritten auf den synchronisiert aufgenommenen Kamerabildern liegt für jedes Kamerabild eine Menge von 2D-Punkten $\mathcal{U} = \{U_1, \dots, U_p\}$ vor, die aufgrund der Kenntnis der intrinsischen Kameraparameter aus Gleichung (2.5) in den jeweiligen (normierten) Kamerabildkoordinaten angegeben werden können. Abbildung 3.1 zeigt ein Paar synchronisierter Kamerabilder eines Infrarottrackingsystems. Im Folgenden werden die p_1 2D-Punkte $\mathcal{U}^1 = \{U_1^1, \dots, U_{p_1}^1\}$ des ersten Kamerabildes mit der 1 als Index und die p_2 2D-Punkte \mathcal{U}^2 des zweiten Kamerabildes mit 2 als Index versehen. Analog gilt dies für die weiteren Kamerabilder, bei der Verwendung von mehr als zwei Kameras.

Ausgehend von den 2D-Punkt Mengen \mathcal{U}^1 und \mathcal{U}^2 wird in den folgenden Abschnitten erläutert, welche Methoden sich eignen, um daraus 3D-Punkte zu rekonstruieren. In Abschnitt 3.1 wird zunächst der hier verwendete Begriff Punkt und dessen Zusammenhang mit entsprechenden Markern definiert. Für die zu rekonstruierenden 3D-Punkte bietet es sich an, nicht nur die 3D-Koordinaten, sondern weitere Informationen, die während der Rekonstruktion gewonnen werden können, wie die den 3D-Punkt erzeugenden 2D-Punkte, zu speichern. Die gesamten Parameter werden in einem 3D-Punktmodell zusammengefasst, das in Abschnitt 3.3 beschrieben wird.

Unter der Voraussetzung, dass das Korrespondenzproblem bezüglich der 2D-Punkt Mengen \mathcal{U}^1 und \mathcal{U}^2 bereits gelöst ist, wird in Abschnitt 3.2 beschrieben, wie sich die Koordinaten der 3D-Punkte berechnen lassen und wie sich der bei der Rekonstruktion

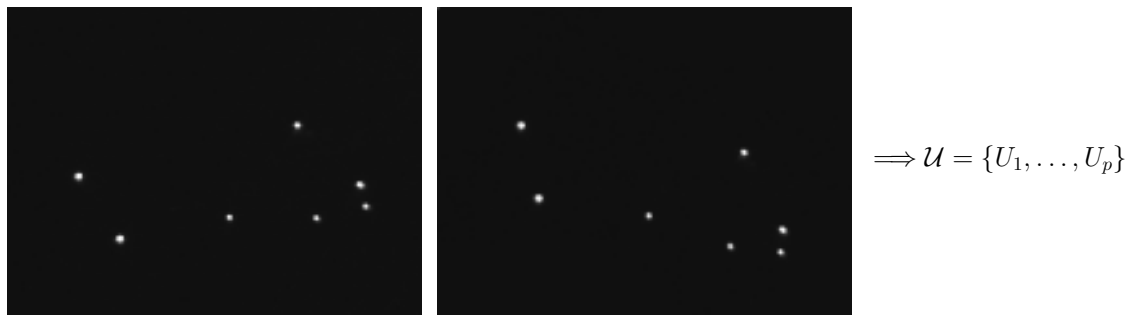


Abbildung 3.1. Durch Infrarotfilter aufgenommenes Bildpaar eines Stereokamerasystems zeigt ein mit Markern versehenes Objekt. Nach der Segmentierung lässt sich eine Menge von 2D-Punkten aus den weiß erscheinenden Markern extrahieren.

entstehende Fehler angeben lässt. Abschnitt 3.4 geht dann auf das Korrespondenzproblem und die für dessen Lösungen anwendbaren beziehungsweise verwendeten Kriterien ein.

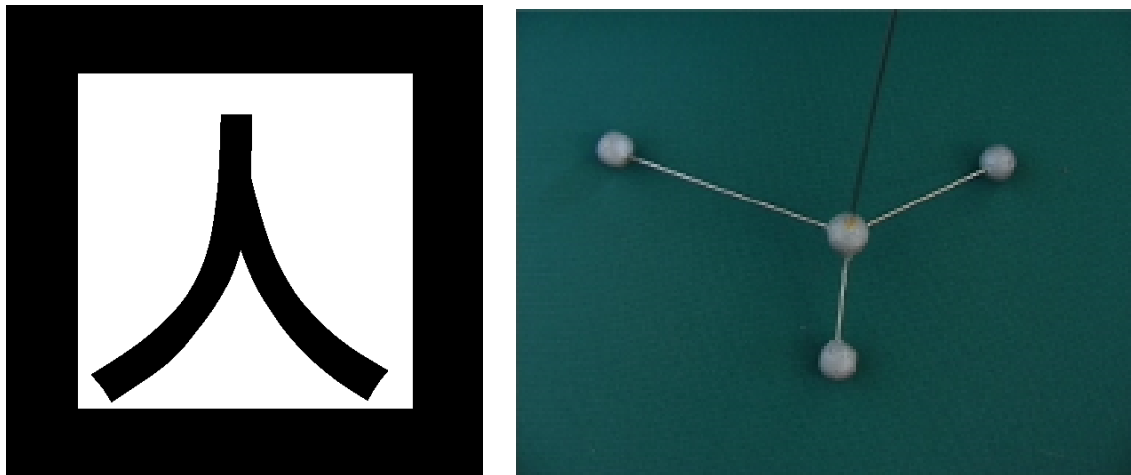
Das Kapitel schließt mit einer Diskussion in Abschnitt 3.5, welche Werte als Qualitätsangabe eines rekonstruierten 3D-Punktes gespeichert werden können, um so zur Laufzeit Aussagen über die - auch sich möglicherweise durch äußere Einflüsse verändernde - Präzision des Trackingsystems machen zu können.

3.1 Marker und Punkte

Die in dieser Arbeit in erster Linie berücksichtigten Marker sind kugel- beziehungsweise kugelförmig. Ihnen kann daher nur eine Position und keine Orientierung zugeordnet werden. Die einzelnen Marker sind untereinander nicht unterscheidbar. Andere Arten von Markern, wie zum Beispiel ARToolKit Marker in Abbildung 3.2(a) sind insofern nicht berücksichtigt, da deren Position und Orientierung auch aus einem Kamerabild berechnet werden kann und daher die Anwendung der hier beschriebenen Algorithmen keinen Sinn macht. Dieser Vorteil nicht kugelförmiger Marker steht unter anderem den Nachteilen gegenüber, dass sie meist deutlich größer sein müssen und außerdem nicht frei rotiert werden können, da typischerweise ein Muster auf einer Fläche erkannt werden muss.

Bevor in den folgenden Kapiteln weiter die Begriffe Punkt und Marker verwendet werden, wird eine Namenskonvention getroffen, um leichter eine Unterscheidung zwischen dem Betrachten der realen Szene - dem Betrachten der Marker - und dem abstrakten Vorgehen auf mathematischer Ebene - der Verarbeitung von Punkten - ausdrücken zu können.

Definition 1 *Der Begriff **Marker** steht hier für ein physikalisch existierendes, annähernd kugelförmiges Objekt, welches verfolgt werden soll. Der Begriff **Punkt** beziehungsweise **3D-Punkt** hingegen ist die Beschreibung im mathematischen Sinn. In einem perfekten System, insbesondere ohne Verdeckungen, Störungen oder Doppeldeutigkeiten, ordnet die 3D-Rekonstruktion jedem Marker genau einen Punkt zu. Marker besitzen hier keine Koordinaten, sondern nur die ihnen zugeordneten Punkte.*



(a) flacher, identifizierbarer ARToolkit
Marker

(b) kugelförmige retroreflektierende Marker

Abbildung 3.2. Vergleich zwischen verschiedenen Markern.

Im Fall des passiven Infrarottrackings ist der Marker also eine Kugel, die mit retroreflektierender Folie überzogen ist, siehe Abbildung 3.2(b). Sobald es sich nur noch um die Daten handelt, die ein Marker erzeugt, wird der Begriff Punkt verwendet. Bei der 3D-Rekonstruktion kann es durch Störungen oder Doppeldeutigkeiten mehr Punkte als tatsächlich vorhandene Marker geben. Umgekehrt kann es durch Verdeckungen einzelner Marker dazu führen, dass Marker im Sichtbereich der Kameras sind, jedoch weniger Punkte rekonstruiert werden. Die Anzahl der rekonstruierten Punkte muss also nicht mit der Anzahl der Marker in der Szene übereinstimmen. Wie eine eindeutige Zuordnung von Markern zu ihren korrekt rekonstruierten Punkten dennoch in den meisten Fällen zu erreichen ist und welche Probleme dabei auftreten können, ist in den folgenden Abschnitten beschrieben.

3.2 Berechnung der 3D-Koordinaten

Für die Rekonstruktion eines 3D-Punktes, also die Berechnung dessen 3D-Koordinaten, häufig auch als **Triangulierung** bezeichnet, ist in diesem Abschnitt die Existenz einer Korrespondenz von 2D-Punkten vorausgesetzt, das heißt die 2D-Punkte der beiden Kamerabilder können einander zugeordnet werden. Zu Vereinfachung sind im Falle eines Zweikamerasystems diese korrespondierenden 2D-Punkte $U_{(i)}^1 \in \mathcal{U}^1$ und $U_{(i)}^2 \in \mathcal{U}^2$ in diesem Abschnitt ohne den unteren Nummerierungsindex angegeben. Da diese Punkte bereits jeweils in Kamerakoordinaten vorliegen, also die jeweiligen intrinsischen Kamerakalibrierungsparameter schon angewendet wurden, können die 2D-Punkte U^1 und U^2 in 3D-Koordinaten

$$\bar{U}^1 = \begin{pmatrix} U^1 \\ -f_1 \end{pmatrix} \in \mathbb{R}^3 \quad \text{und} \quad \bar{U}^2 = \begin{pmatrix} U^2 \\ -f_2 \end{pmatrix} \in \mathbb{R}^3 \quad (3.1)$$

angegeben werden, da sie räumlich gesehen auf der zu den jeweiligen xy -Ebenen parallelen Kamerabildebene liegen, wobei f_1 und f_2 die Brennweiten der Kameras sind, welche den Abstand zwischen Kamerabildebene und Projektionszentrum beschreiben. Diese 3D-Punkte werden dann mit den nach der extrinsischen Kamerakalibrierung bekannten Positionen und Orientierungen der Kameras in ihre Repräsentation in das Weltkoordinatensystem (WKS)

$$\hat{U}^1 = R^1 \cdot \bar{U}^1 + T^1 \quad \text{und} \quad \hat{U}^2 = R^2 \cdot \bar{U}^2 + T^2 \quad (3.2)$$

transformiert. Dabei bezeichnet T^1 die Translation vom Ursprung des WKS zum Projektionszentrum P_1 der ersten Kamera und R^1 die Rotation von Kamera- in Weltkoordinaten und Analoges für T^2 , P_2 und R^2 . Durch die Punkte \hat{U}^1 und P_1 lässt sich dann eine Gerade g_1 aufspannen, sowie durch die Punkte \hat{U}^2 und P_2 eine Gerade g_2 .

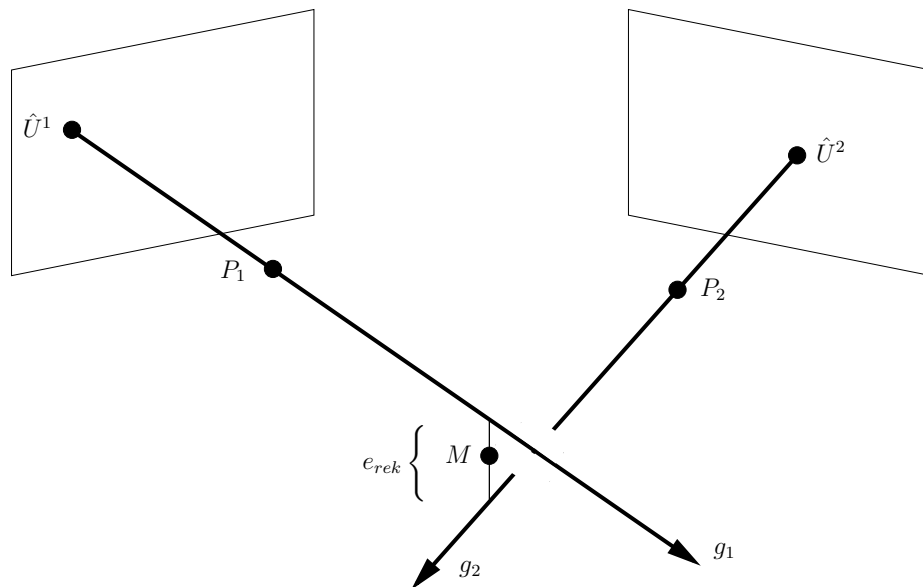


Abbildung 3.3. 3D-Rekonstruktion und Rekonstruktionsfehler aus zwei Kameras mit Projektionszentren P_1 und P_2 .

In einem perfekt kalibrierten System müssten sich diese beiden Geraden im zu rekonstruierenden 3D-Punkt schneiden. In der Realität ist dies jedoch nicht der Fall, sodass der 3D-Punkt beispielsweise als Mittelpunkt der kürzesten Strecke zwischen den beiden Geraden definiert werden kann, siehe Abbildung 3.3. Für die Berechnung dieses angenäherten Schnittpunktes werden zunächst die Richtungsvektoren

$$\vec{r}_1 = \overrightarrow{P_1 - \hat{U}^1} \quad \text{und} \quad \vec{r}_2 = \overrightarrow{P_2 - \hat{U}^2} \quad (3.3)$$

der beiden Geraden g_1 und g_2 berechnet. Der zu \vec{r}_1 orthogonale Vektor \vec{n}_1 in der durch \vec{r}_1 und \vec{r}_2 aufgespannten Ebene lässt sich mit Hilfe des doppelten Vektorprodukts und des Entwicklungssatzes für das Vektorprodukt einfach durch

$$\vec{n}_1 = \vec{r}_1 \times (\vec{r}_1 \times \vec{r}_2) = \vec{r}_1 \cdot (\vec{r}_1 \cdot \vec{r}_2) - \vec{r}_2 \cdot (\vec{r}_1 \cdot \vec{r}_1) \quad (3.4)$$

berechnen. Als nächster Schritt wird die zu \vec{n}_1 orthogonale Ebene durch den Punkt P_1 - womit die Ebene auch die Gerade g_1 enthält - mit der Geraden g_2 geschnitten. Dabei ist die Ebene durch die Gleichung

$$\overrightarrow{X - P_1} \cdot \vec{n}_1 = 0 \quad (3.5)$$

definiert. Setzt man in Gleichung (3.5) die parametrische Darstellung der Geraden

$$g_2 = \{X \in \mathbb{R}^3 : X = P_2 + t_2 \cdot \vec{r}_2, \text{ mit } t_2 \in \mathbb{R}\} \quad (3.6)$$

ein, ergibt sich die Gleichung

$$\overrightarrow{P_2 + t_2 \cdot \vec{r}_2 - P_1} \cdot \vec{n}_1 = 0, \quad (3.7)$$

die sich nach t_2 auflösen lässt. Somit kann der Parameter

$$t_2 = \frac{\overrightarrow{P_1 - P_2} \cdot \vec{n}_1}{\vec{r}_2 \cdot \vec{n}_1} \quad (3.8)$$

berechnet werden, der den zu der Geraden g_1 am nächsten gelegenen Punkt M_2 auf g_2 durch

$$M_2 = P_2 + t_2 \cdot \vec{r}_2 \quad (3.9)$$

angibt. Der Nenner in Gleichung (3.8) kann nur dann Null werden, wenn \vec{r}_2 orthogonal zu \vec{n}_1 ist, was genau dann zutrifft, wenn die beiden Geraden g_1 und g_2 parallel sind. Da ein aus solchen Geraden zu rekonstruierender 3D-Punkt keinen Sinn macht, da er unendlich weit entfernt liegen müsste, wird schon bei Gleichung (3.4) die Berechnung abgebrochen, wenn $\vec{r}_1 \times \vec{r}_2 = \vec{0}$ und damit auch $\vec{n}_1 = \vec{0}$.

Analog erfolgt die Berechnung für den zu der Geraden g_2 am nächsten gelegenen Punkt M_1 auf g_1 . Dazu wird in diesem Fall ein Parameter t_1 berechnet, so dass sich M_1 aus

$$M_1 = P_1 + t_1 \cdot \vec{r}_1 \quad (3.10)$$

berechnen lässt. Der angenäherte Schnittpunkt ergibt sich dann als die Mitte

$$M = \frac{M_1 + M_2}{2} \quad (3.11)$$

der beiden nächsten Punkte auf den beiden Geraden zur jeweils anderen. Dieses Vorgehen für die Bestimmung von M ist unter anderem in ähnlicher Weise in [Kohler 96] beschrieben.

3.2.1 Rekonstruktionsfehler

Basierend auf der Berechnung der beiden jeweils zur anderen Geraden am nächsten gelegenen Punkte M_1 und M_2 aus den Gleichungen (3.9) und (3.10) lässt sich die Distanz

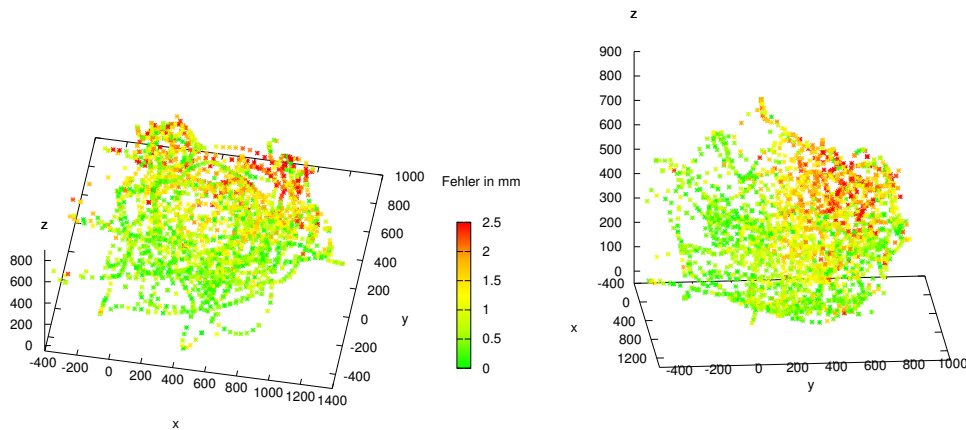
$$e_{rek} = |M_1 - M_2| \quad (3.12)$$

zwischen den beiden Punkten (siehe Abbildung 3.3) als Gütekriterium für die Rekonstruktion nutzen und wird in den folgenden Abschnitten als **Rekonstruktionsfehler** bezeichnet. Dieser Rekonstruktionsfehler entspricht anderen Bezeichnungen, wie dem „3d point residual“ des im Bereich der Biomechanik häufig verwendeten C3D-Dateiformats zur Speicherung von 3D-Informationen, welches im Anhang B kurz beschrieben wird.

Bei der Interpretation dieses Rekonstruktionsfehlers muss beachtet werden, dass ein hoher Rekonstruktionsfehler zwar auf eine schlechte Qualität der Rekonstruktion schließen lässt, dies im umgekehrten Fall aber nicht zutrifft. Ein geringer Rekonstruktionsfehler ist notwendig aber nicht hinreichend für eine gute Qualität der Rekonstruktion, da die beiden windschiefen Geraden g_1 und g_2 einen geringen Abstand haben können, ohne dass der daraus resultierende angenäherte Schnittpunkt M an der richtigen Position liegt. Der Rekonstruktionsfehler lässt sich aber dennoch nutzen, um während der Verwendung eines Systems Aussagen über dessen Zustand treffen zu können. Er kann beispielsweise bei dauerhafter Beobachtung genutzt werden, um eine über eine gewisse Zeit eintretende Verschlechterung der Rekonstruktion festzustellen. Eine solche Verschlechterung könnte durch leicht in Schwingung versetzte Kameras verursacht werden. Für Qualitätsmessungen, die nicht während der Verwendung des Systems erfolgen, sondern gesondert durchgeführt werden, eignen sich andere Kriterien, wie die exakte räumliche Positionierung von Markern oder Distanzmessungen zwischen solchen, siehe Abschnitt 3.5.

Die Evaluierung der Genauigkeit von Trackingsystemen ist sowohl in der Literatur, als auch bei den Herstellern kommerzieller Trackingsysteme nicht vereinheitlicht. Kritik an dem Vorgehen, die Genauigkeit eines Trackingsystems nur durch die Angabe statistischer Werte wie RMS (root mean square) oder Mittelwerte anzugeben, wurde in [Frantz 04] formuliert. Die dort formulierten Mindestanforderungen für Genauigkeitsangaben sind zum einen eine ausgewählte Menge charakterisierender statistischer Angaben, die sowohl Aussagekraft, als auch Genauigkeit beschreiben und zum anderen die entsprechenden Protokolle für die Messungen, auf denen die Auswertung basiert. Insbesondere ist der Messfehler der Trackingsysteme im Interaktionsvolumen nicht gleichverteilt, was anhand der folgenden Beispiele verdeutlicht wird.

Abbildung 3.4 zeigt das Beispiel einer Messung von 2600 3D-Punkten innerhalb eines Interaktionsvolumens von ca. $1500 \text{ mm} \times 1500 \text{ mm} \times 750 \text{ mm}$ mit einem durchschnittlichen Rekonstruktionsfehler von $e_{rek} = 0,98 \text{ mm}$. Der maximale Rekonstruktionsfehler beträgt $4,30 \text{ mm}$ und die Standardabweichung liegt bei $0,68 \text{ mm}$. Die Einfärbung eines 3D-Punktes hängt von der Größe des zugehörigen Rekonstruktionsfehlers ab. Die beiden verwendeten Kameras sind an den in mm angegebenen Positionen $P_1 = (-632, 1897, 927)$ und $P_2 = (2042, 2016, 974)$ platziert. Rekonstruiert man aus den beiden optischen Achsen der Kameras, analog zum Vorgehen in Abschnitt 3.2, einen 3D-Punkt als Zentrum des Interaktionsbereichs, so liegt dieser bei ca. $(365, 165, 70)$. Die optischen Achsen haben



(a) Sicht von oben, gegenüber der Kameras, die auf die Punktmenge gerichtet sind.

(b) Sicht von der Seite, Kameras sind rechts davon in positiver y-Richtung positioniert.

Abbildung 3.4. Farblich dargestellter Rekonstruktionsfehler an 2600 3D-Messpunkten.

dabei an der engsten Stelle eine Distanz von 284 mm. An der eingefärbten Punktmenge kann man feststellen, dass die Punkte mit größerem Rekonstruktionsfehler überwiegend auf der positiven Seite der y-Achse liegen. Diese Punkte sind den Kameras näher, was zunächst einen geringeren Rekonstruktionsfehler erwarten lässt, wenn man davon ausgeht, dass die Koordinaten von Markern, die näher an der Kamera sind, präziser bestimmt werden können und damit die 3D-Rekonstruktion eine bessere Qualität hat. Allerdings spielt es aufgrund der Linsenverzeichnung auch eine Rolle, wo sich die 2D-Koordinaten der 3D-Punkte auf den jeweiligen Kamerabildern befinden. Die Linsenverzeichnung wird zwar bei einer intrinsischen Kamerakalibrierung annähernd berechnet, bleibt aber insbesondere bei der Verwendung von Objektiven mit kurzer Brennweite, wie hier, fehlerbehaftet.

Statt des Rekonstruktionsfehlers bieten sich auch andere Methoden zur Qualitätskontrolle an. Wie in [Dorf Müller 99b] vorgeschlagen, kann auch die Projektion des rekonstruierten 3D-Punktes auf die Kameraebenen ausgewertet werden, indem diese mit der entsprechenden Position des erzeugenden 2D-Punktes verglichen wird, siehe Abbildung 3.5. Für eine Verwendung dieses Kriteriums während des normalen Betriebs zur Qualitätskontrolle sind allerdings zusätzliche Berechnungen, nämlich die Projektionen auf die Kameraebenen, erforderlich. Abbildung 3.6 zeigt den Fehler zwischen originalen und projizierten Punkten, der bei der Projektion der 2600 3D-Punkte aus Abbildung 3.4 auf die Kamerabildebenen entsteht. Die 2D-Punkte sind nicht über die ganze Kamerabildfläche verteilt, sondern nur um das Bildzentrum, was sich aus dem Interaktionsvolumen ergibt. Für Kamera 1 liegt der durchschnittliche Fehler bei 0,15 Pixeln mit einem maximalen Fehler von 0,85 Pixeln und einer Standardabweichung von 0,11 Pixeln. Für Kamera 2

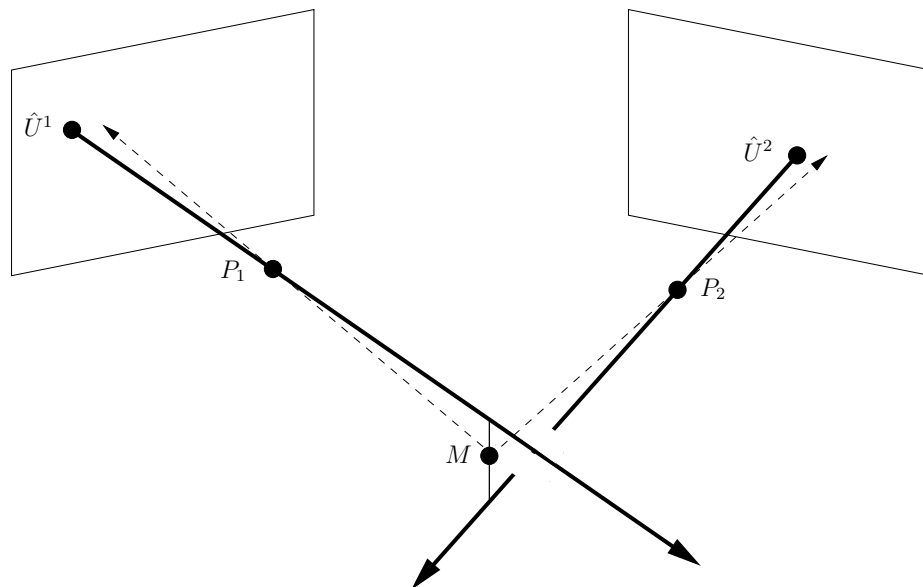


Abbildung 3.5. 3D-Rekonstruktion eines Punktes mit dessen Projektion auf die Kameraebenen.

liegt der durchschnittliche Fehler bei 0,12 Pixeln mit einem maximalen Fehler von 0,60 Pixeln und einer Standardabweichung von 0,09 Pixeln.

Addiert man die beiden Fehler auf den Kameraebenen und ordnet diese Summe jedem 3D-Punkt zu, erhält man Abbildung 3.7. Der durchschnittliche Wert für die Summe der Fehler ist 0,27 Pixel bei einem Maximum von 1,33 Pixeln und einer Standardabweichung von 0,20 Pixeln. Die Verteilung zwischen höheren und geringeren Fehlern ähnelt insgesamt der Verteilung des Rekonstruktionsfehlers, der in Abbildung 3.4 dargestellt wurde.

Der Vorteil dieser Fehlerangabe auf Kameraebene gegenüber des Rekonstruktionsfehlers liegt in der Möglichkeit den Fehler für jede Kamera getrennt zu betrachten und mögliche Abhängigkeiten auf der 2D-Ebene, wie beispielsweise eine schlecht kalibrierte Linienverzeichnung, zu erkennen. Andererseits erscheint ein großer Rekonstruktionsfehler eines von den Kameras weit entfernten Punktes als Fehler auf der Kameraebene genauso groß, wie ein kleinerer Rekonstruktionsfehler eines entsprechend näheren Punktes.

Da der Rekonstruktionsfehler praktisch ohne zusätzlichen Aufwand berechnet werden kann, ist er in dieser Arbeit das bevorzugte Qualitätskriterium.

3.2.2 Alternative Berechnungen des angenäherten Schnittpunkts

Bisher wurde beschrieben, wie sich auf einfache Weise 3D-Punkte als Mittelpunkt der kürzesten Strecke zwischen den zwei rückprojizierten Geraden rekonstruieren lassen. Dieses Verfahren ist vermutlich das naheliegendste, jedoch weder das einzige noch notwendigerweise beste. Unter anderem schlagen Hartley und Zisserman [Hartley 04] weitere Verfahren vor, um 3D-Punkte zu rekonstruieren.

Alternativ zur oben beschriebenen Bestimmung des 3D-Punktes kann man die beiden

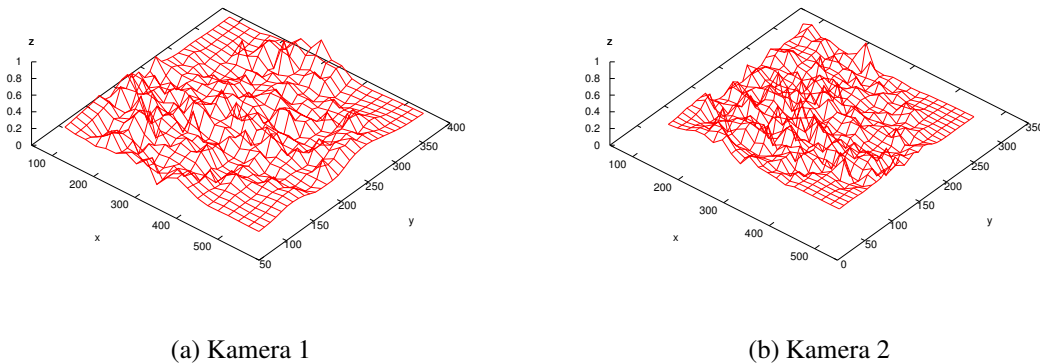


Abbildung 3.6. Durchschnittlicher Fehler der Projektion auf die mit Linsenverzeichnung kalibrierten Kameras in Pixeln.

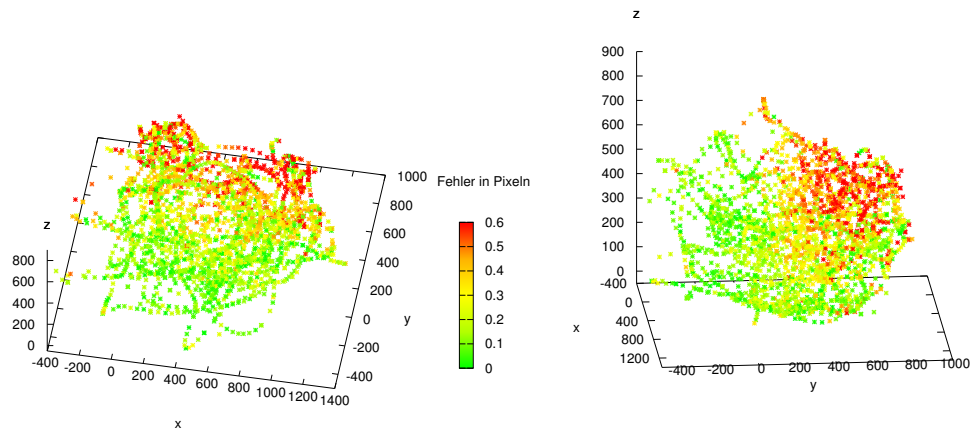
Punkte M_1 und M_2 aus den Gleichungen (3.10) und (3.9) entsprechend ihrer Entfernung von den jeweiligen Projektionszentren P_1 und P_2 durch

$$M = \frac{d_1}{d_1 + d_2} M_1 + \frac{d_2}{d_1 + d_2} M_2 \quad \text{mit} \quad d_1 = |M_1 - P_1| \quad \text{und} \quad d_2 = |M_2 - P_2| \quad (3.13)$$

gewichten. Je näher also der zu berechnende Punkt M an Kamera 1 ist, umso stärker wird der Punkt M_1 für dessen Berechnung gewichtet und umgekehrt. Diese Berechnung lässt sich mit einem nur geringeren Mehraufwand gegenüber dem arithmetischen Mittel umsetzen. Abbildung 3.8 zeigt farblich dargestellt die Größe der Verschiebung der 3D-Punkte, wenn sie statt durch arithmetische Mittelung mit der eben beschriebenen gewichteten Mittelung gewonnen werden. Im Durchschnitt beträgt diese Verschiebung bei den 2600 Messpunkten 0,05 mm mit einem Maximum von 0,64 mm und einer Standardabweichung von 0,06 mm. Außerdem lässt sich im Vergleich mit Abbildung 3.4 feststellen, dass die Verschiebung in den Bereichen größer ist, in denen auch der Rekonstruktionsfehler größer ist.

Der Rekonstruktionsfehler selbst ist für beide Ansätze identisch. Bildet man auch für die durch Gewichtung gewonnenen 3D-Punkte die Summe der beiden Fehler bei Projektion auf die Kamerabildebene, liegt der durchschnittliche Wert für diese Summe bei 0,27 Pixel. Dieser Wert unterscheidet sich damit um weniger als 0,01 Pixel von der arithmetischen Mittelung. Ebenso unterscheidet sich die auf der 2D-Ebene gemessene Qualität kaum, was sich mit der Feststellung im vorausgegangenen Abschnitt, dass rekonstruierte 3D-Punkte, die sich näher bei den Kameras befinden, nicht notwendigerweise einen geringeren Rekonstruktionsfehler haben, vereinbaren lässt.

Ähnlich zum eben beschriebenen Ansatz lassen sich auch die Abstände der den 3D-Punkt erzeugenden 2D-Punkte U_1 und U_2 von den jeweiligen Hauptpunkten (Principal



(a) Sicht von oben, gegenüber der Kameras, die auf die Punktmenge gerichtet sind.

(b) Sicht von der Seite, Kameras sind rechts davon in positiver y-Richtung.

Abbildung 3.7. Farblich dargestellte Summe der 2D-Fehler, die bei der Projektion der 2600 rekonstruierten 3D-Punkte auf die Kameraebenen entstehen.

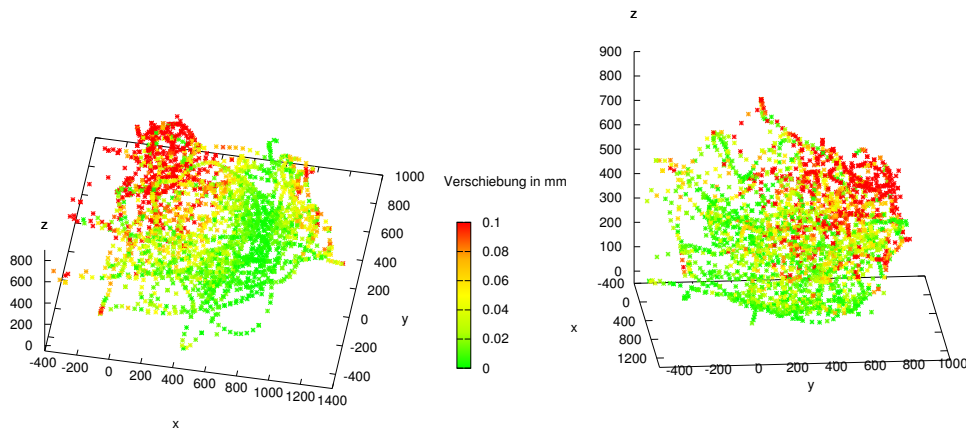
Point) als Gewichte nutzen. Dieser Ansatz hat dann seine Berechtigung, wenn man davon ausgeht oder nachweisen kann, dass die Rekonstruktion an den Bildrändern an Qualität verliert, beispielsweise aufgrund schlechterer Ausleuchtung oder schlecht kalibrierter Linsenverzeichnung.

In beiden Fällen ist jedoch nicht garantiert, dass sich auch die Distanzen $e_1 = |U_1 - \tilde{U}_1|$ und $e_2 = |U_2 - \tilde{U}_2|$ zwischen den M erzeugenden 2D-Punkten U_1 und U_2 und den Projektionen \tilde{U}_1 und \tilde{U}_2 von M auf die Kameraebene reduzieren. Um diesen Fehler zu minimieren bietet sich unter anderem die Hartley-Sturm Triangulierung [Hartley 97] an, bei der 2D-Punkte \tilde{U}_1 und \tilde{U}_2 gesucht werden, sodass

$$e_{2D} = e_1^2 + e_2^2 \quad (3.14)$$

minimiert wird, unter der Voraussetzung des Epipolar Constraints $\tilde{U}_1^T F \tilde{U}_2 = 0$, mit F als Fundamentalmatrix. Ein deutlicher Nachteil eines solchen Ansatzes ist die wesentlich höhere Rechenzeit, die für Projektion und Optimierung benötigt wird.

In [Hartley 97, Hartley 04] finden sich weitere Ansätze für das Triangulierungsproblem, wie die Linear-Eigen, die Linear-LS, Iterative-Eigen und Iterative-LS Methode. Da zwar möglicherweise die Qualität der Rekonstruktion, nicht aber das eigentliche Vorgehen in den folgenden Kapiteln von der Wahl des Triangulierungsalgorithmus abhängt, wird die Wahl eines alternativen Verfahrens an dieser Stelle offengelassen. Die Entscheidung für einen Triangulierungsalgorithmus hängt davon ab, ob hohe Präzision oder geringe Verarbeitungszeit im Vordergrund steht. Insgesamt gilt jedoch, dass bei einer guten Kamerakalibrierung die Wahl der Art der Triangulierung an Bedeutung verliert.



(a) Sicht von oben, gegenüber der Kameras, die auf die Punktmenge gerichtet sind.

(b) Sicht von der Seite, Kameras sind rechts davon in positiver y-Richtung.

Abbildung 3.8. Farblich dargestellte Verschiebung der 2600 3D-Messpunkte durch gewichtete Mittelung in Abhängigkeit der Entfernung zu den Hauptpunkten statt arithmetischer Mittelung.

3.2.3 Rekonstruktion mit mehr als zwei Kameras

Die Verwendung eines Kamerasystems mit mehr als zwei Kameras wirft einige weitere Fragestellungen auf. Zwei Kameras bleiben die minimale Anzahl, die für eine zuverlässige Rekonstruktion eines Punktes notwendig sind. Insbesondere ist es auch für ein **Multikamerasystem** wichtig, dass jeweils nur zwei Kameras ausreichen, um einen Punkt zu rekonstruieren, da das Interaktionsvolumen meist möglichst groß sein soll. Ausgehend von dieser Überlegung bieten sich mindestens drei Ansätze für ein Multikamerasystem an:

- Die Rekonstruktion erfolgt für alle 3D-Punkte zu einem Zeitpunkt immer nur aus den zwei gleichen Kamerabildern. Die zwei ausgewählten Kamerabilder können zum nächsten Zeitpunkt wechseln.
- Die Rekonstruktion erfolgt für die einzelnen 3D-Punkte aus jeweils nur zwei Kamerabildern.
- Die Punkte werden aus allen verfügbaren Kamerabildern rekonstruiert.

Der erste Fall ist ausgehend von dem in dieser Arbeit beschriebenen Ansatz für Zweikamerasysteme einfach umzusetzen. Es können beispielsweise immer die beiden Kameras ausgewählt werden, in denen die größte Anzahl an 2D-Punkten extrahiert wurde. Die

Überlegungen zur Triangulierung aus den beiden ausgewählten Kamerabildern entsprechen dann denen in den vorangegangenen Abschnitten. Der Nachteil dieses Vorgehens liegt darin, dass alle Punkte, die in einem oder beiden ausgewählten Kamerabildern verdeckt sind, nicht rekonstruiert werden können. Insofern eignet sich ein solcher Ansatz nur dann, wenn nur ein Objekt mit wenigen Markern verfolgt werden soll und dabei das Interaktionsvolumen im Vergleich zu einem Zweikamerasystem vergrößert werden soll.

Auch beim zweiten aufgelisteten Verfahren können die Überlegungen zur Triangulierung aus den vorangegangenen Abschnitten herangezogen werden. Allerdings wirft dieses Vorgehen bereits einige Fragestellungen auf, wie die Korrespondenzanalyse umzusetzen ist, die in Abschnitt 3.4 diskutiert wird. Bei diesem Ansatz wird zwar keine Information verschenkt, jedoch ist je nachdem, wie die Korrespondenzanalyse eingesetzt wird, mit zahlreichen Mehrdeutigkeiten zu rechnen. Um diese Mehrdeutigkeiten zu reduzieren, bietet es sich an, Informationen aus mehr als zwei Kamerabildern gleichzeitig zu gewinnen, was dann dem dritten aufgelisteten Vorgehen entspricht.

Beim dritten Fall liegt ein beträchtlicher Anteil der Arbeit in der Korrespondenzanalyse. Geht man wie bisher zunächst davon aus, dass die Korrespondenzanalyse bereits erfolgt ist, stellt sich wieder die Frage nach einem geeigneten Triangulierungsalgorithmus. Die einfachste Methode bei zwei Kameras, die kürzeste Verbindung zwischen zwei Geraden und dann den Mittelpunkt auf dieser zu suchen, lässt sich zumindest auch auf ein Dreikamerasystem einfach übertragen. Dabei werden zunächst drei Mittelpunkte gebildet, um aus diesen den Schwerpunkt des gebildeten Dreiecks als rekonstruierten Punkt auszuwählen. Der Begriff des Rekonstruktionsfehlers muss für Multikamerasysteme angepasst werden. Einen umfassenden Einblick in die Theorie für drei oder mehr Kameras wird unter anderem in [Faugeras 95], [Sonka 99] und [Hartley 04] gegeben. Da es in den folgenden Kapiteln dieser Arbeit um die Weiterverarbeitung der rekonstruierten Punkte geht, wird das Thema der Rekonstruktion aus mehreren Kameras hier nicht weiter vertieft.

3.3 3D-Punktmodelle

Der Begriff **3D-Punktmodell** steht in dieser Arbeit für die Zusammenfassung von Eigenschaften von 3D-Punkten, die durch deren Rekonstruktion und Verfolgung zur Verfügung stehen und mit den Koordinaten gespeichert werden. Aus der Rekonstruktion ergeben sich zwei naheliegende Eigenschaften, die einem 3D-Punkt immer zugeordnet werden können, wobei zunächst von einem Zweikamerasystem ausgegangen wird:

- e_{rek} der Rekonstruktionsfehler im Raum, siehe Gleichung (3.12) in Abschnitt 3.2 und
- $U_i^1 \in \mathcal{U}^1$ und $U_j^2 \in \mathcal{U}^2$, die korrespondierenden 2D-Punkte, aus denen der 3D-Punkt rekonstruiert wird, die im Folgenden auch **Wurzelpunkte** genannt werden.

Sowohl der Rekonstruktionsfehler, als auch die 2D-Wurzelpunkte dienen in Abschnitt 3.4 der Korrespondenzanalyse. Falls man für die Rekonstruktion der 3D-Punkte statt einer einfachen Mittelung ein Verfahren wie die Hartley-Sturm-Triangulierung gewählt hat, bei dem der Rekonstruktionsfehler e_{rek} automatisch immer Null ist, siehe Abschnitt 3.2.2,

wird der Rekonstruktionsfehler an dieser Stelle durch ein anderes Gütekriterium, wie zum Beispiel e_{2D} aus Gleichung (3.14) ersetzt.

Im Kapitel 4 werden die 3D-Punkte unabhängig von den beschriebenen Algorithmen entweder als End- oder Zwischenergebnis klassifiziert, das heißt einer bestimmten Gruppe von Punkten zugeordnet. Daher ist eine weitere sinnvolle Eigenschaft im 3D-Punktmodell ein

- **Klassifikationsindex** c eines 3D-Punktes, mit $c = -1$ für einen unklassifizierten Punkt und ansonsten $c \geq 0$, je nach der gewünschten Klassifikation.

Für Multikamerasysteme lassen sich einem 3D-Punkt die drei für Zweikamerasysteme eingangs aufgelisteten Eigenschaften entsprechend zuordnen:

1. $X = (x, y, z)$, die Koordinaten des Punktes,
2. e_{rek} der Rekonstruktionsfehler im Raum, der für Multikamerasysteme entsprechend definiert werden muss, siehe Abschnitt 3.2.3 und
3. $U_i^1 \in \mathcal{U}^1 \dots U_j^n \in \mathcal{U}^n$, die korrespondierenden 2D-Punkte, aus denen der 3D-Punkt rekonstruiert wird, wobei darauf zu achten ist, dass im Allgemeinen nicht aus jedem Kamerabild ein 2D-Punkt vorliegt, sowie dem
4. Klassifikationsindex, der wie oben beschrieben verwendet wird.

Ausgehend von diesem im Wesentlichen aus vier Eigenschaften bestehenden grundlegenden Modell für 3D-Punkte lassen sich je nach Anwendung neue Eigenschaften finden, die in den entsprechenden folgenden Kapiteln beschrieben werden. Sollen zum Beispiel einzelne Punkte über eine gewisse Zeitspanne verfolgt werden, werden auch die letzten Positionen dieses Punktes gespeichert. Auf Basis der alten Positionen lassen sich dann auch neue Positionen präzisieren, beziehungsweise kurze Verdeckungen überbrücken.

3.4 Korrespondenzanalyse zwischen den 2D-Punkten

Um aus den für jedes Kamerabild zur Verfügung stehenden 2D-Punktmenge \mathcal{U}^1 und \mathcal{U}^2 die Korrespondenzen zu ermitteln, steht als mächtigstes Hilfsmittel die Epipolare Geometrie zur Verfügung, siehe unter anderem in [Trucco 98], [Sonka 99], [Faugeras 01] und [Hartley 04], deren Verwendung in Abschnitt 3.4.1 beschrieben wird. Weitere Kriterien für die Korrespondenzanalyse sind im anschließenden Abschnitt 3.4.2 beschrieben.

3.4.1 Epipolarkriterium

Soweit 2D-Punkte U^1 und U^2 der entsprechenden Kamerabilder vorliegen, steht die Fundamentalmatrix F als Darstellung für das **Epipolarkriterium** durch

$$(U^1)^T F U^2 = 0 \tag{3.15}$$

zur Verfügung. Falls die Punkte U^1 und U^2 bereits durch die hier bekannten Kamera- kalibrierungsmatrizen normiert wurden, drückt entsprechend die Essential Matrix E das Epipolarkriterium durch

$$(U^1)^T E U^2 = 0 \quad (3.16)$$

aus. Da bei aus Kamerabildern gewonnenen Daten grundsätzlich von Störungen auszu- gehen ist, genügt es für die Suche nach möglichen Korrespondenzen allerdings nicht, 2D-Punkte auf den entsprechenden Epipolarlinien zu suchen. Um die Epipolarlinien wer- den Toleranzbereiche definiert, sodass auch Punkte, die knapp neben einer Epipolarlinie liegen, als Kandidaten für eine Korrespondenz in Frage kommen, siehe Abbildung 3.9.

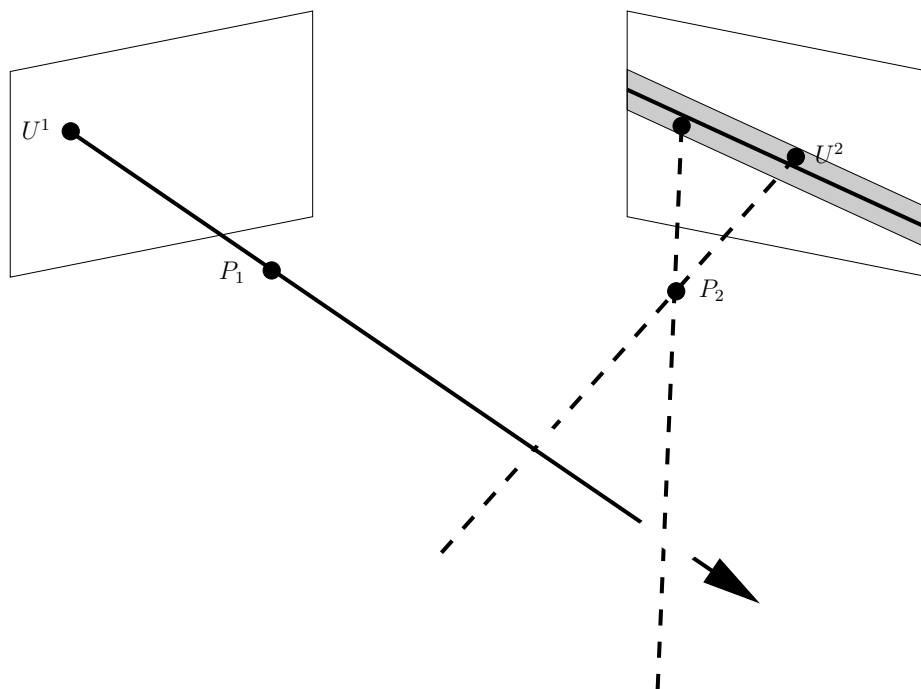


Abbildung 3.9. Toleranzbereich um die Epipolarlinie von U^1 auf der zweiten Bildebene. Zwei Punkte liegen im Toleranzbereich, der gesuchte Punkt U^2 und ein weiterer, sich zufällig innerhalb des Toleranzbereichs befindender, Punkt.

Der Toleranzbereich wird durch den hier als **Epipolartoleranz** bezeichneten Wert de- finiert, der angibt, wie groß der Abstand eines Punktes $U^2 \in \mathcal{U}^2$ von der Epipolarlinie maximal sein darf. Die Epipolarlinie wird durch die Projektion eines Punktes $U^1 \in \mathcal{U}^1$ auf das zweite Kamerabild erzeugt. Dasselbe gilt für einen Punkt $U^1 \in \mathcal{U}^1$, dessen Ab- stand von der Epipolarlinie, die durch einen Punkt $U^2 \in \mathcal{U}^2$ erzeugt wird, gemessen wird. Die Epipolartoleranz eignet sich nicht als eine wie in Abschnitt 3.3 beschriebene Eigen- schaft für einen 3D-Punkt, da sie nicht spezifisch für einen bestimmten 3D-Punkt gewählt wird, sondern für alle Punkte gilt. Sie erzeugt einen „Schlauch“ um die Epipolarlinie, wie in Abbildung 3.9 dargestellt. Punkte, die innerhalb dieses Schlauchs liegen werden als

Epipolarkandidaten bezeichnet. Im Folgenden wird das sogenannte Epipolarkriterium als erfüllt angesehen, wenn sich ein entsprechender Punkt im Toleranzbereich um die Epipolarlinie befindet.

Erzeugt man nun zunächst alle Korrespondenzen, indem zu jedem Punkt $U^1 \in \mathcal{U}^1$ der oder die korrespondierenden Punkte $U^2 \in \mathcal{U}^2$ gesucht werden, bieten sich zwei grundsätzliche Vorgehensweisen an:

- Jedem Punkt U^1 wird höchstens ein Punkt aus \mathcal{U}^2 als Korrespondenz zugeordnet.
- Alle unter der Voraussetzung des Epipolarkriteriums möglichen Korrespondenzen werden gebildet.

Der erste Fall erfordert ein oder mehrere Entscheidungskriterien, falls mehrere Epipolarkandidaten existieren. Ein mögliches Entscheidungskriterium ist der geringste Abstand zur Epipolarlinie, wie es beispielsweise in [Ribo 01] vorgeschlagen wird, beziehungsweise der geringste, in Abschnitt 3.2.1 eingeführte, Rekonstruktionsfehler. Ein solches Vorgehen ist zwar sehr effizient, allerdings auch fehleranfällig, da nicht notwendigerweise die Korrespondenz mit dem geringsten Fehler die richtige ist. Zwei Marker, die in einem Kamerabild genau aufeinander liegen (Eigenverdeckung), können so nicht erfasst werden. Außerdem wurde mit der Epipolartoleranz bereits im ersten Schritt ein Kriterium festgelegt, ab dem Korrespondenzen akzeptiert werden, sodass es Sinn macht, weitere Kriterien zu verwenden, um Mehrdeutigkeiten aufzulösen und falsche Korrespondenzen zu eliminieren.

Akzeptiert man also zunächst, wie im zweiten Fall beschrieben, alle Korrespondenzen, die das Epipolarkriterium erfüllen, können in folgenden Schritten weitere Kriterien angewendet werden. Auch ohne weitere Schritte kann es sinnvoll sein, mit allen Korrespondenzen weiterzuarbeiten und die Korrespondenzanalyse an dieser Stelle zu beenden, insbesondere dann, wenn fehlerhafte Punkte für die weitere Verarbeitung wenig störend sind.

3.4.2 Weitere Kriterien

Typische Kriterien für die Korrespondenzanalyse, wie sie beispielsweise bei [Klette 96] oder [Sonka 99] beschrieben werden, sind:

- **Eindeutigkeitsvermutung:** Jedem Punkt U^1 wird höchstens ein Epipolarkandidat zugeordnet. Die Auswahl des richtigen Epipolarkandidaten erfolgt hier so, dass eine maximale Anzahl von Korrespondenzen entsteht. Eine Skizze des innerhalb dieser Arbeit entstandenen Algorithmus zeigt Abbildung 3.10. In vielen Fällen ist dies ein sinnvolles Kriterium, berücksichtigt jedoch keine Eigenverdeckung.
- **Differentielle photometrische Kompatibilitätsbeschränkung:** Basiert auf der Annahme, dass die Intensitäten der den 2D-Punkten zugrundeliegenden Regionen auf den Kamerabildern ähnlich sind. Versteht man diese Beschränkung so, dass 3D-Punkte, die ähnliche Positionen im Raum einnehmen, ähnliche Größe und Intensität bei ihren Abbildern auf den Kamerabildern aufweisen, so steht eine weitere Hilfe zur Korrespondenzanalyse zur Verfügung.

- Die Ordnungsbeschränkung: 3D-Punkte, die in derselben Tiefe liegen haben im Allgemeinen dieselbe Anordnung der zugehörigen 2D-Punkte auf den Epipolarlinien. Dies verändert sich, wenn die Punkte in unterschiedlicher Tiefe liegen.

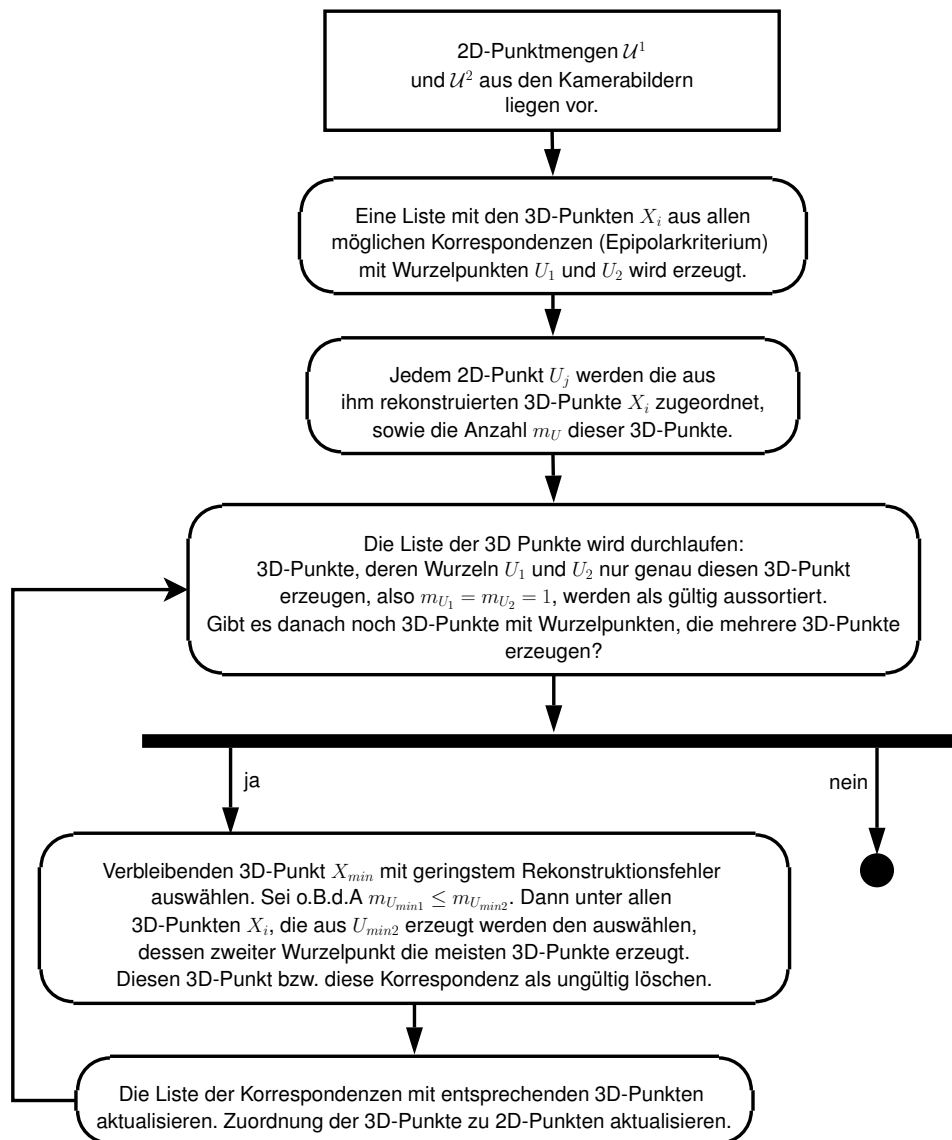


Abbildung 3.10. Algorithmus zur Umsetzung der Eindeutigkeitsvermutung.

Analog zu den 3D-Punktmodellen lassen sich **2D-Punktmodelle** definieren, die außer den Koordinaten zusätzliche Eigenschaften beinhalten. Weitestgehend unabhängig von dem gewählten Segmentierungsalgorithmus lassen sich dann einem 2D-Punktmodell beispielsweise folgende Eigenschaften hinzufügen:

- Ein Größenparameter, der die Größe der Region beschreibt, aus der ein 2D-Punkt berechnet wurde und

- ein Intensitätsparameter, der angibt mit welcher Intensität ein Punkt erkannt wurde oder alternativ
- ein Parameter, der angibt, mit welcher Sicherheit ein 2D-Punkt erkannt wurde.

Es können zahlreiche weitere Eigenschaften bzw. Einschränkungen festgelegt werden, die hauptsächlich von Geometrie und Photometrie abhängen, beziehungsweise die von Objekteigenschaften abhängen, wie der Ordnungsbeschränkung. Die Verwendung der meisten Einschränkungen ist für Trackingverfahren, bei denen es auf möglichst kurze Verarbeitungszeiten ankommt, weniger geeignet, als beispielsweise für Photometrie. Außerdem sind Einschränkungen zu geometrischen Ähnlichkeiten wegen der in dieser Arbeit zumeist verwendeten kugelförmigen Markern, die normalerweise auch dieselbe Größe besitzen, nicht nutzbar. Alle Einschränkungen sind theoretischer Natur und in der Praxis durch ungenaue Kalibrierung oder Störungen nicht für alle Punkte gültig.

Zusammenfassend werden für die Korrespondenzanalyse in dieser Arbeit je nach Anwendung drei Vorgehensweisen bevorzugt:

- die ausschließliche Verwendung des Epipolarkriteriums. Alle Punktepaare, die dieses Kriterium erfüllen, gelten als gültige Korrespondenz.
- die Verwendung des Epipolarkriteriums, wobei jedem Punkt in einem Kamerabild höchstens ein Punkt im anderen Kamerabild zugeordnet wird. Entscheidend ist der kleinste Rekonstruktionsfehler.
- die Verwendung des Epipolarkriteriums, wie beim ersten Ansatz, sodass alle Punktepaare, die dieses Kriterium erfüllen, als Korrespondenz gelten. Dann wird nachträglich kontrolliert, ob es Korrespondenzen gibt, die dieselben 2D-Punkte besitzen. Solche werden nach dem Prinzip, dass eine maximale Anzahl an Korrespondenzen verbleibt, eliminiert. Ein entsprechender Algorithmus wurde in Abbildung 3.10 dargestellt.

3.4.3 Verfolgung und Prädiktion auf 2D-Ebene

In den vorausgegangenen Abschnitten wurde beschrieben, wie Korrespondenzen auf Basis von 2D-Punktemengen $\mathcal{U}^1, \dots, \mathcal{U}^n$ festgelegt und 3D-Punkte rekonstruiert werden. Das so beschriebene Vorgehen wendet keine Identifikation, Verfolgung oder Prädiktion auf den 2D-Ebenen an, beziehungsweise nutzt keinen möglichen Rückfluss der 3D-Information auf die 2D-Ebenen.

Dahingegen wird in [Ribo 01] beschrieben, dass Lineare Prädiktion zusammen mit einer dynamischen Suchfenstergröße verwendet wird, um sich an verschiedene Bewegungen der Objekte anpassen zu können. Die Effizienz dieses Vorgehens wird als ausreichend bezeichnet.

Verfolgung und Prädiktion auf 2D-Ebene, das heißt in den einzelnen Kamerabildebene hat folgende Vor- und Nachteile für die Korrespondenzanalyse:

- Die 2D-Punkte müssen für ihre Verfolgung einen Identifikationsindex erhalten, den sie aber verlieren, sobald sie für einen gewissen Zeitraum nicht sichtbar sind, und dann unter Umständen einen neuen zugewiesen bekommen.
- In vielen Fällen stellt sich die Epipolar Geometrie für die Korrespondenzanalyse als zuverlässiges und effizientes Kriterium dar, sodass die Verfolgung auf 2D-Ebene in vielen Fällen keinen Mehrwert erzeugt.
- Eine Prädiktion kann auch auf Basis der noch zu rekonstruierenden 3D-Punkte stattfinden, was mit der gleichen Problematik der zu vergebenden Identifikationsindizes behaftet ist, oder wie in Kapitel 7 beschrieben, auf Basis der ganzen Modelle.

Um Punkte bereits auf den Kamerabildern zu verfolgen kann auch die Information über den entsprechenden 3D-Punkt im vorherigen Frame genutzt werden. Durch Nutzung dieser Information können auch schon bei der Segmentierung leichter Punkte erkannt werden, die sich ganz oder teilweise gegenseitig verdecken.

Insgesamt kann es also durchaus hilfreich sein, Verfolgung und Prädiktion auf den Kamerabildebene durchzuführen, allerdings bedeutet dies einen Rechenaufwand, der damit automatisch in jedem Bild anfällt. In dieser Arbeit wird aus den oben genannten Gründen auf eine Verfolgung und Prädiktion auf zeitlich aufeinanderfolgenden Kamerabildern verzichtet, wäre aber auch problemlos in den Prozess von Rekonstruktion der 3D-Punkte und Korrespondenzanalyse zu integrieren.

3.4.4 Korrespondenzanalyse bei mehr als zwei Kameras

Bei der Verwendung von mehr als zwei Kameras gibt es für die Korrespondenzanalyse grundsätzlich unterschiedliche Vorgehensweisen, die analog zur Auflistung in Abschnitt 3.2.3 ist. Soll die Korrespondenzanalyse in allen verfügbaren Kameras erfolgen, können die in diesem Abschnitt beschriebenen Vorgehensweisen angewendet werden, allerdings wird das bilineare Epipolarkriterium durch trilineare, beziehungsweise multilineare Verhältnisse, siehe [Sonka 99] und [Hartley 04], ersetzt und die Korrespondenzanalyse damit deutlich komplexer.

3.5 Distanztoleranz

Der Umstand, dass 3D-Punkte nur fehlerbehaftet rekonstruiert werden können, siehe Abschnitt 3.2, spielt in den folgenden Kapiteln eine wichtige Rolle, da Gruppen von Punkten (Modellen) anhand der räumlichen Lage einzelner Punkte relativ zueinander identifiziert werden. Die fehlerbehaftete Positionierung einzelner Punkte führt auch zur fehlerhaften Berechnung derer Distanzen untereinander und macht damit eine Identifikation und Verfolgung von Modellen schwieriger. Da die Berechnung dieser Distanzen also ein grundlegendes Werkzeug für die Identifikation von Punktmengen sein wird, wird folgende Festlegung getroffen:

Definition 2 *Da die Positionen rekonstruierter 3D-Punkte $X_i \in \mathbb{R}^3$ im Allgemeinen mit einem bestimmten Fehler behaftet sind, definiert die **Distanztoleranz** $\delta > 0, \delta \in \mathbb{R}$, wann ein Abstand $d_{ij} = |X_i - X_j|$ zwischen zwei Punkten X_i, X_j einer vorgegebene Distanz $d \in \mathbb{R}$ entspricht, d.h. $d_{ij} \approx d$. Der Abstand d_{ij} entspricht der Distanz d , falls*

$$|d_{ij} - d| < \delta. \quad (3.17)$$

Da das Weltkoordinatensystem, sowie weitere noch verwendete lokale Koordinatensysteme in dieser Arbeit, immer die gleiche Skalierung besitzen, das heißt, sie sind durch eine Rotation und Translation ineinander überführbar, wird δ nicht in Abhängigkeit eines bestimmten Koordinatensystems angegeben, sondern gilt für alle verwendeten Koordinatensysteme gleichermaßen.

Die Einführung der Distanztoleranz als grundlegender Parameter für die in den nächsten Kapiteln beschriebenen Algorithmen unterstützt auch eine Skalierbarkeit der Hardware, also im Fall eines Stereokamerasystems insbesondere die Auswahl an Kameras und Objektiven. Durch die Möglichkeit, die Distanztoleranz auf einen höheren Wert zu setzen, können auch einfachere Kameras und Objektive verwendet werden, die für eine entsprechende Anwendung noch die gewünschte Qualität liefern.

3.6 Zusammenfassung

In diesem Kapitel wurden Grundlagen beschrieben, wie auf mathematischer Ebene aus den 2D-Punkten, die aus den verschiedenen Kamerabildern vorliegen, 3D-Punkte rekonstruiert werden können (Triangulierung). In diesem Zusammenhang wird die begriffliche Unterscheidung zwischen den physikalisch existierenden Objekten (Markern) und der mathematischen Beschreibung (3D-Punkt) getroffen.

Das hier aufgestellte 3D-Punktmodell umfasst neben den x -, y - und z -Koordinaten weitere Elemente, wie den Rekonstruktionsfehler, die Menge der Wurzelpunkte und den Klassifikationsindex. Dieses Punktmodell bildet die Grundlage für die Modellbeschreibungen und Algorithmen in den folgenden Kapiteln.

Für die Triangulierung gibt es verschiedene Methoden, wobei innerhalb dieser Arbeit die Methode des Punktes mit dem geringsten Abstand zwischen zwei Schiefgeraden (Rekonstruktionsfehler) bevorzugt wird. Da die Kameras kalibriert sind, ist der Rekonstruktionsfehler gering, sodass diese Methode hinreichend präzise ist und vor allem sehr effizient. Die genauere Bedeutung des Rekonstruktionsfehlers und seine Eignung als Qualitätsmaß für die Genauigkeit der Rekonstruktion wurde diskutiert.

Um aus jeweils einem 2D-Punkt pro Kamerabild einen 3D-Punkt rekonstruieren zu können, müssen diese Korrespondenzen zunächst ermittelt werden. Verschiedene Methoden, die auch kombiniert werden können, sind in diesem Kapitel beschrieben, wobei die Epipolargeometrie dabei das stärkste Kriterium bildet.

Abschließend wurde der Begriff der Distanztoleranz eingeführt, der der unperfekten Rekonstruktion von 3D-Punkten Rechnung trägt und in den folgenden Kapiteln Verwendung findet.

Kapitel 4

Starre Modelle

Nachdem im letzten Kapitel beschrieben wurde, wie sich aus den 2D-Punkten der verschiedenen Kamerabilder 3D-Punkte rekonstruieren lassen, bildet nun die Existenz einer Menge $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$, $m \in \mathbb{N}$, von 3D-Punkten den Ausgangspunkt dieses Kapitels. Die in den folgenden Abschnitten beschriebenen Algorithmen sind also nicht nur für Kamerasysteme einsetzbar, sondern für jedes Trackingsystem, welches in jedem Zeitschritt 3D-Punktmenge als Ausgabe besitzt. Im Fall der Verwendung anderer Trackingsysteme vereinfachen sich die Algorithmen, da zum Beispiel auf die Betrachtung der 2D-Punkte, aus denen die 3D-Punkte rekonstruiert werden, verzichtet werden kann. Da die Algorithmen jedoch insbesondere für Kamerasysteme entwickelt sind, wird die Entstehung der 3D-Punkte aus 2D-Daten grundsätzlich mit berücksichtigt.

Unabhängig vom verwendeten Trackingsystem beschreiben die Methoden dieses Kapitels, wie einzelne Punkte oder Gruppierungen von Punkten mit geometrischen Eigenschaften in einer 3D-Punktmenge identifiziert werden und wie diese in aufeinanderfolgenden Schritten (bei Kamerasystemen: aufeinanderfolgenden Frames) in jeweils neuen 3D-Punktmenge verfolgt werden. Gruppierungen von Punkten mit verschiedenen geometrischen Eigenschaften werden von nun an als **Modell** bezeichnet und in den folgenden Abschnitten beschrieben. Insgesamt können diese Modelle

- einzelne Marker repräsentieren,
- ein Markerpaar repräsentieren, wobei die beiden Marker relativ zueinander fest positioniert sind, sodass das Modell durch die Distanz der beiden Marker definiert wird,
- aus drei oder mehr Punkten bestehen, die die entsprechende Anzahl an Markern repräsentiert, die relativ zueinander fest positioniert sind, oder
- eine Anzahl nicht fest verbundener Marker repräsentieren, die bestimmte geometrische Randbedingungen erfüllen müssen. Diese Modelle werden in dieser Arbeit als dynamisch bezeichnet und in Kapitel 6 vertieft.

Handelt es sich um die theoretische Beschreibung der Modelle, wird im Folgenden analog zu Definition 1 auf Seite 25 der Begriff n -Punktmodell benutzt. In Ausnahmefällen

wird der Begriff Markermodell benutzt, wenn speziell auf die real existierende und zu verfolgende Anordnung von Markern verwiesen wird. Insbesondere in der englischen Fachliteratur finden sich verschiedene andere Begriffe, die alternativ zum Begriff Modell beziehungsweise „model“ verwendet werden, unter anderem „target“ oder insbesondere für starre Modelle „rigid body“.

Die Anzahl der Marker in einem starren Modell ist entscheidend für die Informationen, die aus dem Modell gewonnen werden können. Während bei einzelnen Markern nur eine Position berechnet werden kann, können bei zwei Markern eine Position und eine Achse bestimmt werden. Bei drei oder mehr Markern lässt sich bereits ein (lokales) Koordinatensystem, wie in Abschnitt 4.3.1 beschrieben, definieren.

Zum leichteren Verständnis dieses Kapitels werden folgende Namenskonventionen für 3D-Punkte getroffen:

Definition 3 Für 3D-Punkte gilt:

- \mathcal{X}_W bezeichnet eine endliche Menge $\{X_W^1, \dots, X_W^m\}$, $m \in \mathbb{N}$ von 3D-Punkten $\in \mathbb{R}^3$ im Weltkoordinatensystem. Diese Punkte sind das Resultat der 3D-Rekonstruktion aus Kapitel 3, also die ermittelten 3D-Punkte.
- X_W bezeichnet allgemein einen 3D-Punkt $\in \mathbb{R}^3$ im Weltkoordinatensystem, stellvertretend für alle Punkte X_W^i , $1 \leq i \leq m$.
- \mathcal{X}_M bezeichnet analog eine endliche Menge $\{X_M^1, \dots, X_M^n\}$, $n \in \mathbb{N}$ von 3D-Punkten $\in \mathbb{R}^3$ in einem bestimmten lokalen Koordinatensystem, zum Beispiel im lokalen Koordinatensystem eines zu verfolgenden Körpers (Modells).
- X_M bezeichnet allgemein einen beliebigen 3D-Punkt $\in \mathbb{R}^3$ in einem bestimmten lokalen Koordinatensystem, stellvertretend für alle Punkte X_M^j , $1 \leq j \leq n$.
- $\mathcal{X}_L, \mathcal{X}_L$ analog für weitere lokale Koordinatensysteme, ein D im Index steht für dynamische Modelle.
- X_1, X_2, \dots werden ohne Angabe des Koordinatensystems verwendet, wenn eindeutig ist, welches Koordinatensystem gemeint ist. Im Allgemeinen ist dies das Weltkoordinatensystem.

Diese Namenskonvention für 3D-Punkte wird für alle 3D-Punktmodelle, wie sie in Abschnitt 3.3 beschrieben wurden, verwendet.

4.1 Identifikation einzelner Marker

Da die einzelnen Marker, die in dieser Arbeit berücksichtigt werden, durch die 3D-Rekonstruktion eine Position erhalten, aber ansonsten nicht unterscheidbar sind, lassen sich diese einzelnen Marker auch nur in speziellen Fällen zuverlässig identifizieren oder erfordern weitere Randbedingungen. Existiert zu einem Zeitpunkt eine 3D-Punktmenge mit mehr als einem Element, lässt sich ein bestimmter Marker ohne weitere Information nicht identifizieren. Erst durch Informationen, wie die Position des Markers im zeitlich vorhergehenden Schritt wird eine Identifikation möglich. Ein weiteres Hilfsmittel um

einen einzelnen Marker zu bestimmen sind Randbedingungen, wie ein Extremwert in einer Koordinate. Dennoch gibt es einige Anwendungsfälle, in denen die Verfolgung, das heißt die Identifikation in jedem Schritt, einzelner Marker besonders interessant ist:

- Die Verfolgung eines einzigen Markers für die Verfolgung eines Objekts ohne Berücksichtigung der Orientierung, zum Beispiel bei Erfassung der Kopfbewegung ohne Blickrichtung.
- Die Verfolgung eines einzelnen Markers zusammen mit einer Gruppe von Markern, wie den Multipunktmodellen, die in Abschnitt 4.3 beschrieben werden. Ein beispielhafter Fall ist eine einfache Kopfverfolgung ohne Orientierung zusammen mit der Verfolgung eines Zeigegeräts mit sechs Freiheitsgraden.
- Die Verfolgung mehrerer einzelner Marker, die nicht fest zueinander positioniert sind, als Grundlage für die in Kapitel 6 beschriebenen dynamischen Modelle. Beispielsweise wird dies für Echtzeit Motion Capturing, die Verfolgung einer Hand ohne spezielles Zeigegerät, oder die Kontrolle von Gelenkarmen genutzt.
- Um das 3D-Punktmodell um eine Eigenschaft erweitern zu können: Einen Wahrscheinlichkeitswert, der angibt, ob ein Punkt möglicherweise bald oder bereits von einem weiteren verdeckt werden könnte.

Die ersten beiden Fälle werden in Abschnitt 4.1.1 beschrieben, wobei der erste Fall weitestgehend trivial ist und nur dann eines geringen Zusatzaufwands bedarf, wenn aufgrund von Störungen (zum Beispiel wechselnde Lichtverhältnisse) zusätzliche 3D-Punkte rekonstruiert werden, für die kein Marker existiert. Die beiden anderen Fälle werden in Abschnitt 4.1.3 behandelt. Dort werden auch die Vor- und Nachteile diskutiert, grundsätzlich einzelne 3D-Punkte zu verfolgen.

Da ein naheliegendes und sinnvolles Mittel für die Verfolgung der 3D-Punkte auf der oder den Positionen aus den letzten Rekonstruktionsschritten beruht, wird das in Abschnitt 3.3 aufgestellte Punktmodell um folgende Eigenschaften erweitert:

5. $L = (X_1, X_2, \dots, X_p)$ eine Liste der letzten p Koordinaten des Punktes und
6. einen **Identifikationsindex** id .

Dabei unterscheidet sich der Identifikationsindex von dem Klassifikationsindex dadurch, dass er ein Merkmal für genau einen bestimmten Punkt darstellt, der Klassifikationsindex jedoch den Punkt einer bestimmten Gruppe zuordnet, also klassifiziert. Insofern wird der Identifikationsindex in jedem Schritt nur einmal vergeben, ein Klassifikationsindex kann jedoch mehrmals vergeben werden.

Die Zuweisung des Identifikationsindex ist bei der Verfolgung mehrerer Punkte problematisch, wenn Punkte immer wieder verdeckt werden. Sobald sie wieder unverdeckt sind, kann man diese unter Umständen von völlig neu hinzukommenden Punkten nicht unterscheiden. Für die Identifikation eines Punktes mit Hilfe der Liste L der vorausgegangenen Positionen gibt es folgende naheliegende Ansätze:

- Vergleich mit der letzten Position oder
- Vergleich mit einer aus L prädizierten Position.
- Einer der ersten beiden Fälle und zusätzliches Modellwissen, falls der Punkt klassifiziert, das heißt einem Modell zugeordnet, ist. Ein solches Modellwissen kann beispielsweise die Auswirkung der Rotationsgeschwindigkeit eines Modells auf die voraussichtliche nächste Position eines einzelnen Punktes sein.

Diese und weitere Ansätze für die Identifikation und Verfolgung der Punkte werden in den folgenden Abschnitten dargestellt.

Soweit die Liste L der vorausgegangenen Positionen, beziehungsweise Koordinaten eines Punktes vorliegt, kann sie auch für eine Filterung genutzt werden. Da sich diese Methoden für die Positionen einzelner Punkte nicht von denen für die Positionen eines ganzen Multipunktmodells unterscheiden, wird dies übergreifend in Kapitel 7 beschrieben.

4.1.1 Verfolgung eines einzelnen Markers

Falls nur ein Marker verfolgt wird, kann der Punkt einfach durch die 3D-Rekonstruktion bestimmt werden, soweit keine Störungen auftreten. Als **Störungen** oder Störpunkte werden in dieser Arbeit solche 3D-Punkte bezeichnet, die zwar rekonstruiert wurden, aber für die es keine physikalische Entsprechung gibt. Bei optischen Systemen kann es dazu beispielsweise durch Lichtreflektionen oder allgemein ungünstige Lichtverhältnisse kommen. Falls keine Störungen auftreten, kann auch die Liste L der letzten Positionen einfach gefüllt werden.

Existieren Störpunkte, sodass eine Menge $\{X_W^1, \dots, X_W^m\}$ rekonstruierter 3D-Punkte mit $m \geq 2$ vorliegt, kann die Liste L verwendet werden, um durch Abstandsmessung zur letzten Position, beziehungsweise zu einer prädizierten Position, besser entscheiden zu können, welches der gesuchte Punkt ist.

Eine weitere Möglichkeit besteht darin, den Rekonstruktionsfehler als Kriterium zur Findung des gesuchten Punktes heranzuziehen. In Abhängigkeit davon, wie die Störung entsteht, zeigt die Praxis, dass der Rekonstruktionsfehler häufig höher ist, als bei korrekt rekonstruierten Markern, sodass bei Punkten mit einem hohen Rekonstruktionsfehler vermutet werden kann, dass es sich hierbei um Störpunkte handelt. Die Annahme, dass Punkte mit geringem Rekonstruktionsfehler automatisch keine Störpunkte sind, ist jedoch problematisch.

4.1.2 Verfolgung eines einzelnen Markers zusammen mit einem Modell

Wird ein einzelner Marker zusammen mit einem Modell verfolgt, wie in Abbildung 4.1(b) dargestellt, so bieten sich vor allem zwei Verfahren an, die auch kombiniert werden können.

Die erste Methode besteht darin, zunächst die Erkennung des Modells auszuführen. Ausgehend davon, dass das Modell identifiziert werden kann, wird bei jedem Punkt der



(a) Shutterbrille für VR mit einem retroreflektierenden Marker zur Positionsbestimmung.

(b) Infrarotkameranahsicht auf Benutzer mit Brille (gelb), die mit einem Marker besetzt ist, und Zeigegerät (grün) mit mehreren Markern (rot).

Abbildung 4.1. Verfolgung der Kopfposition mit einem Marker an Shutterbrille.

Klassifikationsindex des Modells gesetzt, der als zum Modell zugehörig erkannt wurde. Im günstigsten Fall bleibt nach der Erkennung des Modells nur ein nicht klassifizierter Punkt übrig. Zumindest kann bei mehreren nicht klassifizierten Punkten so wie bei der im letzten Abschnitt beschriebenen Erkennung eines einzelnen Punktes vorgegangen werden. Insbesondere kann die letzte Position als Entscheidungskriterium für die Identifikation verwendet werden.

Die zweite Methode setzt voraus, dass es die Anwendung erlaubt, starke Randbedingungen zu definieren. Solche Randbedingungen für einen Punkt können zum Beispiel sein:

- Aus einer Menge von Punkten wird der Punkt X_W^i ausgewählt, der immer die größte x-Koordinate besitzt. Analog für die kleinste x-Koordinate und die größte oder kleinste y- oder z-Koordinate.
- Ähnlich wie obiger Ansatz nur, dass der kleinste oder größte Abstand zu einem bestimmten Punkt im Raum oder zu einer Ebene als Kriterium gilt.

Bei der Verfolgung eines Kopfes zur Blickrichtungserkennung kann man beispielsweise häufig die Annahme aufstellen, dass die Hand unterhalb des Kopfes bleibt, und somit die Kopfposition in der vertikalen Achse immer einen Extremwert im Vergleich zu den anderen Punkten annimmt. In [Dorfmueller 98] werden mit je einem Marker die Kopfposition und eine Handposition verfolgt, wobei ebenfalls mit der Annahme gearbeitet wird, dass sich der Kopf zumindest am Anfang höher als die Hand befindet. Weitere Kriterien, die dort speziell verwendet werden sind, dass der Marker für die Kopfverfolgung ständig

zu sehen ist, der für die Hand nur partiell und dass der Kopfmarker beim Start der Applikation zeitlich als erstes zu sehen ist.

4.1.3 Identifikation mehrerer einzelner Marker

Die Verfolgung mehrerer einzelner Marker ist besonders interessant, wenn diese Marker nicht an einem oder wenigen festen Objekten angebracht sind, sondern an einem flexiblen Objekt, wie dem menschlichen Körper oder einem Körperteil, beziehungsweise an zahlreichen einzelnen Objekten. Die Verfolgung der einzelnen Marker kann als eigenständiges Trackingverfahren angesehen werden, falls das Ergebnis des Trackings beispielsweise Trajektorien der einzelnen Punkte sein sollen. Häufiger ist die Verfolgung der einzelnen Marker Grundlage für die Verfolgung dynamischer Modelle, wie beispielsweise beim klassischen Motion Capturing, insbesondere aber bei Motion Capturing, das in Echtzeit erfolgen soll.

Das Ergebnis, welches die Verfolgung mehrerer einzelner Marker haben soll, ist:

1. Nach jedem Frame liegt eine Liste mit 3D-Punkten vor, von denen jeder einen eindeutigen Identifikationsindex trägt. Die Länge der Liste kann von Frame zu Frame unterschiedlich lang sein.
2. Der zu einem bestimmten Marker rekonstruierte Punkt erhält über alle Frames hinweg immer denselben Identifikationsindex, falls er im entsprechenden Frame vom Trackingsystem rekonstruiert werden kann.
3. Auch nach einer Zeitspanne, in der der Marker verdeckt war, soll sein rekonstruierter Punkt wieder den alten Identifikationsindex erhalten, was impliziert, dass dieser Index nicht bereits an einen anderen Punkt vergeben worden ist.

Vor allem der letzte Punkt ist schwierig zu erfüllen, da die einzelnen Marker für das Trackingsystem in einem statischen Zustand nicht zu unterscheiden sind, sondern hauptsächlich nur durch ihre Bewegung zwischen zeitlich aufeinander folgenden Frames. Soweit die Verdeckung eines Markers nicht zu lange anhält und insbesondere keine Richtungswechsel des Markers auftreten, kann durch die Prädiktion der Position während der Verdeckung versucht werden, diese Phase zu überbrücken. Abbildung 4.2 zeigt die Trajektorien mehrerer verfolgter Punkte, wobei diese nach Unterbrechungen zum Teil wiedererkannt werden und zum Teil einen neuen Identifikationsindex erhalten.

In [Magneau 02a] und [Magneau 02b] wird ein Verfahren zur Identifikation und Positionsbestimmung mehrerer einzelner Marker beschrieben. Die zentrale Idee ist dabei die Erzeugung von Matrizen T , die die Ähnlichkeit zweier Listen von 3D-Punkten ausdrücken. Die beiden Listen stehen für zwei zeitlich aufeinanderfolgende Frames. Die Elemente dieser Matrix setzen sich aus einer Summe von Wahrscheinlichkeiten P^{ij} zusammen, die sich wiederum aus einem Formvergleichsparameter P_L^{ij} und einem Flächenvergleichsparameter P_S^{ij} zweier Dreiecke durch $P^{ij} = P_L^{ij} P_S^{ij}$ zusammensetzen. Die beiden Parameter werden wiederum aus je drei Punkten der entsprechenden Listen gebildet und drücken daher also eine Ähnlichkeit zwischen zwei Dreieckskonstellationen aus. Diese

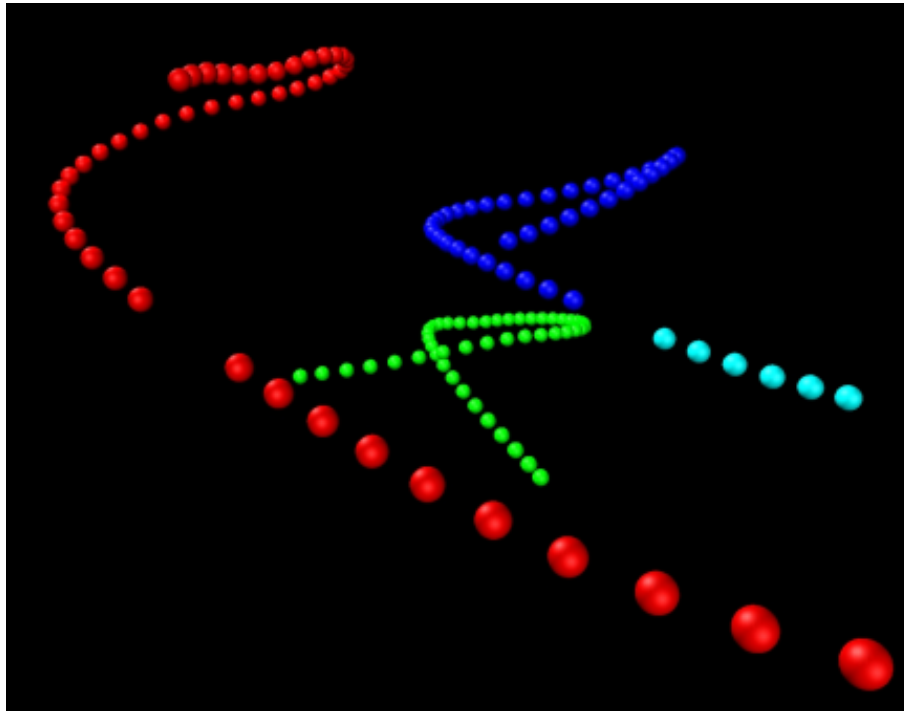


Abbildung 4.2. Trajektorien von drei Markern. Jeder rekonstruierte 3D-Punkt wird als Kugel, der Identifikationsindex als Farbe dargestellt. Die Trajektorie im Vordergrund (rot) wird trotz kurzer Unterbrechung korrekt fortgesetzt, die mittlere Trajektorie (grün) hat keine Unterbrechung, die rechte Trajektorie (blau) erhält nach der Unterbrechung fälschlicherweise einen neuen Index.

Matrizen werden sowohl in einem Vorverarbeitungsschritt zur Bestimmung von Toleranzwerten, als auch zur späteren Identifikation, verwendet.

Der Grundgedanke dieses Ansatzes, zuverlässiges Tracking trotz Markerverdeckungen zu erreichen, ohne obligatorisch dabei die Anzahl der Kameras zu erhöhen, entspricht dem dieser Arbeit. Allerdings ist der Aufwand für die Berechnung dieser Wahrscheinlichkeitsmatrizen relativ hoch.

Algorithmus

Im Folgenden wird ein im Rahmen dieser Arbeit entwickelter Ansatz beschrieben, der auf kein Initialwissen über Anzahl oder Konstellation von Markern zurückgreift. Im initialen Zustand erhält der Algorithmus eine Menge \mathcal{X}_W nicht klassifizierter und identifizierter 3D-Punkte, die im letzten Frame rekonstruiert wurden. Für jeden Punkt $X_W^i \in \mathcal{X}_W$ wird eine Liste L_j veränderlicher aber begrenzter Länge angelegt, deren erstes Element der Punkt selbst ist. Jeder Liste wird ein eindeutiger Identifikationsindex id_j zugewiesen, der auch als Identifikationsindex dem 3D-Punkt zugewiesen wird. Abbildung 4.3 zeigt Indizes, die 3D-Punkten zugeordnet sind und auf die zwei entsprechenden Kamerabilder überlagert wurden. Damit ist die Initialisierung des Algorithmus abgeschlossen.

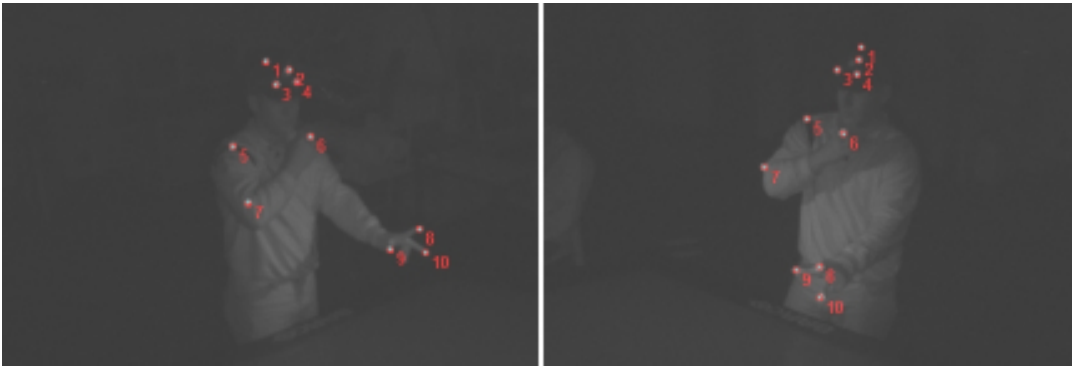


Abbildung 4.3. 3D-Punkten zugeordnete Indizes auf die Kamerabildern überlagert.

Außerdem erhält jede Liste L_j einen **Vertrauenswert** t_j als Parameter, der immer dann auf 1 gesetzt wird, wenn ein Punkt der Liste zugeordnet werden konnte. Kann kein Punkt zugeordnet werden, kann eine prädizierte Position statt eines tatsächlich gefundenen Punktes gesetzt werden, wobei dann der Vertrauenswert um einen Wert t_{red} zwischen 0 und 1 reduziert wird. Ist der Vertrauenswert kleiner oder gleich 0, so wird die Liste L_j aufgelöst und damit steht auch der Identifikationsindex id_j für die Vergabe an eine neue Liste zur Verfügung. Wird die Reduktion des Vertrauenswertes als $t_{red} = 1$ gewählt, so werden Listen sofort aufgelöst, sobald ihnen kein Punkt mehr zugeordnet werden kann. Die Reduktion des Vertrauenswertes erfolgt hier durch Subtraktion, da sich damit einfach eine Anzahl von Schritten bis zur Auflösung der Liste definieren lässt. Alternativ hierzu könnte auch ein Reduktionsfaktor gewählt werden.

Als weiterer Parameter wird der Index i eines der Liste zugeordneten Punktes $X_W^i \in \mathcal{X}_W$ als Variable *lst* (*last index*) abgespeichert, damit bei dem Punkt mit demselben Index im folgenden Frame als erstes überprüft werden kann, ob es sich dabei um den zu verfolgenden Punkt handelt. Die Verwendung dieses Parameters nutzt den Umstand, dass die Kamerabilder immer auf die gleiche Art und Weise abgearbeitet werden, sodass die Wahrscheinlichkeit hoch ist, dass die zu verfolgenden Punkte den gleichen Index in \mathcal{X}_W besitzen wie beim vorhergehenden Frame. Eine Liste L_j besitzt also insgesamt folgende zusätzliche Parameter:

- id - den unveränderlichen Identifikationsindex,
- t - den variablen Vertrauenswert, für den initial $t = 1$ gilt,
- lst - den Index des zugeordneten Punktes aus der letzten Menge $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$. Es gilt also $1 \leq lst \leq m$.
- P - die prädizierte Position des Punktes.

Für alle Listen gemeinsam werden außerdem noch folgende, während des Trackings unveränderliche, Parameter festgelegt:

- d_{max} , ein Grenzwert für die Distanz eines neu zuzuordnenden Punktes zur letzten Position, beziehungsweise zu einer prädizierten Position. Dieser Wert sollte sich

auch an der Framerate des Kamerasystems orientieren. Je höher diese ist, umso kleiner kann d_{max} gewählt werden.

- $t_{red} \in (0, 1]$, der Wert, um den der Vertrauenswert reduziert wird, falls kein Punkt zugeordnet werden kann.
- w_1 und $w_2 \in [0, 1]$, zwei Gewichtungparameter, die im Folgenden bei der Zuordnung von Punkten zu Listen verwendet werden.

Abbildung 4.4 zeigt einen Überblick über die Punklisten und deren Parameter.

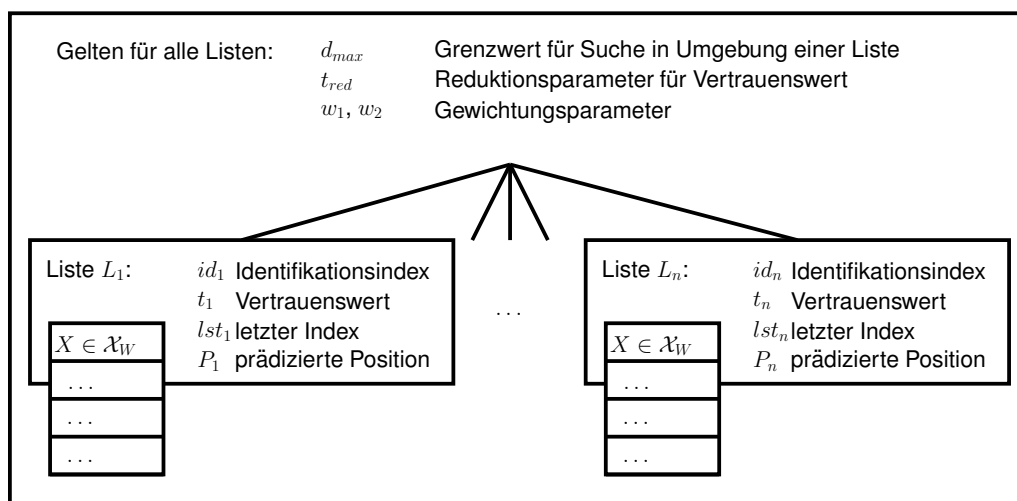


Abbildung 4.4. Aufbau der Liste der Punklisten.

Mit diesen Festlegungen wird in den folgenden Frames jeweils die Menge \mathcal{X}_W der aktuell rekonstruierten 3D-Punkte verarbeitet, sodass jeder Punkt X_W^i einen eindeutigen Identifikationsindex erhält.

Für jede Liste L_j wird zunächst eine Position P_j für den gesuchten Punkt prädiziert. Enthält die Liste nur ein Element, so ist die prädizierte Position gleich der letzten Position. Ist die Liste länger, so wird linear prädiziert, eventuell auch mit höherer Ordnung. Die Methoden für die Prädiktion entsprechen denen aus Kapitel 7.

Um die Punkte $X_W^i \in \mathcal{X}_W$ einer der Listen L_j zuzuordnen, wird zunächst eine Matrix $W \in [0, 1]^{m \times n}$ angelegt, deren Einträge initial alle 0 sind. Dabei ist m die aktuelle Anzahl der Punkte X_W^i und n die Anzahl der vorhandenen Listen L_j .

Die Matrix W , deren Elemente Wahrscheinlichkeitswerte sind, ähnelt auf den ersten Blick der „Probabilistic Matrix“ aus [Magneau 02a] und [Magneau 02b], wird jedoch auf völlig andere Weise gefüllt und verwendet. Je höher der Eintrag W_{ij} ist, desto wahrscheinlicher ist es, dass der Punkt X_W^i der Liste L_j zugeordnet wird. Zunächst wird die Matrix W in zwei Schritten gefüllt:

1. Für jede Liste L_j wird aus der Menge \mathcal{X}_W der Punkt $X_W^{lst_j}$ ausgewählt, falls dieser vorhanden ist. Befindet er sich in der Umgebung der entsprechenden prädizierten

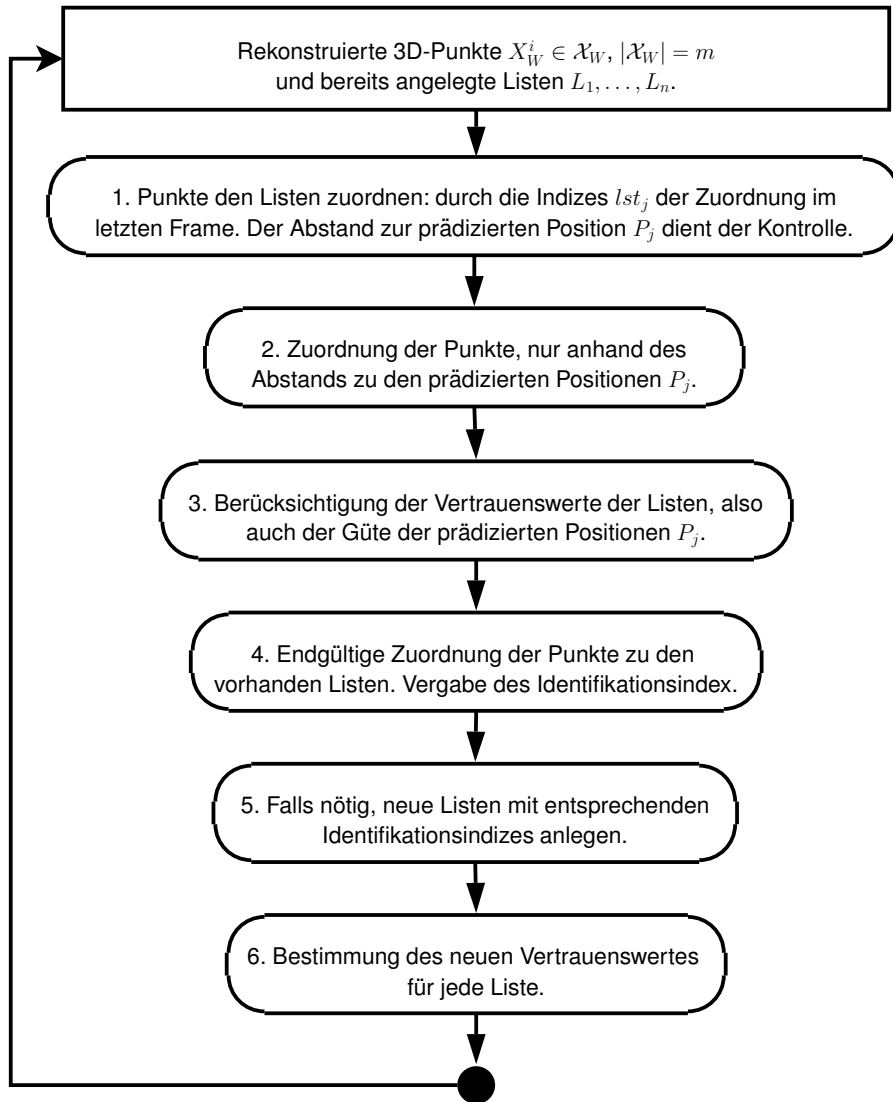


Abbildung 4.5. Algorithmus für die Verfolgung mehrerer 3D-Punkte.

Position P_j , gilt also $d < d_{max}$ mit $d = |P_j - X_W^{lst_j}|$, dann wird $W_{lst_j} = w_1 \cdot \frac{d_{max} - d}{d_{max}}$ gesetzt.

2. Jeder Punkt $X_W^i \in \mathcal{X}_W$ wird daraufhin untersucht, ob er sich in der Umgebung einer prädizierten Position einer Liste L_j befindet. Ist dies der Fall, wird $W_{ij} = w_2 \cdot \frac{d_{max} - d}{d_{max}}$ mit $d = |P_j - X_W^i|$ gesetzt, falls $W_{ij} = 0$ und damit dieses Element im ersten Schritt noch nicht gesetzt wurde. Mehrere von Null verschiedene Werte pro Zeile oder Spalte sind dabei möglich.

Für die Werte w_1 und w_2 bieten sich beispielsweise $w_1 = 1,0$ und $w_2 = 0,8$ an, was dem letzten Index als Auswahlkriterium durch $w_1 > w_2$ einen Einfluss gewährt. Durch die Multiplikation des jeweiligen Gewichtes mit $\frac{d_{max} - d}{d_{max}}$ wird die tatsächliche Entfernung zum prädizierten Punkt berücksichtigt. Im nächsten Schritt wird der aktuelle Vertrauenswert

der Listen L_j berücksichtigt:

3. Die Matrix W wird mit einer Matrix D_t multipliziert, wobei $D_t \in [0, 1]^{n \times n}$ eine Diagonalmatrix ist, deren Diagonaleinträge die Vertrauenswerte t_j der Listen L_j sind. Das Produkt WD_t der beiden Matrizen, als Gewichtung der aktuellen Vertrauenswerte, wird im Folgenden als V bezeichnet, das heißt $V = WD_t$.
4. In der Matrix V wird der maximale Eintrag V_{ij} bestimmt. Der Punkt X_W^i wird der Liste L_j zugeordnet und er erhält den Identifikationsindex id_j der Liste. Die Koordinaten des Punktes werden als aktuelle Position der Liste hinzugefügt. Alle Einträge in der i -ten Zeile und der j -ten Spalte von V werden auf 0 gesetzt. Daraufhin wird erneut der maximale Eintrag in V gesucht, bis alle Einträge 0 sind.
5. Für Punkte X_W^i , die noch keiner Liste zugeordnet werden konnten, werden neue Listen angelegt, jeweils mit Vertrauenswert $t_j = 1$.
6. Als letzter Schritt werden die Vertrauenswerte neu gesetzt. Bei jeder Liste, der ein Punkt zugeordnet werden konnte, wird der Vertrauenswert $t = 1$ gesetzt. Bei allen anderen wird t als $t - t_{red}$ abgespeichert. Der entsprechende prädizierte Punkt P wird als aktuelles Element abgespeichert. Alle Listen, die danach einen Vertrauenswert $t \leq 0$ haben, werden gelöscht.
7. Im nächsten Frame beginnt die Verarbeitung wieder bei Schritt 1.

Abbildung 4.5 zeigt den Ablauf des Algorithmus schematisch.

Dieses Vorgehen erlaubt eine temporäre Fortführung der Punktlisten, auch wenn kein Punkt zugeordnet werden konnte. Wie lange diese Listen ohne erneut gefundenen Punkt fortgeführt werden, hängt von der Wahl von t_{red} ab. Für die Ausgabe oder Weiterverarbeitung der 3D-Punkte kann optional die Menge \mathcal{X}_W der tatsächlich rekonstruierten Punkte erweitert werden, und zwar um die prädizierten Punkte derjenigen Listen, denen kein Punkt zugeordnet wurde. Dies ermöglicht die Ausgabe der vermuteten Position kurzzeitig verdeckter Marker. Eine zusätzliche Speicherung des Vertrauenswerts im Punktmodell ermöglicht es, bei der späteren Verarbeitung der Punkte daraus zu schließen, ob dieser verdeckt war.

Beim Anlegen einer neuen Liste wird auch ein neuer Identifikationsindex vergeben. Entweder wird dabei in aufsteigender Reihenfolge immer ein neuer Index gewählt, oder wieder frei gewordene Indizes werden durch die Auflösung von Listen neu verteilt. In Anhang C befindet sich ein kurzes Beispiel, das anhand eines Schrittes des Algorithmus die Wirkung der einzelnen Parameter verdeutlicht.

Weiterverarbeitung der identifizierten Punkte

Die Verfolgung einzelner Marker dient meist nur als Vorverarbeitung für weitere Trackingalgorithmen, falls nicht ohnehin nur die Trajektorien einzelner Marker von Interesse sind. Die in den folgenden Abschnitten beschriebenen Algorithmen greifen auf die Verfolgung

einzelner Marker nur dann zurück, wenn sie nicht fest zueinander platziert sind, siehe Kapitel 6. In den Fällen, in denen die Marker fest zueinander platziert sind, gibt es durch die festen geometrischen Strukturen zuverlässigere Methoden zur Identifikation der Markerkonstellationen, sodass auf die obige Vorverarbeitung, die eine geringere Zuverlässigkeit hat, verzichtet wird.

4.2 Zweipunktmodelle

Zweipunktmodelle stellen die einfachste Form der unterscheidbaren Modellbeschreibungen auf Punktbasis dar. Die wesentliche Eigenschaft eines Zweipunktmodells ist die Distanz d zwischen den zwei zu erkennenden Punkten. Weitere Eigenschaften eines Zweipunktmodells, wie die letzte Position, ergeben sich erst bei Betrachtung des Modells über eine Zeitspanne, also bei dessen Verfolgung. Diese und weitere mögliche Eigenschaften werden in Abschnitt 4.2.1 vorgestellt.

Durch die Verwendung von genau zwei Punkten ist es zwar möglich, ein solches Zweipunktmodell zu identifizieren und dessen Position zum Beispiel als Mittelpunkt des gefundenen Punktpaars anzugeben, eine räumliche Orientierung des Zweipunktmodells kann jedoch nicht bestimmt werden. Insofern handelt es sich bei der Verwendung eines Zweipunktmodells um ein **3DOF-Trackingverfahren**, also ein Trackingverfahren, bei dem drei Freiheitsgrade, nämlich die Position, bestimmt wird. Die einzigen zusätzlichen Informationen, die berechnet werden können, sind die Richtungsvektoren zwischen den beiden Punkten. Ein eindeutiger Richtungsvektor kann in speziellen Anwendungen, wie zum Beispiel einem Zeigegerät, bei dem wiederum eindeutig ist, wie es ausgerichtet ist, zusätzlich verwendet werden, so dass man in diesem Fall von einem 5DOF-Trackingverfahren sprechen kann.

Die Verwendung von Zweipunktmodellen bietet sich also dort an, wo nur die 3D-Position von Objekten bestimmt werden soll und die Objekte jederzeit eindeutig identifizierbar sein müssen, was mit Einpunktmodellen schwierig umzusetzen ist. Die Anzahl der gleichzeitig zu verfolgenden Zweipunktmodelle ist beschränkt. Je nach Genauigkeit des Trackingsystems, das heißt, je nachdem, wie genau die (relativen) Positionen der vorab bestimmten einzelnen 3D-Punkte sind, müssen die Distanzen d_k der Zweipunktmodelle gewählt werden. Je ungenauer die Bestimmung der 3D-Punkte ist, desto größer müssen die Abstände zwischen den verschiedenen Distanzen d_k gewählt werden. Zusätzlich hat man im Allgemeinen einen Maximalwert für die Größe der festzulegenden Distanzen, der durch die konkrete Anwendung, zum Beispiel durch dessen Interaktionsvolumen, vorgegeben wird. Daraus ergibt sich die maximale Anzahl der Zweipunktmodelle, die definiert werden kann.

In Kapitel 8.3 ist eine Anwendung der Zweipunktmodelle für eine Interaktion mit acht verschiedenen Benutzern beschrieben. Diese Anwendung sowie Teile der hier beschriebenen Algorithmen wurden in [Schwald 05c] publiziert. Jeder Benutzer trägt in diesem Beispiel eine Kappe, auf der zwei fest miteinander verbundene Infrarotmarker befestigt sind, siehe Abbildung 4.6. Diese Abbildung zeigt einen weiteren Vorteil dieses Ansatzes: Die Zweipunktmodelle sind - zumindest bei der Verwendung von Infrarotmarkern - mühelos in der Praxis umzusetzen und an einem Objekt, in diesem Fall der Kappe,



Abbildung 4.6. Eine mit zwei Infrarotmarkern besetzte Kappe.

anzubringen. Für die später beschriebenen Multipunktmodelle sind die entsprechenden Markermodelle komplizierter zu erstellen und nehmen mehr Raum ein, dafür besitzen sie jedoch Redundanzen, das heißt, es müssen nicht unbedingt alle Punkte eines Modells entdeckt werden. Bei den Zweipunktmodellen müssen bis auf kurze Zeitspannen beide Punkte erkannt werden.

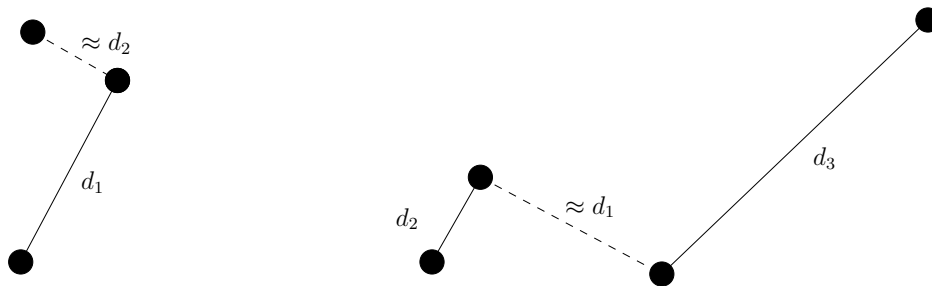


Abbildung 4.7. Mehrdeutigkeiten (gestrichelte Linien) durch drei Zweipunktmodelle (durch durchgezogene Linien gekennzeichnet) und einzelne 3D-Punkte.

Die Identifikation von Zweipunktmodellen durch deren zugeordnete Distanzen ist dann problemlos, wenn die gesuchten Distanzen jeweils nur einmal in der vorliegenden Punktemenge existieren. Bei der Verwendung mehrerer Zweipunktmodelle ist es möglich, dass sich Punktepaare bilden, die eigentlich kein Modell darstellen, aber dieselbe Distanz wie ein solches haben, siehe Abbildung 4.7. Daher genügt es nicht, sich auf die Distanzmes-

sung zwischen Punkten zu beschränken, sondern zeitliche Eigenschaften, wie die letzte Position eines Modells, müssen mit berücksichtigt werden. Probleme können dennoch entstehen, wenn die Modelle dicht beieinander liegen und dabei unbewegt bleiben, da es dann zu einer großen Anzahl von Doppeldeutigkeiten kommen kann, die nur schwer aufzulösen ist. Für dieses Problem gibt es keine allgemeine Lösung, es muss daher applikationsabhängig behandelt werden.

Zusammengefasst lassen sich folgende Merkmale der Zweimarkermodelle, beziehungsweise Zweipunktmodelle in Kombination mit einem typischen Infrarottrackingsystem feststellen:

- 3DOF-Trackingverfahren zur Ermittlung der Position
- Einfache Umsetzung und Integration der Infrarotmarkermodelle
- Je nach (Distanz-)Genauigkeit lassen sich zahlreiche Modelle verfolgen.
- Geeignet, falls die Modelle meist in Bewegung sind. Insbesondere sollten die Modelle nicht statisch auf einem Haufen liegen.

Im folgenden Abschnitt 4.2.1 wird der Algorithmus zur Identifikation und Positionierung der Modelle beschrieben. Eine formale Definition der Zweimarkermodelle wird passend zum Algorithmus aufgestellt. Ergänzend wird in Abschnitt 4.2.2 auf die Glättung der aufeinander folgenden ermittelten Positionen eingegangen. Abschnitt 4.2.3 schließt die Beschreibung der Zweipunktmodelle mit der Diskussion verschiedener Varianten ab.

4.2.1 Algorithmus und Definition für Zweipunktmodelle

Da sich für aus nur zwei Punkten bestehende Modelle kein eindeutiges lokales Koordinatensystem aufspannen lässt, kann bei diesem Ansatz nur eine 3D-Position bestimmt werden, gegebenenfalls noch ein Richtungsvektor zwischen den beiden Punkten mit unbekanntem Vorzeichen. Daher wird in diesem Abschnitt die in Definition 3 beschriebene, vereinfachte Bezeichnung X_i für Punkte X_W^i im Weltkoordinatensystem verwendet und auf Indizes, die das Koordinatensystem angeben, verzichtet.

Die Modellbeschreibung besteht im Fall der Zweipunktmodelle in erster Linie aus der Angabe einer Distanz d_k zwischen den Markern. Die gesamte Menge der m gesuchten Distanzen ist $\mathcal{D} = \{d_k \in \mathbb{R} : 1 \leq k \leq m \in \mathbb{N}\}$. Ausgehend von einer wie in Definition 2 auf Seite 42 vorgegebenen Distanztoleranz δ werden die Distanzen d_k wie folgt gewählt, um Verwechslungen zu vermeiden:

- $d_1 > \delta$ und
- $d_k > d_{k-1} + 2\delta$ für $2 \leq k \leq m$.

Mit diesen Voraussetzungen gliedert sich der Algorithmus für die Detektion, beziehungsweise Verfolgung der Zweipunktmodelle wie in Abbildung 4.8 dargestellt, in fünf Schritte:

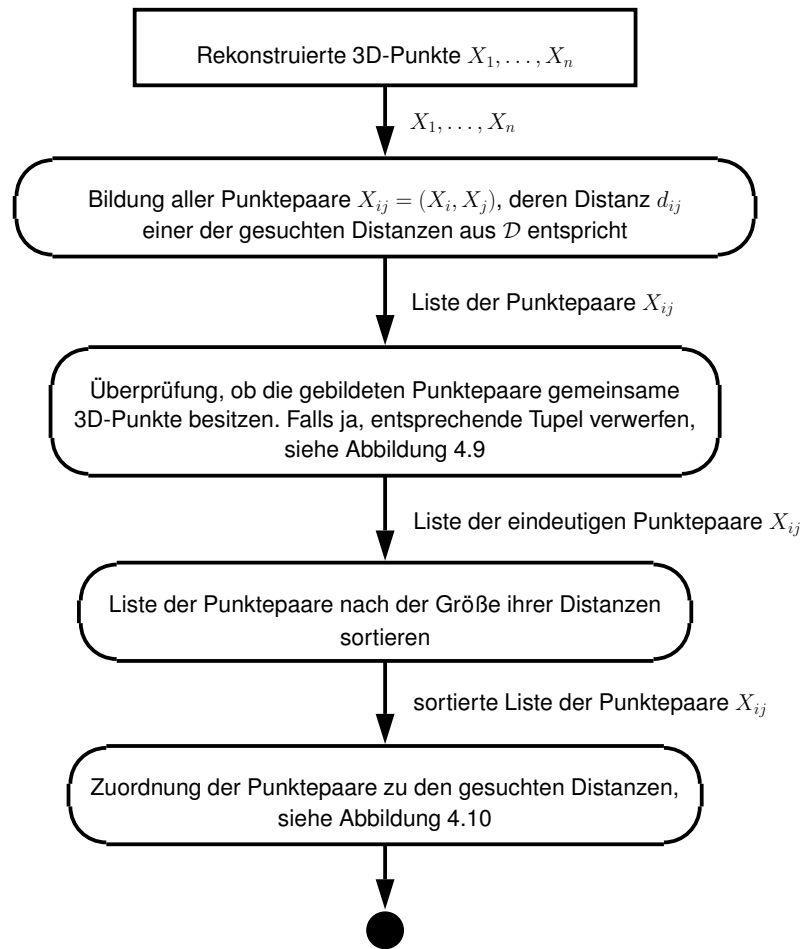


Abbildung 4.8. Algorithmus zur Erkennung mehrerer 3D-Punktpaare.

1. Rekonstruktion der 3D-Punkte X_1, \dots, X_n mit $X_i \in \mathbb{R}^3, n \in \mathbb{N}$ aus den aktuellen Kamerabildern.
2. Bildung aller Punktpaare $X_{ij} := (X_i, X_j)$, für die ein $d_k \in \mathcal{D}$ mit $||X_i - X_j| - d_k| < \delta$ existiert, wobei $X_{ij} = X_{ji}$.
3. Überprüfung, ob Punktpaare X_{ij} und X_{il} mit $j \neq l$ existieren, die aus einem gemeinsamen 3D-Punkt X_i gebildet wurden. Falls ja, wird die Liste der Punktpaare so lange reduziert, bis für alle X_{ij} und X_{hl} gilt: $i \neq h \wedge i \neq l \wedge j \neq h \wedge j \neq l$. Die Reduktion der Liste ist in Abbildung 4.9 genauer beschrieben.
4. Die nun vorliegende Liste der Punktpaare X_{ij} mit Distanzen $d_{ij} = |X_i - X_j|$ wird nach der Größe der d_{ij} in aufsteigender Reihenfolge sortiert.
5. Suche der Modelle anhand der jeweiligen Distanz $d_k \in \mathcal{D}$ in der Liste der Punktpaare X_{ij} , wie in Abbildung 4.10 dargestellt. Berücksichtigt werden dabei für jedes Modell die letzte Position und ein **Vertrauenswert** t_k , der immer wieder angepasst

wird. Für die als detektiert geltenden Zweipunktmodelle wird die Position gespeichert.

Dem eben beschriebenen Ablauf entsprechend, lässt sich für das hier verwendete Zweipunktmodell somit folgende Definition formulieren:

Definition 4 *Ein Zweipunktmodell besteht aus folgenden festen Parametern:*

- d *einer Distanz zur Erkennung der Modelle, $d \in \mathbb{R}^+$,*
- t_{limit} *einem Grenzwert für den Vertrauenswert, $0 < t_{limit} \leq 1$,*
- t_+ *einem Faktor zur Erhöhung des Vertrauenswertes, $t_+ > 1$,*
- t_- *einem Faktor zur Senkung des Vertrauenswertes, $0 < t_- < 1$ und*
- max *einem Grenzwert für die maximale Distanz zur letzten Position des Modells.*

Die während der Detektion zu speichernden und veränderlichen Parameter sind:

- t *ein Vertrauenswert, $0 < t \leq 1$,*
- P *die aktuelle Position, $P \in \mathbb{R}^3$ und*
- P_{alt} *die letzte berechnete Position, $P_{alt} \in \mathbb{R}^3$.*

Der Grenzwert max für die maximale Distanz zur letzten Position des Modells basiert dabei auf der Annahme, dass sich ein ZweimarkermodeLL höchstens mit einer bestimmten, maximalen Geschwindigkeit bewegt und deshalb in Abhängigkeit der Framerate maximal eine Strecke max zwischen zwei Frames zurücklegen kann.

Die Veränderung des Vertrauenswertes erfolgt multiplikativ statt additiv, um den Vertrauenswert nicht linear zu verändern, da der Vertrauenswert auch zur Gewichtung bei der Glättung der Bewegung des Modells verwendet wird und die Bewegung dadurch subjektiv besser wirkt. Die Glättung der Bewegung wird im folgenden Abschnitt beschrieben.

4.2.2 Glättung aufeinander folgender Positionen

Falls ein Punktepaar für ein entsprechendes Zweipunktmodell gefunden wird, dann wird die Position des Modells als Mitte der beiden Punkte definiert. Falls die letzte Position zu weit von der aktuellen entfernt liegt, kann dieses Springen zwischen zwei Positionen unerwünscht sein. Daher bietet sich an dieser Stelle das in Kapitel 7.3 beschriebene Nachführen der Position an, bei dem wiederum P_{alt} verwendet wird. Dies kann wie folgt umgesetzt werden:

Falls $t < t_{limit}$, wird zunächst wie bisher die Position P als Mitte des Punktepaars bestimmt. Um die letzte Position zu berücksichtigen, kann dann die neue Position als Mittelung zwischen der tatsächlich berechneten und der letzten Position durch

$$P_{neu} = t \cdot P + (1 - t) \cdot P_{alt} \quad (4.1)$$

bestimmt werden, wobei t der Vertrauenswert des Modells ist. Ist $t = 1$, so entspricht P_{neu} genau der Mitte des Punktepaars. P_{neu} wird als aktuelle Position P abgespeichert. Diese Technik hat unter anderem folgenden Effekt:

Wenn ein Modell über einen längeren Zeitraum nicht erfasst wird, sinkt der Vertrauenswert t kontinuierlich, indem t durch $t = t_{alt} \cdot t_-$ in jedem Schritt neu berechnet wird.

Kann dann die aktuelle Position P des Modells wieder ohne Unterbrechung erfasst werden, springt die Position im ersten Moment nicht direkt von P_{alt} zu P , sondern sie bewegt sich in mehreren Schritten immer schneller werdend auf die jeweils aktuelle Position zu, wobei t durch $t = t_{alt} \cdot t_+$ erhöht wird, bis der Vertrauenswert $t \geq t_{limit}$. Sobald dies erreicht ist, wird die aktuelle Position P nicht mehr durch die Gleichung (4.1) berechnet und damit durch P_{alt} korrigiert. Wird für t ein Wert größer 1 berechnet, wird $t = 1$ gesetzt.

Dieses Nachführen einer Position ist allerdings nur dann interessant, wenn es bei den Trackingergebnissen auf gleichmäßige Bewegungen ankommt, nicht jedoch, wenn Genauigkeit die größere Rolle spielt. Falls letzteres der Fall ist, sollte auf das Nachführen verzichtet werden, da die Position kurzzeitig stark verfälscht wird.

Falls kein Punktpaar für das Zweipunktmodell gefunden wurde, der Vertrauenswert t dennoch sehr hoch ist, wird die Position mit Hilfe eines einzelnen Punktes bestimmt, falls sich ein solcher in der Nähe der letzten Position befindet. Dieser Vorgang ist in Abbildung 4.10 im rechten Zweig dargestellt. Er trägt ebenfalls dazu bei, bei einem zeitweiligen Verlust eines einzelnen Punktes, Bewegungsabläufe glatter wirken zu lassen, hat aber ebenfalls zur Folge, dass die Ergebnisse verfälscht werden. Auch dieses Vorgehen ist nicht ratsam, falls die Genauigkeit der Resultate im Vordergrund der Applikation steht.

4.2.3 Reduktion der Liste der Punktpaare

In der vorliegenden Punktmenge existieren häufig mehrere Punktpaare, die aufgrund ihrer Distanz als gesuchtes Zweipunktmodell in Frage kommen. Je mehr Modelle verwendet werden, umso wahrscheinlicher ist es, dass zwei nicht zusammengehörige Punkte den gleichen Abstand haben, wie die Punkte eines tatsächlich gesuchten Modells. Weitere Kriterien sind also notwendig, um Mehrdeutigkeiten sinnvoll aufzulösen.

In Abbildung 4.9 ist beschrieben, wie die Anzahl der zu den Distanzen aus $\mathcal{D} = \{d_k \in \mathbb{R} : 1 \leq k \leq m \in \mathbb{N}\}$ passenden Punktpaare in einem frühen Abschnitt des Algorithmus reduziert wird. Die Idee ist, Listen $L_h = \{X_{ij} : i = h \vee j = h\}$ von Punktpaaren $X_{ij} = \{X_i, X_j\}$ aufzubauen, wobei in jeder Liste L_h nur solche Paare sind, von denen einer der beiden Punkte X_h ist. Jedem Punktpaar X_{ij} wird außerdem die Anzahl v_{ij} der Punktpaare, deren Abstände derselben Distanz entsprechen, zugeordnet. Sei also $X_{25} \approx X_{34} \approx d_3$, dann wäre $v_{25} = v_{34} = 2$.

Die Listen werden daraufhin solange schrittweise um Punktpaare reduziert, wie es Listen L_{h_0} gibt, bei denen Mehrdeutigkeiten existieren, das heißt, für die $l_{h_0} = |L_{h_0}| > 1$ gilt. Dabei wird immer zu einer Liste L_{h_0} eine weitere Liste L_i gesucht, die einen gemeinsamen 3D-Punkt mit L_{h_0} hat und unter diesen Listen die größte Anzahl l_i , das heißt die meisten Mehrdeutigkeiten, besitzt. Das entsprechende Punktpaare X_{h_0i} beziehungsweise X_{ih_0} wird von den Listen L_{h_0} und L_i entfernt. Gibt es mehrere Listen mit der größten Anzahl l_i , wird das Punktpaar entfernt, das den größten Wert v_{ij} besitzt. Gibt es mehrere solcher Punktpaare, wird das erste gewählt. Nach Entfernung der Punktpaare, werden die v_{ij} entsprechend der verbleibenden Punktpaare neu berechnet.

Dieses Kriterium soll sicherstellen, dass Punkte nur einer Distanz zugeordnet und daher einig, aber nicht notwendigerweise alle, Mehrdeutigkeiten aufgelöst werden. In Anhang

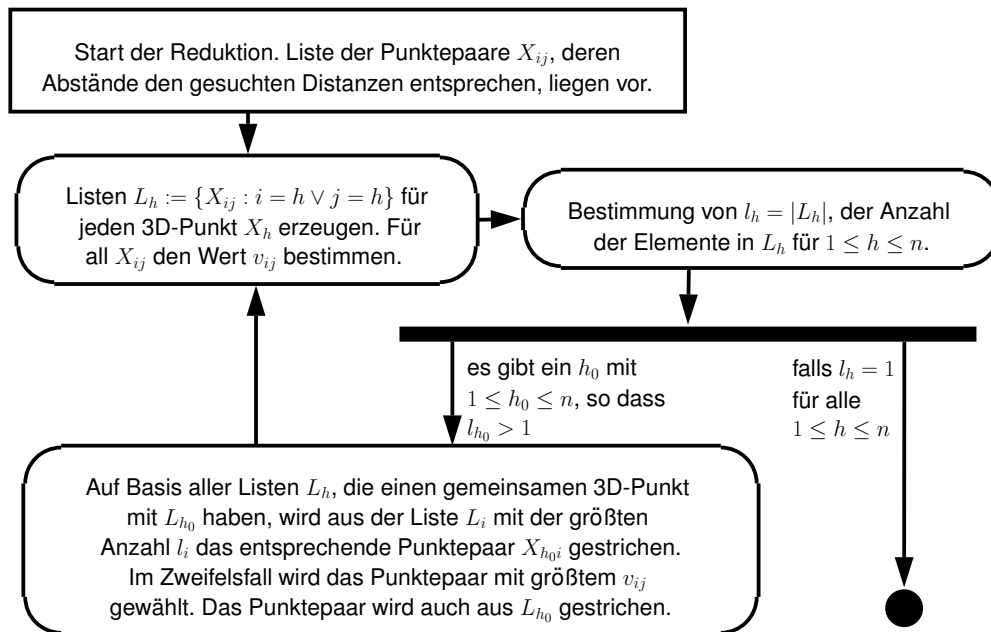


Abbildung 4.9. Reduktion der Liste der Punktepaare.

D befindet sich ein Beispiel für eine solche Reduktion.

In diesem Ansatz spielt die letzte Position P_{alt} eines Zweipunktmodells bei der Reduktion keine Rolle, obwohl dies ein starkes Kriterium für die Auflösung von Mehrdeutigkeiten sein könnte. Man kann beispielsweise unter den Punktepaaren, die zu einem bestimmten Zweipunktmodell passen, dasjenige auswählen, welches P_{alt} am nächsten ist. Dieses Kriterium würde zusätzlich auch das Springen eines Zweipunktmodells von einer bestimmten Position in der ermittelten 3D-Punktemenge zu einer weit entfernten Position in der zeitlich darauf folgenden Punktemenge unwahrscheinlicher machen.

Allerdings hat die Verwendung der letzten Position für die Reduktion einen Nachteil: Gibt es ein Punktepaar, welches nicht das Zweipunktmodell k darstellt, jedoch zufällig die entsprechende Distanz d_k hat, so würde immer wieder die Position dieses Punktepaars als Position des Zweipunktmodells ermittelt werden, sobald es einmal dem Zweipunktmodell zugeordnet wurde. Dies tritt beispielsweise auf, wenn das richtige Punktepaar des Zweipunktmodells in diesem Moment nicht ermittelt werden kann. Wenn dann das falsche Punktepaar über einen gewissen Zeitraum die Position nicht verändert, kann das Zweipunktmodell nicht erkannt werden. Um diesen Zustand zu vermeiden, wird an dieser Stelle des Algorithmus auf ein Vergleich mit P_{alt} verzichtet. Es werden nur Punktepaare aussortiert, die mit anderen Punktepaaren einen Punkt gemeinsam haben. Die letzte Position P_{alt} eines Zweipunktmodells wird jedoch erst zu einem späteren Zeitpunkt, nach Berücksichtigung des Vertrauenswertes t , wie in Abbildung 4.10 dargestellt, verwendet.

4.2.4 Mehrdeutigkeiten

Insgesamt lässt sich für die Erkennung eines Zweipunktmodells festhalten, dass aufgrund der geringen geometrischen Information über das Modell, die nur aus der Distanz

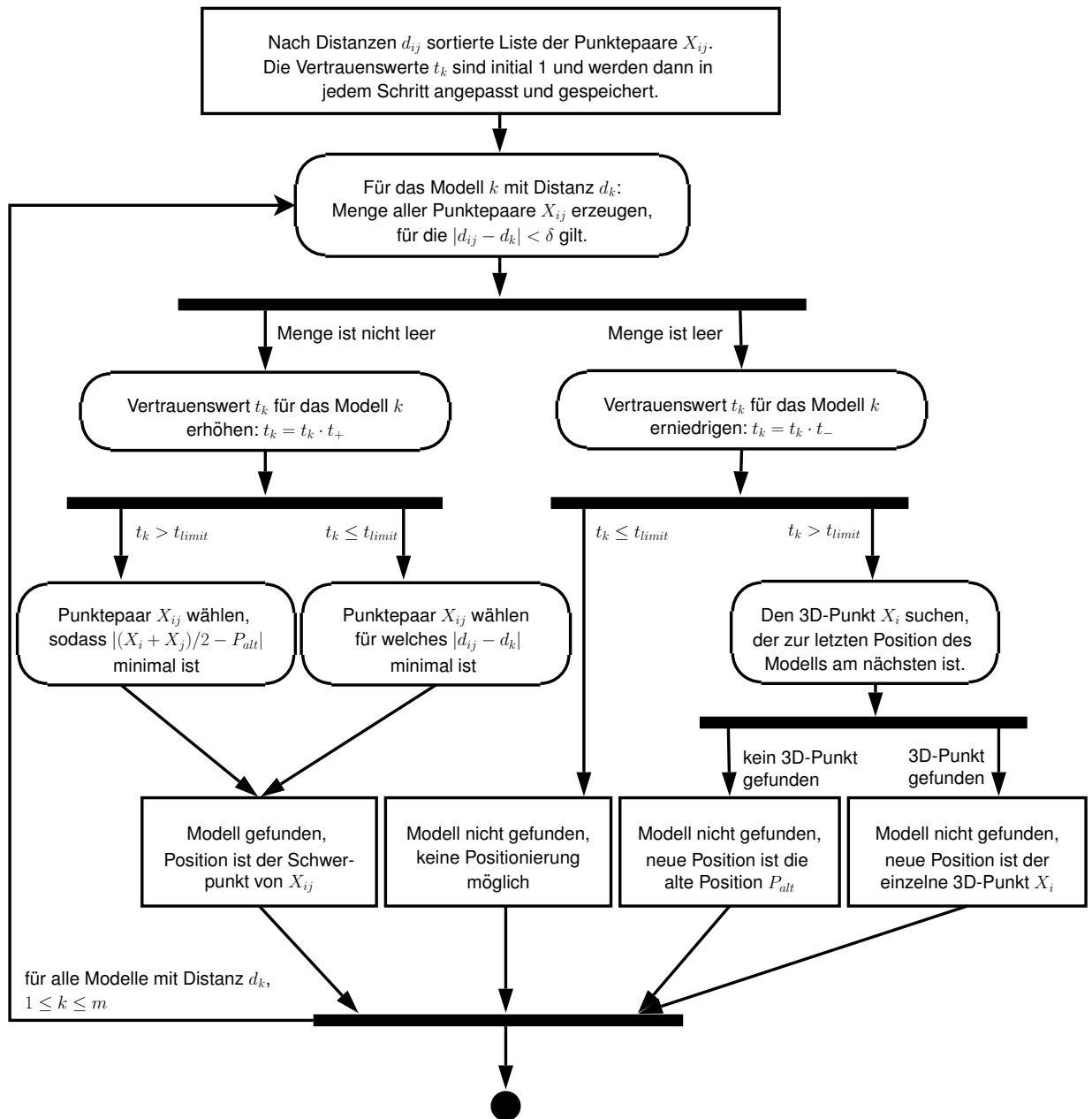


Abbildung 4.10. Erkennung der Modelle aus der sortierten Liste der Punktepaare X_{ij} .

d_k besteht, bei der Verwendung mehrerer Modelle leicht Mehrdeutigkeiten durch ungünstige Positionierung der Modelle auftreten können. Diese Mehrdeutigkeiten lassen sich mit dem beschriebenen Vorgehen einfach auflösen, soweit alle Modelle in Bewegung sind und sich somit Punktkonstellationen, die Mehrdeutigkeiten erzeugen, nur für kurze Zeiträume existieren. Bleiben die Modelle jedoch an festen Positionen, also mit Bewegungen, die geringer als die Distanztoleranz δ sind, können nur schwierig aufzulösende Konstellatio-

nen von Punkten entstehen. Ist solch ein Fall zu erwarten, empfiehlt sich die Verwendung der im folgenden Abschnitt 4.3 beschriebenen Multipunktmodelle.

4.3 Multipunktmodelle

Die hier beschriebenen Multipunktmodelle (MPM) bestehen aus mindestens drei 3D-Punkten, die relativ zueinander fest liegen. Theoretisch ist die Anzahl der Punkte in einem solchen Modell unbeschränkt, in der Praxis liegt die Anzahl der Punkte eines Modells häufig zwischen drei und sechs. Bei nicht symmetrischer Anordnung der Punkte lassen sich schon aus drei Punkten Rotation und Translation eines Modells eindeutig bestimmen, was bei einer symmetrischen Anordnung nicht der Fall ist. Die Punkte eines Modells M werden im Folgenden mit $\mathcal{X}_M = \{X_M^1, \dots, X_M^n\}, n \in \mathbb{N}$ bezeichnet.

Die Verwendung solcher Modelle für die Verfolgung von Objekten bietet sich bei Infrarottrackingsystemen häufig an. Zwei Beispiele für eine Verwendung der in dieser Arbeit beschriebenen Modelle und Algorithmen im medizinischen Anwendungsgebiet befinden sich in den Abschnitten 8.1 und 8.2. Dabei handelt es sich zum einen um die Verfolgung eines transparenten Displays und eines Brillengestells, siehe Abbildung 4.11, für ein AR-System und zum anderen um den Ausgleich der Kopfbewegung bei einer Bildgebung mittels eines Magnetresonanztomographen.



Abbildung 4.11. Brillengestell als Teil eines AR-Systems zur Bestimmung der Kopfposition des Benutzers. Vier Dioden werden als aktive Marker verwendet.

Die Mindestgröße von Multipunktmodellen ist drei, da ab dieser Anzahl auch eine Orientierung eines Modells berechnet werden kann. Sollen mehrere Objekte verfolgt werden,

kann es jedoch auch von Vorteil sein, eine Mindestanzahl von vier zu fordern, um die Gefahr der Verwechslung zwischen Modellen zu verringern oder die Verdeckungsproblematik zu mindern, siehe Abschnitte 4.3.3 und 4.3.5. Werden vier Marker als Mindestanzahl gefordert, muss diese Anzahl von Markern allerdings in jedem Frame erkannt werden, was bei Verdeckungen unter Umständen die Erkennung der einzelnen Modelle verschlechtert.

Im Abschnitt 4.3.3 werden außerdem verschiedene Vorgehensweisen für die Detektion von Modellen aufgezeigt. Die Zuverlässigkeit der Erkennung durch die im Modell gespeicherten Eigenschaften, wie Distanzen oder die vorausgegangene Position und Orientierung, spielt dabei eine große Rolle. Dennoch ist es sinnvoll, sich auf gewisse Kriterien für die Erkennung zu beschränken, um die Latenz zwischen Bildaufnahme und Ausgabe der Transformationen minimal zu halten, indem auf aufwändige Berechnungen verzichtet wird. Eine Definition für Multipunktmodelle wird nach der Beschreibung der Detektionsalgorithmen in Abschnitt 4.3.3 getroffen.

Die Positionierung der 3D-Punkte eines Modells innerhalb des lokalen Modellkoordinatensystems wird in den Abschnitten dieses Kapitels als gegeben vorausgesetzt. Die Ermittlung der Koordinaten dieser Punkte ist in einfachen Fällen, das heißt bei Modellen mit wenigen Markern, unproblematisch, wirft jedoch bei komplexeren Konstellationen einige Fragen auf. Mit diesen Fragestellungen beschäftigt sich Kapitel 5.

In der Literatur zur Detektion mit Infrarottrackingsystemen, beziehungsweise bei existierenden Trackingsystemen, lässt sich eine große Vielfalt an Detektionsalgorithmen erkennen, wobei die Verfahren häufig nur angedeutet werden. In [Dorf Müller 99a], [Ribo 01] und [Dorf Müller-Ulhaas 03] wird beschrieben, wie mit drei nicht symmetrisch platzierten Punkten ein Objekt durch Distanzmessungen detektiert werden kann, beziehungsweise Objekte mit unterschiedlichen Konstellationen unterschieden werden können.

Eine Beschränkung auf drei Punkte ist für einfache Anwendungen vorteilhaft, für zahlreiche Anwendungen jedoch nicht ausreichend. Kommerzielle Systeme wie die von A.R.T. [ART 06b], Northern Digital inc. [NDI 06] und Vicon [Vicon 06] arbeiten mit Modellen, die aus mehr als drei Punkten bestehen können oder sogar müssen. Die Algorithmen, die dort verwendet werden, sind allerdings nicht offengelegt. Bei der DTrack Software für die A.R.T Trackingsysteme werden für die Modelle mindestens vier Marker gefordert. Das Polaris System von NDI kann mit sechs single-face oder 20 multi-face Markern arbeiten.

4.3.1 Berechnung der Modelltransformation aus drei beliebigen Markern

Als Vorbereitung wird eine Definition des Begriffs Korrespondenz zwischen einem Punkt in Weltkoordinaten und einem Punkt in Modellkoordinaten getroffen, der im Folgenden verwendet wird:

Definition 5 Ein Punkt X_W in Weltkoordinaten und ein Punkt X_M in Modellkoordinaten korrespondieren genau dann, wenn $\|X_W - R_M^W \cdot X_M + T_M^W\| < \delta'$, wobei R_M^W die Rotation von Modellkoordinaten nach Weltkoordinaten, T_M^W die Translation von Weltkoordinaten in Modellkoordinaten und $\delta' \in \mathbb{R}$ eine vorgegebene Toleranzgrenze ist.

Die Toleranzgrenze δ' kann dabei als $\delta' = \delta$ gewählt, oder abhängig von δ ausgedrückt werden, beispielsweise $\delta' = 2\delta$. Dabei ist δ die Distanztoleranz aus Definition 2 in Ab-

schnitt 3.5. Die beiden Konstanten δ und δ' drücken beide eine Toleranz für die Messung von Distanzen aus, werden jedoch nicht grundsätzlich gleichgesetzt, da es sich bei δ um ein Maß für die reine Positionsgenauigkeit von Punkten im 3D handelt und bei δ' die Rotationsgenauigkeit einfließt. Um sich bei der Suche nach Korrespondenzen zwischen rekonstruierten Punkten und den in den Modellen gespeicherten Punkten nicht zu stark einzuschränken, sollte im Allgemeinen $\delta' \geq \delta$ gelten.

Sind also drei beliebige Marker eines n -Punktmodells bekannt, das heißt, es existieren drei Korrespondenzen (X_W, X_M) zwischen getrackten Markern X_W in Weltkoordinaten und Punkten X_M des Modells, lassen sich Rotation und Translation des Modells auf einfache Weise bestimmen.

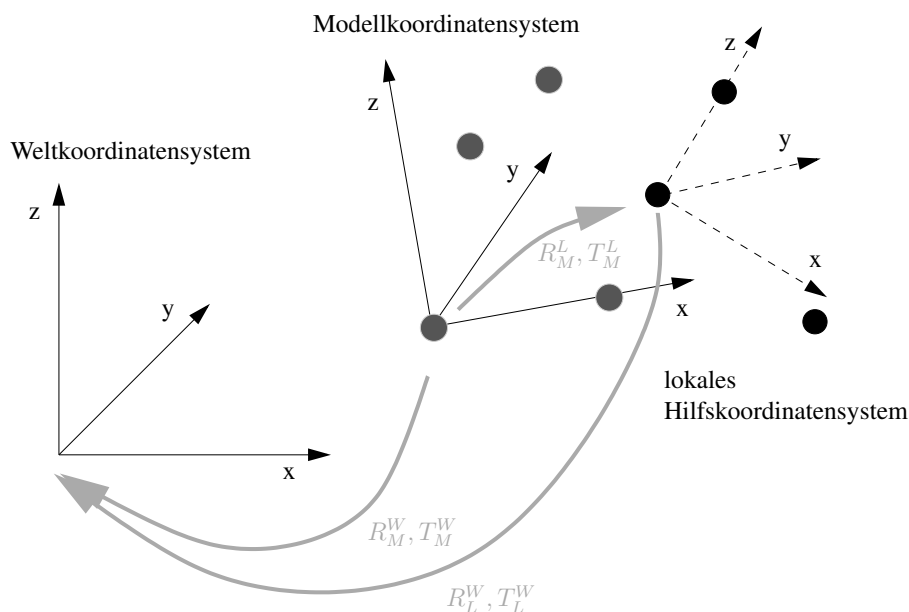


Abbildung 4.12. Transformationen für die Bestimmung der Abbildung von Welt- auf Modellkoordinatensystem. Dem Modell sind hier sieben Punkte zugeordnet, wobei nur die drei schwarz markierten Punkte die tatsächlich getrackten Marker darstellen.

Seien $((X_W^1, X_M^1), (X_W^2, X_M^2), (X_W^3, X_M^3))$ die drei Korrespondenzen, aus denen die Rotation R_M^W von Modellkoordinaten nach Weltkoordinaten („Rotation von M nach W “) und die Translation T_M^W , der Vektor in Weltkoordinaten, der auf den Ursprung des Modellkoordinatensystems zeigt („Translation nach M in W “), berechnet werden sollen. Dann gilt

$$X_W = R_M^W \cdot X_M + T_M^W. \quad (4.2)$$

Das Vorgehen besteht nun daraus, ein drittes, wohldefiniertes (Hilfs-) Koordinatensystem aus den drei rekonstruierten Punkten sowohl in Welt- als auch in Modellkoordinaten aufzuspannen und daraus die gesuchte Transformation zu berechnen, siehe Abbildung 4.12.

Für das neue Koordinatensystem L in Modellkoordinaten wird zunächst die x -Achse durch

$$\mathbf{x} = \frac{X_M^2 - X_M^1}{\|X_M^2 - X_M^1\|} \quad (4.3)$$

festgelegt. Auf analoge Weise wird eine temporäre y -Achse

$$\mathbf{y}_{\text{temp}} = \frac{X_M^3 - X_M^1}{\|X_M^3 - X_M^1\|} \quad (4.4)$$

definiert. Sind die beiden Vektoren \mathbf{x} und \mathbf{y}_{temp} linear unabhängig, wird die z -Achse orthogonal zu den beiden anderen Achsen mit Hilfe des Kreuzprodukts durch

$$\mathbf{z} = \mathbf{x} \times \mathbf{y}_{\text{temp}} \quad (4.5)$$

berechnet. Sind \mathbf{x} und \mathbf{y}_{temp} linear abhängig, so sind die drei Punkte kollinear und es muss eine andere Konstellation von Markern gefunden werden.

Da \mathbf{y}_{temp} zwar orthogonal zu \mathbf{z} ist, nicht notwendigerweise aber zu \mathbf{x} , wird noch

$$\mathbf{y} = \mathbf{z} \times \mathbf{x} \quad (4.6)$$

gesetzt. Aus diesen drei orthogonalen Vektoren ergeben sich sofort

$$R_L^M := \begin{pmatrix} | & | & | \\ \mathbf{x} & \mathbf{y} & \mathbf{z} \\ | & | & | \end{pmatrix} \quad \text{und} \quad T_L^M := X_M^1 \quad (4.7)$$

mit

$$X_M = R_L^M \cdot X_L + T_L^M. \quad (4.8)$$

Analog hierzu werden aus X_W^1 , X_W^2 und X_W^3 die Rotation R_L^W und die Translation T_L^W mit

$$X_W = R_L^W \cdot X_L + T_L^W \quad (4.9)$$

definiert. Setzt man die umgeformte Gleichung (4.8)

$$X_L = \underbrace{(R_L^M)^{-1}}_{R_M^L} (X_M - T_L^M) \quad (4.10)$$

in Gleichung (4.9) ein, erhält man schließlich

$$X_W = \underbrace{R_L^W \cdot R_M^L}_{R_M^W} \cdot X_M + \underbrace{T_L^W - R_L^W \cdot R_M^L \cdot T_L^M}_{T_M^W}. \quad (4.11)$$

Die auf diese Weise bestimmte Rotation R_M^W und Translation T_M^W zwischen Welt- und Modellkoordinatensystem wird nicht nur als Resultat für das entsprechende Modell ausgegeben, sondern kann auch wie im folgenden Abschnitt 4.3.3 beschrieben, für weitere Berechnungen, wie das Auffinden noch nicht erkannter Modellpunkte, genutzt werden. In beiden Fällen gilt es zu berücksichtigen, dass die berechnete Transformation fehlerbehaftet ist und insbesondere Fehler in der Rotation große Abweichungen bei transformierten Punkten erzeugen können.

Rotationsfehler

Zur Verdeutlichung des Einflusses des Rotationsfehlers geht man zunächst davon aus, dass die berechnete Rotation R_M^W einen Fehler von $\frac{\vec{\varepsilon}}{|\vec{x}|}$ in der ersten Spalte, also der x-Achse des lokalen Koordinatensystems, aufweist. Dabei ist $\vec{\varepsilon}$ ein Vektor mit $|\vec{\varepsilon}| < \delta$ und \vec{x} ist der Vektor zwischen zwei rekonstruierten Punkten, die die x-Achse erzeugen. Die Konstante δ steht für die in Definition 2 auf Seite 42 eingeführte Distanztoleranz. Analoges gilt für die y- und die z-Achse.

Diese Annahme für den Fehler in einer Spalte der Rotationsmatrix kann wie folgt hergeleitet werden: In Gleichung (4.3) wird beschrieben, wie die x-Achse berechnet wird. Nach Definition des Distanzfehlers δ ist dabei die Länge des Fehlervektors $\vec{\varepsilon}$ im Zähler $|\vec{\varepsilon}| < \delta$. Da die x-Achse normiert wird, ergibt sich ein Fehler von $\frac{\vec{\varepsilon}}{|\vec{x}|}$. Diese Abschätzung wird für die anderen Achsen übernommen, auch wenn diese über das Kreuzprodukt berechnet werden. Insgesamt ist das Wesentliche bei dieser Beschreibung des Fehlers nur eine Abhängigkeit des Fehlers von den Distanzen zwischen den drei Punkten.

Für jeden Punkt $P = (p_x, p_y, p_z)^T$, der mit der fehlerbehafteten Rotation transformiert wird, beträgt der Fehler dann

$$\varepsilon_{rot} \approx \left(\frac{p_x}{|\vec{x}|}, \frac{p_y}{|\vec{y}|}, \frac{p_z}{|\vec{z}|} \right) \cdot \vec{\varepsilon}. \quad (4.12)$$

In Abhängigkeit der Komponenten von P und der Länge der Vektoren \vec{x} , \vec{y} und \vec{z} , kann der Fehler größer als δ sein, insbesondere dann, wenn der Abstand von P zum Ursprung des lokalen Koordinatensystems zunimmt.

Dies bedeutet auch, je kleiner der Abstand zwischen den Punkten im Modell ist, die das Koordinatensystem definieren, desto größer ist der Fehler.

4.3.2 Berechnung der Modelltransformation aus mehreren beliebigen Markern

Sind mehr als drei Marker eines n -Punktmodells bekannt, das heißt, es existieren mehr als drei Korrespondenzen (X_W^i, X_M^i) zwischen getrackten Punkten X_W^i in Weltkoordinaten und Punkten X_M^i des Modells, lassen sich Rotation und Translation des Modells nicht mehr wie in Abschnitt 4.3.1 beschrieben konstruieren, wenn dabei alle Korrespondenzen berücksichtigt werden sollen. Statt dessen bieten sich zwei Möglichkeiten an:

- Berechnung von Rotation und Translation aus allen dreielementigen Untermengen der vorliegenden Korrespondenzen. Anschließend Mittelung der Translationen und Rotationen. Die Mittelung von Rotationen ist allerdings problematisch, siehe Abschnitt 7.1.1.
- Berechnung von Rotation und Translation als Lösung eines Minimierungsproblems, welches sich mit linearen Methoden lösen lässt.

Der erste Ansatz ist wenig elegant und mit zunehmender Anzahl von Korrespondenzen rechenintensiv, da bei $m \geq 3$ Korrespondenzen $\binom{m}{3}$ dreielementige Untermengen existieren. Der zweite Ansatz ist die Lösung des Minimierungsproblems

$$\min_{R,T} \sum_{i=1}^m \|X_W^i - R \cdot X_M^i - T\|^2, \quad (4.13)$$

mit m der Anzahl der bekannten Korrespondenzen. Dieses ist unter anderem in [Trucco 99] beschrieben. Zunächst werden die Punkte X_M^i und X_W^i um ihre Schwerpunkte $\bar{X}_M = 1/m \sum_{i=1}^m X_M^i$ und $\bar{X}_W = 1/m \sum_{i=1}^m X_W^i$ verschoben:

$$\hat{X}_M^i = X_M^i - \bar{X}_M \quad \text{und} \quad \hat{X}_W^i = X_W^i - \bar{X}_W. \quad (4.14)$$

Es sei darauf hingewiesen, dass es nicht ausreicht, die Schwerpunkte der Punkte zu berechnen, um die Translation zu bestimmen. Die korrekte Berechnung erfolgt weiter unten. Das Minimierungsproblem (4.13) ist äquivalent zum Problem

$$\min_R \sum_{i=1}^m \|\hat{X}_W^i - R \cdot \hat{X}_M^i\|^2, \quad (4.15)$$

welches minimiert wird, wenn die $Spur(RK)$ (die Summe der Hauptdiagonalelemente) maximiert wird [Kanatani 93], mit

$$K = \sum_{i=1}^m \hat{X}_W^i \cdot (\hat{X}_M^i)^T. \quad (4.16)$$

Durch eine Singulärwertzerlegung von $K = VDU^T$ erhält man die minimierende Rotationsmatrix $R = VU^T$, welche die $Spur$ maximiert. Die optimale Translation T lässt sich dann durch $T = \bar{X}_W - R \cdot \bar{X}_M$ berechnen. In Abschnitt 6.3 wird dieser Algorithmus in einem anderen Zusammenhang näher beschrieben.

In der Praxis zeigt sich, dass das zweite Verfahren bei einer geringen Anzahl von Korrespondenzen nicht notwendigerweise bessere Ergebnisse liefert. Bei einer größeren Anzahl von Korrespondenzen ist es auf jeden Fall zu bevorzugen.

4.3.3 Modellerkennungsstrategien

Für die Erkennung von Multipunktmodellen gibt es keine allgemeine optimale Strategie, mit der man vorgehen kann. Vielmehr hängt die Wahl der richtigen Strategie davon ab, auf welche Eigenschaften bei einer Detektion besonderer Wert gelegt wird. Bei Modellen $\mathcal{X}_M = \{X_M^1, \dots, X_M^n\}$, $n \in \mathbb{N}$ mit $n \geq 4$ stellt sich bereits die Frage, ob drei der vier Marker für eine Erkennung ausreichen können oder ob vier Marker notwendig sind. Diese Fragestellung weist auf verschiedene Anforderungen an einen Trackingalgorithmus hin, die in Abbildung 4.13 dargestellt sind. Eine hohe Zuverlässigkeit bei der Erkennung eines Modells steht der Stabilität der Verfolgung eines Modells und einem geringen Rechenaufwand gegenüber. Eine hohe Zuverlässigkeit bedeutet hier, dass

- es sich bei einem erkannten Modell wirklich um das gesuchte handelt und
- dass die Transformation des Modells präzise berechnet wird, zum Beispiel dadurch, dass möglichst viele erkannte 3D-Punkte des Modells verwendet werden.

- Außerdem sollte die Latenz bei der Erkennung möglichst gering sein, so dass eine berechnete Transformation auch zum aktuellen Zeitpunkt passt.

Die Stabilität der Verfolgung eines Modells bedeutet hier,

- dass in jedem Frame eine Transformation für ein Modell angegeben wird, auch dann, wenn es unter Umständen in diesem Frame unzuverlässig oder gar nicht erkannt wurde und somit die Transformation zum Beispiel nur aus den vorherigen Transformationen prädiiziert werden kann.
- Außerdem steht Stabilität hier dafür, dass sich durch die berechneten Transformationen gleichmäßige Bewegungen eines Objekts ergeben, zum Beispiel durch starke Filterung.

Eine hohe Zuverlässigkeit ist beispielsweise für Tracking im medizinischen Bereich unverzichtbar, wohingegen bei VR-Anwendungen im Unterhaltungsbereich eine möglichst geringe Verarbeitungszeit und damit Latenz, sowie eine hohe Erkennungsrate als wichtiger angesehen werden.

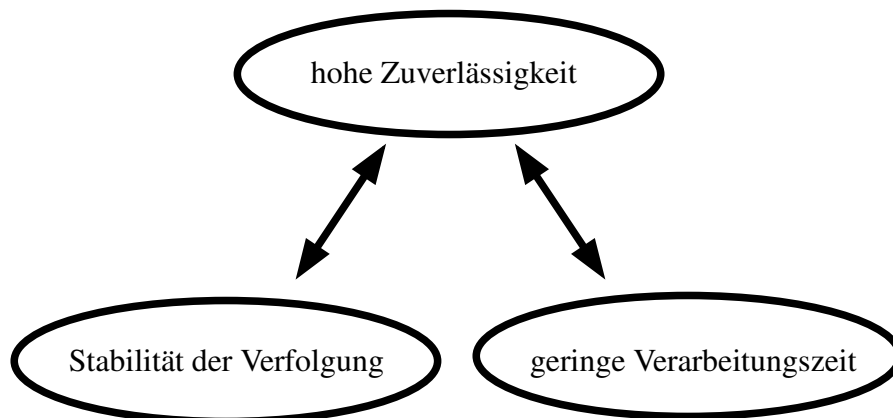


Abbildung 4.13. Verschiedene Anforderungen an Trackingstrategien für Multipunktmodelle, die nicht alle gleichermaßen erfüllt werden.

Es gibt weitere Kriterien als die Mindestanzahl der erkannten Marker eines Modells, die darauf Einfluss haben, inwiefern die dargestellten Anforderungen an eine Modellerkennungsstrategie erfüllt werden. Wie in den vorausgegangenen Kapiteln, können beispielsweise vorhergehende Positionen und Orientierungen bei der Erkennung berücksichtigt werden, um Verwechslungen zu vermeiden.

Der im Folgenden präsentierte Trackingalgorithmus [Schwald 05a] für die Erkennung und Verfolgung eines Modells steht an keiner bestimmten Stelle zwischen den aufgezählten Anforderungen, sondern lässt sich durch Parameter so konfigurieren, dass er den Erfordernissen einer Anwendung angepasst werden kann. Eine einfachere Form des Algorithmus findet sich in [Figueiredo 02].

Vorverarbeitung

Vor der Detektion jedes Modells werden aus dessen Modellpunkten X_M^1, \dots, X_M^n mit $n \geq 3$ die Distanzen d_M^i zwischen den Modellpunkten berechnet. Da immer wieder auf diese Distanzen zurückgegriffen wird, werden sie einmalig beim Anlegen des Modells berechnet, gespeichert und sortiert, sodass sie später jederzeit zur Verfügung stehen und nicht erneut berechnet werden müssen.

Ausgangspunkt für die Detektion der Multipunktmodelle ist, wie am Anfang dieses Kapitels beschrieben, eine Menge $\{X_W^1, \dots, X_W^m\}$, $m \in \mathbb{N}$ von zu einem Zeitpunkt rekonstruierten 3D-Punkten, die im Weltkoordinatensystem vorliegen. Der erste Schritt besteht daraus, das gesuchte Modell anhand von drei Korrespondenzen zwischen den Modellpunkten X_M^i und den rekonstruierten Punkten X_W^j zu identifizieren. Dazu werden die Distanzen zwischen den getrackten Punkten berechnet und mit den gespeicherten Distanzen des Modells verglichen. Für die Gleichheit zweier Distanzen gilt die in Abschnitt 3.5 eingeführte Distanztoleranz δ . Der Begriff getrackter Punkt wird im Folgenden synonym zum Begriff rekonstruierter Punkt verwendet. Wichtig ist nur die Unterscheidung dieser Punkte von den Modellpunkten.

Detektion ohne Vorwissen

Soweit keine Information über das Modell von einem früheren Verarbeitungsschritt vorliegt,

- beispielsweise beim ersten Frame, der verarbeitet wird,
- oder weil überhaupt keine weitere Information verwendet werden soll,

lassen sich Korrespondenzen für die Modellpunkte finden, wie in Abbildung 4.14 beispielhaft skizziert und im Folgenden genauer beschrieben:

1. Für die Distanz $d_W^1 = |X_W^1 - X_W^2|$ zwischen den ersten beiden getrackten Punkten wird eine entsprechende, im Modell gespeicherte, Distanz d_M^i gesucht, das heißt, $|d_W^1 - d_M^i| < \delta$ soll gelten.
2. Falls es keine solche Distanz d_M^i gibt, wird systematisch ein neues Paar von getrackten Punkten gewählt, bis eine korrespondierende Distanz der im Modell gespeicherten Distanzen gefunden werden kann, oder keine Punktepaare mehr vorliegen. Im letzteren Fall konnte das Modell nicht gefunden werden. Ohne Beschränkung der Allgemeinheit (o.B.d.A.) sei $d_M^1 = |X_M^1 - X_M^2|$ die gesuchte Distanz.
3. Dann wird die Distanz $d_W^2 = |X_W^1 - X_W^3|$ zwischen dem ersten und einem dritten getrackten Punkt bestimmt oder zwischen dem zweiten und einem dritten Punkt.
4. Die Distanz d_W^2 wird nun unter allen im Modell gespeicherten Distanzen d_M^i gesucht, die aus einem der bereits ausgewählten Punkte X_M^1 oder X_M^2 und einem weiteren Punkt erzeugt werden. Falls solch eine Distanz nicht gefunden wird, muss ein neuer dritter Punkt (Schritt 3) oder ein neues initiales Punktepaar (Schritt 1) gesucht werden.

5. O.B.d.A. sei d_M^2 eine zu d_W^2 passende Distanz. Damit lassen sich drei getrackte Punkte X_W^1, X_W^2, X_W^3 drei Modellpunkten X_M^1, X_M^2, X_M^3 eindeutig zuordnen. Zur Kontrolle müssen noch die dritten, bislang nicht berücksichtigte Distanzen $d_W^3 = |X_W^2 - X_W^3|$ und $d_M^3 = |X_M^2 - X_M^3|$ auf Ähnlichkeit überprüft werden. Ist die Überprüfung erfolgreich, stehen die Korrespondenzen zwischen den drei getrackten Punkten und den Modellpunkten fest. War die Überprüfung nicht erfolgreich, wird bei Schritt 4 wieder angesetzt, bis das Modell identifiziert ist oder alle Distanzen überprüft wurden.

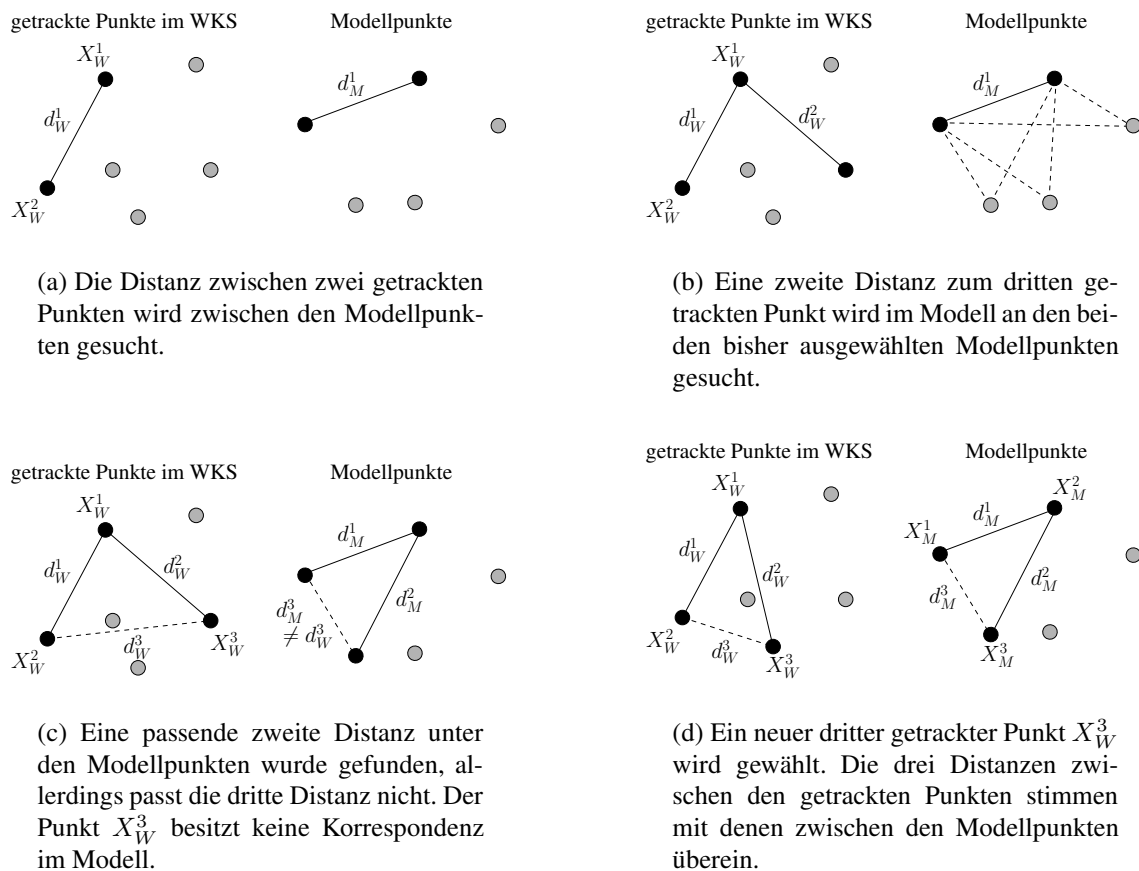


Abbildung 4.14. Suche nach Korrespondenzen zwischen sechs getrackten Punkten im Weltkoordinatensystem (WKS) und fünf Modellpunkten anhand von Distanzen. Die jeweils aktuell relevanten Punkte sind schwarz eingefärbt, die anderen grau.

Nach einem erfolgreichen Durchlauf des Detektionsalgorithmus können aus den Korrespondenzen $((X_W^1, X_M^1), (X_W^2, X_M^2), (X_W^3, X_M^3))$ entsprechend Abschnitt 4.3.1 Rotation und Translation des Modells bestimmt werden. Die getrackten Punkte X_W^1, X_W^2, X_W^3 erhalten einen, in Abschnitt 3.3 eingeführten, Klassifikationsindex des Modells, sodass diese Punkte bei der weiteren Verarbeitung als diesem Modell zugehörig markiert sind. Der Klassifikationsindex ist zum Beispiel einfach die Nummer des Modells innerhalb einer

Menge von Modellen. Somit kann an dieser Stelle die Detektion eines Modells abgebrochen werden, da die gewünschten Informationen vorliegen. Ein Abbruch des Algorithmus zu diesem Zeitpunkt stellt die minimale Form des Detektionsalgorithmus dar. Allerdings drängen sich weitere Überlegungen zur Detektion bezüglich verschiedener Aspekte, wie Optimierung und Robustheit, auf:

- Vollständigkeit der Zuordnung: Weitere zum Modell gehörende Punkte sollen ebenfalls den Klassifikationsindex erhalten.
- Genauigkeit: Sind weitere Punkte bekannt, lassen sich Rotation und Translation aus einer größeren Menge von Korrespondenzen berechnen, siehe Abschnitt 4.3.2.
- Optimierung: Weitere Punkte können einfach mit Hilfe der berechneten Transformation des Modells bestimmt werden, oder aufwändiger über Distanzen.
- Optimierung: Wenn Rotation und Translation im letzten Verarbeitungsschritt berechnet wurden, lässt sich dann diese Information für ein schnelleres und unter Umständen sichereres Erkennen des Modells nutzen?
- Robustheit: Sollen mehr als drei Punkte für die Identifikation eines Modells gefordert werden, und wenn ja, zu welchem Zweck?
- Robustheit: Es könnten weitere Konstellationen von getrackten Punkten existieren, die zum Modell passen, aber nicht als weitere Punkte erkannt werden. Das Modell wird in diesem Fall also zweifach erkannt. Es muss entschieden werden, welches das richtige Modell ist.
- Stabilität: Wann lohnt sich eine Detektion auf Basis von nur zwei Markern?

Für die Erkennung weiterer Korrespondenzen zwischen Modellpunkten X_M^i und getrackten Punkten X_W^j bietet es sich zunächst an, die für das Modell berechnete Rotation R_M^W und Translation T_M^W zu verwenden. Die Modellpunkte X_M^i , die noch keine korrespondierenden getrackten Punkte in Weltkoordinaten haben, werden dafür einfach ins Weltkoordinatensystem durch

$$X_W^{M^i} = R_M^W \cdot X_M^i + T_M^W \quad (4.17)$$

transformiert. In der Umgebung des Punktes $X_W^{M^i}$ wird dann nach einem Punkt $X_W^j \in \{X_W^1, \dots, X_W^m\}$ gesucht, der noch zu keinem Modell gehört, das heißt, dem noch kein Klassifikationsindex zugewiesen wurde. Bei der Definition der Größe der Umgebung ist es im Allgemeinen zu restriktiv, die Distanztoleranz δ als Radius zu wählen und sollte daher etwas größer gewählt werden. Die Begründung ist analog zu den Überlegungen zum Rotationsfehler in Gleichung (4.12) in Abschnitt 4.3.1. Wird nur ein Punkt in der Umgebung gefunden, lässt sich dieser einfach anhand der Distanzen zwischen den nun feststehenden Punkten mit der gegebenen Distanztoleranz δ validieren. Ist kein Punkt in der Umgebung, muss davon ausgegangen werden, dass es keine Korrespondenz zu dem entsprechenden Modellpunkt gibt. Gibt es mehrere Punkte in der Umgebung, ist dies häufig

auf Störungen zurückzuführen, da ansonsten die Punkte sehr dicht beieinander wären, was in der Praxis seltener vorkommt. Unabhängig davon, warum mehrere Punkte in der Umgebung sind, kann mit Hilfe der Distanzen zwischen den restlichen Punkten ein am besten passender Punkt gefunden werden. Bei der Verwendung eines Infrarottrackingsystems mit passiven Markern sind mehrere Punkte, die tatsächlich aus Markern rekonstruiert wurden, in einer solchen Umgebung deshalb schon unwahrscheinlich, weil die zu wählende Distanztoleranz (wenige Millimeter) meist deutlich kleiner ist, als die physikalische Größe von Markern (häufig mehr als zehn Millimeter).

Die Suche nach weiteren Korrespondenzen ausschließlich durch Distanzmessungen ist zuverlässiger, aber auch mit mehr Rechenaufwand verbunden. Im Wesentlichen wird dabei wie bei der Suche nach dem dritten der drei initialen Punkte vorgegangen, was bedeutet, dass jeder getrackte Punkt, der noch nicht einem Modell zugeordnet ist, geprüft werden muss. Neu gefundene Punkte in Weltkoordinaten, die mit einem Modellpunkt korrespondieren, erhalten ebenfalls den Klassifikationsindex des Modells. Der in Kapitel 5 beschriebene Algorithmus zum sequentiellen Anlernen von Multipunktmodellen ist auf diese Zuweisung des Klassifikationsindex an alle getrackten Punkte, die mit einem Modellpunkt korrespondierenden, angewiesen.

Die Ansätze zur Suche nach mehr als drei Korrespondenzen zwischen getrackten Punkten und Modellpunkten eröffnen auch die Möglichkeit, ein Modell überhaupt erst dann als erkannt gelten zu lassen, wenn mindestens vier Korrespondenzen vorliegen. Der Vorteil ist die deutlich höhere Sicherheit, das Modell wirklich gefunden zu haben. Auf der anderen Seite wird das Modell dann eben auch nicht erkannt, wenn beispielsweise aufgrund von Verdeckungen nur drei Korrespondenzen vorhanden sind, obwohl diese für die Bestimmung von Rotation und Translation ausreichen würden. Die Entscheidung, welche Mindestanzahl von Punkten für eine Validierung notwendig sein soll, ist wie eingangs erwähnt, abhängig von den gewünschten Eigenschaften des Trackingalgorithmus und stellt eine Konfigurationsmöglichkeit des Algorithmus dar.

Die bislang beschriebene Form der Detektion eines Modells basiert auf der Annahme, dass keine Information über das Modell zum letzten Zeitpunkt vorliegt. In zahlreichen Situationen sind jedoch die letzte Rotation und Translation eines Modells bekannt und auch die Korrespondenzen, die für deren Berechnung notwendig waren. Dieses Vorwissen ermöglicht ein weiteres, effizienteres Vorgehen.

Detektion mit Vorwissen

Der Ausgangspunkt für die Detektion besteht in dieser Situation neben der Menge $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$, $m \in \mathbb{N}$, der zu einem Zeitpunkt rekonstruierten 3D-Punkte, aus der Menge

$$\{(X_W^{1,last}, X_M^{1,last}), (X_W^{2,last}, X_M^{2,last}), \dots, (X_W^{p,last}, X_M^{p,last})\}, \quad (4.18)$$

der $p \in \mathbb{N}$ zum letzten Zeitpunkt gefundenen Korrespondenzen zwischen Modellpunkten und getrackten Punkten eines Modells. Daher kann in einer vordefinierten Umgebung eines Punktes $X_W^{j,last}$ nach aktuell getrackten Punkten gesucht werden, um daraus neue Korrespondenzen zwischen getrackten Punkten und Modellpunkten zu bilden. Dazu wird

vorab ein $\varepsilon \in \mathbb{R}$ festgelegt, welches die Umgebung

$$U_\varepsilon^{i,last} := \{X \in \mathbb{R}^3 : |X - X_W^{j,last}| < \varepsilon\} \quad (4.19)$$

eines Punktes $X_W^{j,last}$ definiert. Wählt man dabei ε sehr gering, besteht die Gefahr, dass insbesondere bei schnellen Bewegungen keine Punkte in den Umgebungen gefunden werden. Ein zu hoher Wert für ε birgt den Nachteil, dass zu viele unerwünschte Punkte in den Umgebungen gefunden werden können. Ein ausgewogener Wert, der sich aus der Update-rate des Trackingsystems und den zu erwartenden schnellsten Bewegungen eines Modells ableiten lässt, sollte hier gewählt werden. Sind mehrere der letzten Posen eines Modells gespeichert, sodass sich Geschwindigkeiten und Richtungen einzelner Modellpunkte berechnen lassen, können diese genutzt werden, um statt in der Umgebung der letzten Position, in der Umgebung einer prädierten Position nach dem neuen Punkt X_W^i zu suchen. In der folgenden Beschreibung des Algorithmus werden die Umgebungen nicht prädiert, um die Darstellung zu vereinfachen:

1. Beginnend bei $X_W^{1,last}$ wird in dessen Umgebung $U^{1,last}$ nach einem aktuell getrackten Punkt $X_W^{j_1} \in \mathcal{X}_W$ gesucht. Existiert ein solcher Punkt, wird vorläufig eine neue Korrespondenz $(X_W^{j_1}, X_M^{1,last})$ gespeichert, wobei $X_M^{1,last}$ der korrespondierende Punkt zu $X_W^{1,last}$ vom letzten Frame ist. Ansonsten wird in der Umgebung der nächsten Punkte $X_W^{j,last}$ nach einem entsprechenden Punkt gesucht.
2. Nach demselben Prinzip wie in Schritt 1 wird nach einer zweiten Korrespondenz $(X_W^{j_2}, X_M^{2,last})$ in den Umgebungen der verbleibenden Punkte gesucht. Existiert eine solche Korrespondenz, wird überprüft, ob

$$\left| \underbrace{|X_W^{j_1} - X_W^{j_2}|}_{d_W^1} - \underbrace{|X_M^{1,last} - X_M^{2,last}|}_{d_M^{1,last}} \right| < \delta \quad (4.20)$$

ist, wobei die Differenz $d_M^{1,last}$ zwischen den Modellpunkten nicht neu berechnet werden muss, sondern bereits im Modell gespeichert ist. Falls diese Bedingung nicht erfüllt ist, wird die Korrespondenz wieder verworfen und eine neue zweite Korrespondenz gesucht. Falls keine zweite Korrespondenz gefunden werden kann, wird eine neue erste Korrespondenz gesucht (1. Schritt).

3. Eine dritte Korrespondenz $(X_W^{j_3}, X_M^{3,last})$ wird wieder wie in Schritt 1 beschrieben bestimmt. Existiert eine solche Korrespondenz, wird überprüft, ob

$$\left| \underbrace{|X_W^{j_1} - X_W^{j_3}|}_{d_W^2} - \underbrace{|X_M^{1,last} - X_M^{3,last}|}_{d_M^{2,last}} \right| < \delta \quad \text{und} \quad (4.21)$$

$$\left| \underbrace{|X_W^{j_2} - X_W^{j_3}|}_{d_W^3} - \underbrace{|X_M^{2,last} - X_M^{3,last}|}_{d_M^{3,last}} \right| < \delta \quad (4.22)$$

ist. Trifft dies zu, ist die Detektion erfolgreich beendet, ansonsten muss eine neue dritte Korrespondenz gesucht werden, beziehungsweise bei Schritt 2 oder Schritt 1 neu angefangen werden.

Wie beim Detektionsalgorithmus ohne Vorwissen im vorhergehenden Abschnitt, können, nach erfolgreichem Durchlaufen des Detektionsalgorithmus, aus den Korrespondenzen entsprechend Abschnitt 4.3.1 Rotation und Translation des Modells bestimmt werden. Dabei erhalten die getrackten Punkte $X_W^{j_1}$, $X_W^{j_2}$, $X_W^{j_3}$ aus den ermittelten Korrespondenzen den Klassifikationsindex des Modells und weitere Korrespondenzen können detektiert werden.

Der Algorithmus kann zusätzlich noch effizienter gestaltet werden, wenn man die Indizes der getrackten Punkte berücksichtigt. Im Allgemeinen bleiben diese von Frame zu Frame gleich, da die Kamerabilder immer nach dem gleichen Schema abgearbeitet werden und damit Marker häufig in der gleichen Reihenfolge erkannt werden. Insofern ist es effizient, wenn man den getrackten Punkt mit dem Index des getrackten Punktes zum letzten Zeitpunkt als erstes auf seine Positionierung innerhalb der entsprechenden Umgebung überprüft.

Kombinierter Detektionsalgorithmus

Beide in diesem Abschnitt beschriebenen Detektionsalgorithmen, der ohne Vorwissen und der mit Vorwissen, lassen sich sinnvoll zu einem allgemeineren Trackingalgorithmus zusammenfassen, der in Abbildung 4.15 skizziert ist:

1. Als erstes wird überprüft, ob Korrespondenzen zwischen Modellpunkten und getrackten Punkten von der letzten Detektion vorliegen. Ist dies der Fall, wird der Detektionsalgorithmus mit Vorwissen aufgerufen.
2. Liegt kein Vorwissen vor, zum Beispiel beim Starten des Trackingsystems, oder war der Detektionsalgorithmus mit Vorwissen nicht erfolgreich, wird der Detektionsalgorithmus ohne Vorwissen durchlaufen.
3. Falls keine Korrespondenzen gefunden wurden, wird der Trackingalgorithmus hier abgebrochen, das Modell kann nicht detektiert werden. Falls drei Korrespondenzen aufgestellt werden konnten, wird aus diesen die Transformation des Modells zum Weltkoordinatensystem bestimmt, siehe Abschnitt 4.3.1. Die getrackten Punkte der Korrespondenzen erhalten einen Klassifikationsindex, um sie als Teil des Modells zu markieren.
4. Ausgehend von den bereits gefundenen Korrespondenzen werden weitere Korrespondenzen zwischen noch nicht klassifizierten, getrackten Punkten und Modellpunkten gesucht. Dabei kann die bereits berechnete Transformation verwendet werden, oder eine Suche nur über Distanzen durchgeführt werden. Die getrackten Punkte der Korrespondenzen erhalten einen Klassifikationsindex, um sie als Teil des Modells zu markieren.
5. Falls sich die Anzahl der Korrespondenzen erhöht hat, kann erneut die Transformation des Modells zum Weltkoordinatensystem auf Basis aller Korrespondenzen bestimmt werden, siehe Abschnitt 4.3.2.

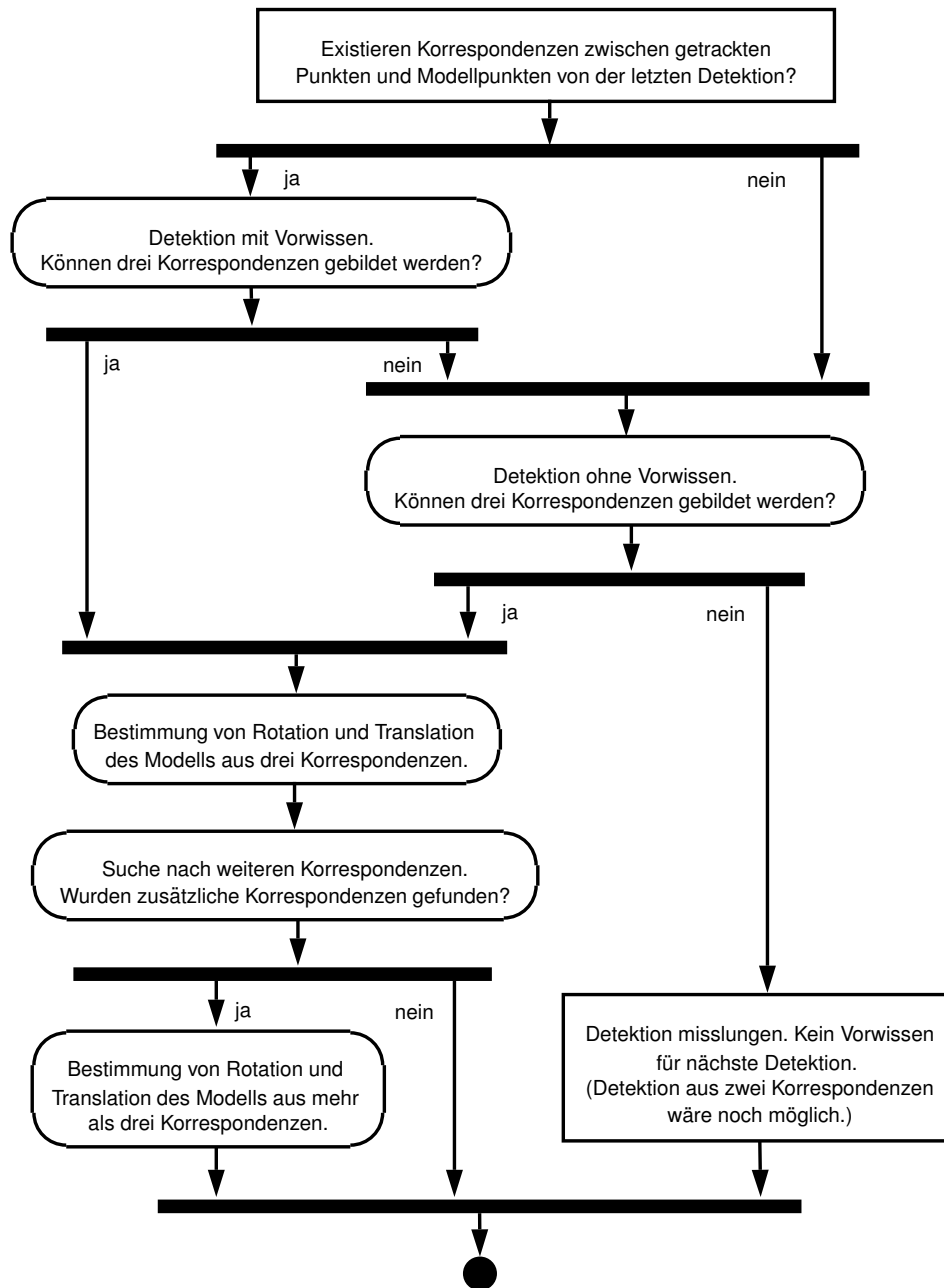


Abbildung 4.15. Trackingalgorithmus für Multipunktmodelle basierend auf Detektion mit und ohne Vorwissen aus dem letzten Frame.

Der Klassifikationsindex, der die Zugehörigkeit eines 3D-Punktes zu einem Modell ausdrückt, kann auch an die 2D-Punkte vergeben werden, die diesen 3D-Punkt erzeugen, da sie im 3D-Punktmodell gespeichert werden, siehe Abschnitt 3.3. Dies ermöglicht eine wiederholte, möglicherweise bessere Lösung des Korrespondenzproblems zwischen den 2D-Punkten der Kamerabildern oder eine Prädiktion mit Modellwissen auf den Kame-

rabildebenen. In der hier beschriebenen Umsetzung wird auf ein solches Vorgehen nicht zurückgegriffen, da dies nur in Einzelfällen eine Verbesserung bringt, aber einen erhöhten Rechenaufwand bedeutet und damit eine größere Latenz in jedem Frame verursacht.

Definition Multipunktmodell

Wie bereits beschrieben, bestehen die Multipunktmodelle im Wesentlichen aus einer Menge $\mathcal{X}_M = \{X_M^1, \dots, X_M^n\}$, $n \in \mathbb{N}$ von 3D-Punkten. Die vorgestellten Detektionsalgorithmen erfordern jedoch weitere Informationen über das Modell, sodass sich folgende Beschreibung eines Multipunktmodells formulieren lässt:

Definition 6 *Ein Multipunktmodell (MPM) besteht aus folgenden, fest definierten Parametern:*

\mathcal{X}_M die Menge $\{X_M^1, \dots, X_M^n\}$ mit $n \geq 3 \in \mathbb{N}$ der 3D-Punkte und
id ein eindeutiger Identifikationsindex für das Modell.

Aus der Menge \mathcal{X}_M der 3D-Punkte ergibt sich die zu speichernde Menge $\mathcal{D}_M = \{d_M^{jk} \in \mathbb{R} : 1 \leq j \leq n-1, j < k \leq n\}$ der $n_D = \frac{n(n-1)}{2}$ Distanzen zwischen den Modellpunkten X_M^i . Dabei gilt $d_M^{jk} = |X_M^j - X_M^k|$. Die während der Detektion zu speichernden und veränderlichen Parameter sind:

\mathcal{K}^{last} die Menge der zuletzt bestimmten Korrespondenzen zwischen getrackten Punkten X_W^j und Modellpunkten X_M^i . Für die Modellpunkte wird hier nur der entsprechende Index i gespeichert.

R^{last} die zuletzt berechnete Rotation des Modells und

T^{last} die zuletzt berechnete Translation des Modells.

Detektion aus zwei Korrespondenzen

Eine der grundlegenden Voraussetzungen für die Erkennung und Verfolgung eines Multipunktmodells mit den hier aufgezeigten Verfahren ist die Bestimmung von mindestens drei Korrespondenzen zwischen getrackten Punkten und Modellpunkten in jedem Frame. Dennoch lässt sich der Trackingalgorithmus so erweitern, dass in speziellen Fällen eine Detektion auf Basis von nur zwei Korrespondenzen durchgeführt wird. Da bei dieser Art der Detektion insbesondere die Rotation nicht voll bestimmt werden kann, ist eine solche Erweiterung nur dann sinnvoll, wenn beim Tracking zugunsten einer kontinuierlichen Erkennung Einbußen bei der Präzision in Kauf genommen werden können.

Für eine solche Detektion wird gefordert, dass die Distanz $d_W^1 = |X_W^1 - X_W^2|$ zwischen zwei getrackten Punkten einer im Modell gespeicherten Distanz d_M^{jk} entspricht. Falls der Ursprung des Modellkoordinatensystems aber nicht zufällig als Mitte dieser beiden Punkte gewählt wurde, lässt sich daraus weder die Position, noch die Rotation des Modells bestimmen. Aus diesem Grund macht die Detektion auf Basis von nur zwei Korrespondenzen nur dann Sinn, wenn Korrespondenzen, Rotation und Translation von der letzten Detektion vorliegen. Die Detektion hat dann folgenden Ablauf:

1. Analog zum Detektionsalgorithmus mit Vorwissen werden zwei Korrespondenzen $(X_W^{j_1}, X_M^{1,last})$ und $(X_W^{j_2}, X_M^{2,last})$ zwischen den aktuell getrackten Punkten und denen des letzten Zeitpunktes gesucht. Die Distanz $|X_W^{j_1} - X_W^{j_2}|$ muss einer im Modell gespeicherten Distanz entsprechen.
2. Ist der erste Schritt erfolgreich, kann die zuletzt berechnete Rotation R^{last} als Annäherung für die aktuelle Rotation verwendet werden, um damit auch die Translation anzunähern.

Da ein solches Vorgehen die Präzision der Detektion reduziert, ist diese Ergänzung zum Trackingalgorithmus nur sinnvoll, wenn als Anforderung an das Trackingverfahren eine kontinuierliche Erkennung (vergleiche Abbildung 4.13: Stabilität der Verfolgung) wichtiger ist, als höchst mögliche Präzision (vergleiche Abbildung 4.13: hohe Zuverlässigkeit).

4.3.4 Detektion mehrerer Modelle

In den vorausgegangenen Abschnitten wurde beschrieben, wie sich der Trackingalgorithmus für ein Modell zusammensetzt und konfigurieren lässt. Dabei wurde getrackten Punkten, die einem Modell zugeordnet werden konnten, jeweils der Identifikationsindex des Modells zugeordnet, sodass sie kein zweites Mal untersucht werden. Dieser Identifikationsindex erweist sich nun auch als nützlich, um Punkte, die bereits einem Modell zugewiesen wurden, für die Verarbeitung bei der Detektion der weiteren Modelle auszuklammern. Für die Verfolgung mehrerer Modelle bestehen also folgende Voraussetzungen:

- Es werden m Punkte X_W^1, \dots, X_W^m aus den Kamerabildern rekonstruiert.
- Es werden q Modelle, bestehend aus jeweils n_q Punkten $X_{Mq}^1, \dots, X_{Mq}^{n_q}$, verfolgt, für die jeweils die Distanzen $d_{Mq}^{i,j} = |X_{Mq}^i - X_{Mq}^j|$ zwischen den Punkten des jeweiligen Modells berechnet und gespeichert sind.
- Die Distanztoleranz δ ist vorgegeben.
- Alle getrackten Punkte, die einem Modell zugeordnet werden konnten, werden mit dem in Abschnitt 3.3 beschriebenen Klassifikationsindex im 3D-Punktmodell versehen.

Sequentielle Detektion mehrerer Modelle

Die einfachste Umsetzung zur Erkennung und Verfolgung mehrerer Modelle ist die sequentielle Detektion der einzelnen Modelle. Die Erkennung der Modelle wird hierbei in einer festen Reihenfolge nacheinander abgearbeitet. Bereits mit einem Klassifikationsindex versehene rekonstruierte Punkte werden für die Detektion eines nächsten Modells nicht mehr berücksichtigt. Dieses Vorgehen ist einfach zu implementieren und funktioniert in den meisten Fällen problemlos.

Vor der Verarbeitung der Modelle liegt wie gewohnt eine Anzahl m rekonstruierter 3D-Punkte X_W^i vor. Da die Distanzen zwischen diesen getrackten Punkten in jedem Frame berechnet werden müssen und die Anzahl $\frac{m(m-1)}{2}$ der Distanzen in Abhängigkeit der Markeranzahl m quadratisch steigt, stellt sich die Frage, wie effizient ein solcher Algorithmus noch ist. Um die Verarbeitungszeit des Trackingalgorithmus zu evaluieren wurde die Implementierung des Trackingalgorithmus als Bestandteil eines Infrarottrackingsystems [Schwald 04b] untersucht. Um die Verarbeitungszeit des Prozesses zu evaluieren, wurden zwei Modelle mit drei, eines mit vier und eines mit sieben Markern gleichzeitig verfolgt. Die Trackingsoftware wurde mit einem Athlon XP 1800+ mit 1GB RAM und einer Framerate von 21 fps getestet, wobei in jedem fünften Frame die Verarbeitungszeit gemessen wurde. Die Modelle wurden innerhalb des Interaktionsvolumens bewegt, sowie aus dem Interaktionsvolumen entfernt, sodass die Anzahl der Modelle erkennbar variierte. Tabelle 4.1 zeigt die gemittelten Ergebnisse von ≈ 700 Messungen. Dabei wurde genau ein Modell in 267 Frames erkannt, wobei die Verarbeitungszeit durchschnittlich 5,9 ms betrug, durchschnittlich 9,9 Markerpositionen rekonstruiert wurden und dabei 45 Distanzen berechnet und weiterverarbeitet wurden. Wenn alle vier Modelle erkannt wurden, liegt die durchschnittliche Anzahl der erkannten Punkte mit 18,9 über der Gesamtzahl der Marker, was auf Störpunkte zurückzuführen ist, siehe Abschnitt 4.1.1. Die Erkennung wurde davon nicht beeinflusst.

Anzahl erkannter Modelle	in x Frames erkannt	durchschn. Δ in ms	durchschnittliche Anzahl	
			von Punkten	von Distanzen
0	85	4,8	1,4	0
1	267	5,9	9,9	45
2	148	6,2	14,3	91
3	117	6,6	17,0	136
4	104	7,0	18,9	171

Tabelle 4.1. In ≈ 700 Messungen wurden zwischen 0 und 4 Modelle getrackt. Δ ist die Verarbeitungszeit für die entsprechende Anzahl von getrackten Modellen, beziehungsweise Punkten und Distanzen.

Die Verarbeitungszeit Δ umfasst die Bildverarbeitung, die Rekonstruktion der Marker aus zwei Kamerabildern und den Trackingalgorithmus. Offensichtlich steigt die Verarbeitungszeit mit der Anzahl der Modelle, beziehungsweise der rekonstruierten 3D-Punkte an, ist aber beispielsweise mit einer Zeit von $\approx 7,0$ ms für eine Konstellation mit vier Modellen und 19 Markern hinreichend gut. Insbesondere steigt die Verarbeitungszeit Δ nicht im selben Maße an, wie die Anzahl der Distanzen zwischen rekonstruierten Punkten bei zunehmender Anzahl von Markern und Modellen. Die Verarbeitungszeit steigt im Vergleich von einem zu vier Modellen um $\approx 20\%$ an, während die Anzahl der Distanzen zwischen getrackten Punkten um $\approx 280\%$ ansteigt.

Die relativ geringe Verarbeitungszeit ist zusammen mit der hohen Detektionsrate in diesem Test zufriedenstellend. Innerhalb dieses Tests war die Zuverlässigkeit der Detektion

sehr hoch, allerdings hängt dies auch von der geeigneten Wahl der Modelle ab, da bestimmte Konstellationen von Markern Probleme bereiten, die in Abschnitt 4.3.5 genauer diskutiert werden.

Parallele Detektion mehrerer Modelle mit gemeinsamer Distanzliste

Eine etwas komplexere Umsetzung zur Erkennung mehrerer Modelle besteht darin, die Modelle nicht getrennt voneinander und hintereinander abzuarbeiten, sondern aus den in den Modellen gespeicherten Distanzen eine gemeinsame Liste zu generieren. Eine solche Liste lässt auf effiziente Weise erkennen, ob eine Distanz zwischen zwei getrackten Punkten zwei verschiedenen Modellen zugeordnet werden kann. Da die Modelle vor der Detektion schon bekannt sein müssen, kann eine solche Liste als Vorverarbeitungsschritt einmalig wie folgt erzeugt werden:

1. Alle Distanzen werden in einer Liste gespeichert, wobei jeder Distanz, ähnlich den Punktmodellen aus Abschnitt 3.3, weitere Informationen zugeordnet sind:
 - der Klassifikationsindex des Modells und
 - die beiden Modellpunkte, die die Distanz bilden.
2. Diese Liste wird nach der Größe der Distanzen sortiert.

Bei der Detektion ohne Vorwissen wird dann nach der ersten Distanz d_W^1 nicht unter den in einem Modell gespeicherten Distanzen gesucht, sondern in der gemeinsamen Liste. Für jede, aufgrund der Distanztoleranz δ in Frage kommende Distanz, wird dann in dem Modell, welches dem Klassifikationsindex entspricht, nach drei Korrespondenzen gesucht. Gelingt dies in mehr als einem Modell, müssen weitere Kriterien angewendet werden. Insbesondere falls kein Vorwissen vorliegt, kann beispielsweise das Modell ausgewählt werden, welches den kleinsten maximalen Fehler bei den Differenzen zwischen den Distanzen d_M^i im Modell und den korrespondierenden Distanzen d_W^j aufweist.

Ein solches Vorgehen ist naheliegend, soweit kein Vorwissen existiert. Ist Vorwissen vorhanden, ist es jedoch sinnvoll, dieses anzuwenden, bevor eine Suche über die gesamte Liste durchgeführt wird. Ausgangspunkt ist wieder die Menge der getrackten 3D-Punkte $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$, $m \in \mathbb{N}$. Die Punktmodelle aus Abschnitt 3.3 werden so erweitert, dass sie temporär mehrere potentielle Klassifikationsindizes enthalten können:

1. Parallel werden alle Modelle, für die Vorwissen vorhanden ist, in der gesamten Punktmenge \mathcal{X}_W detektiert. Eine erfolgreiche Detektion ist zunächst nur vorläufig gültig. Außerdem erhalten wie gewohnt alle getrackten Punkte, für die Korrespondenzen in einem Modell gefunden wurden, den entsprechenden Klassifikationsindex.
2. Nach diesem Schritt sind die Punkte aus \mathcal{X}_W mit keinem, einem oder mehreren Klassifikationsindizes versehen. Jedes Modell, das über die Mindestanzahl von Korrespondenzen zu getrackten Punkten, die wiederum eindeutig diesem Modell zugeordnet wurden, verfügt, gilt als detektiert. Im Allgemeinen ist die Mindestanzahl

drei, kann aber nach Bedarf größer definiert werden. Ein Punkt ist einem Modell eindeutig zugeordnet, wenn er nur einen Klassifikationsindex erhalten hat.

3. Bleiben Modelle übrig, für die zwar eine Mindestanzahl an Korrespondenzen zu getrackten Punkten vorhanden ist, allerdings nicht hinreichend viele getrackte Punkte eindeutig zugewiesen werden können, gibt es folgende Möglichkeiten:
 - Das Modell wird als detektiert akzeptiert. Einem getrackten Punkt könnten also mehrere Modelle zugeordnet sein.
 - Das Modell wird nicht als detektiert akzeptiert.
 - Anhand weiterer Kriterien wird entschieden, zu welchem Modell nicht eindeutig zugewiesene 3D-Punkte gehören. Dies kann beispielsweise wieder durch den Vergleich von Distanzen mit anderen bereits eindeutig zugewiesenen Punkten erfolgen.
4. Nach diesen Schritten erfolgt die eingangs beschriebene Detektion nach den Modellen, die nicht gefunden wurden, beziehungsweise für die kein Vorwissen vorlag. Dabei werden nun nur noch die getrackten Punkte untersucht, die keinen Klassifikationsindex besitzen, und es werden nur die Distanzen aus der gesamten Distanzliste berücksichtigt, die nicht bereits einem Modell zugeordnet sind.

Auf die hier beschriebene Weise können Doppeldeutigkeiten besser aufgelöst werden als bei der sequentiellen Verarbeitung, allerdings ist die Verarbeitungszeit durchschnittlich höher, da über größere Mengen von Punkten und Distanzen gesucht wird. Weitere Überlegungen, wie Doppeldeutigkeiten insbesondere auch innerhalb eines Modells behandelt werden können, befinden sich im folgenden Abschnitt.

4.3.5 Verwechselbare Distanzen

Wie in den vorausgegangenen Abschnitten dargelegt, birgt die Detektion basierend auf Distanzmessungen auch eine Problematik: einzelne oder sogar Kombinationen von Distanzen können verwechselbar sein. Insbesondere symmetrische Anordnungen von Markern führen zu solchen Konstellationen. Dennoch wirkt sich nicht jede solcher Konstellationen gleichermaßen negativ auf die Detektion aus. Bei einer Detektion stehen den $\frac{m(m-1)}{2}$ Distanzen der m getrackten Punkte die Summe $\frac{1}{2} \sum_{j=1}^p n_j(n_j - 1)$ der Distanzen der p zu trackenden Modelle mit jeweils n_1, \dots, n_p Modellpunkten gegenüber. Im Wesentlichen gibt es die folgenden ungünstigen Konstellationen von Markern, wobei der Begriff der Gleichheit von Distanzen wieder in Abhängigkeit der Distanztoleranz δ , siehe Abschnitt 3.5, zu verstehen ist:

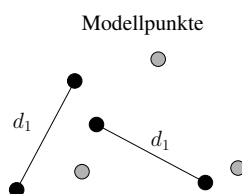
1. Innerhalb eines Modells:
 - (a) Zwei gleiche Distanzen, die keinen gemeinsamen erzeugenden Punkt haben.

- (b) Zwei gleiche Distanzen, die einen gemeinsamen erzeugenden Punkt haben. Die Distanz, die die beiden anderen Punkte verbindet ist ungleich den beiden anderen. Die Punkte formen also ein gleichschenkliges Dreieck. Die Problematik mit drei oder mehr Distanzen mit einem gemeinsamen erzeugenden Punkt lässt sich auf die Problematik mit zwei solcher Distanzen zurückführen.
- (c) Drei gleiche Distanzen, die jeweils genau einen erzeugenden Punkt mit einer anderen Distanz gemeinsam haben. Die drei erzeugenden Punkte formen also ein gleichseitiges Dreieck.
- (d) Zwei Konstellationen aus jeweils drei Punkten beziehungsweise Distanzen, die sich gleichen.

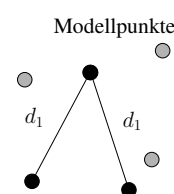
2. Zwischen Modellen:

- (a) Gleiche Distanzen in unterschiedlichen Modellen.
- (b) Konstellationen aus jeweils drei Punkten zweier unterschiedlicher Modelle, die sich gleichen.

Außerdem gibt es weitere ungünstige Konstellationen, die sich aber auf Kombinationen der oben genannten Fälle zurückführen lassen. So treten innerhalb einer rechteckigen Anordnung von vier Markern dreimal Fall 1a und sechsmal Fall 1d auf, wobei bei letzterem sich vier Konstellationen von Markern gleichen, siehe Abbildung 4.17. Bei einer quadratischen Anordnung von vier Markern tritt zusätzlich noch mehrfach der Fall 1b ein, sodass dies eine noch ungünstigere Konstellation von Markern ist. Die Abbildungen 4.16 und 4.18 zeigen exemplarisch weitere verschiedene ungünstige Konstellationen.



(a) Fall 1a: Wenig problematisch innerhalb eines Modells.



(b) Fall 1b: Zwei gleiche Distanzen an einem gemeinsamen Punkt.

Abbildung 4.16. Ungünstige Konstellationen von Punkten für die Detektion.

Die oben aufgeführten Fälle unterscheiden sich jedoch wesentlich in ihrer Problematik. Die Fälle 1a und 2a sind im Allgemeinen bei einer Detektion wenig problematisch und spielen nur in Spezialfällen eine Rolle, wie beim initialen Anlernen eines Multipunktmodells, siehe Abschnitt 5.1.1, und bei einer Detektion, die auch mit nur zwei Korrespondenzen funktionieren soll, siehe Detektion aus zwei Korrespondenzen in Abschnitt 4.3.3. Die anderen Fälle sind kritischer.

Fall 1b ist der am häufigsten auftretende, kritische Fall, wie die praktische Erfahrung gezeigt hat. Gibt es bei einer solchen Konstellation keine weitere Information, zum Beispiel durch eine weitere Korrespondenz oder die Transformation des Modells im letzten

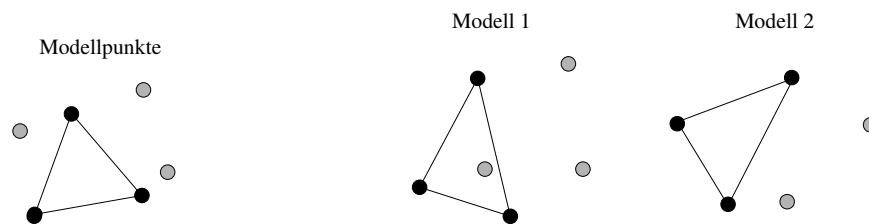


(a) Dreimal Fall 1a innerhalb einer rechteckigen Anordnung von vier Punkten.

(b) Fall 1d: Die Konstellation aus drei Punkten kommt in der rechteckigen Anordnung viermal vor. Der Fall 1d tritt also sechsmal auf.

Abbildung 4.17. Ungünstige Konstellationen von Punkten für die Detektion in rechteckiger Anordnung.

Frame, kann ein Modell zwar noch erkannt, aber dennoch falsch positioniert und orientiert werden. Der Fall 1d ist ähnlich und kommt insbesondere bei den oben beschriebenen rechteckigen beziehungsweise quadratischen Konstellationen vor. Dieselbe Problematik weist auch der Fall 1d auf. Der ungünstigste Fall ist 2b, da hier zusätzlich das falsche Modell erkannt werden kann.



(a) Fall 1c: Drei Punkte sind so angeordnet, dass sie ein gleichseitiges Dreieck bilden.

(b) Fall 2b: Konstellationen aus jeweils drei Punkten zwei unterschiedlicher Modelle, die sich gleichen.

Abbildung 4.18. Ungünstige Konstellationen von Punkten für die Detektion.

Die einfachste Methode die Problematik dieser Markerkonstellationen zu umgehen ist, Symmetrien innerhalb von Modellen und Ähnlichkeiten zwischen Modellen zu vermeiden. Dies ist bei ein oder zwei Modellen mit einer geringen Anzahl von Markern kein Problem sein. Bei einer größeren Anzahl von Markern und/oder Modellen können diese Voraussetzungen jedoch schon komplexere Überlegungen bei der Anordnung von Markern erfordern. In jedem Fall ist es daher sinnvoll ein Analysewerkzeug bereitzustellen, das im Vorfeld problematische Konstellationen erkennt und diese anzeigt, unter Umständen eine Änderung der Konstellation vorschlägt oder sogar einem erweiterten Detektionsalgorithmus Parameter zur Verfügung stellt, sodass problematische Bereiche bei der Detektion gesondert behandelt werden können. Für solch ein Analyseprogramm bestehen folgende Voraussetzungen:

- Die zu verfolgenden Modelle sind bekannt. Die 3D-Koordinaten der Modellpunkte liegen vor, entweder als Planungsdaten, wie ein Modell gebaut werden soll, oder als Punkte eines bereits vermessenen Modells.
- Da in der Realität Distanzen fast nie wirklich exakt gleich sind, hängt es von der für die Detektion gewählten Distanztoleranz δ ab, ob Konstellationen ungünstig sind oder nicht. Daher muss die Analyse abhängig von der Distanztoleranz sein und muss neu durchgeführt werden, sobald sich Modelle oder die Distanztoleranz selbst ändern.

Ein Programm zur Analyse lässt sich dann einfach implementieren, indem systematisch die oben aufgelisteten Fälle untersucht werden. Soweit sich die Analyse auf eine reine Anzeige problematischer Konstellationen beschränkt, um eine Vermeidung ungünstiger Konstellationen zu unterstützen, steht damit alles zur Verfügung.

Hilfreicher für den Anwender ist es, wenn aus der Analyse von Modellen auch ein Lösungsvorschlag resultiert. Dazu werden die Punkte manipuliert, die die meisten und vor allem die schwerwiegendsten Probleme verursachen. Iterativ können dann die Multipunktmodelle verändert werden, indem immer der aktuell problematischste Punkt verschoben wird, danach eine neue Analyse durchgeführt wird, bis die Analyse ein zufriedenstellendes Resultat liefert. Diese Funktionalität ist in einer Planungsphase für Multipunktmodelle nützlich, allerdings nicht, wenn die Modelle schon existieren, da es in der Praxis mühsam oder nicht möglich sein kann, Marker noch einmal zu versetzen.

Neben der reinen Anzeige problematischer Konstellationen und der Berechnung modifizierter Modelle bietet sich noch ein dritter Lösungsansatz an: die Erweiterung der Multipunktmodelle durch Parameter, die problematische Konstellationen speichern und während der Detektion bewerten können. Die Beschreibung der Multipunktmodelle, siehe Abschnitt 4.3.3, wird dazu so erweitert, dass eine Liste ungünstiger Konstellationen zusätzlich gespeichert werden kann. Jedes Element dieser Liste besteht

- aus einer Untermenge der Modellpunkte,
- einem Parameter p_{Fall} , der die Art der oben aufgeführten Problemfälle beschreibt,
- sowie einer Liste von Verweisen auf andere Modelle oder Konstellationen im eigenen Modell, falls es sich um verwechselbare Konstellationen handelt.

Wurde bei einer Detektion mit oder ohne Vorwissen die Mindestanzahl der Korrespondenzen zwischen getrackten Punkten und Modellpunkten gefunden, wird überprüft, ob die Menge der Modellpunkte dieser Korrespondenzen eine der Untermengen, die in der Liste der ungünstigen Konstellationen abgespeichert ist, enthält. Ist dies der Fall, wird in Abhängigkeit von p_{Fall} eine Aktion ausgelöst, zum Beispiel bei $p_{Fall} = 1b$ eine weitere Korrespondenz gesucht, oder bei $p_{Fall} = 2a$ die möglicherweise vorhandene letzte Position des anderen zu verwechselnden Modells berücksichtigt.

Dieser Ansatz birgt weitere Möglichkeiten, die Zuverlässigkeit des Trackings zu erhöhen, ist aber auch mit einer höheren Verarbeitungszeit verbunden, sodass im Sinne der Abbildung 4.13 je nach Applikation entschieden werden muss, ob sich der Aufwand für diese Technik lohnt.

4.4 Zusammenfassung

Dieses Kapitel baut auf der Existenz einer Menge von 3D-Punkten auf, die zu einem Erfassungszeitpunkt rekonstruiert wurden. Die Punkte liegen in dem 3D-Punktmodell vor, das im vorausgegangenen Kapitel definiert wurde.

Eine Verfolgung einzelner Marker macht die Erweiterung des 3D-Punktmodells um den Identifikationsindex notwendig, sodass rekonstruierte Punkte nicht nur klassifiziert, sondern auch als einzelnes Objekt verfolgt werden können. Algorithmen zur Verfolgung eines einzelnen oder mehrerer einzelner Marker sind beschrieben. Die Identifizierung einzelner Marker ist für die Verfolgung der starren Modelle von geringerer Bedeutung, da die Modelleigenschaften der starren Modelle andere zuverlässige Kriterien zur Erkennung bieten.

Zweipunktmodelle sind einfach zu beschreiben und haben als wichtigstes Erkennungsmerkmal den Abstand zwischen den zwei entsprechenden Markern. Aus ihnen kann eine Position und eine Richtungsgerade berechnet werden, sodass es sich bei deren Verfolgung im Wesentlichen um ein 3DOF-Trackingverfahren handelt. Auch wenn die Anzahl der gleichzeitig verwendbaren Zweipunktmodelle beschränkt ist, eignen sich diese Modelle für Anwendungen mit mehreren Benutzern.

Wichtigster Bestandteil dieses Kapitels sind die Multipunktmodelle, die vor allem aus einer Menge fest zueinander platzierter Punkte besteht. Bei den Anforderungen an die Eigenschaften eines Algorithmus zur Verfolgung der Multipunktmodelle steht eine hohe Zuverlässigkeit der Verfolgung einer hohen Erkennungsrate (Stabilität) und einer geringen Verarbeitungszeit gegenüber. Der hier entwickelte Detektionsalgorithmus kann entsprechend einer Gewichtung zwischen diesen Eigenschaften konfiguriert werden. Er gliedert sich in zwei Hauptbestandteile: die Erkennung ohne Vorwissen (Registrierung des Modells) und die Erkennung mit Vorwissen (Verfolgung des Modells) über die vorausgegangene Position und Orientierung.

Bei der konkreten Umsetzung eines oder mehrerer Multipunktmodelle ist darauf zu achten, dass für deren Detektion Symmetrien und insbesondere sich gleichende Konstellationen von drei und mehr Markern, innerhalb eines oder zwischen verschiedenen Modellen, ungünstig sind. Eine Übersicht dieser ungünstigen Konstellationen, sowie darüber, wie sie bei der Detektion berücksichtigt oder noch besser vermieden werden können wurde gegeben.

Kapitel 5

Anlernen starrer Modelle

Bei den n -Punktmodellen, die auf festen Konstellationen beruhen, also den Zweipunktmodellen aus Abschnitt 4.2 und den Multipunktmodellen aus Abschnitt 4.3, müssen die Informationen über deren Geometrie, also die Distanzen, beziehungsweise Koordinaten der 3D-Punkte des Modells bekannt sein, um sie identifizieren und verfolgen zu können. Für Zweipunktmodelle lässt sich das Anlernen sehr einfach umsetzen, indem ein einzelnes Zweipunktmodell im Interaktionsbereich des kalibrierten Stereokamerasystems platziert oder bewegt wird. Die zu vermessende Distanz kann dann zu einem bestimmten Zeitpunkt ermittelt werden, oder aber aus mehreren Messungen gemittelt werden. Eine Vermessung von Hand ist im Allgemeinen ebenfalls hinreichend.

Auch für drei Marker stellt eine einfache Vermessung mit dem Stereokamerasystem oder von Hand kein Problem dar. Mit zunehmender Anzahl der Marker, die zu einem Modell gehören, ist eine Vermessung von Hand praktisch ausgeschlossen. Je nachdem wie und auf was die Marker angeordnet sind, kann auch die Vermessung mit dem Stereokamerasystem schwierig werden. Kann das Objekt, auf dem die Marker platziert sind, so im Interaktionsbereich positioniert werden, dass alle Marker zum selben Zeitpunkt sichtbar sind, ist ein einfaches Anlernen wie oben beschrieben möglich. Ist die Anordnung der Marker jedoch so, dass eine Erfassung aller Marker zum selben Zeitpunkt nicht möglich ist, bleiben zwei Lösungen, um das Modell anzulernen:

- Die Anzahl der verwendeten Kameras im Stereosystem muss erhöht werden. Die zusätzlichen Kameras müssen dabei so platziert werden, dass ein Anlernen zu einem Zeitpunkt möglich ist, das heißt alle Marker auf einmal sichtbar sind.
- Ein Verfahren muss verwendet werden, bei dem das Anlernen schrittweise erfolgt und das Modell gedreht beziehungsweise bewegt werden kann.

Der erste Fall ist sinnvoll, wenn die Anzahl der Kameras ohnehin auch für das eigentliche Tracking erhöht wird. In allen anderen Fällen bietet sich ein Verfahren an, bei dem Multipunktmodelle schrittweise durch Bewegen im Interaktionsbereich angelernt werden [Schwald 04a], wie es im Folgenden präsentiert wird.

Die beschriebene Situation, dass eine Anordnung von Markern ein solches Verfahren erfordert, ist nicht rein theoretisch, sondern tritt in der Praxis beispielsweise dann ein,



Abbildung 5.1. AR-System für minimal-invasive Chirurgie. Infrarotkameras verfolgen das mit Markern besetzte AR-Display.

wenn ein Objekt, auf dem die Marker direkt platziert werden, von allen Seiten erkennbar sein soll. Ein abstraktes Beispiel ist ein würfelförmiges Objekt, welches in jeder Pose von einem Zweikamerasystem erkannt werden soll. Das Platzieren von Markern auf jeder Seite macht ein Anlernen zu einem Zeitpunkt unmöglich. Ein weiteres Beispiel ist dem medizinischen AR-Projekt MEDARPA [Medarpa 06] entnommen, siehe Abbildungen 5.1 und 5.2, bei dem ein transparentes Display verfolgt wird, um aus der Sicht des Arztes präoperativ gewonnene Daten direkt auf dem Patienten überlagert sehen zu können. Eine genauere Beschreibung dieser Anwendung befindet sich in Abschnitt 8.1.

Das Prinzip, Struktur aus Bewegung (structure from motion, SfM) zu gewinnen, wird im Bereich Computer Vision häufig verwendet. Zahlreiche Problemstellungen wurden von Azarbayejani und Pentland [Azarbayejani 95] mit Hilfe von SfM bearbeitet, unter anderem die Rekonstruktion eines Objekts aus einem Video, ein Thema, welches immer wieder aufgegriffen wird, zum Beispiel in [Gibson 03]. In diesen Ansätzen ist eine Millimetergenauigkeit und die Möglichkeit der Interaktion in Echtzeit weniger wichtig als das Ergebnis eines gut aussehenden 3D-Modells einschließlich der Texturen.

Die Erzeugung von 3D-Modellen aus komplexen Objekten für Visualisierungszwecke mit einem 3D-Scanner in Echtzeit wurde in [Rusinkiewicz 02] beschrieben. Das System erlaubt dem Benutzer ein zu rekonstruierendes Objekt in der Hand zu drehen und dabei das stetig verbesserte virtuelle Modell als direkte Ausgabe des 3D-Scanners zu betrachten. Das Integrieren einer Benutzerinteraktion in den Lernprozess zur Erzeugung eines besseren Modells ist dem in dieser Arbeit verfolgten Ansatz ähnlich, das Vorgehen und das Resultat sind jedoch aufgrund der unterschiedlichen Anwendungen verschieden.

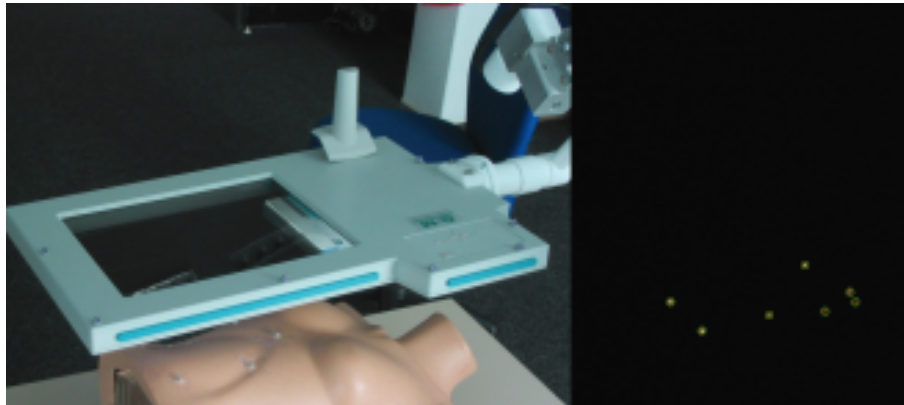


Abbildung 5.2. Links: AR-Display mit Infrarotdioden als Marker aus einer der Kamera-perspektiven. Rechts: Kamerabild mit den Überlagerungen der detektierten Marker.

Ein Verfahren automatisch neue Positionen zuvor unbekannter Punkte anzulernen wurde in [Kanbara 01] präsentiert. Das auf einem Stereokamerasystem basierende AR-System verfolgt Marker und natürliche Merkmale. Die Marker werden mit einem auf Prädiktion basierenden hybriden Verfahren verfolgt, welches Farbinformation und die Daten eines inertialen Sensors verarbeitet. Das Ziel ist dabei nicht, die vollständige Beschreibung eines Modells zu erreichen, sondern neue Merkmale für die Verfolgung zu erzeugen, beziehungsweise diese wieder zu löschen, wenn sie nicht mehr genutzt werden.

5.1 Lernalgorithmus für Multipunktmodelle

Der Lernalgorithmus für die in Abschnitt 4.3 beschriebenen Multipunktmodelle unterteilt sich in verschiedene Lernphasen [Figueiredo 02], die in Abbildung 5.3 skizziert sind. In der ersten Phase werden Distanzen zwischen Punkten, die innerhalb einer gewissen Zeitspanne registriert wurden, analysiert, um Initialwerte für die Definition eines neuen Modells zu erhalten. Die initiale Definition des Modells wird daraufhin in einem zweiten Schritt über eine kurze Zeitspanne validiert. Diese beiden Schritte tragen der Tatsache Rechnung, dass die Messungen der 3D-Punkte durch das Trackingsystem fehlerbehaftet sind, siehe Abschnitt 3.2, und möglicherweise zusätzliche Störungen nicht auszuschließen sind.

Ausgehend von einem - durch die validierten Initialwerte definierten - Dreipunktmodell können schrittweise neue Punkte hinzugefügt werden, bis keine neuen Punkte mehr gefunden werden oder das Modell eine vordefinierte Größe erreicht hat. Während dieser Lernphase bewegt der Benutzer das Markermodell so, dass jeder Marker zumindest für einen kurzen Zeitraum erfasst werden kann.

5.1.1 Distanzmessung

In dieser ersten Lernphase werden 3D-Punkte in einer vordefinierten Anzahl von n_1 Frames, beispielsweise $n_1 = 100$, rekonstruiert und in Weltkoordinaten gespeichert. In

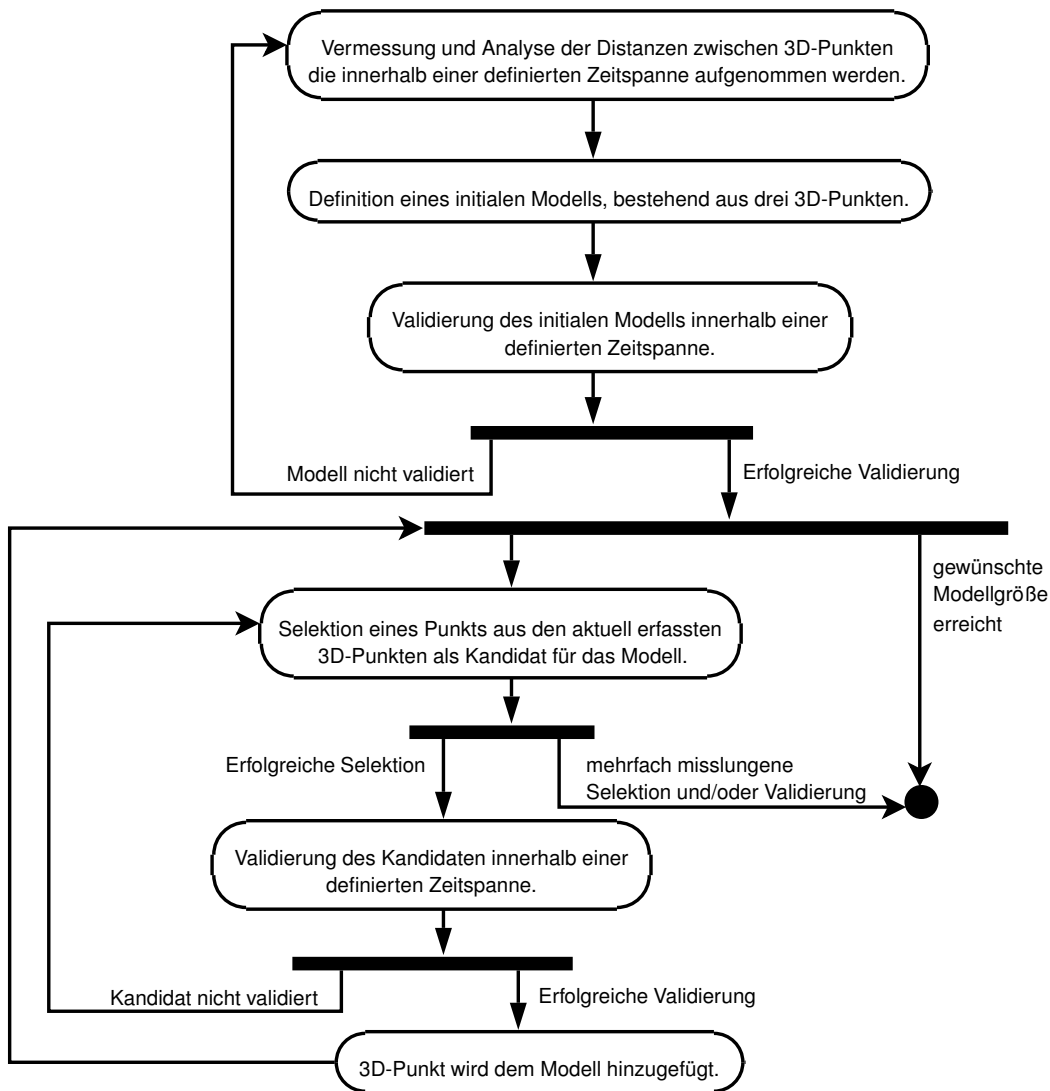


Abbildung 5.3. Der Lernalgorithmus für Multipunktmodelle.

jedem Frame $j \in \{1, \dots, n_1\}$ werden dabei aus den m_j 3D Punkten dieses Frames $m_j(m_j - 1)/2$ Distanzen berechnet. Daraufhin werden die Distanzen gruppiert, indem alle Distanzen, deren Größe sich innerhalb eines gewissen Intervalls I_p mit einer vordefinierten Länge $|I_p|$ befinden, zu einer Untermenge von Distanzen zusammengefasst werden. Da die Bildung der Intervalle, beziehungsweise der Untermengen, zum Ausgleich der Messungenauigkeiten dient, bietet es sich an, die Intervallgröße in Abhängigkeit von der in Abschnitt 3.5 eingeführten Distanztoleranz zu wählen und beispielsweise $|I_p| = 2\delta$ für alle Intervalle I_p zu setzen.

Die Benutzerinteraktion beschränkt sich in diesem Schritt darauf, das Markermodell so im Interaktionsbereich zu platzieren, dass mindestens drei Marker gut erfasst werden können. Das Modell kann auch leicht bewegt werden, wichtig ist jedoch, dass hauptsächlich dieselben Marker in den meisten der n_1 Frames erfasst werden. Nachdem die Distanzen der 3D-Punkte aus den n_1 Frames berechnet wurden, enthält ein ideales Intervall n_1 Dis-

tanzen, das heißt, eine Distanz wurde in jedem Frame erfasst und es gibt keine zweite Distanz ähnlicher Größe. Dennoch bietet es sich an, auch hier eine gewisse Toleranzgrenze einzubeziehen, sodass Intervalle I_p , in denen zwischen $\frac{9}{10}n_1$ bis n_1 Distanzen enthalten sind, also $\frac{9}{10}n_1 \geq \#I_p \geq n_1$, als „gut“ qualifiziert werden. Intervalle, die mehr als n_1 Distanzen enthalten, werden als „unakzeptabel“ qualifiziert, da hier die Wahrscheinlichkeit hoch ist, dass ähnliche Distanzen zwischen Markern vorliegen. Ähnliche Distanzen zwischen Markern verursachen während der Detektion von Modellen nicht notwendigerweise Probleme, siehe Abschnitt 4.3.5, aber sie stellen keine guten Initialwerte für das Anlernen von Modellen dar.

5.1.2 Modellerzeugung und -validierung

Nach der Gruppierung der Distanzen in den entsprechenden Intervallen werden in einem nächsten Frame $n_1 + 1$ die rekonstruierten 3D-Punkte X_W^1, \dots, X_W^m betrachtet. Wiederum werden alle Distanzen zwischen den m 3D-Punkten bestimmt. Falls eine dieser neu gebildeten Distanzen innerhalb eines als „gut“ qualifizierten Intervalls liegt, werden die zwei 3D-Punkte X_W^i und X_W^j , um deren Distanz es sich handelt, als die ersten beiden Punkte des anzulernenden Modells gewählt. Ein dritter Punkt X_W^k wird daraufhin so ausgewählt, dass beide Distanzen $\overline{X_W^i X_W^k}$ und $\overline{X_W^j X_W^k}$ innerhalb eines „guten“ Intervalls liegen. Kann ein Punkt X_W^k gefunden werden, der diese Bedingungen erfüllt, wird ein Modell bestehend aus den drei Punkten X_W^i, X_W^j und X_W^k definiert. Um das lokale Koordinatensystem des Modells nicht wahllos durch die aktuelle Position der drei Punkte im Weltkoordinatensystem festzulegen, wird der Ursprung in den Punkt X_W^i gelegt, indem alle drei Punkte um $-X_W^i$ transliert werden, also

$$X_M^1 := X_W^i - X_W^i = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{und} \quad X_M^2 := X_W^j - X_W^i, \quad X_M^3 := X_W^k - X_W^i. \quad (5.1)$$

Die hieraus resultierenden Punkte X_M^1, X_M^2 und X_M^3 werden als Modellpunkte gespeichert. Lassen sich keine drei Punkte finden, die die oben genannten Bedingungen erfüllen, müssen entweder nochmals Distanzen angelernt werden, wie in Abschnitt 5.1.1 beschrieben, oder es werden Punkte ausgewählt, die Distanzen bilden, deren entsprechende Intervalle I nicht als „gut“ qualifiziert wurden, für die also gilt $\#I_p < \frac{9}{10}n_1$.

Nach der Definition und Speicherung des Modells folgt über mehrere Frames hinweg eine Validierungsphase. In einer vordefinierten Anzahl n_2 von Frames wird das Modell mit den in Abschnitt 4.3.3 beschriebenen Verfahren registriert und verfolgt. Wird das Modell in einem vorab festgelegten Prozentsatz der n_2 Frames gefunden, gilt das Modell als validiert. Während der Validierungsphase kann und sollte das Modell leicht bewegt werden, um falsch angelernte Punkte, die durch eine feste Positionierung des Modells begünstigt werden, zu vermeiden. Ein Beispiel für das Auftreten eines falsch oder schlecht angelernten Punktes ist die Rekonstruktion eines 3D-Punktes aus einer Überlappung zweier Marker in einem Kamerabild. Durch Bewegung des Modells in der Phase der Distanzanalyse, siehe Abschnitt 5.1.1, können solche Phänomene stark reduziert werden.

5.1.3 Hinzufügen neuer Punkte

Das Hinzufügen weiterer 3D-Punkte zum Modell basiert im Wesentlichen auf der Verfolgung des Modells durch den in Abschnitt 4.3.3 beschriebenen Algorithmus. Ein 3D-Punkt, der nicht zum Modell gehört aber unter Umständen hinzugefügt werden soll, wird dabei ebenfalls als neuer Kandidat verfolgt. Falls der Punkt über eine längere Zeitspanne in Relation zum Modell fest bleibt, wird er dem Modell hinzugefügt.

Dazu wird nach der Detektion eines Modells mit $p \geq 3$ Modellpunkten X_M bei jedem Punkt X_W eines Frames, der als zu einem Modell gehörend identifiziert wurde, der Klassifikationsindex des Modells gesetzt. Gibt es unter den n Punkten X_W^1, \dots, X_W^m eines Frames mindestens einen Punkt, der nach der Detektion des Modells nicht klassifiziert wurde, wird dieser Punkt X_W^{kand} Kandidat für einen neuen Modellpunkt. Dieser Punkt wird mit der durch das detektierte Modell bekannten Transformation von Welt- in Modellkoordinaten in seiner Repräsentation in Modellkoordinaten X_M^{kand} abgespeichert. Als nächster Schritt werden in einer vordefinierten Anzahl n_3 von Frames zwei Bedingungen überprüft:

- ob das Modell mit p Punkten gefunden wurde und, falls dies zutrifft,
- ob der Kandidat X_M^{kand} einen korrespondierenden Punkt in Weltkoordinaten besitzt, der nicht zum Modell gehört. Der Begriff des korrespondierenden Punktes ist dabei im Sinne von Definition 5 aus Abschnitt 4.3.1 zu verstehen.

Sind beide Bedingungen in einem vordefinierten Prozentsatz der n_3 Frames erfüllt, wird der Kandidat X_M^{kand} dem Modell hinzugefügt. Das Hinzufügen neuer Punkte kann solange wiederholt werden, bis eine festgelegte Anzahl von Modellpunkten erreicht ist oder bis in mehreren Versuchen kein Kandidat mehr gefunden, beziehungsweise validiert werden kann.

5.2 Evaluation des Lernalgorithmus

Für eine Evaluierung des Lernalgorithmus wurde das am Anfang dieses Kapitels erwähnte, mit Markern versehene, transparente Display verwendet. Auf dem Display sind sieben Infrarotdioden fest angebracht, die jeweils in bestimmten Ausrichtungen des Displays alle von zwei Kameras zu einem Zeitpunkt erfasst werden können. Die Evaluation wird anhand verschiedener Vorgehensweisen, die dann verglichen werden, wie folgt durchgeführt:

1. Das Anlernen der sieben Marker in verschiedenen Ausrichtungen des Displays, aber ohne das Objekt während der Lernphasen zu bewegen. Diese Tests gelten als Referenz für die folgenden Tests und entsprechen der Anlernprozedur der meisten artverwandten Trackingsysteme. Die Tests nach dieser Vorgehensweise sind in Abschnitt 5.2.1 beschrieben.
2. Das Anlernen der sieben Marker, wobei das Objekt auf verschiedene Arten bewegt wird. Die Marker sind meistens für beide Kameras sichtbar, aber teilweise auch verdeckt. Die Tests nach dieser Vorgehensweise sind in Abschnitt 5.2.2 beschrieben.

3. Drei weitere Marker werden so am Display angebracht, dass das Display nun in weitere Stellungen gebracht werden kann, in denen es verfolgt werden kann. In diesem Sinne erhöhen die zusätzlichen Marker die Bewegungsfreiheit des Displays, wenn man dessen Verfolgbarkeit als Voraussetzung nimmt. Mit dieser Markerkonstellation ist es nicht mehr möglich, alle zehn Marker zu einem Zeitpunkt mit beiden Kameras zu erfassen. Das Objekt muss während der Lernphase bewegt werden, um alle zehn Marker anlernen zu können. Die Tests nach dieser Vorgehensweise sind in Abschnitt 5.2.3 beschrieben.

Der dritte Test zeigt den Nutzen des hier präsentierten Lernalgorithmus im Vergleich zu Ansätzen, bei denen ein Bewegen des Objekts nicht möglich ist. Die Bewegung des Objekts während des Anlernens ist notwendig, um alle Marker erfassen zu können, auch wenn nur zwei Kameras während des Lernprozesses vorgesehen sind. Die drei neuen Marker wurden auf der Rückseite des Displays nicht nur für die Evaluation angebracht, sondern auch um die Bewegungsfreiheit des Displays, wie in Abschnitt 5.2.3 beschrieben, zu erhöhen.

Das Zweikamerasystem ist, wie in Abbildung 5.1 dargestellt, mit einem Basisabstand von ca. 2 m zwischen den Kameras auf einem Gestell montiert. Die Distanz der Kameras zum zu verfolgenden Objekt, dem AR-Display, beträgt ungefähr 1,6 m. Die intrinsischen Parameter wurden mit dem Verfahren [Zhang 98] kalibriert. Die Autokalibrierung der extrinsischen Kameraparameter, ähnlich zu [Azarbayejani 95], wurde mit 200 Punkten durchgeführt. Der Rekonstruktionsfehler betrug durchschnittlich 0,14 mm und maximal 0,43 mm.

(mm)	Versuch 1	Versuch 2	manuell
Mittelwert 1. Distanz	151,072	151,11	149,7
Mittelwert 2. Distanz	240,419	240,349	240,0
Mittelwert 3. Distanz	284,238	284,158	283,3
Standard Abweichung 1. Distanz	0,2957	0,3210	-
Standard Abweichung 2. Distanz	0,3767	0,3658	-
Standard Abweichung 3. Distanz	0,5301	0,5196	-
Maximale Abweichung 1. Distanz	1,0723	0,8102	-
Maximale Abweichung 2. Distanz	1,1692	1,2488	-
Maximale Abweichung 3. Distanz	1,6682	1,3482	-

Tabelle 5.1. Distanzmessungen (je Versuch 100 Messungen) mit Mittelwert, Standardabweichung und maximaler Abweichung in mm.

Um die Genauigkeit des Systems abschätzen zu können, wird ein Dreimarkermodell in demselben Interaktionsbereich bewegt, in dem die sieben, beziehungsweise zehn Marker am AR-Display angelernt werden sollen. Die drei Marker sind in einem rechtwinkligen Dreieck mit handvermessenen Distanzen von 149,7 mm, 240,0 mm und 283,3 mm angebracht. Zwei Versuche mit jeweils 100 Messungen werden mit den in Tabelle 5.1 angegebenen Ergebnissen durchgeführt. Die Ergebnisse in den folgenden Abschnitten basieren

auf der Kalibrierung des Trackingsystems, welche mit den zwei Versuchen evaluiert wurde.

Die hier ermittelte Genauigkeit des Systems wird für die folgenden Tests als gegeben angenommen. Daher wird der Lernalgorithmus in Bezug auf die Reproduzierbarkeit der Ergebnisse ausgewertet, indem Mittel- und Maximalwerte verschiedener Durchführungen des Anlernens eines Modells verglichen werden. Ein Vergleich mit realen Distanzen oder Positionen wird nicht durchgeführt, da eine manuelle Vermessung der Distanzen zwischen sieben, beziehungsweise zehn Markern schwer mit einer höheren Präzision als mit der des Trackingsystems umzusetzen ist. Dies ist insbesondere der Fall, da die Marker auf der Vorder- und Rückseite des Displays liegen.

5.2.1 Anlernen mit unbewegtem Objekt

In den acht Versuchen dieses Abschnitts wird das Modell während des jeweiligen Lernprozesses nicht bewegt. Zwischen jedem Versuch wird das Modell jeweils neu positioniert, wobei darauf geachtet wird, dass in jedem Versuch alle sieben Marker von beiden Kameras erfasst werden können.

In allen acht Tests wird das Modell ohne Probleme angelernt. Abbildung 5.4 zeigt die graphische Oberfläche zur Steuerung des Lernalgorithmus für den fünften Versuch mit den ersten beiden Schritten der Distanzmessungen, beschrieben in Abschnitt 5.1.1, und der Erzeugung des Modells, beschrieben in 5.1.2. Nach der Auswertung der Distanzen in jeweils $n_1 = 100$ Frames wurden die Distanzen 295, 12 mm, 594, 94 mm und 337, 06 mm als Initialwerte für das Modell ausgewählt. Das Modell wurde daraufhin in 30 von 30 Frames erfolgreich getrackt. Die drei 3D-Punkte, die das daraus resultierende Dreipunktmodell formen, sind in Abbildung 5.4 unten rechts aufgeführt.

Betrachtet man bei der in Abbildung 5.4 dargestellten graphischen Oberfläche die aus den Distanzen gebildeten Intervalle auf der linken Seite, fällt auf, dass die ersten beiden Intervalle 200 Distanzen enthalten. Da aber Distanzen nur in $n_1 = 100$ Frames gemessen wurden, weist dies auf die Existenz sehr ähnlicher Distanzen in der Konstellation der Marker hin. Diese Konstellation der Marker verursacht nicht notwendigerweise Probleme bei der Detektion des Modells. Als Basis für das Anlernen des Modells werden diese Distanzen jedoch ignoriert.

Nachdem also ein Dreipunktmodell definiert ist, können weitere Punkte entsprechend Abschnitt 5.1.3 hinzugefügt werden. Abbildung 5.5 zeigt aus der Sicht einer der beiden Kameras, wie 3D-Punkte schrittweise hinzugefügt werden, bis die vier noch nicht angelernt Punkte Teil des Modells sind. Auf dem Kamerabild werden detektierte Punkte mit dem inneren Kreis und die zum Modell gehörenden Punkte mit dem äußeren Kreis überlagert.

Das Modell wird für jeden Versuch in einer anderen Pose angelernt, sodass der Ablauf des Anlernens für jeden Versuch leicht verschieden ist. Insbesondere die Reihenfolge der hinzugefügten Punkte und die lokalen Koordinatensysteme, die auf Basis der ersten drei Punkte definiert werden, unterscheiden sich in den verschiedenen Versuchen. Daher werden alle acht Modelle mit einer Rotation und einer Translation in ein gemeinsames Koordinatensystem transformiert, um die Ergebnisse vergleichbar zu machen. Die Kon-

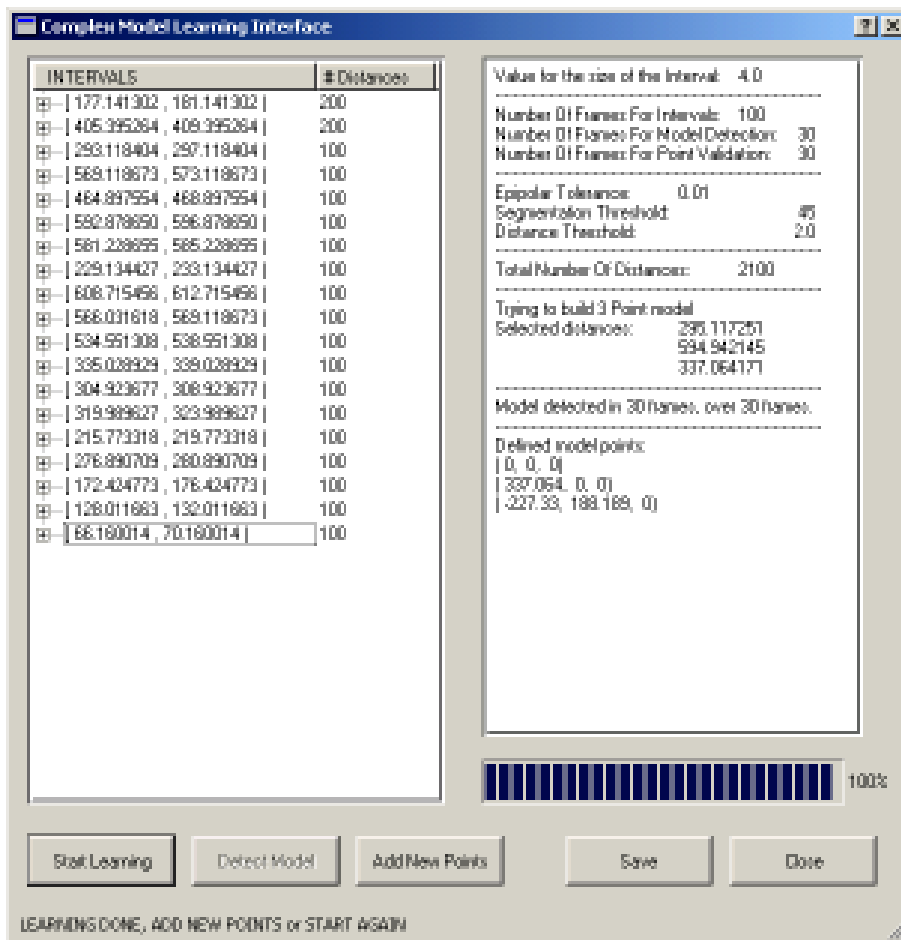


Abbildung 5.4. Die graphische Oberfläche des implementierten Lernalgorithmus zeigt Distanzmessungen (links) und die daraus abgeleitete Definition eines Dreipunktmodells (rechts).

sequenz ist, dass alle angelernten Modelle aus den acht Versuchen einen exakt gleichen Punkt $(0.0, 0.0, 0.0)^T$ besitzen, der daher auch keine Abweichungen vom Mittelwert zwischen den Versuchen aufweist. Tabelle 5.2 zeigt die Ergebnisse aus den acht Versuchen.

Die Distanzfehler zwischen den Modellpunkten entsprechen ungefähr der Größenordnung der Ergebnisse der Distanzmessungen, die in Tabelle 5.1 aufgelistet sind und als Referenz aufgenommen wurden. Die Mittelwerte der einzelnen Punkte formen wiederum eine Referenz für Vergleiche mit den Resultaten, die in den folgenden Abschnitten beschrieben werden.

5.2.2 Anlernen mit bewegtem Objekt

In den sechs Versuchen dieses Abschnitts wird das Modell langsam in großen Teilen des Interaktionsbereichs bewegt. Alle Marker können meist von beiden Kameras gleichzeitig erfasst werden. Die Bewegung selbst wird auf verschiedene Arten durchgeführt:

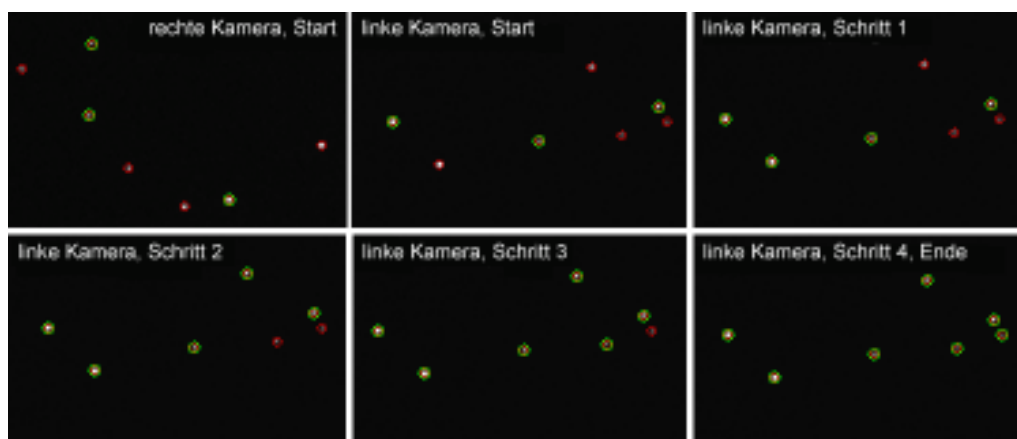


Abbildung 5.5. Schrittweises Hinzufügen neuer 3D-Punkte.

(mm)	x	y	z	Mittlerer Fehler	Maximaler Fehler
Punkt 0	0,0	0,0	0,0	0,0	0,0
Punkt 1	567,920	0,0	0,0	0,126	0,333
Punkt 2	-2,245	217,841	0,0	0,300	0,542
Punkt 3	336,964	-5,302	2,615	0,224	0,383
Punkt 4	567,506	179,297	1,722	0,274	0,381
Punkt 5	164,780	-57,034	2,104	0,317	0,531
Punkt 6	34,686	-58,350	2,183	0,301	0,462
Mittel/Max				0,220	0,542

Tabelle 5.2. Mittelwerte und mittlere, beziehungsweise maximale Distanzfehler des angelernten Modells in mm aus acht Versuchen.

Im ersten Versuch wird das Modell schnell rotiert und transliert, im zweiten wird das Modell hauptsächlich transliert und in den anderen Versuchen gleichmäßig und langsam transliert und rotiert. Tabelle 5.3 zeigt die Fehlermessungen dieser Versuche im Vergleich zu den Ergebnissen aus Abschnitt 5.2.1. Wie bereits in den vorausgegangenen Abschnitten beschrieben, werden die Punkte in ein gemeinsames Koordinatensystem transformiert und nachträglich sortiert, sodass Vergleiche zwischen den Koordinaten der Punkte aus verschiedenen Versuchen möglich sind.

Der mittlere Fehler der 3D-Punkte beträgt im Vergleich zu den Mittelwerten der Versuche mit dem unbewegten Modell gerundet 0,396 mm. Im Test mit dem unbewegten Modell wurde jeder Punkt ohne Probleme sofort akzeptiert, das heißt, er wurde in diesem Fall in mindestens 25 von 30 Frames in einer hinreichend konstanten Position relativ zum bereits angelernten Teil des Modells erfasst. In einigen Versuchen aus dieser Testreihe wurden mehrere automatisch hintereinander erfolgende Anläufe benötigt, bevor ein Punkt akzeptiert werden konnte. In den Versuchen 2 bis 4 wurden zwischen vier und sechs Anläufe benötigt, um die vier Punkte hinzuzulernen. In den Versuchen 1, 5 und 6 waren

(mm)	Versuch 1	Versuch 2	Versuch 3	Versuch 4	Versuch 5	Versuch 6
Punkt 0	0,0	0,0	0,0	0,0	0,0	0,0
Punkt 1	0,115	0,504	0,294	0,384	0,169	0,328
Punkt 2	0,914	0,664	0,110	0,554	0,064	0,361
Punkt 3	0,181	0,599	0,149	0,210	0,250	0,148
Punkt 4	0,772	0,410	0,302	0,597	0,885	0,571
Punkt 5	0,767	0,461	0,290	0,482	0,309	1,008
Punkt 6	1,120	0,519	0,356	0,167	0,511	1,171
Mittel	0,553	0,451	0,214	0,342	0,313	0,512

Tabelle 5.3. Distanzfehler für jeden Versuch und Punkt im Vergleich zum jeweiligen Mittelwert der Punkte aus Tabelle 5.3.

mehr als 20 Anläufe notwendig. Im letzten Abschnitt 5.2.1 gab es in jedem Versuch nur vier Anläufe, also genau der Anzahl der anzulernenden Punkte entsprechend. Die höhere Anzahl benötigter Anläufe ist damit zu begründen, dass während der Validierungsphase für einen neu hinzuzufügenden Punkt, siehe Abschnitt 5.1.3, das Objekt so bewegt wurde, dass das Modell oder der anzulernende Punkt teilweise nicht erkannt werden konnte. Bei den durchgeführten Tests wurden Infrarotdioden als Marker verwendet, welche nur einen beschränkten Abstrahlwinkel haben, sodass diese in bestimmten Ausrichtungen aufgrund einer zu schwachen Intensität vom Trackingsystem nicht erkannt werden können. Eine höhere Anzahl von Anläufen ist in diesen Fällen also kein Anzeichen für ein schlechtes Endergebnis des Anlernprozesses.

5.2.3 Anlernen mit bewegtem Objekt und teilweise verdeckten Markern

Nach den vorangegangenen Tests wurden drei weitere Marker auf der Rückseite des Displays angebracht, um eine Drehung des Displays in eine vertikale Stellung zu ermöglichen, sodass die Hinterseite des Displays zu den Kameras zeigt, ohne dabei die Verfolgung des Displays aufgeben zu müssen, siehe Abbildung 5.6. Auf diese Weise kann das Display nicht nur über einem Patienten benutzt, sondern auch an dessen Seite platziert werden. Bei dieser Konstellation der Marker ist es nicht möglich, alle zehn Marker zu einem Zeitpunkt mit beiden Kameras zu erfassen.

Analog zu der Beschreibung in Abschnitt 5.2.2 werden sechs Versuche mit den zehn Markern durchgeführt. Im ersten und im zweiten Versuch wird das Anlernen der Marker auf der Rückseite des Displays begonnen, womit in der Anfangsphase im Wesentlichen die drei neuen Marker zu sehen sind. Während der Phase des Anlernens zusätzlicher Punkte, siehe Abschnitt 5.1.3, wird das Display langsam gedreht, sodass das bisher angelegte Modell am Display verfolgt werden kann und gleichzeitig schrittweise noch nicht hinzugefügte Marker in den Sichtbereich der Kameras gelangen.

Die Versuche drei und vier werden in umgekehrter Abfolge durchgeführt, das initiale Modell wird also auf der Vorderseite des Displays angelernt, um dann das Display so zu drehen, bis alle Marker, insbesondere auch auf der Rückseite, erfasst werden können.

Im Gegensatz zu den ersten vier Versuchen, bei denen das Display gleichmäßig bewegt

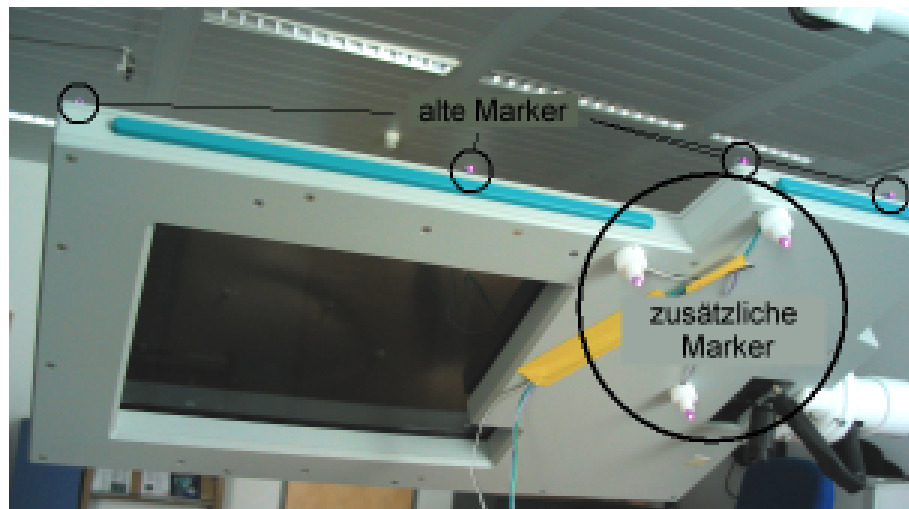


Abbildung 5.6. Drei weitere Marker sind auf der Rückseite des Displays angebracht. Im Bild sind vier der sieben alten Marker sichtbar, drei sind verdeckt.

wird, wird in den Versuchen fünf und sechs das Display nur jeweils einmal bewegt. Wie in den Versuchen drei und vier werden die Marker auf der Vorderseite des Displays für die initiale Anlernphase genutzt. Das Display wird nicht bewegt, bis alle sieben Marker auf der Vorderseite angelernt wurden. Daraufhin wird der Lernprozess gestoppt und das Display wird so positioniert, dass alle Marker auf der Rückseite, sowie mindestens drei Marker der Vorderseite, von beiden Kameras erfasst werden können. In dieser Stellung wird der Lernprozess fortgesetzt.

Wie in den vorausgegangenen Abschnitten werden die angelernten Punkte der verschiedenen Versuche in ein gemeinsames Koordinatensystem transformiert und danach so sortiert, dass die ersten sieben Punkte des Zehnpunktmodells den sieben Punkten aus den anderen Abschnitten entsprechen und mit ihnen verglichen werden können. Das Ergebnis dieses Vergleichs ist in Tabelle 5.4 aufgeführt. Die Qualität der Ergebnisse des Anlernens der sieben Punkte, die mit den Punkten aus den vorherigen Tests korrespondieren, ist im Vergleich zu den vorigen Versuchen ähnlich.

Tabelle 5.5 zeigt die Ergebnisse der sechs Versuche mit jeweils zehn Markern ohne Vergleich zu vorherigen Tests. Offensichtlich sind die Ergebnisse für die zusätzlich angebrachten Marker schlechter als für die ersten sieben, insbesondere das Ergebnis für Punkt 9 hebt sich von den anderen ab.

Wirft man einen genaueren Blick auf die Ergebnisse der einzelnen Versuche in Tabelle 5.6, dann weisen Versuch 3 und Versuch 5 eine stärkere Abweichung im Vergleich zu den anderen Versuchen auf.

Bei den größeren Fehlerwerten in vereinzelt Versuchen sind zwei Umstände zu beachten: Erstens die Abstrahlcharakteristik der verwendeten Infrarotdioden und zweitens die Art, wie die Marker platziert wurden. Infrarotdioden emittieren Licht nicht gleichmäßig in alle Richtungen, insbesondere strahlen die verwendeten Dioden nicht kugelförmig

(mm)	Versuch 1	Versuch 2	Versuch 3	Versuch 4	Versuch 5	Versuch 6
Punkt 0	0,0	0,0	0,0	0,0	0,0	0,0
Punkt 1	0,245	0,186	0,270	0,304	0,376	0,200
Punkt 2	0,614	0,221	0,506	0,212	0,311	0,267
Punkt 3	0,140	0,267	0,162	0,443	0,380	0,395
Punkt 4	1,149	0,287	0,195	0,291	0,352	0,234
Punkt 5	0,730	0,766	0,185	0,319	0,425	0,225
Punkt 6	0,587	0,824	0,171	0,377	0,430	0,203
Mittel	0,495	0,364	0,213	0,280	0,325	0,218

Tabelle 5.4. Distanzfehler für jeden Versuch und die ersten sieben Punkte des Modells im Vergleich zu den Mittelwerten der Punkte aus Tabelle 5.2.

(mm)	x	y	z	Mittlerer Fehler	Maximaler Fehler
Punkt 0	0,0	0,0	0,0	0,0	0,0
Punkt 1	568,102	0,0	0,0	0,142	0,427
Punkt 2	-2,248	217,970	0,0	0,287	0,742
Punkt 3	337,072	-5,280	2,682	0,277	0,389
Punkt 4	567,496	179,345	1,924	0,418	0,953
Punkt 5	164,763	-57,117	2,123	0,435	0,766
Punkt 6	34,774	-58,326	2,317	0,387	0,816
Punkt 7	151,271	-50,064	63,948	0,682	1,020
Punkt 8	227,946	5,626	64,547	0,876	1,688
Punkt 9	101,802	141,326	68,114	3,309	6,692
Mittel/Max				0,681	6,692

Tabelle 5.5. Mittelwerte und mittlere, beziehungsweise maximale Distanzfehler des angelernten Modells in mm aus acht Versuchen.

ab, sondern nur in einem begrenzten Winkel. Daher verursachen Infrarotdioden in ungünstigeren Positionen mehr Probleme als beispielsweise passive retroreflektierende Marker. Die drei zusätzlichen Marker sind um mehr als 90° rotiert im Vergleich zu den bereits platzierten sieben Markern, was aufgrund der obigen Erläuterungen größere Abweichungen bei den Messungen durch das Trackingsystem erzeugt. Vermeidet man diese eher problematischen Konstellationen durch eine andere Platzierung der aktiven Marker beziehungsweise durch die Verwendung passiver Marker, liefert die iterative Lernprozedur gute Ergebnisse.

5.3 Definition lokaler Koordinatensysteme

Beim Anlernen eines neuen Modells wird das lokale Koordinatensystem (LKS) des Modells - welches auch als Modellkoordinatensystem bezeichnet wird - zunächst wahllos

(mm)	Versuch 1	Versuch 2	Versuch 3	Versuch 4	Versuch 5	Versuch 6
Punkt 7	0,840	0,916	0,716	1,020	0,201	0,395
Punkt 8	0,842	0,404	0,567	1,273	1,688	0,480
Punkt 9	0,499	2,301	6,692	1,559	6,619	2,182
Mittel	0,727	1,208	2,659	1,284	2,836	1,020
Punkt 7	0,696	0,982		0,991		0,400
Punkt 8	0,340	0,679		0,749		0,618
Punkt 9	0,781	1,94		1,624		2,275
Mittel	0,606	1,202		1,122		1,098

Tabelle 5.6. Distanzfehler der drei zusätzlich angebrachten Marker: in der oberen Hälfte wurde der Vergleich zum Mittel einschließlich Versuch 3 und 5 berechnet. In der unteren Hälfte wurde der Vergleich zum Mittel ohne die beiden Versuche berechnet.

definiert. Wie in Gleichung (5.1) beschrieben, wird der Ursprung in einen der drei Punkte des initialen Modells gelegt. Die Orientierung des Modells wird definiert, indem die x -Achse vom Ursprungspunkt durch einen weiteren der drei Punkte gelegt wird und der verbleibende dritte Punkt als Punkt in der xy -Ebene gespeichert wird. Damit ist auch die z -Achse eindeutig definiert, woraus sich dann auch die y -Achse in dem euklidischen Koordinatensystem ableiten lässt.

Auf der graphischen Oberfläche des Anlernalgorithmus in Abbildung 5.4 sind die Punkte

$$X_M^1 = \begin{pmatrix} 0,0 \\ 0,0 \\ 0,0 \end{pmatrix}, \quad X_M^2 = \begin{pmatrix} 337,06 \\ 0,0 \\ 0,0 \end{pmatrix} \quad \text{und} \quad X_M^3 = \begin{pmatrix} -227,33 \\ 188,189 \\ 0,0 \end{pmatrix} \quad (5.2)$$

abgebildet, die auf diese Weise aus den Distanzen 295,12 mm, 594,94 mm und 337,06 mm als Modellpunkte bestimmt und durch Verfolgung über einen Zeitraum von $n_2 = 30$ Frames validiert wurden.

Sind die drei Punkte so angeordnet, dass sie ein rechtwinkliges beziehungsweise nahezu rechtwinkliges Dreieck formen, bietet es sich an, den Ursprung in den Punkt zu legen, der die Ecke mit dem rechten Winkel bildet. Ansonsten ist die Definition des LKS beim Anlernen weitestgehend wahllos und muss vom Benutzer später neu definiert werden.

Für die Definition eines neuen LKS bieten sich je nach Objekt, welches mit Hilfe des an ihm angebrachten Markermodells verfolgt werden soll, unter anderem folgende Möglichkeiten an:

- Bestimmung des LKS durch einen Kalibrierungsvorgang
- Bestimmung des LKS durch freie Platzierung relativ zu den Punkten des Modells

Umsetzungen für diese beiden Methoden werden in den folgenden beiden Abschnitten näher erläutert.

5.3.1 Bestimmung eines lokalen Koordinatensystems durch Kalibrierung

Ein typisches Beispiel für die Notwendigkeit einer Kalibrierung eines LKS ist die Verfolgung eines medizinischen Instruments. Zum einen müssen am Instrument die Marker angebracht werden, was typischerweise am Griff geschieht, zum anderen ist meist die Spitze des Instruments der eigentliche Punkt, der verfolgt werden soll. In [Wesarg 04] wurde ein Kalibrierungsverfahren für das in Kapitel 8.1 beschriebene AR-System vorgestellt, das die Vermessung der Transformation zwischen Sensorik und Spitze eines Instruments ermöglicht. Das Kalibrierungsverfahren wird bei diesem Ansatz mit einem elektromagnetischen Sensor im Griff des Instruments angewandt, ist aber für alle Trackingsysteme, die sechs Freiheitsgrade ermitteln können, verwendbar, insbesondere auch für ein Stereokamerasystem, das Multipunktmodelle verfolgt.



Abbildung 5.7. Kalibrierung einer Brachietherapienadel mit integriertem elektromagnetischem 6DOF-Sensor. Die Instrumentenspitze ist an einem Punkt fixiert, während das Instrument in verschiedene Richtungen ausgelenkt und rotiert wird.

Abbildung 5.7 zeigt, wie bei diesem Verfahren die Spitze eines zu vermessenden medizinischen Instruments an einem Punkt fixiert wird, indem sich die Spitze in einer Vertiefung befindet. Dann wird das Instrument mit fixierter Spitze rotiert und in verschiedene Richtungen ausgelenkt, bis hinreichend viele Messungen vorhanden sind. Die Auswertung erfolgt an neun verschiedenen Fixierpunkten, die in einem quadratischen Gitter angeordnet sind. An jedem Punkt werden 500 Messwerte aufgenommen. Aus den Messwerten ergibt sich für die Berechnung einer Transformation von Sensor (altes LKS) zu Instrumentenspitze (neues LKS) ein überbestimmtes lineares Gleichungssystem, das sich mit Standardmethoden lösen lässt. Abbildung 5.8 zeigt eine Projektion der Messwerte auf die xy -Ebene vor und nach der Anwendung der berechneten Transformation.

Die berechnete Transformation ermöglicht die Verschiebung des Ursprungs des LKS an die Spitze des Instruments. Es gibt zahlreiche solcher Kalibrierungsverfahren, die in Abhängigkeit des zu kalibrierenden Objekts ausgewählt werden müssen.

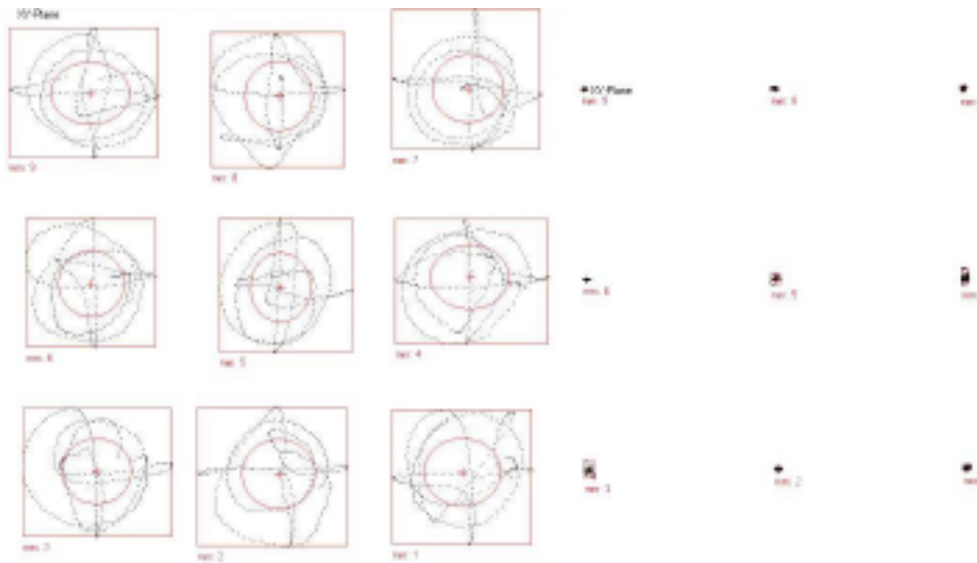
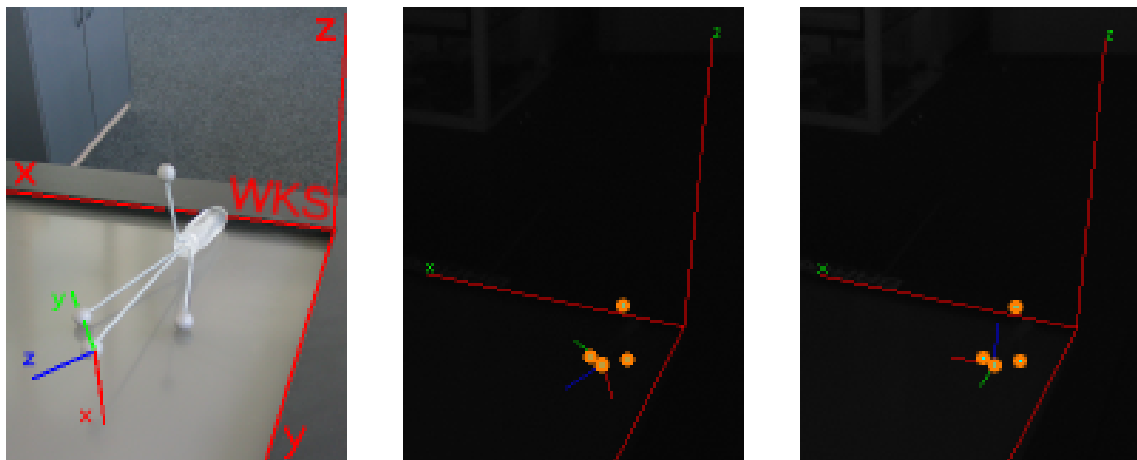


Abbildung 5.8. Aufsicht auf die Messungen mit der an neun verschiedenen Punkten fixierten Instrumentenspitze. Auf der linken Seite sind die Originalmesswerte des Sensors abgebildet, auf der rechten Seite dieselben Messungen, nachdem auf alle Messpunkte die berechnete Transformation von Sensor (altes LKS) zu Instrumentenspitze (neues LKS) angewendet wurde.

Ein einfacherer Kalibrierungsvorgang kann bei einem typischen VR-Handgerät, welches mit Hilfe eines Multipunktmodells verfolgt wird, gewählt werden, siehe Abbildung 5.9. Bei einem solchen Gerät spielt die exakte Ausrichtung und Platzierung des LKS normalerweise eine geringere Rolle, da meist nur im Virtuellen interagiert wird und damit kleinere Ungenauigkeiten, die ansonsten besonders durch Berührung mit realen Objekten auffallen, irrelevant sind. Der Kalibrierungsvorgang besteht einfach daraus, dass das Modell, dessen LKS neu definiert werden soll, so im WKS platziert wird, dass die Orientierung des WKS mit der neuen Orientierung des Modells identisch ist. Die momentane Rotation R_M^W des Modells kann durch das Trackingsystem erfasst werden. Daraufhin werden die Modellpunkte $\mathcal{X}_M = \{X_M^1, \dots, X_M^n\}$ mit R_M^W transformiert, sodass die transformierten Punkte $\hat{X}_M^i = R_M^W \cdot X_M^i$ als neue Modellpunkte gespeichert werden können. Falls zusätzlich der Ursprung des LKS verschoben werden soll, bietet sich das im folgenden Abschnitt beschriebene Vorgehen an.

5.3.2 Bestimmung eines lokalen Koordinatensystems durch freie Platzierung

Die Bestimmung eines LKS ist einfach durchzuführen, wenn es sich relativ zu den Punkten des Modells angeben lässt. Abbildung 5.10 zeigt die Implementierung einer graphischen Oberfläche zur freien interaktiven Platzierung eines LKS. Die Ausgabe der Punkte erfolgt zum einen in Form ihrer Koordinaten und zum anderen als Visualisierung in einer 3D-Umgebung, umgesetzt in einem VRML-Viewer, der eine frei wählbare Sicht des Benutzers erlaubt. Die Modellpunkte werden als Billardkugeln repräsentiert, das ak-



(a) Gewünschte Platzierung des Interaktionsgeräts im WKS. Das WKS und das aktuelle LKS sind eingezeichnet.

(b) Die gewünschte Platzierung des Interaktionsgeräts aus Sicht einer Kamera mit Infrarotfilter.

(c) Das LKS wurde geändert, indem es am WKS ausgerichtet wurde.

Abbildung 5.9. Ein Interaktionsgerät, an dem Marker fest angebracht sind, wird so im WKS platziert, dass das neue LKS daran ausgerichtet werden kann. Die Kamerabilder sind mit zusätzlichen Informationen überlagert.

tuelle LKS wird in drei Farben angezeigt: Rot für die x -Achse, Grün für die y -Achse und Blau für die z -Achse.

Eine einfache Möglichkeit der Transformation der Modellkoordinaten ist das Platzieren des Ursprungs des LKS in einem der Modellpunkte. Falls dies nicht erwünscht ist, kann der Ursprung über Linearkombinationen der Modellpunkte bestimmt werden und seine Koordinaten im noch aktuellen LKS angegeben werden. Weitaus schwieriger stellt sich zumeist die Bestimmung der Orientierung des LKS dar. Eine Rotation, die nicht auf 90° Winkeln basiert, kann von den meisten Benutzern unabhängig von der Repräsentation der Rotation, ob in Euler Winkeln oder als Achse und Drehwinkel, nicht direkt angegeben werden.

Sollen die Achsen zumindest teilweise durch die Modellpunkte verlaufen, lässt sich die Rotation einfach angeben, indem zwei Punkte, ein Start- und ein Endpunkt, für die Richtung der x -Achse angegeben werden. Daraufhin wird eine vorläufige zweite Achse, also y - oder z -Achse bestimmt, wieder mit einem Start- und einem Endpunkt gewählt. Diese dient nur dazu, die dritte Achse orthogonal zu den anderen zu bestimmen. Automatisch wird dann aus der dritten Achse und der ersten Achse die zweite Achse so korrigiert, dass alle Achsen zueinander orthogonal sind. Auf diese Weise ist die Orientierung des LKS vollständig definiert.

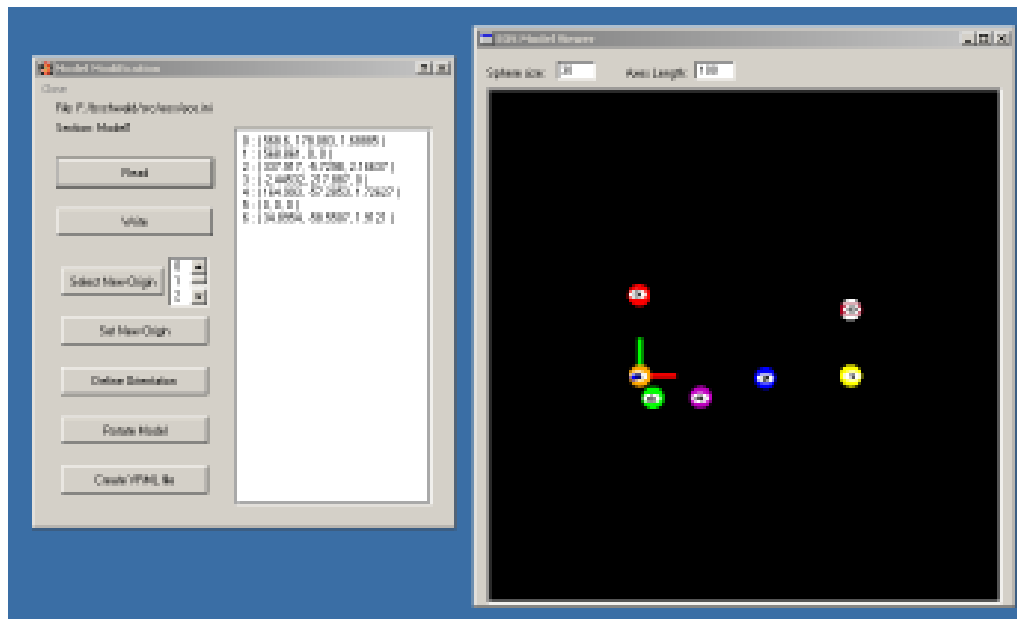


Abbildung 5.10. Graphische Oberfläche für die interaktive, freie Platzierung eines lokalen Koordinatensystems relativ zu den Modellpunkten.

5.4 Zusammenfassung

Insgesamt stellt der hier beschriebene Lernalgorithmus für starre Multipunktmodelle ein essentielles Element eines Trackingsystems dar, insbesondere wenn ein Stereokamerasystem mit nur wenigen Kameras benutzt werden soll und dabei komplexere Modelle, mit mehr als vier oder fünf Markern, die beispielsweise auf gegenüberliegenden Seiten eines Objekts angebracht werden, eingesetzt werden sollen. Ein installiertes und kalibriertes Trackingsystem vorausgesetzt, ist das Anlernen eines neuen Modells zusammen mit der in Abschnitt 5.3 beschriebenen Transformation in ein lokales Koordinatensystem ein Vorgang, der innerhalb weniger Minuten durchzuführen ist.

Eine grundlegende Idee des Lernalgorithmus ist die Verwendung des Detektionsalgorithmus aus Abschnitt 4.3.3 beim Anlernen eines Modells für die Validierung und das sukzessive Hinzufügen neuer Punkte zum Modell.

Eine Voraussetzung für den Erfolg des Lernalgorithmus ist eine geeignete Platzierung der Marker an einem Objekt. Es ist notwendig, dass es keinen Marker gibt, der in allen Positionierungen des Modells in beiden Kameras gleichzeitig zusammen mit nur zwei weiteren Markern erkannt werden kann. Mindestens vier Marker müssen während des Hinzufügens neuer Punkte eines Modells zu erkennen sein, um das bereits angelegte Modell identifizieren und registrieren zu können (mindestens drei Marker sind hierfür notwendig) und um einen neuen Punkt hinzufügen zu können. In Abhängigkeit der Art von Markern, zum Beispiel aktiven Dioden oder passiven retroreflektierenden Kugeln, kann deren Platzierung einen Einfluss auf die Genauigkeit des Trackings während der Lernphase haben, wie in Abschnitt 5.2 beschrieben.

Kapitel 6

Dynamische Modelle

Der Begriff **Dynamisches Modell** (DM) ist im Kontext dieser Arbeit als Gegensatz zu einem starren Modell zu verstehen. Die entscheidende Eigenschaft der starren Modelle und insbesondere der Multipunktmodelle ist, dass die Punkte, die ein Modell bilden, relativ zueinander fest positioniert sind. Bei dynamischen Modellen bilden die Marker hingegen keine feste Geometrie. Dynamische Modelle bestehen physikalisch ebenfalls aus einer festen Anzahl von $n \in \mathbb{N}$ Markern und deren Repräsentation als Punkte X_D^1, \dots, X_D^n . Ein dynamisches Modell kann zwar temporär durch die Koordinaten der Punkte des Modells angegeben werden, aber nicht dadurch vollständig beschrieben werden. Dementsprechend müssen andere Modellbeschreibungen gefunden werden.

Typische Beispiele für dynamische Modelle bilden Handmodelle oder Körpermodelle. Abbildung 6.1 zeigt mit Markern besetzte Handschuhe zur Verfolgung von Handpositionen und Erkennung von Gesten (beispielsweise Zeigerichtung). Der typische Ansatz für die Verfolgung der Hand mit Hilfe eines solchen Handschuhs beruht auf der Gruppierung von Markern und deren Analyse [Schwald 05b]. Diese Trackingtechnologie wird in dieser Arbeit als Gruppenmodell bezeichnet und in Abschnitt 6.1 vertieft.

Während sich Handmodelle je nach Gesten auch mit einer geringen Anzahl von Markern erzeugen lassen, sind Menschmodelle weitaus aufwändiger und bedürfen einer großen Anzahl von Markern. Diese Technik zur Verfolgung menschlicher Bewegungen wird als (Human) Motion Capturing bezeichnet und wird insbesondere in der Tricktechnik bei Filmen und für Animationen in Computerspielen eingesetzt. Hier wird meist mit einer großen Anzahl von Kameras in speziellen Umgebungen gearbeitet, wobei eine vollständige Echtzeitverarbeitung eine geringere Rolle spielt, da nachgearbeitet werden kann. Innerhalb dieser Arbeit ist im Gegensatz dazu die Verarbeitung in Echtzeit von besonderem Interesse.

Dynamische Modelle für menschliche Körper oder deren Gliedmaßen, sowie für Roboterarme oder Ähnliches haben eine Gemeinsamkeit: sie können über Gelenke definiert werden. Die Gruppe der Gelenkmodelle wird in Abschnitt 6.2 kurz beschrieben.

Neben markerbasierten Ansätzen zur Verfolgung menschlicher Körper(-teile) werden auch immer mehr markerlose Ansätze verfolgt. Aktuelle Ansätze sind unter anderem in [Malerczyk 04] und [Stenger 04] beschrieben. Der Vorteil, der durch den Verzicht auf Marker entsteht, steht meist den Nachteilen einer geringeren Präzision und Robustheit,

beziehungsweise eines sehr eingeschränkten Interaktionsbereichs gegenüber. Besonders interessant sind markerlose Ansätze bei Kopfbewegungen [Yang 01], da eine Platzierung von Markern am Kopf sehr schlecht umzusetzen ist, soweit nicht andere Objekte, wie Kappen oder Brillen, als Markerträger verwendet werden können.



Abbildung 6.1. Zwei Handschuhe mit je drei angenähten Markern für eine Verfolgung mit Hilfe Dynamischer Modelle. Die Marker am Mittelfinger sind an den beiden Handschuhen auf unterschiedliche Weise angebracht.

Neben den bereits erwähnten Arten von Modellen wird in Abschnitt 6.3 der Typ der dynamischen Flächenmodelle beschrieben. Bei diesen Modellen wird die Menge der Marker als eine Beschreibung für eine Oberfläche aufgefasst. Das Kapitel schließt in Abschnitt 6.4 mit einer Diskussion über die Möglichkeiten, dynamische und starre Modelle innerhalb einer Anwendung zu kombinieren.

Im Gegensatz zu den Multipunktmodellen besitzen dynamische Modelle keine entsprechend eindeutigen geometrischen Eigenschaften. Die in Abschnitt 4.1.3 beschriebene Verfolgung mehrerer einzelner Marker ist als Vorverarbeitungsschritt für die Verfolgung von dynamischen Modellen im Allgemeinen sinnvoll. Bei der Verfolgung der einzelnen Marker erfolgt eine Zuordnung von Identifikationsindizes. Dennoch muss auch bei der Berücksichtigung von Identifikationsindizes der einzelnen Marker damit gerechnet werden, dass Indizes nach Markerverdeckungen neu oder falsch zugewiesen werden, oder dass Marker zeitweise nicht vorhanden sind. Dies erfordert wie bei Multipunktmodellen eine Redundanz in den Modellen oder, wie insbesondere bei den Zweipunktmodellen, eine zeitweise Verfolgung mit unzureichenden Informationen und höherem Fehlerrisiko.

6.1 Dynamische Gruppenmodelle

Ein in dieser Arbeit sogenanntes dynamisches Gruppenmodell zeichnet sich in erster Linie dadurch aus, dass die dynamisch platzierten Marker des Modells stets in Relation zum Interaktionsvolumen dicht beieinander platziert sind und daher im Allgemeinen durch Gruppierung aller in einer Szene vorhandener Marker identifiziert werden können. Gelingt keine eindeutige Identifikation, sodass mehrere Gruppen für das tatsächliche Modell in Frage kommen, werden weitere Eigenschaften, wie der letzte Identifikationsindex der einzelnen Marker, oder prädierte Daten verwendet.

Der Algorithmus für die Erkennung von dynamischen Gruppenmodellen gliedert sich in drei Schritte. Nach der Gruppierung der Marker und der Identifikation der Gruppen folgt die Interpretation der Punktconstellation innerhalb der Gruppen, um einen Zustand erkennen zu können, siehe Abbildung 6.2. Wird das dynamische Gruppenmodell zur Verfolgung einer Hand eingesetzt, ist ein solcher Zustand eine Geste. Wird nur ein dynamisches Gruppenmodell verfolgt und können Störungen ausgeschlossen werden, können Gruppierung und Identifikation übersprungen werden. Die konkrete Umsetzung für ein dynamisches Gruppenmodell wird in Abschnitt 6.1.1 für die menschliche Hand beschrieben.

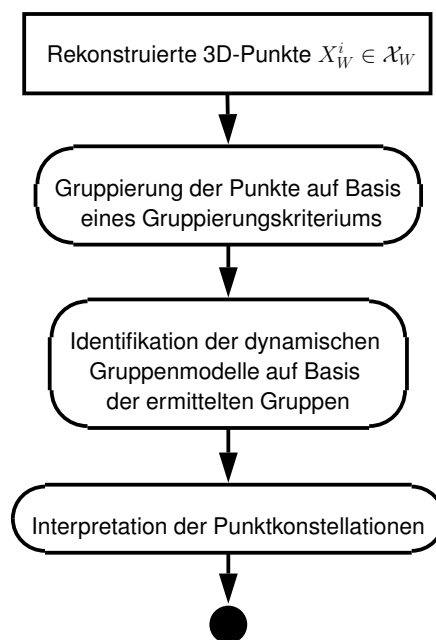


Abbildung 6.2. Phasen für die Verfolgung und Erkennung von dynamischen Gruppenmodellen.

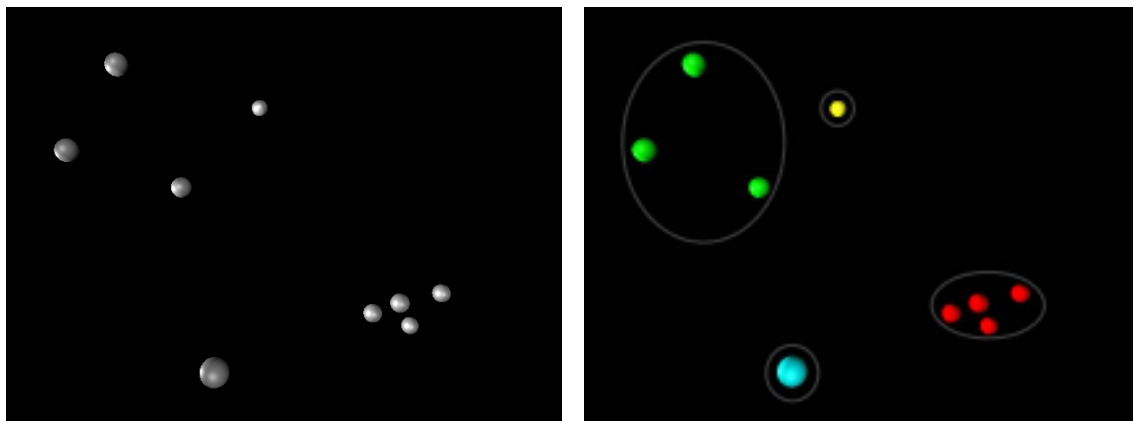
Um dynamische Gruppenmodelle zu erkennen, wird also zunächst die Menge aller getrackten Punkte $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$ in (disjunkte) Untermengen, hier als Gruppe bezeichnet, unterteilt. Die Zugehörigkeit eines Punktes zu einer Gruppe erfolgt durch das Setzen des Klassifikationsindex des Punktes. Bereits klassifizierte Punkte, beispielsweise

durch eine vorausgegangene Erkennung von Multipunktmodellen, können dabei von der Gruppierung ausgeschlossen werden.

Das Kriterium für die Gruppierung kann unterschiedlich gewählt werden. Allerdings steht zumindest bei jedem initialen Suchvorgang nach dem Modell, also wenn noch keine Verfolgung stattgefunden hat oder das Modell nach einer Verfolgung wieder verloren wurde, aufgrund der geringen zur Verfügung stehenden Information, nämlich einer Menge nicht unterscheidbarer 3D-Punkte, nur eine geringere Auswahl an Kriterien zur Verfügung. Zwei naheliegende Kriterien sind:

- Für jeden Punkt X_D einer Gruppe gibt es mindestens einen Punkt X'_D , sodass $|X_D - X'_D| < d$.
- Für beliebige Punkte X_D, X'_D einer Gruppe gilt $|X_D - X'_D| < d$.

Dabei ist der Parameter $d \in \mathbb{R}$ ein der Gruppe fest zugeordneter Wert. Das erste Kriterium ist relativ schwach und kann, je nachdem wie groß d gewählt wurde, dazu führen, dass Gruppen mit sehr vielen Elementen gebildet werden. Beispielsweise würden Marker, die sich alle auf einer Linie befinden, aber dabei zwei benachbarte Marker immer einen geringeren Abstand als d haben, als eine Gruppe erkannt werden. Im Gegensatz dazu ist das zweite Kriterium sehr stark. Wird hier der Wert von d zu klein gewählt, bildet im Extremfall jeder Punkte eine eigene Gruppe. Abbildung 6.3 zeigt eine Gruppierung von Punkten nach dem zweiten Kriterium mit einem sinnvoll gewählten d .



(a) Ungruppierte Menge von 3D-Punkten.

(b) Die Menge nach der Gruppierung. Punkte einer Gruppe tragen dieselbe Farbe.

Abbildung 6.3. Gruppierung von 3D-Punkten aufgrund der räumlichen Distanzen zwischen ihnen. Die Punkte werden durch die Kugeln repräsentiert.

Wie eingangs erwähnt, kann eine Verfolgung von einzelnen Markern, die die Detektion der dynamischen Gruppenmodelle vorbereitet, sinnvoll genutzt werden. Jedem Marker wird dabei ein Identifikationsindex zugewiesen, der die Suche nach der richtigen Gruppe stark vereinfacht, indem

- entweder eine Gruppe zunächst aus den Punkten zusammengesetzt wird, die den Index der Punkte aus der letzten Detektion tragen, und dann die Punkte auf das Gruppenkriterium überprüft werden, oder
- die Identifikationsindizes der Punkte einer Gruppe nachträglich mit denen aus der letzten Detektion verglichen werden, um sicherzustellen, dass es sich um die richtige Gruppe handelt.

In beiden Fällen spielt der Schwachpunkt der Identifikation der einzelnen Marker, nämlich dass dicht beieinander liegende Punkte nicht unterschieden werden können und daher Indexvertauschungen wahrscheinlicher werden, für die Identifikation der Gruppe eine untergeordnete Rolle. Der Grund dafür ist, dass dicht beieinander liegende Punkte ohnehin in einer Gruppe angenommen werden.

Insgesamt stellt das Tracking der dynamischen Gruppen eine Methode dar, die im Vergleich zu festen Multimarkermodellen unpräziser und weniger robust ist. Allerdings kann sie auch als Grundlage für andere Formen der Interaktion genutzt werden, wie in den folgenden Abschnitten dargestellt wird. Auf Basis der bisher angestellten Überlegung kann nun eine Definition dynamischer Gruppenmodelle erfolgen.

Definition 7 *Ein dynamisches Gruppenmodell besteht aus den folgenden Parametern:*

- $n \in \mathbb{N}$ Anzahl der zum Modell gehörenden Punkte,
- $d \in \mathbb{R}^+$ einer maximalen Distanz zwischen 3D-Punkten, zusammen mit einem entsprechenden Gruppierungskriterium.

Darüber hinaus kann es weitere Parameter geben, die entweder zu einer verbesserten Detektion beitragen oder aber auch die Erkennung von Zuständen eines dynamischen Gruppenmodells erlauben.

Bei einem Handmodell sind solche Zustände beispielsweise bestimmte Gesten (zum Beispiel Zeigen, Greifen oder Loslassen). Diese weiteren Parameter sind meist applikationsabhängig. Eine mögliche Variation dieses Modells besteht darin, d dem Zustand entsprechend dynamisch anzupassen.

6.1.1 Handgruppenmodelle

Die Verfolgung von Hand- beziehungsweise Fingerbewegungen stellt ein sinnvolles Anwendungsgebiet für dynamische Gruppenmodelle dar. Die Verfolgung beziehungsweise Deutung von Gesten ist insbesondere im VR- und AR-Bereich interessant, darüber hinaus auch für Ergonomieuntersuchungen oder zum spielerischen Erlernen von Zeichensprache, beispielsweise für Stumme. Eine allgemeine Betrachtung von 3D-Interaktion findet sich in [Bowman 00], sowie speziell die Interpretation von Handgesten für die Mensch-Maschine-Interaktion in [Pavlovic 97]. Im Folgenden werden vorhandene Ansätze präsentiert und ein eigener Ansatz zur Verfolgung der Hand und zur Deutung verschiedener Gesten näher erläutert.

In [Rusdorf 04a] wird das Tracking der menschlichen Hand zur Bedienung virtueller Steuerelemente wie einem Drehrad (Jog-dial) oder einem Schieberegler (Slider) eingesetzt. Verwendet wird ein Modellansatz mit drei Punkten. Die drei Marker sind dabei

Fingerhüte, die mit retroreflektierendem Material überzogen sind. Das Gruppierungskriterium ist, dass sich alle drei Punkte innerhalb einer Kugel vordefinierter Größe befinden müssen. Eine Identifikation einzelner Finger findet über Annahmen über die einzelnen Distanzen statt. In [Rusdorf 04b] wird weiter ausgeführt, wie sich die aus der Verfolgung gewonnene Information für eine kontextabhängige Steuerung in immersiven VR-Umgebungen verwenden lässt.

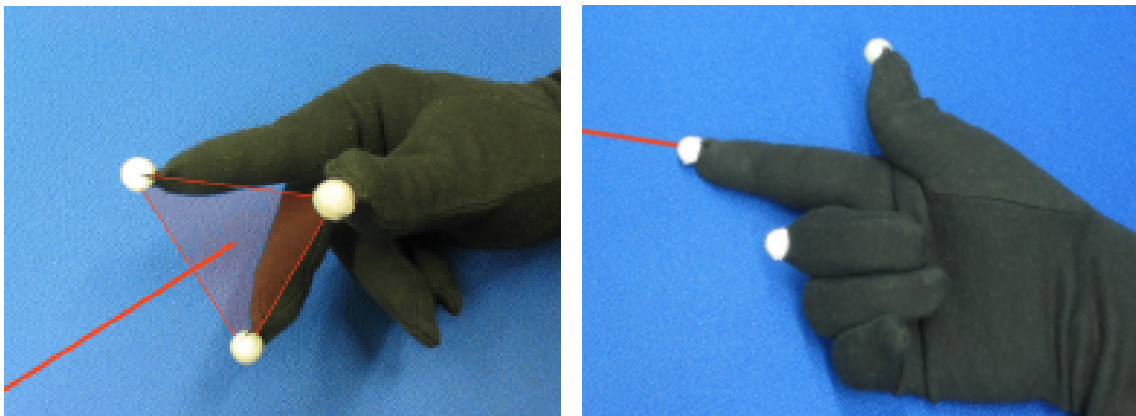
Während in [Rusdorf 04b] sowohl statische als auch dynamische Gesten beschrieben sind, findet in [Kim 04a] und [Kim 04b] für die Modellierung von 3D-Objekten eine Beschränkung auf statische Gesten statt. Statt eines Infrarottrackingsystems basiert der dort beschriebene Ansatz auf einem Zweikamerasystem mit einer Schwarzlichtausleuchtung, die Marker sind Fingerhüte aus weißem Papier. Es werden vier Marker an vier verschiedenen Fingern verwendet, um vier verschiedene Gesten ausdrücken zu können: Greifen, Rotieren, Zeigen und Pausieren.

Bei dem im Rahmen dieser Arbeit entwickelten Ansatz wird zunächst eine Gruppierung durchgeführt, wie sie weiter oben beschrieben wurde. Die Gruppierung erlaubt insbesondere auch eine Erkennung von zwei Händen und ermöglicht die Kombination mit anderen Trackingalgorithmen, wie die Erkennung eines einzelnen Punktes für die Kopfposition, siehe Abschnitt 4.1.1. Als Gruppierungskriterium wird folgende Regel verwendet: Für beliebige Punkte X_D, X'_D einer Gruppe gilt $|X_D - X'_D| < d$, wobei aus praktischen Erfahrungen $d = 150$ mm gewählt werden kann. Ziel ist es, eine Zeigegeste zu erfassen, die auf drei Markern und einer Handhaltung, wie in Abbildung 6.4(a) dargestellt, beruht:

1. Nach Durchführung der Gruppierung ist die Menge $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$ der getrackten Punkte in p disjunkte Untermengen $\mathcal{X}_{U_1}, \dots, \mathcal{X}_{U_p}$ unterteilt.
2. Daraufhin werden die Untermengen \mathcal{X}_{U_i} hinsichtlich der Anzahl ihrer Elemente untersucht. Gibt es nur eine Untermenge mit drei Elementen wird zunächst davon ausgegangen, dass dies die richtige Untermenge ist.
3. Gibt es keine Untermenge mit drei Elementen, gilt das dynamische Modell als nicht erkannt. Bei mehr als einer Untermenge mit drei Elementen wird die letzte, beziehungsweise eine prädierte, Position des Handmodells als Kriterium hinzugenommen.
4. Aus den drei Markern, die das dynamische Handgruppenmodell bilden, wird eine Zeigerichtung berechnet, indem eine Ebene zwischen den Markern aufgespannt wird und die zur Ebene orthogonale Gerade als Zeigerichtung berechnet wird, siehe Abbildung 6.4(a).

6.1.2 Interaktion durch Handgruppenmodelle

Die im letzten Abschnitt beschriebene neue Art der Bestimmung einer Zeigerichtung hat gegenüber der ansonsten üblichen Art eine Zeigegeste zu interpretieren verschiedene Vorteile. Ein spezieller Handschuh für die konventionelle Zeigegeste, mit ausgestrecktem



(a) Zeigegeste basierend auf der zur Ebene durch die drei Marker orthogonalen Gerade.

(b) Handschuh für verbreitete Art der Zeigegeste: entlang des Zeigefingers.

Abbildung 6.4. Ein auf verschiedene Arten mit drei Infrarotmarkern besetzter Handschuh für Mensch-Maschine-Interaktion. Die virtuelle Zeigerichtung und deren Erzeugung sind mit rot eingezeichnet.

Zeigefinger und möglicherweise abgespreiztem Daumen, ist in Abbildung 6.4(b) dargestellt. Ein Vorteil der Zeigegeste durch Ebene und Orthogonale (Abbildung 6.4(a)) ist, dass die Erzeugung der Zeigerichtung für den Benutzer weniger missverständlich ist. Im Gegensatz dazu gibt es bei der konventionellen Zeigerichtung häufig Irritationen, ob die Zeigerichtung entlang des Zeigefingers oder durch das Auge des Anwenders und seine Fingerspitze verläuft. Ein weiterer Vorteil ist, dass die Ebene durch die Finger durch gezielte Veränderung der Fingerstellung (insbesondere des Daumens) verändert werden kann und somit gezielt Veränderungen der Zeigerichtung vorgenommen werden können, ohne dabei die Position der Hand stark verändern zu müssen. Bei zu großen Abständen zum Bereich, auf den gezeigt werden soll, kann diese Sensitivität allerdings auch als störend empfunden werden.

Da die Sensitivität stark zunimmt, je dichter die Fingerspitzen beieinander sind, und im Extremfall, wenn die Marker sich berühren, kaum mehr auseinander zu halten sind, kann die Zeigegeste auch durch einen minimalen Abstand zwischen den Punkten eingeschränkt werden. Das bedeutet, ein zusätzlicher Parameter $d_{min} \in \mathbb{R}^+$ wird dem dynamischen Gruppenmodell zugefügt, der eine Zeigegeste nur dann als gültig zulässt, wenn für beliebige Punkte X_D, X'_D einer Gruppe $|X_D - X'_D| \geq d_{min}$ gilt.

Neben der Zeigegeste ist für die meisten Anwendungen mindestens eine weitere Geste erforderlich, die ein Ereignis auslösen kann, sodass in Kombination mit der Zeigegeste Selektion und/oder Manipulation von Objekten ermöglicht wird. Insbesondere wenn die zweite Hand, siehe Abschnitt 6.1.3, für diese, im Folgenden Selektionsgeste genannte, Interaktion genutzt wird, bietet sich ein einfaches Konzept an: das Öffnen und wieder Schließen der Hand. Abgebildet auf die Bedienung einer Maus als Eingabegerät entspricht

das Öffnen der Hand dem Drücken einer Maustaste, das geöffnet Halten dem gedrückt Halten der Maustaste und das wieder Schließen dem Loslassen der Maustaste. Wird nur eine einhändige Interaktion umgesetzt, kann diese statische Geste als dynamische Geste erweitert werden. Eine dynamische Geste ist im Gegensatz zu einer statischen Geste über Bewegungsabläufe definiert und kann nicht zu einem Zeitpunkt, sondern nur in einem Zeitintervall, erkannt werden. In diesem Fall entspricht der Mausklick dann beispielsweise der Bewegung eines schnell hintereinander ausgeführten Schließens und Öffnens.

Je nachdem welche Komplexität der Interaktion von einer entsprechenden Anwendung gefordert wird, muss das Interaktionskonzept angepasst werden. Sind mehrere Gesten erforderlich, um verschiedene Aktionen auszulösen, so kann eine jeweils unterschiedliche Anzahl von Markern für jede Geste genutzt werden oder auch bestimmte Konstellationen von mehreren Markern sowie dynamischen Gesten, wie beispielsweise die in [Rusdorf 04b] beschriebene V-förmige oder kreisförmige Bewegung.

6.1.3 Interaktion mit zwei Händen

Die Zeigegeste als eigenständige Interaktion lässt sich für einfache Interaktionen nutzen, für viele VR-Anwendungen sind jedoch weitere Interaktionsmuster, wie eine Selektion oder eine Manipulation, erforderlich. Eine Selektion lässt sich durch eine einfache dynamische Geste realisieren, indem an der zu selektierenden Stelle oder dem zu selektierenden Objekt eine gewisse Zeitspanne verweilt wird. Um weitere Interaktionsmuster umsetzen zu können, müssen weitere Handgesten definiert werden. Dies kann wie in den oben erwähnten Ansätzen mit einer Hand implementiert werden oder muss, wie hier bevorzugt, durch die zweite Hand erfolgen.

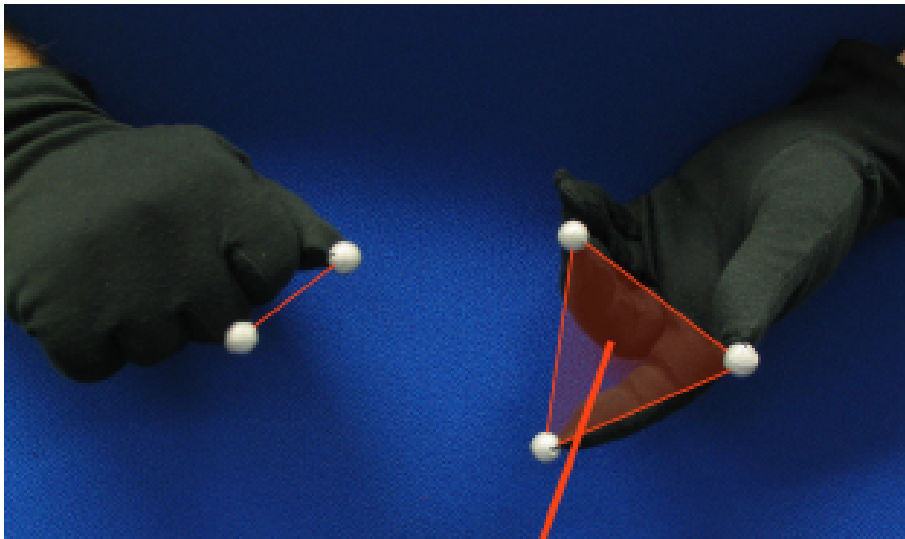


Abbildung 6.5. Interaktion mit zwei Handschuhen.

Eine beidhändige Interaktion wurde auf Basis von elektromagnetischem Tracking unter anderem von [Nishino 98] und [Grossman 01] umgesetzt, wobei bei elektromagnetischen Systemen die unterschiedlichen Sensoren automatisch unterscheidbar sind. In

[Fiorentino 03] wurde die Präzision von Anwendern bei 3D-Positionierungsaufgaben unter Einsatz beidhändiger Interaktion mit einem optischen Trackingsystem und starren Modellen untersucht. Ein System mit einem durch Multipunktmodelle optisch verfolgten Stift und einem Würfel für die beidhändige Interaktion, wurde in [van Liere 04] präsentiert. Die Firma A.R.T. [ART 06a] bietet seit kurzem ein System zur Verfolgung von zwei Händen und jeweils drei Fingern an. Dabei werden zur Identifikation der Hände starre und für die Finger dynamische Modelle eingesetzt. Als Marker werden aktive Infrarotdioden verwendet.

Für Anwendungen, bei denen es ausreicht, ein Touch-Panel zu verwenden, bieten neuere Ansätze die Möglichkeit an Stelle von nur einem, zahlreiche Druckpunkte zu erfassen [Han 05]. Ebenso wie bei dem Ansatz mit Infrarotmarkern lassen sich einzelne Druckpunkte keiner Hand und keiner bestimmten Geste zuordnen, wenn diese nur zu einem Zeitpunkt betrachtet werden. Zur Identifikation und Erfassung von Gesten könnten daher dynamische Modelle analog zu dem in dieser Arbeit beschriebenen Ansatz verwendet werden, wobei man diese dann nur im Zweidimensionalen definieren würde.

Bei der beidhändigen Interaktion mit dem hier beschriebenen Ansatz eröffnen vor allem Gruppierung und die Verfolgung der beiden Hände die Möglichkeit zur Unterscheidung [Schwald 05b]. Dabei bietet es sich an, der einen Hand nur die Aufgabe der Zeigerichtung und der anderen Hand die Auswahl des Interaktionsmusters zu überlassen. Dies hat den Vorteil, dass die Hand, die die Zeigerichtung bestimmt, nicht bewegt werden muss, wenn sie auf ein Objekt zeigt und eine Aktion ausgelöst werden soll, siehe Abbildung 6.5. Als Gesten zum Auslösen der verschiedenen Aktionen können beispielsweise die in [Kim 04a] beschriebenen Ansätze verwendet werden. Im Folgenden wird eine beidhändige Interaktion beschrieben, bei der eine Hand ausschließlich die Zeigegeste umsetzen soll und die andere Hand selektieren soll.

Eine Grundlage für diese Verwendung von zwei dynamischen Handgruppenmodellen stellt die Unterscheidung der beiden Gruppen zu jedem Zeitpunkt dar. Eine unterschiedliche Anzahl von Markern für jede Gruppe kann die Unterscheidung zwar in einigen Fällen erleichtern, ist aber, aufgrund der zu erwartenden häufigen und möglicherweise länger anhaltenden Verdeckungen, kein wirklich zuverlässiges Kriterium. Im folgenden Ansatz wird davon ausgegangen, dass die Anzahl der Marker der beiden Gruppen gleich ist. Diese Anzahl sei drei, wobei je ein Marker auf Daumen, Zeige- und Mittelfinger angebracht ist.

Das wichtigste Kriterium für die Identifikation der Modelle in der Menge der Gruppen ist neben der Größe die Verfolgung der Gruppe. Um die Gruppe verfolgen zu können, muss allerdings zunächst eine Initialisierung stattfinden, das heißt, es muss einer der beiden Hände die Zeigegeste zugewiesen werden und der anderen die Selektion. Da die Zeigegeste allein im Allgemeinen keine Aktion auslöst, wird der Hand, deren Marker als erstes als Gruppe aus drei Punkten erkannt wird, die Zeigegeste zugewiesen. Auf diese Weise lässt sich jederzeit die Zeigegeste neu zuweisen, indem alle Marker verdeckt werden und dann die Hand, die die Zeigegeste ausführen soll, als erstes interagiert.

Sind die Funktionalitäten zugewiesen, werden die Handgruppen verfolgt, wobei sich folgende Kriterien für die Identifikation bei der Verfolgung anbieten:

1. Anzahl der Punkte in einer Gruppe.

2. Entfernung zu einer präzidierten Position. Im einfachsten Fall ist die präzidierte Position die vorausgegangene.
3. Vergleich der Identifikationsindizes der Marker, die das Gruppenmodell im letzten Frame gebildet haben, mit den Identifikationsindizes der Marker in der entsprechenden zu untersuchenden Gruppe.
4. Vergleich zur vorausgegangenen Konstellation der Marker einer Gruppe, beispielsweise durch durchschnittliche Distanz zwischen den Markern oder Größe der Dreiecksfläche.

Das erste Kriterium mit der richtigen Anzahl der Punkte kann stark formuliert werden, indem man eben genau die richtige Anzahl der Marker fordert, hier also drei. Dies hat allerdings den Nachteil, dass ein Modell nicht erkannt wird, wenn beispielsweise kurzzeitig ein Marker verdeckt ist. Um die Robustheit der Erkennung der dynamischen Gesten zu erhöhen, ist es daher meist sinnvoll, eine gering abweichende Anzahl von Punkten zuzulassen. Wird eine Zeigegeste mit nur zwei Punkten aufgrund ihrer eindeutigen Position im Interaktionsvolumen erkannt, kann immer noch ihre Position berechnet werden. Die Zeigerichtung muss dann aus der vorhergehenden abgeleitet werden.

Das zweite und das dritte Kriterium sind sehr zuverlässig, wenn die Gruppen hinreichend weit voneinander entfernt sind, das vierte Kriterium ist sinnvoll als zusätzliches Kriterium, aber nicht als hinreichendes Kriterium. Gilt ein Modell als detektiert, wird bei den das Modell erzeugenden Punkten der Klassifikationsindex des Modells gesetzt, um diese für die Suche nach weiteren Modellen auszuschließen. Abbildung 6.6 zeigt den Ablauf für die hier beschriebene Interaktion mit zwei Händen.

Insbesondere bei der Verwendung von zwei oder mehr dynamischen Gruppenmodellen muss eine erfolgreiche Detektion eines Modells nicht notwendigerweise eine Aktion auslösen. Wird beispielsweise eine Zeigegeste mit nur zwei statt drei Markern erkannt, besteht eine Möglichkeit darin, die Zeigerichtung nicht auszugeben, da sie ja auch aktuell nicht bestimmt werden kann. Das Modell dennoch als erkannt gelten zu lassen hat den Vorteil, dass das Modell leichter weiter verfolgt werden kann.

6.2 Dynamische Gelenkmodelle

Im Gegensatz zu den dynamischen Gruppenmodellen sind bei den (dynamischen) Gelenkmodellen die Marker nicht notwendigerweise dicht beieinander platziert, sondern können weit verteilt sein. Dafür sind einzelne, die Marker repräsentierende, Punkte in der Modellbeschreibung auf verschiedene Arten miteinander verbunden, was durch ein Modell starrer Verbindungen zwischen Gelenkpunkten ausgedrückt werden kann. Die Marker werden als Gelenkpunkte interpretiert. Dabei sind alle Punkte eines Gelenkmodells direkt durch eine oder durch mehrere aufeinander folgende Verbindungen miteinander verbunden.

Typische Beispiele für Objekte, die mit den Gelenkmodellen verfolgt werden können sind der gesamte menschliche Körper oder einzelne Extremitäten, sowie mechanische

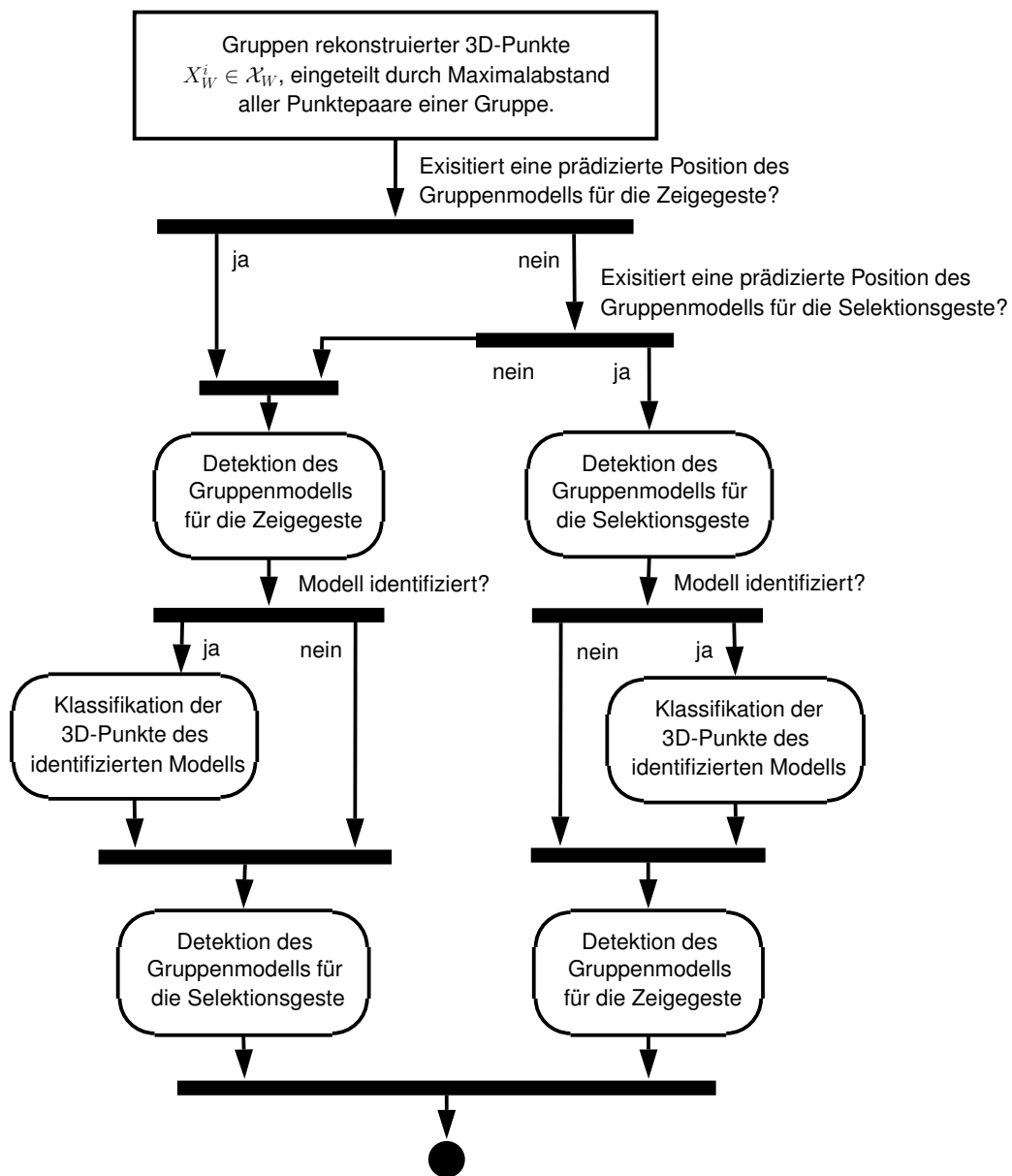


Abbildung 6.6. Schematischer Ablauf der Verfolgung von zwei Handgruppenmodellen, davon ein Modell für eine Zeigegeste, das zweite Modell für Selektion.

Arme. Die Verfolgung mechanischer Arme kann insbesondere zur Qualitätskontrolle eingesetzt werden.

In den folgenden Abschnitten werden zwei verschiedene Ausprägungen von Gelenkmodellen beschrieben: Gelenkmodelle für Arme oder Finger und Gelenkmodelle für den menschlichen Körper.

6.2.1 Arm- und Fingergelenkmodelle

Für die Verfolgung eines menschlichen Arms oder Fingers, beziehungsweise eines mechanischen Arms, genügt es meist, ein Modell mit drei oder vier Gelenken zu definieren. Die Bewegungsfreiheit der Gelenke ist meist eingeschränkt, insbesondere ist dies bei einem Finger der Fall, was die Erkennung eines Modells oder Teile dieses Modells einfacher macht. In [Dorfmueller-Ulhaas 01] wird ein Ansatz für ein Fingertracking zur Interaktion beim Schachspiel in einer AR-Umgebung vorgestellt. Die Marker sind dabei in Form eines Exoskeletts auf dem Zeigefinger angebracht, was den Vorteil hat, dass zumindest zwischen den durch das Exoskelett verbundenen Markern die Abstände konstant gehalten werden können. Als Interaktion wurde Greifen und Loslassen von Objekten durch Finger beugen und Finger strecken, sowie verschieben und rotieren von Objekten durch Bewegung des gesamten Fingers implementiert.

Die Verwendung eines Exoskeletts bedeutet bei einem Finger eine geringe Einschränkung, ist für einen menschlichen Arm jedoch schon weniger geeignet. Werden Marker nicht mechanisch miteinander verbunden, so muss die Beschreibung der Verbindungen zwischen einzelnen Markern flexibler gestaltet werden. Um dennoch Verbindungen zwischen verschiedenen Gelenkpunkten beziehungsweise Markern zu erkennen, bietet sich ein Kugelschalenmodell an. Dabei ist ausgehend von einem Gelenkpunkt bekannt, dass sich ein mit diesem Gelenkpunkt verbundener nächster Gelenkpunkt in einer bestimmten Entfernung befindet, allerdings mit einer gewissen vorgegebenen Toleranz. Der Bereich, in dem sich der zweite Gelenkpunkt befinden kann, entspricht also der Vorstellung, dass er in einer Kugelschale um den ersten Gelenkpunkt liegt. In einfacheren Fällen, insbesondere wenn sich die Abstände zwischen Gelenkmodellen stark unterscheiden, lässt sich damit das vollständige Gelenkmodell einfach erfassen.

Der Ansatz mit den Kugelschalen funktioniert insbesondere ohne Vorwissen über die aktuellen Positionen der Gelenkpunkte. Existiert jedoch dieses Vorwissen und werden dazu die einzelnen Punkte wie in Abschnitt 4.1.3 beschrieben verfolgt und mit einem Identifikationsindex versehen, so lässt sich in den meisten Fällen das Gelenkmodell einfach anhand der Identifikationsindizes verfolgen, und die Kugelschalen können ausschließlich zur Überprüfung verwendet werden. Fehlen einzelne Gelenke durch die Verdeckung des entsprechenden Markers, bietet sich die Verwendung Inverser Kinematik an, um die Position des entsprechenden Gelenkpunktes zu bestimmen.

6.2.2 Körpermodelle

Die Erfassung und Analyse menschlicher Bewegung durch Kameras wird auf viele verschiedene Arten praktiziert. Einen Überblick über die zahlreichen Ansätze sowohl mit Stereokamerasystemen als auch einzelnen Kameras geben unter anderem [Gavrila 99], [Moeslund 01], [Shafaei 06] und [Shen 06]. Die Erfassung des Körpers durch Marker, insbesondere Infrarotmarker, stellt darunter eine technisch vergleichsweise einfache, aber vor allem sehr präzise und effiziente Methode dar. Kommerzielle Systeme wie [Vicon 06], die mit einer großen Anzahl von Kameras arbeiten, werden in der Film- und Computerspielindustrie eingesetzt, um menschliche Bewegungsabläufe auf virtuelle Charaktere

übertragen zu können. Die Erkennung und Verarbeitung der menschlichen Bewegungsabläufe in Echtzeit stellt dabei kein vordringliches Ziel dar, da die Daten ohnehin erst später verwendet werden und zusätzlich noch nachbearbeitet werden können. Echtzeitverarbeitung wird dennoch zumindest in reduzierter Form meist unterstützt, um unter anderem die Aufnahme der Daten direkt kontrollieren zu können.

Für typische VR-Anwendungen sind die existierenden Systeme meist nicht geeignet, da sie einen großen technischen und finanziellen Aufwand bedeuten. Auf der anderen Seite ist die Verfolgung des ganzen Körpers mit nur zwei Kameras, die eine vergleichsweise geringe Framerate haben (≤ 60 Hz), auch hinsichtlich der Software beziehungsweise der Algorithmen komplizierter, da mehr Verdeckungen ausgeglichen werden müssen und schnelle Bewegungen große Veränderungen zwischen den aufgenommenen Bildern verursachen.

Ein einfaches Human Motion Capturing System, das mit nur zwei Kameras arbeitet und die Bewegung in Echtzeit analysiert, kann mit wenigen Markern realisiert werden. Mit einer Anzahl von mindestens 13 Markern beziehungsweise Gelenkpunkten (drei für jede der vier Extremitäten und einen für den Kopf), lässt sich ein grobes Menschmodell festlegen. Falls die vorausgegangene Pose bekannt ist, kann die Verfolgung des Menschmodells, ähnlich wie in Abschnitt 6.2.1 beschrieben, durch die Verfolgung der einzelnen Marker in einigen Fällen bewältigt werden. Ein Schlüsselproblem bleibt, welche Informationen dem Modell initial hinzugefügt werden und wie. Feste Abstände zwischen den Gelenkpunkten sind aufgrund unterschiedlicher Körpergrößen von verschiedenen Benutzern ungeeignet. Daher bietet sich im Allgemeinen eine Initialpose, wie aufrechtes Stehen mit ausgebreiteten Armen zur Vermessung von Distanzen zwischen Gelenkpunkten an. In [Hornung 04] wird Motion Capturing ohne Vorabwissen über das Skelett durchgeführt, allerdings werden statt einzelner Punkte in der Nähe von Gelenken starre Modelle an verschiedenen Körperteilen verwendet.

6.3 Dynamisches Flächenmodell

Die hier als dynamisches Flächenmodell beschriebene Verwendung von Markern stellt eine seltener vorkommende Form des Trackings dar. Der Begriff des dynamischen Flächenmodells steht hier für die Verarbeitung aller sichtbarer Marker zu einer Oberfläche. Ein Beispiel hierfür, bei dem auch die Verarbeitung in Echtzeit eine Rolle spielt, ist eine Kontrolle der Patientenlagerung im medizinischen Umfeld.

In zahlreichen medizinischen Vorgängen müssen Patienten ohne jegliche Bewegung über einen längeren Zeitraum entweder auf einem Operationstisch oder dem Tisch eines bildgebenden Geräts, zum Beispiel eines Computertomographen (CT) oder Magnetresonanztomographen liegen. Um dies sicherzustellen, werden Patienten entweder fixiert, oder aber deren Bewegung wird verfolgt, siehe Abschnitte 8.1 und 8.2. Für diesen Zweck ist neben anderen Methoden die Verwendung eines optischen Trackingsystems mit Infrarotmarkern geeignet, wobei die Marker auf der Patientenoberfläche verteilt werden. Abbildung 6.7 zeigt, wie im medizinischen Bereich verwendete, in CT-Bildern sehr gut sichtbare, sogenannte CT-Spots auf einem Patienten angebracht sind und in dem abgebildeten Fall mit einem Registrierungsgerät eingemessen werden. Um das Registrierungs-

gerät zu verfolgen, können Marker am oberen Ende des Griffs angebracht werden. Die Verfolgung erfolgt dann durch ein Multipunktmodell und das lokale Koordinatensystem wird in die Spitze des Instruments verschoben. In der Abbildung 6.7 wurde alternativ hierzu ein elektromagnetischer Sensor in den Griff integriert, um das Registrierungsgerät zu verfolgen. Vergleicht man die mit dem Registrierungsgerät ausgemessenen aktuellen Positionen der als Marker fungierenden CT-Spots mit den Positionen der CT-Spots in einem vorab aufgenommenen CT-Datensatz, lassen sich die CT-Daten auf den Patienten überlagern und so während eines Eingriffs für eine VR- oder AR-Anwendung nutzen.

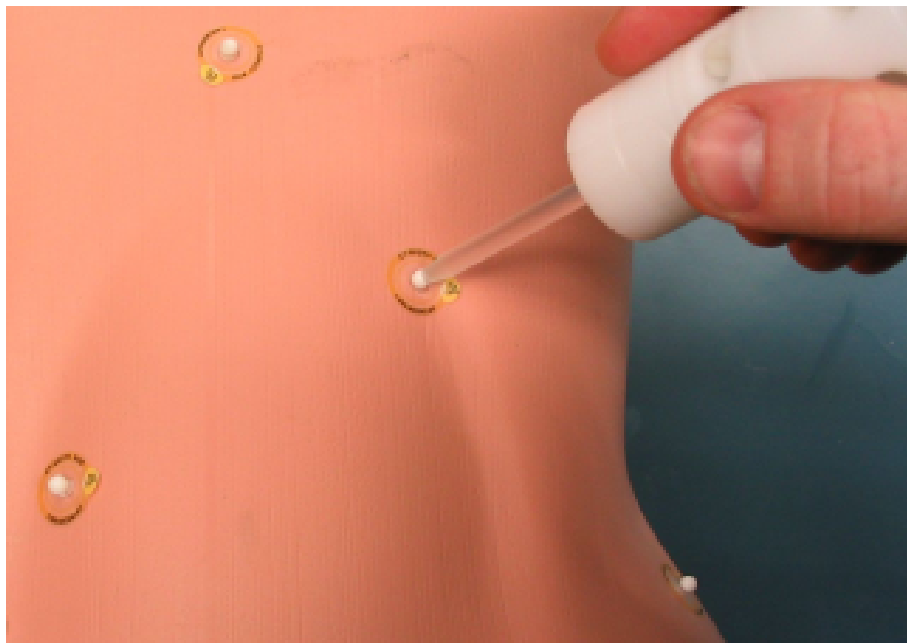


Abbildung 6.7. Kugelförmige Markierungen, die auf der Patientenoberfläche angebracht werden und im CT gut sichtbar sind.

Die bessere Lösung wäre hier, die Marker nicht zu einem bestimmten Zeitpunkt mit einem Registrierungsgerät einzumessen, sondern optisch zu verfolgen. Damit können Bewegungen des Patienten sowie Deformationen seiner Körperoberfläche automatisch erfasst werden und der CT-Datensatz entsprechend angepasst werden. Für den Einsatz eines Infrarottrackingsystems müssten die CT-Spots so modifiziert werden, dass sie retroreflektierende Eigenschaften erhalten.

Die Verfolgung eines dynamischen Flächenmodells beruht hier auf der Annahme, dass alle sichtbaren beziehungsweise keinem anderen Modell zugeordneten Punkte, die vom Trackingsystem erfasst werden, auf einer Fläche platziert sind. Die feste Anzahl der 3D-Punkte, die das Flächenmodell bilden, sind in einer initialen Positionierung als Bestandteil des Modells gespeichert. Wird dieser Ansatz verwendet, um einen oben beschriebenen CT-Datensatz korrekt auf den Patienten abzubilden, so sind die initialen Positionen der Marker im Allgemeinen die, die im CT-Datensatz in dessen lokalem Koordinatensystem ermittelt werden. Während des Trackings können dann relativ zu der initialen Positi-

on Transformationen angegeben werden. Eine Berechnung von Deformationsparametern wäre ebenfalls möglich, wird aber im folgenden Verfahren zunächst nicht umgesetzt.

Sei $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$ die Menge der getrackten Punkte und $\mathcal{X}_D = \{X_D^1, \dots, X_D^n\}$ die Menge der initialen Positionen der zum Flächenmodell gehörenden Punkte. Lassen sich einzelne getrackte Punkte X_W^i jeweils einem Modellpunkt zuordnen, das heißt, existieren $p \leq n$ Korrespondenzen (X_W^i, X_D^i) zwischen 3D-Punkten, so kann eine Transformation bestehend aus einer Rotationsmatrix R und einer Translation T durch Minimierung von

$$err = \sum_{i=1}^p \underbrace{\|RX_W^i + T - X_D^i\|}_{\text{Residuum}}^2 \quad (6.1)$$

gefunden werden. Ohne Beschränkung der Allgemeinheit korrespondieren dabei hier X_W^1 und X_D^1 , X_W^2 und X_D^2 , \dots , sowie X_W^p und X_D^p . Durch die Berechnung von R und T kann allerdings keine Deformation der Fläche ausgedrückt werden.

Die Optimierung kann analog zu dem in [Schwald 04b] beschriebenen Abgleichalgorithmus zwischen verschiedenen Trackingsystemen erfolgen: err aus Gleichung (6.1) wird dabei mit derselben Vorgehensweise wie in Abschnitt 4.3.2 minimiert.

Die Ergebnisse für R und T werden auf jede Korrespondenz (X_W^i, X_D^i) angewendet und falls der Abstand

$$d = |RX_W^i + T - X_D^i| \quad (6.2)$$

einen gewissen Grenzwert überschreitet, wird diese Korrespondenz in einem zweiten Iterationsschritt nicht berücksichtigt, um den Einfluss fehlerhafter oder stark gestörter Korrespondenzen zu vermeiden. Die nach dem zweiten Iterationsschritt resultierende Rotation R und Translation T sind das Endergebnis. Dieser Algorithmus ist eine Variante des **Iterative Closest Point** (ICP) Algorithmus, der auf Besl und McKay [Besl 92] zurückgeht und innerhalb der letzten Jahre in zahlreichen Publikationen diskutiert wurde.

Eine alternative allerdings auch rechenintensivere Methode ist eine nichtlineare Optimierung nach Levenberg-Marquardt (LM) [Marquardt 63],[Press 02]. Die Erfahrungen in [Schwald 04b] haben jedoch gezeigt, dass sich die Ergebnisse mit diesem zweiten Ansatz nur unwesentlich von denen des ersten Ansatzes unterscheiden.

Der alternative Algorithmus auf Basis der LM-Methode funktioniert wie folgt: Der erste Schritt ist eine lineare Optimierung mit dem Ziel, hinreichend gute Initialwerte für die LM-Optimierung zu erzeugen. Aus den korrespondierenden Punktpaaren (X_W^i, X_D^i) wird die Matrix $A \in \mathbb{R}^{3p \times 12}$ mit

$$A^T = \begin{pmatrix} x_W^1 & 0 & 0 & x_W^2 & \cdots & x_W^p & 0 & 0 \\ y_W^1 & 0 & 0 & y_W^2 & \cdots & y_W^p & 0 & 0 \\ z_W^1 & 0 & 0 & z_W^2 & \cdots & z_W^p & 0 & 0 \\ 1 & 0 & 0 & 1 & \cdots & 1 & 0 & 0 \\ 0 & x_W^1 & 0 & 0 & \cdots & 0 & x_W^p & 0 \\ 0 & y_W^1 & 0 & 0 & \cdots & 0 & y_W^p & 0 \\ 0 & z_W^1 & 0 & 0 & \cdots & 0 & z_W^p & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & x_W^1 & 0 & \cdots & 0 & 0 & x_W^p \\ 0 & 0 & y_W^1 & 0 & \cdots & 0 & 0 & y_W^p \\ 0 & 0 & z_W^1 & 0 & \cdots & 0 & 0 & z_W^p \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \quad (6.3)$$

und dem Vektor $b^T = (x_D^1, y_D^1, z_D^1, x_D^2, \dots, z_D^p)$, $b \in \mathbb{R}^{3p}$ erzeugt. Unter anderem wird in [Golub 89] beschrieben, wie das überbestimmte lineare Gleichungssystem $Ax = b$ durch eine QR-Zerlegung gelöst werden kann und neun Komponenten des Ergebnisses $x = (x_1, x_2, \dots, x_{12})$ die vorläufige Lösung R' mit

$$R' = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_5 & x_6 & x_7 \\ x_9 & x_{10} & x_{11} \end{pmatrix} \quad (6.4)$$

formen. Die Matrix R' ist keine Rotationsmatrix, da sie im Allgemeinen nicht orthonormal ist. Die eigentliche Rotationsmatrix R erhält man wiederum durch eine Singulärwertzerlegung von R' . Die Translation $T = (x_4, x_8, x_{12})^T$ wird aus den drei verbleibenden Komponenten erzeugt. Diese Translation T und die Rotation R werden als Initialwert für die Optimierung mit der LM-Methode mit *Residuum* aus Gleichung (6.1) als Kostenfunktion verwendet. Die daraus resultierende Translation T und Rotation R sind das Endergebnis.

Lassen sich die Korrespondenzen (X_W^i, X_D^i) nicht ohne weiteres aus vorhergehenden Konstellationen oder geometrischen Bedingungen bestimmen, muss mit mehreren verschiedenen Mengen von Korrespondenzen optimiert werden oder eine Optimierung vorab durchgeführt werden, die die Korrespondenzen bestimmt.

Existieren die Korrespondenzen zwischen den initialen Punkten und den aktuell getrackten Punkten, und wurde eine Transformation zwischen diesen bestimmt, so lässt sich auch eine Deformation des dynamischen Flächenmodells bestimmen. Dabei können unter anderem die bei der Optimierung für die Transformation ermittelten Residuen zwischen den korrespondierenden Punkten verwendet werden.

6.4 Kombination starrer und dynamischer Modelle

In der bisherigen Betrachtung der verschiedenen starren und dynamischen Modelltypen wurden diese weitestgehend nur getrennt voneinander behandelt. In diesem Abschnitt soll daher näher diskutiert werden, welche Arten von Modellen sich innerhalb eines Tracking-

systems sinnvoll und zuverlässig kombinieren lassen. Die meisten existierenden Tracking-systeme beschränken sich auf einen Modelltyp.

Die innerhalb dieser Arbeit präsentierten Modelle eignen sich unterschiedlich gut zur Nutzung mit mehreren Modellen des eigenen Typs oder in Kombination mit anderen Modelltypen. Die Tabelle 6.1 zeigt eine Bewertung, ob sich bestimmte Modelltypen gut kombinieren lassen, dafür eher bedingt geeignet oder ungeeignet sind. Dabei hängt die Eignung zum einen von den Gegebenheiten der speziellen Anwendung ab und zum anderen davon, ob diese Kombination überhaupt sinnvoll ist oder ob eine andere Kombination von Modellen grundsätzlich sinnvoller ist.

Modelltyp	starre Modelle			dynamische Modelle		
	S1	S2	S3	DGP	DGK	DF
S1	○					
S2	+	+				
S3	+	○	+			
DGP	+	○	+	+		
DGK	-	-	+	-	-	
DF	-	-	+	-	-	-

Tabelle 6.1. Sinnvolle Kombinationsmöglichkeit verschiedener Modelle für die Verfolgung von Objekten. Dabei steht S1 für einzelne Marker, S2 für Zweipunktmodelle, S3 für Multipunktmodelle, DGP für dynamische Gruppenmodelle, DGK für dynamische Gelenkmodelle und DF für dynamische Flächenmodelle. Eine gut umzusetzende und sinnvolle Kombination wird mit + markiert, eine umsetzbare, aber eher ungeeignete Kombination mit ○ und eine ungeeignete Kombination mit –.

Dem Eintrag in der Tabelle entsprechend lassen sich zwei einzelne Marker nur mit sehr selten auftretenden Verdeckungen verfolgen, da sie eben nicht unterscheidbar sind und deshalb insbesondere bei einer gleichzeitigen Verdeckung der beiden Marker danach kaum mehr identifiziert werden können. Der wesentlich sinnvollere Ansatz, um nur die jeweilige Position von zwei verschiedenen Objekten zu erfassen, ist die Verwendung von zwei Zweipunktmodellen (siehe beispielsweise Abschnitt 8.3) oder gar zwei Multipunktmodellen (siehe beispielsweise Abschnitt 8.1), die sich jeweils problemlos miteinander kombinieren lassen. Einpunktmodelle lassen sich ansonsten sinnvoll mit Zwei- und Multipunktmodellen, beziehungsweise dynamischen Gruppenmodellen, kombinieren, da diese Modelle identifiziert werden können und daraus auf den einzelnen Marker geschlossen

werden kann. Im VR-Bereich wird teilweise nur ein einzelner Marker zur Erfassung einer Kopfposition verwendet und ein anderer Modelltyp für eine direkte und indirekte Interaktion mit der Hand. Eine Kombination der Verfolgung eines einzelnen Markers mit einem dynamischen Gelenkmodell ist im Allgemeinen ungeeignet, da sich bei Gelenkmodellen einzelne Punkte schlecht identifizieren lassen.

Eine Kombination von Zweipunktmodellen mit Multipunktmodellen ist möglich, allerdings meist wohl eher ausschließlich durch Zweipunktmodelle oder ausschließlich durch Multipunktmodelle umsetzbar. Dahingegen ist die Kombination von dynamischen Gruppenmodellen und Multipunktmodellen relativ problemlos, da die Multipunktmodelle sehr gut identifiziert werden können. Sinnvoll ist diese Kombination beispielsweise, um die Verfolgung und Erkennung dynamischer Handgesten mit der Verfolgung fester Körper zu kombinieren. Die Kombination von Multipunktmodellen mit Gelenkmodellen ist unter Umständen sinnvoll, um beispielsweise Körperteile und den Kopf des Benutzers mit einer Kappe, auf der ein festes Modell angebracht ist, zu verfolgen. Allerdings ist bei Gelenkmodellen eine Kombination mit anderen Modellen in vielen Fällen problematisch, da sie meist auf spezielle Situationen zugeschnitten sind und durch andere Modelle stark gestört werden können. Daher ist auch in den meisten Fällen eine Kombination von zwei dynamischen Gelenkmodellen ungeeignet. Im Gegensatz dazu ist die Kombination aus zwei dynamischen Gruppenmodellen, wie in Abschnitt 6.1.3 beschrieben, weitestgehend unproblematisch und sinnvoll.

Dynamische Flächenmodelle sind insgesamt relativ ungeeignet für die Kombination mit anderen Modelltypen. Ausnahmen bilden hier wieder die Multipunktmodelle. Im medizinischen Kontext besteht ein sinnvolles Szenario aus der Verfolgung der Patientenoberfläche durch ein dynamisches Flächenmodell mit gleichzeitiger Verfolgung eines medizinischen Instruments durch ein Multipunktmodell.

In der dieser Arbeit zugrunde liegenden Implementierung ist eine Kombination aus der Verfolgung eines einzelnen Markers (S1), von Multipunktmodellen (S3) und dynamischen Gruppenmodellen (DGP) umgesetzt. Dabei ist der Ablauf für die Erkennung der verschiedenen Modelle wie folgt:

1. Die Menge $\mathcal{X}_W = \{X_W^1, \dots, X_W^m\}$ der rekonstruierten 3D-Punkte wird aus den 2D-Bildpunkten erzeugt und mit einem Identifikationsindex versehen.
2. Daraufhin wird versucht, die vorgegebenen Multipunktmodelle zu detektieren. Jeder 3D-Punkt X_W^i , der einem Multipunktmodell zugeordnet werden konnte, erhält den Klassifikationsindex des entsprechenden Modells. Bereits klassifizierte Punkte werden für die Erkennung weiterer Modelle nicht berücksichtigt.
3. Auf Basis der verbleibenden, also nicht klassifizierten, 3D-Punkte wird versucht, dynamische Gruppenmodelle zu detektieren. Jeder 3D-Punkt X_W^i , der einem dynamischen Gruppenmodell zugeordnet werden konnte, erhält den Klassifikationsindex des entsprechenden Modells. Auch diese klassifizierten Punkte werden für die Erkennung weiterer Modelle nicht berücksichtigt. Für spezielle Anwendungen, wie die in Abschnitt 6.1.3 beschriebene zweihändige Interaktion, kann innerhalb

dieses Schritts eine besondere Verarbeitung der dynamischen Gruppenmodelle untereinander erforderlich sein. Die Identifikationsindizes der 3D-Punkte können für die Verfolgung berücksichtigt werden.

4. Zuletzt wird die Erkennung eines einzelnen Markers auf den nicht bereits klassifizierten 3D-Punkten durchgeführt. Der Identifikationsindex der 3D-Punkte, die letzte Position sowie weitere Randbedingungen bezüglich der Positionierung sind weitere Kriterien für die eindeutige Identifikation.

Insgesamt gilt, dass sich insbesondere die Multipunktmodelle mit anderen Modelltypen gut kombinieren lassen, da diese sehr gut identifiziert werden können.

6.5 Zusammenfassung

Dynamische Modelle bestehen aus einer Menge von Markern, aber im Gegensatz zu den starren Modellen sind diese nicht fest zueinander positioniert. Daher kann ein dynamisches Modell auch nicht wie ein Multipunktmodell durch die Koordinaten von 3D-Punkten beschrieben werden. Die dynamischen Modelle werden hier in drei verschiedene Kategorien unterteilt: Gruppenmodelle, Gelenkmodelle und Flächenmodelle.

Bei der Verfolgung von Gruppenmodellen wird vor der eigentlichen Erkennung eines Modells eine Gruppierung aller rekonstruierten Punkte vorgenommen. Insbesondere für die direkte Interaktion mit den Händen sind Gruppenmodelle geeignet. Ein Algorithmus für eine zweihändige Interaktion mit einer speziellen Zeigegeste wurde präsentiert.

Bei Gelenkmodellen ist die Gruppierung von Punkten als erster Verarbeitungsschritt nicht sinnvoll, da diese im Allgemeinen nicht an einer Stelle gehäuft sind. Gelenkmodelle werden typischerweise zur Verfolgung der menschlichen Gliedmaßen verwendet. Gelenkbeziehungsweise Roboterarme stellen einen weiteren sinnvollen Anwendungsbereich dar.

Dynamische Flächenmodelle sind insbesondere im medizinischen Kontext sinnvoll, um beispielsweise die mit Markern bestückte Oberfläche eines Patienten zu rekonstruieren und vor allem deren Veränderungen zu verfolgen.

In einigen Anwendungen ist es außerdem von Interesse, verschiedene Typen von Modellen kombinieren zu können. Ein Schema, welche Modelltypen sinnvoll miteinander kombiniert werden können, befindet sich am Ende dieses Kapitels.

Kapitel 7

Filterung und Prädiktion

Dieses Kapitel beschäftigt sich mit Filterung und Prädiktion der Trackingergebnisse, die in dieser Arbeit aus Rotationen und Translationen bestehen. In vorausgegangenen Kapiteln wurde auf die Prädiktion von Translationen beziehungsweise Punkten hingewiesen, wobei diese dort nicht näher beschrieben wurde oder im Fall der Prädiktion vereinfachend auf den letzten Wert zurückgegriffen wurde. Um die grundsätzlich verwendeten Techniken für Filterung und Prädiktion nicht mehrfach zu erläutern, sind sie in diesem Kapitel zusammengefasst.

Für Filterung und Prädiktion stehen unterschiedliche, bekannte Methoden zur Verfügung, die Vor- und Nachteile, wie schwächeres oder stärkeres Rauschen und geringere oder höhere Latenz, mit sich bringen. Von den in dieser Arbeit berücksichtigten Methoden wird die Verwendungsweise in den folgenden Abschnitten kurz erläutert. Grundlegende und weiterführende Literatur findet sich unter anderem in [Zikan 94a], [Zikan 94b], [Welch 95] und [Azuma 97]. Analog zu der in Abschnitt 4.3.3 beschriebenen Problematik für Trackingstrategien, nicht alle positiven Aspekte gleichzeitig erreichen zu können, verhält es sich auch bei den Zielen, die Transformationen möglichst genau, rauschfrei und ohne Latenz auszugeben und bei Bewegungen eine im mathematischen Sinne glatte („nicht kantige“) Ausgabe zu erzeugen.

Im Fall einer VR-Applikation, in der vor allem langsame Bewegungen verfolgt werden, wie beispielsweise bei der Verfolgung von medizinischen Instrumenten, ist der Effekt einer Transformationsvorhersage durch Prädiktion weniger wichtig, als präzise und möglichst rauschfreie Angaben über eine aktuelle, nahezu statische Transformation. Auch eine gewisse Latenz durch die Filterung ist in einem solchen Fall akzeptabel. Dahingegen erfordern VR-Applikationen mit schnelleren Bewegungen eine geringe Latenz, wobei ein gewisses Rauschen und eine etwas höhere Ungenauigkeit bei der Wiedergabe durch ein Renderingssystem kaum auffallen.

Im Folgenden wird angenommen, dass n Messergebnisse in Listen (Queues) einer vordefinierten maximalen Länge n_{max} vorliegen. Dabei werden Vektoren (oder je nach Zusammenhang Punkte) mit $T_i \in \mathbb{R}^3$ und Rotationen mit R_i bezeichnet, wobei die Rotationen auf verschiedene Arten dargestellt werden können, siehe Anhang A. Die indizierten Variablen T_{n+1} und R_{n+1} beschreiben, dass es sich um den jeweils neuesten Wert handelt. Je kleiner der Index, um so älter ist die Transformation.

7.1 Mittelung und Lineare Prädiktion

Eine der einfachsten Formen der Filterung für Vektoren ist die Bildung des arithmetischen Mittels

$$\bar{T} = \frac{1}{n} \sum_{i=1}^n T_i \quad (7.1)$$

der letzten n gespeicherten Vektoren. Das Mittel ist sehr einfach und schnell zu berechnen und unterdrückt ein Rauschen sehr stark, sodass Positionen, die durch Messungen, die an einer festen Stelle betrachtet werden, auch präzise im Rahmen der Messgenauigkeit des Trackingsystems ermittelt werden können. Allerdings verursacht dieses Vorgehen auch eine große Latenz, die vor allem bei schnelleren Bewegungen auffällt. Bei einer Mittelung der Messungen aus den n letzten Frames kann die Mittelung das Erreichen der tatsächlichen Position um $n - 1$ Frames verzögern (wenn man ein Objekt nach einer Bewegung fest positioniert und dann wartet, bis der konstante Messwert erreicht wird). Bei einer Updaterate von 50Hz und $n = 3$ entspricht dies also schon 40 Millisekunden.

Die Bestimmung eines solchen Mittels \bar{R} für Rotationen ist nicht auf dieselbe einfache Art wie bei Translationen möglich. Summiert man die letzten n Rotationen in Matrixform auf, so lässt sich die resultierende Matrix natürlich ohne weiteres durch die Anzahl der Matrizen teilen. Das Problem ist, dass diese so gemittelte Matrix im Allgemeinen nicht mehr die Eigenschaften einer Rotationsmatrix besitzt. Daher muss die zu dieser Matrix nächste Rotationsmatrix gefunden werden oder ein ganz anderer Ansatz verfolgt werden. Abschnitt 7.1.1 zeigt verschiedene Vorgehensweisen auf.

Während die Bildung des arithmetischen Mittels also zumindest für die Translation eine vergleichsweise einfache Methode zur Filterung der Transformationen ist, steht auf ähnliche Art und Weise die lineare Prädiktion zur Abschätzung der nächsten Transformation zur Verfügung. Während bei vielen Trackingsystemen für Filterung und Prädiktion der Kalman Filter, siehe Abschnitt 7.2, verwendet wird, wird unter anderem in [Zikan 94a], [Zikan 94b] berichtet, dass die Lineare Prädiktion in bestimmten Fällen bessere Ergebnisse liefert.

Für Vektoren erfolgt die Lineare Prädiktion m -ter Ordnung einfach durch die Bildung der gewichteten Summe

$$T_{n+1} = \sum_{j=0}^m w_j T_{n-j} \quad \text{mit} \quad w_j = (-1)^j \sum_{k=j}^m \frac{a^k}{j!(k-j)!} \quad (7.2)$$

mit einem festen Inkrement a . Wählt man als Inkrement $a = 1$, ergibt sich als Linearer Prädiktor erster Ordnung

$$T_{n+1} = \underbrace{T_n}_{\text{Position}} + \underbrace{T_n - T_{n-1}}_{\text{Geschwindigkeit}} = 2T_n - T_{n-1} \quad (7.3)$$

und als Linearer Prädiktor zweiter Ordnung

$$T_{n+1} = \underbrace{T_n}_{\text{Position}} + \underbrace{T_n - T_{n-1}}_{\text{Geschwindigkeit}} + \frac{1}{2} \underbrace{(T_n - 2T_{n-1} + T_{n-2})}_{\text{Beschleunigung}} = \frac{5}{2}T_n - 2T_{n-1} + \frac{1}{2}T_{n-2}. \quad (7.4)$$

Für Rotationen ist diese Art der Prädiktion auf Basis der gewichteten Summierung von Rotationen wesentlich problematischer. In [Auer 00] wird dies aufgegriffen und als prädizierte Rotation erster Ordnung

$$R_{n+1} = R_n R_{n-1}^T R_n \quad (7.5)$$

angegeben. Für die prädizierte Rotation zweiter Ordnung ergeben sich auf diese Weise allerdings bereits zwei mögliche Lösungen. Im Abschnitt 7.1.1 werden neben Ansätzen zur reinen Mittelung von Rotation auch solche zu deren Linearkombination präsentiert. Mit der Möglichkeit, Linearkombinationen von Rotationen zu bilden, lassen sich auch die Ansätze der linearen Prädiktion vergleichsweise einfach auf Rotationen übertragen.

Im Gegensatz zur arithmetischen Mittelung reduziert die lineare Prädiktion die Latenz, zumindest bei einer Bewegung ohne starke Richtungsveränderungen. Nachteil ist dafür ein zunehmendes Rauschen an festen Positionen und größere Fehler bei Richtungsänderungen. In [Auer 00] wird zur Reduktion des Rauschen bei der Prädiktion vorgeschlagen, statt des aktuellen Geschwindigkeitsvektors das arithmetische Mittel der letzten Geschwindigkeitsvektoren zu verwenden. Für die lineare Prädiktion erster Ordnung bedeutet dies statt der Gleichung (7.3)

$$\begin{aligned} T_{n+1} &= T_n + \frac{(T_n - T_{n-1}) + (T_{n-1} - T_{n-2}) + (T_{n-2} - T_{n-3})}{3} \\ &= \underbrace{T_n}_{\text{Position}} + \underbrace{\frac{1}{3}(T_n - T_{n-3})}_{\text{gemittelte Geschwindigkeit}} \\ &= \frac{4}{3}T_n - \frac{1}{3}T_{n-3} \end{aligned} \quad (7.6)$$

zu verwenden, hier mit einer Mittelung der letzten drei Geschwindigkeitsvektoren ausgedrückt. Diese Mittelung erhöht allerdings die Trägheit bei einem Richtungswechsel, sodass in dieser Arbeit ein anderer Ansatz vorgeschlagen wird. Die Überlegungen für den folgenden Filter besteht aus zwei Feststellungen:

- Die Bildung des arithmetischen Mittels reduziert bei geringen Bewegungen, oder wenn das zu verfolgende Objekt fest steht, das Rauschen der Messungen, was in diesen Fällen am wichtigsten ist. Eine Prädiktion hat wenig oder negativen Effekt.
- Bei Bewegungen, zumindest solchen ohne schnelle Richtungswechsel, ist eine Prädiktion sinnvoll.

Daher bietet es sich an, die beiden Techniken wie in [Zachmann 00] vorgeschlagen zu kombinieren und in Abhängigkeit der aktuellen Geschwindigkeit und unter Umständen

der Richtungsveränderung zu nutzen. Sei $v_n = T_n - T_{n-1}$, der aktuelle Richtungsvektor mit der Geschwindigkeit $s_n = |v_n|$ und s_{limit} ein vorab festgelegter Grenzwert, so wird bei $s_n \leq s_{limit}$ gemittelt und bei $s_n > s_{limit}$ prädiziert. Unterscheidet sich v_n in der Richtung sehr stark von v_{n-1} (zum Beispiel wenn der Winkel zwischen den beiden Vektoren einen gewissen Wert überschreitet), so kann die Prädiktion ausgesetzt werden. Um bei den Geschwindigkeiten um s_{limit} den Übergang von der einen Technik zur anderen weniger extrem zu gestalten, können beide Methoden gleichzeitig mit einer von s_n abhängigen Gewichtung angewendet werden. Ein solcher Ansatz ähnelt den bekannten Prädiktor-Korrektor Methoden, beispielsweise dem Kalman Filter, der in Abschnitt 7.2 behandelt wird.

7.1.1 Mittelung von Rotationen

Während sich Vektoren sehr einfach beispielsweise durch das arithmetische Mittel mitteln lassen, stellt sich dies für Rotationen komplizierter da. Eine arithmetische Mittelung von Rotationsmatrizen hat im Allgemeinen zur Folge, dass die gemittelte Matrix R_M keine Rotationsmatrix mehr ist, also $R_M \notin SO_3$. Eine Beschreibung der verschiedenen Darstellungsformen von Rotationen befindet sich in Anhang A.

In [Curtis 93] wird dargestellt, wie sich eine sinnvoll gemittelte Rotation R_M aus zwei Rotationen R_1 und R_2 berechnen lässt, wobei

$$R_M = XY \quad \text{und} \quad XDY = R_1 + R_2 \quad (7.7)$$

eine SVD der Summe der beiden Rotationsmatrizen R_1 und R_2 ist. Außerdem gilt für die den Rotationsmatrizen entsprechenden Quaternionen q_1, q_2 und q_M

$$q_M = \begin{cases} \frac{q_1+q_2}{\lambda}, & \text{falls } \langle q_1, q_2 \rangle \geq 0 \\ \frac{q_1-q_2}{\mu}, & \text{sonst} \end{cases} \quad (7.8)$$

wobei $\lambda = |q_1 + q_2|$ und $\mu = |q_1 - q_2|$ ist. Die Mittelung von $n \geq 2$ Rotationen lässt sich ebenso durch eine Aufsummierung $R_S = \sum_{i=1}^n R_i$ der Rotationsmatrizen und der anschließenden Anwendung einer SVD auf R_S bestimmen. Die SVD bestimmt dabei die zu R_S nächste Rotationsmatrix im Sinne der Frobenius Norm. Der Ansatz mit den Quaternionen lässt sich allerdings nicht ohne weiteres auf $n \geq 2$ zu mittelnde Rotationen übertragen.

Ist neben der Mittelung von Rotationen auch insbesondere die Interpolation von Rotationen von Interesse, bieten sich weitere Methoden an. Verschiedene Ansätze mit Bezierkurven und Splines, die unter anderem in [Shoemake 85] und [Belta 02] beschrieben sind, existieren. In [Belta 02] wird dabei zur Erzeugung gleichmäßiger Trajektorien für sich bewegende feste Körper eine SVD-basierte Projektion für die Interpolation auf der Lie-Gruppe der Transformationsmatrizen A mit

$$A = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \quad (7.9)$$

genutzt, wobei A als Blockmatrix zu verstehen ist mit R einer Rotationsmatrix und T einem Translationsvektor. Die Menge aller solcher Matrizen A beschreibt die Bewegung-

gen fester Körper und kann alternativ zur getrennten Betrachtung von R und T verwendet werden.

Weitere Ansätze, die die Eigenschaften von Lie-Algebren zur Interpolation von Transformationen nutzen, finden sich in [Alexa 02] und [Govindu 04], wobei in [Alexa 02] insbesondere die daraus entstehenden Möglichkeiten zur Linearkombination von Transformationen hervorgehoben werden. Definitionen verschiedener invarianter Notationen für mittlere oder durchschnittliche Rotationen finden sich in [Moakher 02].

7.2 Kalman Filter

Eine in vielen Bereichen wie Signalverarbeitung, Spracherkennung und Computer Vision weit verbreitete Methode zur Filterung von Daten ist das **Kalman Filter** [Kalman 60]. Das Kalman Filter ist ein rekursiver Kleinste-Fehlerquadrat-Algorithmus, dessen Zweck es ist, den Zustand eines Systems durch Messungen, die zufällig verteilte Fehler besitzen, zu schätzen. Für den Computer Vision Bereich ist [Welch 95] eine gute Einführung in dieses Thema. Typisch sind unter anderem die Filterung von Kopfpositionen [Jung 00], [Cordea 01] und hybrides Tracking [Ribo 03] oder allgemein der Versuch, dynamische Fehler beim Tracking zu reduzieren [Azuma 97]. Dabei wird meist das Extended Kalman Filter (EKF) verwendet, da dieses nicht-lineare Prozesse der zu schätzenden, sich verändernden Zustände zulässt.

Das Kalman Filter schätzt einen Prozess in einem Verfahren aus zwei Schritten:

- Das Filter schätzt den Prozesszustand zu einem gewissen Zeitpunkt (Prädiktion). Die dafür notwendigen Gleichungen werden als „Time Update“ Gleichungen zusammengefasst. Durch sie wird unter anderem eine a priori-Schätzung für den nächsten Schritt bestimmt.
- Daraufhin erhält das Filter eine neue Messung, die im Allgemeinen zumindest etwas verrauscht ist. Mit den an dieser Stelle verwendeten „Measurement Update“ Gleichungen wird zusammen mit der a priori-Schätzung unter anderem eine a posteriori-Schätzung berechnet (Korrektur).

Abbildung 7.1 zeigt eine Skizze dieses Verfahrens. Die Prozesse, für die das Kalman Filter in dieser Arbeit vorgesehen sind, sind die Filterungen von Translationen und Rotationen. In Abhängigkeit des zu filternden Prozesses können in einfachen Fällen Zustand und Messung auch identisch sein. Die Filterung der Translation lässt sich beispielsweise so implementieren, dass der Zustand $s_k = (x_k, y_k, z_k)^T$ mit der Messung $m_k = (x_k, y_k, z_k)^T$ identisch ist, der Zustand selbst ist also ebenfalls eine Translation. Bei einer weiteren sinnvollen Implementierung umfasst der Zustand $s_k = (x_k, y_k, z_k, vx_k, vy_k, vz_k)^T$ zusätzlich noch den aktuellen Geschwindigkeitsvektor. Im ersten Fall ist als Ergebnis der Filterung bei Bewegungen eine Latenz zu erwarten, beim Verharren an einer festen Position jedoch wenig Rauschen, ähnlich wie bei der reinen arithmetischen Mittelung in Abschnitt 7.1. Im zweiten Fall ist als Ergebnis der Filterung bei Bewegungen eine deutlich geringere Latenz zu erwarten, beim Verharren an einer festen Position allerdings mehr

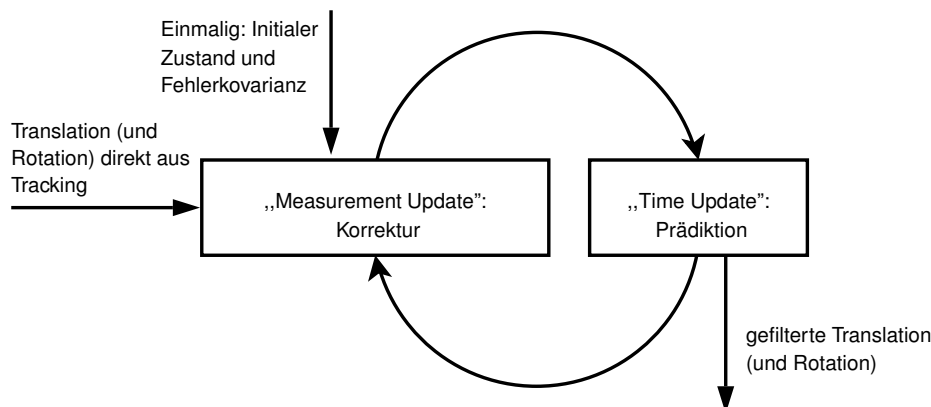


Abbildung 7.1. Der Kreislauf des Kalman Filters zur Filterung von Rotationen und Translationen.

Rauschen, ähnlich zur Beobachtung bei der Linearen Prädiktion in Abschnitt 7.1, siehe [Welch 01].

Bei den fehlerbehafteten Messungen wird davon ausgegangen, dass sich der Fehler ausschließlich durch zufälliges, unabhängiges, normalverteiltes weißes Rauschen modellieren lässt. Insbesondere ist also das Rauschen zu einem Zeitpunkt k unabhängig vom Rauschen zum Zeitpunkt $k - 1$.

In [Dorfmueller-Ulhaas 03] werden die geschätzten Transformationsparameter des EKF außer für die Filterung der Transformationen auch als Input für die Segmentierung verwendet. Eine im Raum prädierte Transformation wird also auf 3D-Punkte angewendet, die wiederum auf die 2D-Kameraebenen projiziert werden, um dort einen Anhaltspunkt für die Segmentierung im nächsten Schritt zu liefern. Dieses Vorgehen erlaubt unter Umständen eine schnellere Segmentierung und eine Hilfe beim Lösen des Korrespondenzproblems zur Vermeidung falscher Triangulierungen im nächsten Schritt, siehe Abschnitt 3.4.3.

Insgesamt ist das Kalman Filter eine sinnvolle Methode zur Filterung und Prädiktion der Transformationen, hat allerdings auch die Probleme anderer Methoden, dass es nicht für jede Situation automatisch die beste Lösung liefert, insbesondere bei den Gegensätzen schneller Bewegungen und Stillstand.

7.3 Nachführung der Transformation

Abgesehen von möglichst geringem Rauschen bei festen Positionen und einer möglichst geringen Latenz bei Bewegungen, gibt es noch weitere Eigenschaften, die eine Filterung in Abhängigkeit der Anwendung besitzen sollte. Eines dieser Ziele resultiert aus dem, bei optischen Trackingsystemen typischen, Auftreten von Verdeckung. Vorausgesetzt ist, dass ein zu verfolgendes Objekt nach einer Verdeckungsphase wieder identifiziert wird, was bei den Zwei- und Multipunktmodellen im Allgemeinen kein Problem darstellt. Während der Verdeckungsphase ist die Transformation des Objekts konstant geblieben oder würde

beispielsweise anhand der letzten Geschwindigkeit und Richtung weitergeführt. In beiden Fällen entsteht im Allgemeinen ein Sprung in den Messungen, sobald das Objekt wieder verfolgt wird, oder eine sehr langsame Korrektur der Messungen, falsch stark gefiltert wird.

In Anwendungen, in denen es nur auf aktuelle Genauigkeit ankommt und nicht auf eine glatte Bewegung, kann dieser Sprung vernachlässigt werden. Immer dann, wenn Bewegungsabläufe stetig und glatt wirken sollen, muss in solchen Situation gefiltert werden. Eine Möglichkeit besteht darin, beispielsweise ein Kalman Filter, siehe Abschnitt 7.2, zu verwenden, das entsprechend eingestellt wird. Allerdings bietet sich auch eine wesentlich einfachere Methode an.

Sei T die aktuell ermittelte Transformation und T_{alt} die letzte ausgegebene Transformation. Um einen Sprung zu vermeiden wird nicht T ausgegeben, sondern ein

$$T_{neu} = w \cdot T + (1 - w) \cdot T_{alt}, \quad (7.10)$$

wobei $w \in [0, 1]$ ein Gewichtungparameter ist. Wählt man $w = \frac{1}{2}$ konstant, entspricht dies einfach einer arithmetischen Mittelung. Wird w jedoch dynamisch verändert, ergeben sich andere Effekte. Setzt man w nach einer Phase der Verdeckung sehr gering an, zum Beispiel $w = \frac{1}{10}$ und steigert diesen Wert in den folgenden Frames kontinuierlich bis $w = 1$ erreicht wird, so hat man den Effekt, dass sich das Objekt von der Position vor der Verdeckung immer schneller auf die aktuelle Position zu bewegt, die dabei auch noch verändert werden kann. Entsprechendes gilt für Rotationen, allerdings mit mehr Aufwand, siehe Abschnitt 7.1.1.

7.4 Zusammenfassung

In diesem Kapitel sind verschiedene Methoden zur Filterung und Prädiktion der in den vorausgegangenen Kapiteln beschriebenen Trackingergebnissen zusammengefasst. Da bei Mixed Reality-Anwendungen häufig eine ruhige und stabile Ausgabe von Transformationen wichtiger ist als die bestmögliche Präzision der Trackingergebnisse, bieten sich verschiedene Verfahren an, die hier vorgestellt wurden. Dies sind unter anderem die einfache Mittelung, die Lineare Prädiktion und der Kalman Filter.

Während sich (gewichtete) Mittel von Translationen einfach bilden lassen, ist das Vorgehen bei Rotationen weniger intuitiv. Verschiedene Methoden zur Mittelung von Rotationen wurden daher kurz vorgestellt.

In einigen Anwendungen ist es außerdem für den Anwender angenehmer, wenn selbst nach einem temporären Verlust eines Objekts durch das Trackingsystem dessen virtuelle Darstellung keine sofortige Änderung der Transformation bei der Wiedererkennung des Objekts vollzieht, sondern die Transformation nachgeführt wird, wie es am Ende dieses Kapitels beschrieben ist.

Kapitel 8

Anwendungen

In diesem Kapitel werden Anwendungen beschrieben, in denen die Trackingverfahren aus dieser Arbeit Verwendung gefunden haben, beziehungsweise für die einzelne Aspekte erarbeitet und implementiert wurden. Die in den Abschnitten 8.1 und 8.2 beschriebenen Anwendungen machen dabei von den in Abschnitt 4.3 beschriebenen Multipunktmodellen Gebrauch. In Abschnitt 8.1 wird außerdem auf den Anlernalgorithmus aus Kapitel 5 zurückgegriffen. In Abschnitt 8.3 ist eine Anwendung der Zweipunktmodelle für eine größere Anzahl von Benutzern beschrieben.

8.1 MEDARPA - MEDical Augmented Reality for PATients

Begleitend zum medizinisch ausgerichteten AR-Projekt MEDARPA (MEDical Augmented Reality for PATients) [Schwald 02b], [Schnaider 03], [Schwald 04d], [Schwald 04e] wurden verschiedene Schwerpunkte dieser Arbeit erschlossen und entsprechend umgesetzt. Dabei handelt es sich um die Weiterentwicklung der Multipunktmodelle aus Abschnitt 4.3 und deren automatisches Anlernen aus Kapitel 5. MEDARPA wurde von der European Association for Computer Graphics (Eurographics) mit dem „EG2005 Medical Prize“ (2. Platz) ausgezeichnet.

Die Zielstellung des Projekts MEDARPA war es, minimal-invasive Operationstechniken durch AR-Technologie zu unterstützen. Minimal-invasiv bedeutet, dass nur durch kleine Öffnungen am Patientenkörper operiert wird, statt dem Körper große Inzisionen zuzufügen. Während die in den letzten Jahren zunehmend akzeptierten minimal-invasiven Methoden klare Vorteile für den Patienten bedeuten, zum Beispiel ein vermindertes Trauma nach dem Eingriff, schnellere Erholung nach der Operation und bessere kosmetische Resultate, haben sie auch einen großen Nachteil: die beschränkte Sicht des Arztes auf den Bereich der Intervention, da in den Körper des Patienten nur durch kleine Zugänge vorgedrungen wird. Die Verwendung von AR bietet dabei die Möglichkeit, das Problem der beschränkten Sicht zu reduzieren. Abbildung 8.1 zeigt einen Testaufbau des Systems in einer klinischen Umgebung. Die drei klinischen Szenarien, für die das MEDARPA-System entwickelt wurde, sind:

- Herzchirurgie: Unterstützung bei der Suche nach einer optimalen Port-Platzierung für einen total endoskopischen Koronararterien Bypass (TECAB).



Abbildung 8.1. MEDARPA-System bei einem Testaufbau in einem Operationssaal des Frankfurter Klinikums.

- Bronchoskopie: Bronchoskopische Punktion verdächtiger Regionen.
- Brachitherapie: Assistenz für die Navigation von Brachitherapie-Kathetern im Patienteninneren. Brachitherapie ist eine Bestrahlung ganz nah an oder innerhalb von Tumoren.

Zahlreiche AR-Anwendungen nutzen Head Mounted Displays um eine erweiterte Sicht für den Benutzer zu ermöglichen. Da dies die Sicht des Arztes auf den Patienten aber aufgrund eines eingeschränkten Sichtfelds und unzureichender Transparenz deutlich verschlechtert, wurde in MEDARPA ein transparentes Display als Ausgabegerät entwickelt. Dieses, an einem Gelenkarm befestigte AR-Fenster, bereits in Abbildung 5.2 dargestellt, kann während der Intervention einfach über den Patienten geschwenkt oder wieder aus dem Operationsbereich entfernt werden.

Alle Komponenten dieses Systems sind an einem Trolley angebracht, an dem sich auch der Gelenkarm mit dem Display befindet, siehe Abbildung 8.2. Die Sicht des operierenden Arztes auf den Patienten wird durch ein vorab mit einem Computertomographen (CT) aufgenommenes 3D-Modell des Patienteninneren, sowie zusätzliche virtuelle Informationen, wie eine virtuelle Verlängerung eines Instruments, erweitert. Um diese Informationen räumlich korrekt in Bezug auf den Patienten darstellen zu können, müssen Position und Orientierung des Patienten, des Instrumentes, des Displays und die Augenposition des Arztes kontinuierlich in Echtzeit bestimmt werden. Diese Aufgaben übernimmt ein hybrides Trackingsystem, bestehend aus einem kommerziellen elektromagnetischen Subsystem und dem auf dieser Arbeit basierenden optischen Trackingsystem. Die Verwendung der beiden Subsysteme erlaubt es, die unterschiedlichen Stärken der Systeme - Kabellosigkeit des optischen Trackings und keine Verdeckungsproblematik des elektromagnetischen Trackings - zu nutzen.

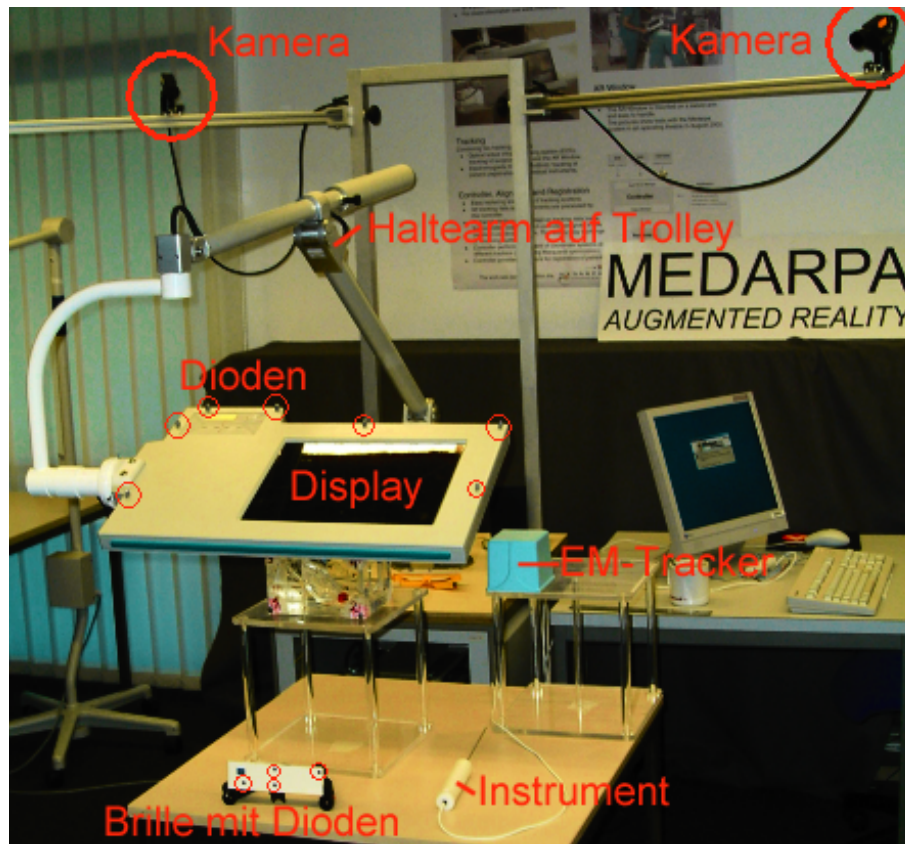


Abbildung 8.2. Die für das Infrarottracking relevanten Bestandteile des MEDARPA Systems.

Das optische Trackingsystem basiert auf einem Zweikamerasystem mit Infrarotfiltern, das, wie in Abbildung 8.2 ersichtlich, auf zwei Armen am Trolley befestigt ist. Optisch verfolgt werden eine Augenposition des Arztes sowie das Display. Eine mechanische Verfolgung des Displays durch Messgeräte in den Gelenken des Gelenkarms wäre möglich, aber in der erforderlichen Qualität und Genauigkeit sehr schwierig umzusetzen. In das Gehäuse des Displays sind sieben Infrarotdioden eingelassen, auf einem Brillengestell, das zur Bestimmung einer Position der Augen des Arztes dient, sind vier weitere Dioden befestigt, siehe Abbildung 4.11. Die Infrarotdioden fungieren als aktive Marker und machen das Trackingsystem weniger anfällig gegen ungünstige Lichtverhältnisse in der Umgebung des Systems, als bei einem Ansatz mit passiven Markern. Da die verwendeten Infrarotdioden einen Abstrahlwinkel von immerhin ca. 90° haben, allerdings nicht rundum gleichmäßig abstrahlen können, sind sie in unterschiedlichen Neigungen angebracht, sodass das Multipunktmodell insgesamt besser zu verfolgen ist. Das Anlernen des Multipunktmodells auf dem Display ist ausführlich in Abschnitt 5.2 beschrieben. Das kalibrierte optische Trackingsystem weist in dem Interaktionsvolumen von $1,5 \text{ m} \times 0,8 \text{ m} \times 1,0 \text{ m}$ eine statische Positionsgenauigkeit von $\approx 1,0 \text{ mm}$ auf.

Das elektromagnetische Tracking wird für die Verfolgung medizinischer Instrumente

eingesetzt, da unterhalb des AR-Fensters, dort wo das Instrument normalerweise am Patienten verwendet wird, ein optisches Tracking kaum möglich ist. Für die Navigation mit einem Instrument im Patienten ist es am wichtigsten, dass die Lagerung des Patienten relativ zum Instrument möglichst exakt bestimmt wird. Aus diesem Grund wird der Patient ebenfalls mit dem elektromagnetischen Tracking erfasst, um den weiter unten beschriebenen Abgleich zum optischen Trackingsystem dafür nicht berücksichtigen zu müssen. Diese und weitere Überlegungen zu einer hybriden Verwendung zweier Trackingsysteme mit sechs Freiheitsgraden sind ausführlicher in [Schwald 04c] beschrieben.

Um die Lagerung des Patienten zu bestimmen, und damit den vorab aufgenommenen Volumendatensatz des Patienteninneren richtig auf den Patienten abzubilden, werden vor der Aufnahme der Volumendaten mit dem CT kleine, für die Verwendung im CT vorgesehene und im Volumendatensatz gut sichtbare, Marker auf dem Patienten angebracht. Die Koordinaten dieser Markierungen lassen sich dann automatisch segmentieren, sodass deren Koordinaten in einem lokalen Koordinatensystem vorliegen. Die Markierungen selbst werden vor Verwendung des MEDARPA-Systems mit einem Registrierungsgerät, in das ein elektromagnetischer Sensor eingelassen ist, eingemessen, sodass deren Koordinaten auch im Koordinatensystem des elektromagnetischen Trackingsystems vorliegen und später im gemeinsamen globalen Koordinatensystem angegeben werden können.

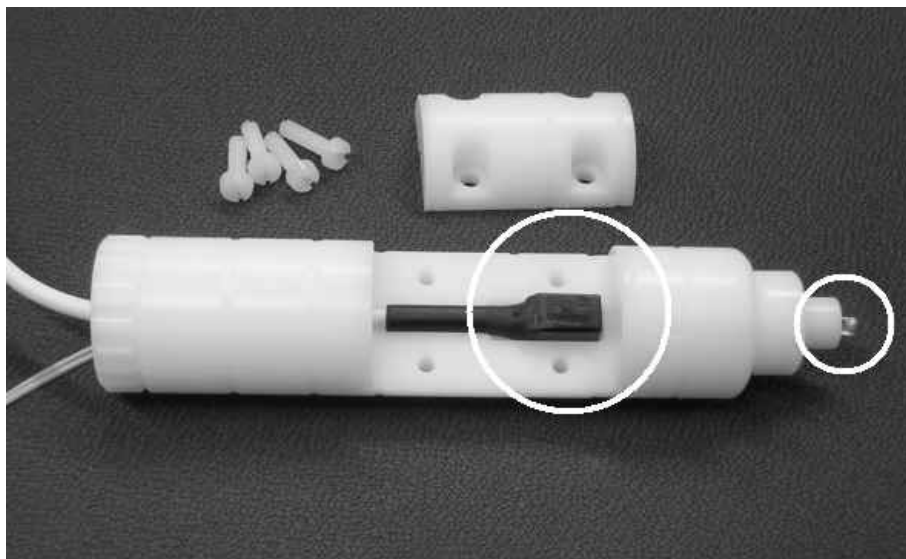


Abbildung 8.3. Für den Abgleich der Trackingsysteme ist im Griff ein elektromagnetischer Sensor eingelassen und an dessen Ende eine Infrarotdiode befestigt.

Um beide Trackingsysteme parallel für die Verfolgung unterschiedlicher Objekte einsetzen zu können, müssen beide Trackingsysteme Daten in einem gemeinsamen Koordinatensystem ausgeben. Um dies zu erreichen, werden die beiden Trackingsysteme in einer Prozedur abgeglichen, bei der an einem Griff sowohl ein elektromagnetischer Sensor, als auch eine Infrarotdiode befestigt sind, siehe Abbildung 8.3. Während dieser Prozedur wird der Griff innerhalb des Interaktionsbereichs beider Trackingsysteme geschwenkt, so-

dass mit einem Zeitstempel versehene Positionsdaten beider Trackingsysteme vorliegen. Die Transformation zwischen den beiden Trackingsystemen sowie ein Zeitversatz werden anhand dieser Datenbasis daraufhin optimiert. Eine detaillierte Beschreibung dieses Verfahrens, sowie eine Auswertung findet sich in [Schwald 04b].

Neben der Verfolgung von Brillengestell und Display unter Verwendung der Multi-punktmodelle, siehe Abschnitt 4.3, muss das optische Trackingsystem also noch während des Abgleichs einen einzelnen Marker verfolgen. Dabei kommt es besonders auf die Zuverlässigkeit des Systems an, da sich Fehler bei der Berechnung der Transformationen aus den Daten des einzeln aufgenommenen Markers auf spätere Ergebnisse auswirken. Sind alle Kalibrierungs- und Abgleichvorgänge erfolgreich abgeschlossen, ist das AR-System einsatzbereit. Abbildung 8.4 zeigt den Blick des Arztes durch das Display, hier auf ein antropomorphes Phantom. Das medizinische Instrument mit einem weißen Griff wird grün überlagert und virtuell verlängert, sodass sich Schnittpunkte mit Zielregionen bestimmen lassen.



Abbildung 8.4. Sicht des Arztes durch das AR-Display mit grüner Überlagerung des Instruments.

Das MEDARPA-System wurde 2003 in drei Krankenhäusern installiert. Ende 2003 und im Laufe von 2004 folgte daraufhin eine Evaluationsphase mit den bereitgestellten Prototypen in den Kliniken [Wesarg 04], [Röddiger 04]. In einer ungestörten Umgebung ist das System nach Aussage von beteiligten Ärzten einfach zu handhaben. Nach der Installation des Systems werden durchschnittlich sechs Minuten für die Kalibrierung des optischen Systems sowie den Abgleich- und Patientenregistrierungsvorgang benötigt. Das System ist für die Unterstützung bei der Planung von Interventionen und zu Trainingszwecken ge-

eignet. Tests unter klinischen Bedingungen haben für eine Punktionsnadel mit einer Länge von ≈ 15 cm ergeben, dass eine Navigationsgenauigkeit von ca. 6 mm erreicht wird. Dieses Ergebnis ist schlechter als unter Laborbedingungen, aber dennoch eine Hilfe für die ansonsten praktisch blind durchzuführenden Interventionen. Der Fehler entsteht insbesondere durch die Länge und die Elastizität der Nadel, die sich bei Druck gegen das zu durchdringende Gewebe verbiegt.

8.2 Korrektur von Magnetresonanz-Tomographie-Aufnahmen

Die Magnetresonanz-Tomographie (MRT) ist eine Form der Bildgebung für das Patienteninnere in der heutigen Medizin. Dabei stellen die Bewegungen des aufzunehmenden Patienten einen limitierenden Faktor in zahlreichen Anwendungen der Magnetresonanz-Bildgebung (MRI, Magnetic Resonance Imaging) dar. Für die Aufnahme des Kopfes muss der Patient seine Kopfposition über längere Zeit unverändert lassen, um eine gute Aufnahme zu ermöglichen. Eine Methode, um die nahezu unvermeidbaren leichten Kopfbewegungen bei der Bildaufnahme auszugleichen, ist die Verwendung eines optischen Trackingsystems [Lerner 06]. In dem in [Dold 04] beschriebenen Ansatz wird ein Infrarottrackingsystem basierend auf der in dieser Arbeit entwickelten Methoden eingesetzt.



Abbildung 8.5. Aufbau des Infrarottrackingsystems vor einem MR-Scanner.

Abbildung 8.5 zeigt den Aufbau des Infrarottrackingsystems vor dem Magnetresonanz-Tomographen im Kontext des MRI-MARCB (MRI System Development Correcting Motion Artifacts for Cardiac and Brain Diagnostic Applications) Projekts. Die großen magnetischen Felder ($\geq 1,5$ Tesla) innerhalb von MRT-Systemen erfordern dabei einen hinreichend großen Abstand des Kamerasystems vom MRT-System, um Funktionalität und Sicherheit zu gewährleisten. Dementsprechend wurden Kameras mit einer großen Brennweite benutzt, um trotz eines großen Abstands der Kameras vom Interaktionsvolumen, Kopfbewegungen möglichst genau zu erfassen.

Die verwendeten Marker sind passive, retroreflektierende Kugeln, die mit einer speziellen Flüssigkeit gefüllt sind, sodass sie sowohl vom optischen Trackingsystem, als auch

vom MRT-System erkannt werden und die Koordinatensysteme beider Systeme in Einklang gebracht werden können.

Verfolgt wird nur ein Multipunktmodell, welches sich auf einem Mundstück befindet, sodass es fest mit dem Patienten verbunden werden kann, siehe Abbildung 8.6. Dies ermöglicht es, noch während der Aufnahme der Daten die Positionierung des Bildvolumens vorausschauend anzupassen. Die für diese speziellen Tests ermittelten Resultate für das Trackingsystem ergaben eine Positionsgenauigkeit von $\approx 0,1$ mm (RMS). Der Mittelwert für die zu messende Reproduzierbarkeit von Rotation und Translation betrug $0,008^\circ$ beziehungsweise $0,067$ mm. Störungen in den MRT-Aufnahmen, die durch Wechselwirkungen zwischen dem MRT-System und dem Trackingsystem auftreten könnten, wurden nicht festgestellt.



Abbildung 8.6. Eine Testperson mit einem Mundstück innerhalb der Kopfspule, aber außerhalb des Scanners.

Die Auswertung der unter Einsatz des Trackingsystems aufgenommenen Testobjekte zeigte insgesamt die grundsätzliche Verwendbarkeit eines externen optischen Trackingsystems. Die Tests wurden an einem Rotationsphantom und in In-Vivo-Experimenten an gesunden Personen durchgeführt. Außerdem waren die Ergebnisse der korrigierten Bilder trotz eines geringen Qualitätsverlusts durch das Rauschen der Trackingergebnisse gut genug, um den Aufwand der Durchführung solch einer vorausschauenden Kopfkorrektur zu rechtfertigen.

8.3 Multiuser-Anwendung im Cybernarium

Das im Jahr 2004 im Cybernarium in Darmstadt installierte Virtual-Reality-Exponat „Unter Spannung - der interaktive Weg des Stroms“ beinhaltet die Implementierung der Zweipunktmodelle aus Kapitel 4.2. Dieses Exponat aus dem Edutainment Bereich soll einer Gruppe von sieben oder acht Mitspielern, insbesondere Schülern, helfen, die Frage zu beantworten, wie der Strom in die Steckdose kommt [Schwald 05c].

Die sieben oder acht Mitspieler sind dabei Teil des Stromnetzes. Sie stehen vor einer Projektionsleinwand, auf der für jeden Mitspieler ein Element des Stromnetzes, zum Beispiel ein Kraftwerk, ein Umspannungswerk oder ein Mietshaus, abgebildet ist. Jeder Mitspieler trägt eine Kappe, auf der zwei fest miteinander verbundene Marker angebracht sind. Bewegt sich ein Mitspieler und damit auch eine Kappe, bewegt sich auch ein dem Mitspieler zugeordnetes Element des Stromnetzes auf der Leinwand, siehe Abbildung 8.7. Befinden sich zwei passende Elemente des Stromnetzes in unmittelbarer Nähe, baut sich eine Verbindung zwischen ihnen auf. Ziel ist es, das Stromnetz aufzubauen, sodass der Strom aus dem Kraftwerk bis in die Wohnungen fließen kann.



Abbildung 8.7. Sieben Personen bewegen ein jeweils ihnen zugeordnetes Objekt an der Projektionsleinwand mit Hilfe der Kappen.

Der Aufbau dieses Exponats besteht aus einer ca. 4 m × 6 m großen Projektionsleinwand, zwei PCs und zwei Kameras mit Infrarotscheinwerfern, die am oberen Rand der Projektionsleinwand befestigt sind. Die Kameras sind so ausgerichtet, dass sie beide einen ca. 3 m × 4 m großen Bereich vor der Leinwand erfassen können. Dies ist der Interaktionsbereich, in dem sich die Mitspieler bewegen können. Auf einem PC läuft die

Rendering Applikation, auf dem anderen das Tracking, beide sind durch ein Netzwerk verbunden.

Als erster Schritt müssen die physikalischen Modelle, bestehend aus zwei Markern, erstellt werden. Um die Gefahr einer Verwechslung der Modelle gering zu halten, aber um die Modelle auch nicht zu groß werden zu lassen, wurden für die acht Modelle Distanzen von 70 mm bis 175 mm im Abstand von jeweils 15 mm gewählt. Kleinere Modelle wurden nicht verwendet, da sie sich als ungünstig aufgrund zu häufiger Selbstverdeckung erwiesen. Eine Größe von 175 mm für ein Modell erlaubt gerade noch eine Befestigung auf einer Kappe, ohne dass die Marker weit abstehen. Die beiden Marker werden so befestigt, dass sie ideal von den Kameras erfasst werden, wenn der Benutzer zur Leinwand sieht. Abbildung 4.6 aus Kapitel 4.2 zeigt ein Beispiel für solch eine Kappe, die Marker sind hintereinander angeordnet.

Die Distanzen können im größten Teil des Interaktionsvolumens mit einer Genauigkeit von ca. 3 mm gemessen werden. An den Randbereichen, insbesondere je weiter man sich von der Leinwand und damit den Kameras entfernt, sind die Ergebnisse häufig schlechter. Für die Distanztoleranz wurde ein Wert von $\delta = 5$ mm gewählt, der zum einen damit großzügig gewählt ist, andererseits die Verwechslungsgefahr von Modellen ausschließt.

Für die Verwendung eines Infrarottrackingsystems für dieses Exponat spricht die Anforderung, dass die Mitspieler kabellos verfolgt werden sollen. Außerdem ist der Raum, in dem das Exponat installiert ist, abgedunkelt, um eine gute Projektion auf der Leinwand zu ermöglichen, was den Einsatz eines Infrarottrackingsystems besonders sinnvoll macht.

8.4 Zusammenfassung

Die in diesem Kapitel aufgeführten Anwendungen beschreiben den praktischen Einsatz der Implementierung einiger der in dieser Arbeit beschriebenen Punktmodelle und Algorithmen innerhalb größerer Projekte.

Im Rahmen des medizinischen Augmented Reality Projekts MEDARPA wurden insbesondere die Multipunktmodelle und deren automatisches Anlernen entwickelt. Innerhalb dieses Projekts wurde auch die hybride Nutzung des optischen Trackingsystems mit einem elektromagnetischen Trackingsystem umgesetzt. Außerdem wurden die Multipunktmodelle für die Korrektur von MRT Aufnahmen eingesetzt.

Als Technologie für ein interaktives Spiel mit bis zu acht Benutzer wurden die Zweipunktmodelle eingesetzt. Die Marker befinden sich dabei auf den Kappen der beteiligten Spieler.

Kapitel 9

Zusammenfassung und Ausblick

9.1 Zusammenfassung

Das Ziel dieser Arbeit bestand darin, auf Basis der durch ein Stereokamerasystem zur Laufzeit rekonstruierten Menge von 3D-Punkten, Algorithmen zu entwickeln, die in Echtzeit das Tracking von Objekten und/oder Personen vor allem für den Mixed Reality Bereich zulassen. Diese Algorithmen sollten vor allem für technisch weniger aufwändige Systeme, also insbesondere für im VR- und AR-Bereich übliche Zweikamerainfrarot-trackingsysteme, verwendbar sein. Dabei sollten die durch verschiedene Anwendungen vorgegebenen Anforderungen an die Trackingalgorithmen berücksichtigt werden und soweit möglich deren kombinierte Nutzung unterstützt werden.

Die im Rahmen dieser Arbeit entwickelten Verfahren erfüllen diese Voraussetzungen und wurden in Algorithmen und Modelle für das Tracking starrer und dynamischer Objekte unterteilt. Ein besonderer Schwerpunkt lag auf der Beschreibung des neu entwickelten und parametrisierbaren Algorithmus und dem Entwurf der Multipunktmodelle, die vielfältig einsetzbar sind. Durch die Festlegung verschiedener Parameter kann der Algorithmus dabei so modifiziert werden, dass er bestimmte erwünschte Eigenschaften, wie geringe Verarbeitungszeit oder Robustheit, stärker oder schwächer berücksichtigt.

Das neu entwickelte Verfahren zum automatischen Anlernen von Multipunktmodellen durch freie Bewegung des zu verfolgenden Objekts erlaubt ein Anlernen komplexer Modelle auch mit einem Trackingsystem mit nur zwei Kameras. Neben den Multipunktmodellen wurden als starre Modelle Ein- und Zweimarkerverfahren mit den entsprechenden Algorithmen präsentiert. Die Zweipunktmodelle eignen sich dabei insbesondere für Anwendungen, in denen die Interaktion zahlreicher Benutzer erforderlich ist.

Die im Rahmen dieser Arbeit entwickelten dynamischen Gruppenmodellen bieten die Möglichkeit einer direkteren Form der Interaktion, beispielsweise durch Verfolgung der Hände und einzelner spezieller Gesten. Die Modelle bestehen dabei nicht aus einer festen Anordnung von Markern sondern einer veränderlichen.

Die Möglichkeiten zur kombinierten Nutzung der verschiedenen Arten von Modellen dieses Konzepts innerhalb einer Anwendung wurden beschrieben. Die gemeinsame Nutzung des 3D-Punktmodells, welches Zusatzinformationen wie die letzte Position, den Klassifikationsindex oder den Rekonstruktionsfehler enthält, unterstützt dabei die

Möglichkeit zur Kombination.

Weitere Grundlagen für eine gesamte Trackingverarbeitungs pipeline, also Kalibrierung der Kameras und Bildverarbeitung, Rekonstruktion der 3D-Punkte sowie die Filterung der Trackingergebnisse, wurden unter Berücksichtigung der Beeinflussung auf die zu entwickelnden Trackingalgorithmen kurz dargestellt.

Die praktische Verwendbarkeit der in dieser Arbeit beschriebenen und implementierten Verfahren wurde anhand verschiedener umgesetzter Anwendungen dokumentiert.

9.2 Ausblick

Neue Formen der Mensch-Maschine-Interaktion gewinnen in vielen Bereichen zunehmend an Bedeutung. Dies drückt sich unter anderem in den jüngeren wissenschaftlichen Bereichen Ambient Intelligence beziehungsweise Pervasive Computing aus, bei denen Computertechnologie unscheinbar in das Umfeld der Benutzer integriert werden soll. Optisches Tracking ist dabei ein Mittel, um neue Interaktionsformen mit dem Computer zu ermöglichen.

Aber auch im klassischen VR-Bereich, beziehungsweise im AR-Bereich, ist die Interaktion mit den virtuellen Welten oft noch von klassischen Eingabegeräten, wie Maus und Tastatur, bestimmt. Insbesondere bei einer Interaktion im Dreidimensionalen bietet das optische Tracking die Grundlage für natürlichere Formen der Interaktion. In diesem Zusammenhang ist die Weiterentwicklung der beidhändigen Interaktion mit den hier beschriebenen Handgruppenmodellen, unter Umständen auch für mehrere Benutzer gleichzeitig, ein Bereich für zukünftige Arbeiten.

Optische Trackingsysteme, insbesondere Infrarottrackingsysteme, sind seit wenigen Jahren bereits auch kommerziell erhältlich, allerdings sind diese Systeme relativ teuer, technisch aufwändig und meist auf nur eine Art des Trackings im Sinne der in dieser Arbeit beschriebenen Modelle spezialisiert. Sehr preisgünstige Systeme haben im Allgemeinen einen extrem eingeschränkten Funktionsumfang, wie zum Beispiel die Verfolgung nur eines einzelnen Punktes. Daher ist eine Anforderung für zukünftige optische Trackingsysteme eine höhere Flexibilität in der Verwendbarkeit mit geringerem technischen Aufwand zu erreichen. Diese Flexibilität drückt sich in dieser Arbeit durch die Kombination der verschiedenen Modelltypen beziehungsweise Trackingalgorithmen innerhalb eines Systems dar. Das Konzept dieser Kombinierbarkeit von Modelltypen ist in zukünftigen Arbeiten fortzuführen und zu erweitern.

Das automatische Anlernen von Modellen wurde im Fall der Multipunktmodelle auch im Hinblick auf komplexere Strukturen und Systeme mit wenigen Kameras umfassend beschrieben. Für dynamische Modelle ist das Anlernen ebenfalls wichtig aber auch schwieriger, da sich die dynamischen Modelle weniger verallgemeinern lassen, als die statischen Modelle. Anlernalgorithmen, die mit nur sehr wenig oder gar keinen Randbedingungen auskommen, um die Strukturen dynamischer Modelle aufzubauen, sind ein weiterer Punkt für zukünftige Untersuchungen und Entwicklungen.

Anhang A

Darstellung von Rotationen

Rotationen sind im Mixed Reality Bereich unverzichtbar und häufig beschrieben, dennoch bereiten sie häufig Probleme oder sorgen beispielsweise bei der Definition von Schnittstellen für Missverständnisse. Diese Missverständnisse entstehen zum einen durch die verschiedenen Repräsentationsmöglichkeiten von Rotationen und zum anderen durch einfache Fragestellungen wie: eine mittlere Translation zwischen zwei Translationen zu bestimmen ist trivial, aber was ist die mittlere Rotation zwischen zwei Rotationen? Diese Frage ist insbesondere für Filterung oder Prädiktion interessant und wurde daher in Abschnitt 7.1.1 aufgegriffen.

Ziel dieses Anhangs ist es nicht, das Thema Rotationen erschöpfend zu bearbeiten, sondern die für diese Arbeit relevanten Aspekte zusammenzustellen und den Begriff Rotation, wie er hier verwendet wird, zu verdeutlichen. Insbesondere steht der Begriff Rotation hier immer für Rotationen im Raum (\mathbb{R}^3), Rotationen in der Ebene spielen keine Rolle. Es gibt zahlreiche weiterführende Literatur, wobei eine gute Zusammenstellungen der Konversionen zwischen verschiedenen Repräsentationen von Rotationen unter anderem in [Shoemake 85] und [Eberly 06] zu finden sind.

Eine Rotation ist die Drehung eines festen Körpers im Raum (hier immer euklidischer Vektorraum), die in verschiedenen Repräsentationen dargestellt werden kann. Im Wesentlichen und am häufigsten verwendet sind dies:

- Rotationsmatrizen $R \in SO_3 \in \mathbb{R}^{3 \times 3}$,
- Euler-Winkel (α, β, γ) mit $\alpha, \beta, \gamma \in [0, 2\pi]$,
- Achse und Drehwinkel $A = (x, y, z)^T \in \mathbb{R}^3, \alpha \in [0, 2\pi]$ und
- (Einheits-)Quaternionen $q = ((x, y, z), w)$ mit $(x, y, z)^T \in \mathbb{R}^3, w \in \mathbb{R}$.

Die verschiedenen Repräsentationen haben Vor- und Nachteile, sodass eine Verwendung unterschiedlicher Repräsentationen je nach Kontext sinnvoll ist. Die in dieser Arbeit wichtigste Darstellungsform ist die Rotationsmatrix, sodass der Ausdruck Rotation hier synonym für deren Repräsentation als Rotationsmatrix verwendet wird. Die spezielle orthogonale Gruppe

$$SO_3 = \{R \in GL_3 | R^T R = I \wedge R = I\} \quad (\text{A.1})$$

ist die Gruppe der Rotationen, was auch die für die praktische Berechnung effiziente Eigenschaft $R^{-1} = R^T$ impliziert. Außerdem bedeutet dies, dass in einer Rotationsmatrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} | & | & | \\ x & y & z \\ | & | & | \end{pmatrix} \quad (\text{A.2})$$

spaltenweise die zueinander orthogonalen, normierten Richtungsvektoren x, y und z der Hauptachsen des gedrehten Koordinatensystems stehen. Multipliziert man beispielsweise R mit $e_1 = (1, 0, 0)^T$ erhält man x , die Darstellung von e_1 im neuen Koordinatensystem.

Eine Verwendung von Euler-Winkeln bietet sich nur in wenigen Fällen an. Ihr Vorteil liegt in der - im Vergleich zu den anderen - einfachen Interpretationsmöglichkeit der Drehung um drei Achsen und ihrer minimalen Darstellungsform aus drei Parametern. Daher sind die Euler Winkel besonders dann sinnvoll, wenn von einem Benutzer Eingaben gemacht werden müssen oder sollen. Ansonsten sind die anderen Repräsentationen zu bevorzugen.

Schon der Begriff Euler-Winkel ist irreführend, da Euler-Winkel eigentlich eine zusammengesetzte Drehung um drei Achsen bedeuten, deren erste und letzte Drehung um dieselbe Achse geht, zum Beispiel x-y-x (erste um die x-Achse, dann um die y-Achse und dann wieder um die x-Achse). Eine Drehung x-y-z, wie sie meist verwendet wird, wird eigentlich als Cardan-Winkel bezeichnet. Zusammen gibt es zwölf verschiedene Anordnungen, auf welche Art gedreht werden kann, wobei deren Reihenfolge nicht einfach vertauscht werden kann. Weitere Probleme sind das Gimbal Lock (wenn durch die zweite Drehung die dritte Achse nahezu parallel zur ersten gedreht wird) und die Doppeldeutigkeiten bei der Konversion von Rotationsmatrizen zu Euler-Winkeln.

Konversion	A	M	D	F
Achse/Drehwinkel zu Matrix	13	15		2
Matrix zu Achse/Drehwinkel	8	7	1	2
Achse/Drehwinkel zu Quaternion	1	5	1	2
Quaternion zu Achse/Drehwinkel		4		2
Quaternion zu Matrix	12	12		
Matrix zu Quaternion	6	5	1	1

Tabelle A.1. Operationsanzahl für die Konversion zwischen Repräsentationen, Angaben gemäß [Eberly 06].

Die (Einheits-)Quaternionen sind ebenso wie die Rotationsmatrizen schwieriger zu interpretieren, aber sehr gut zu verwenden. Einheitsquaternionen sind normierte Vektoren mit vier Parametern, lassen sich bis auf das Vorzeichen eindeutig aus den Rotationsmatrizen bestimmen und deren Multiplikation ist äquivalent zu der der Rotationsmatrizen, allerdings viel effizienter, siehe Tabelle A.3. Über die Verwendung von Quaternionen

nen in der Computergraphik findet sich Grundlegendes zum Beispiel in [Shoemake 85], [Shoemake 91], [Shoemake 92].

Die Repräsentation durch Achse und Drehwinkel ist den Quaternionen sehr ähnlich und etwas anschaulicher, Bedarf für Transformation oder Komposition mehr Aufwand, da immer zunächst in eine andere Darstellungsform konvertiert werden muss.

Repräsentation	A	M	
Rotationsmatrix	6	9	
Achse und Drehwinkel	12	18	
Quaternion	24	32	generische Quaternionmultiplikation
Quaternion	17	24	spezielle Quaternionmultiplikation
Quaternion	18	21	mit Konversion zu Matrix

Tabelle A.2. Operationsanzahl für die Transformation eines Vektors, Angaben gemäß [Eberly 06].

Die Anzahl an Operationen für Konversion (Tabelle A.1), Transformation (Tabelle A.2) und Komposition (Tabelle A.3) sind in den entsprechenden Tabellen dargestellt, die [Eberly 06] entnommen sind. Dabei werden Addition und Subtraktion (A), Multiplikation (M), Division (D) und teure Evaluierungen (F) berücksichtigt. Da bei den in dieser Arbeit dargestellten Trackingalgorithmen hauptsächlich Vektoren transformiert werden und weniger verschiedene Rotationen komponiert werden, ist daher auch die Rotationsmatrix die wichtigste Repräsentation.

Repräsentation	A	M	D	F
Rotationsmatrix	18	27		
Quaternion	12	16		
Achse und Drehwinkel (mit Konversion zu Quaternion)	14	30	2	6

Tabelle A.3. Operationsanzahl für die Komposition zweier Rotationen (Multiplikation), Angaben gemäß [Eberly 06].

Anhang B

3D-Punkt Residuum

Das C3D- (Coordinate 3D) Dateiformat [Dainis 05] dient der Speicherung von 3D-Koordinaten und zugehöriger Parameter für eine Messreihe. Es ist eine Komponente einer Familie von Dateiformaten, die ursprünglich Ende der 1980-er Jahre für das AMASS (ADTech Motion Analysis Software System) Photogrammetrie Softwaresystem entwickelt wurde, um die durch ein Kamerasystem gewonnenen Daten zu speichern. AMASS speichert seine Ausgabedaten in C3D-Dateien und wurde von Andrew Dainis entwickelt. Motion Capturing Systeme, z.B die von Vicon [Vicon 06], unterstützen das C3D-Format.

Die wesentliche Idee beim C3D-Format ist, dass alle 3D-Koordinaten oder numerischen Daten, die beispielsweise von biomechanischen Messsystemen kommen können, für jede aufgenommene Messung zusammen mit verschiedenen Parametern in einer Datei gespeichert werden. Das Format selbst umfasst viele Parameter und wird ausführlich in [Dainis 05] beschrieben.

Das Interesse gilt der Speicherung der 3D-Koordinaten mit den Informationen über die Messung:

- Genauigkeit (durchschnittlicher Fehler oder Residuum) und
- der Angabe, welche Kameras die Daten erfasst haben.

Bei einem Zweikamerasystem ist der letzte Punkt nicht interessant, da beide Kameras benötigt werden. In diesem Fall entspricht das 3D-Punkt-Residuum des C3D-Formats, welches eine relative Genauigkeit der 3D-Messung eines Punktes angibt, der Hälfte des Rekonstruktionsfehlers aus Abschnitt 3.2.1, siehe Abbildung B.1(a).

Grundsätzlich gilt, dass eine ungenaue Messung oder schlechte Kalibrierung große Residuen erzeugt, aber bei Messungen mit zwei Kameras kleine Residuen nicht notwendigerweise bedeuten, dass die Messungen sehr genau waren.

Eine Messung mit drei Kameras bezieht einen dritten Strahl ein, der normalerweise in der Nähe des berechneten Markerzentrums aus zwei Strahlen verläuft, siehe Abbildung B.1(b). Mit einer Kleinste-Fehlerquadrate-Methode lässt sich dann ein Punkt berechnen, sodass die Summe der Abstände zwischen diesem Punkt und den Strahlen minimal ist. Auch bei drei Kameras bedeuten kleine Residuen nicht notwendigerweise, dass die Messungen sehr genau waren. Dies ist jedoch wahrscheinlicher als bei der Verwendung von

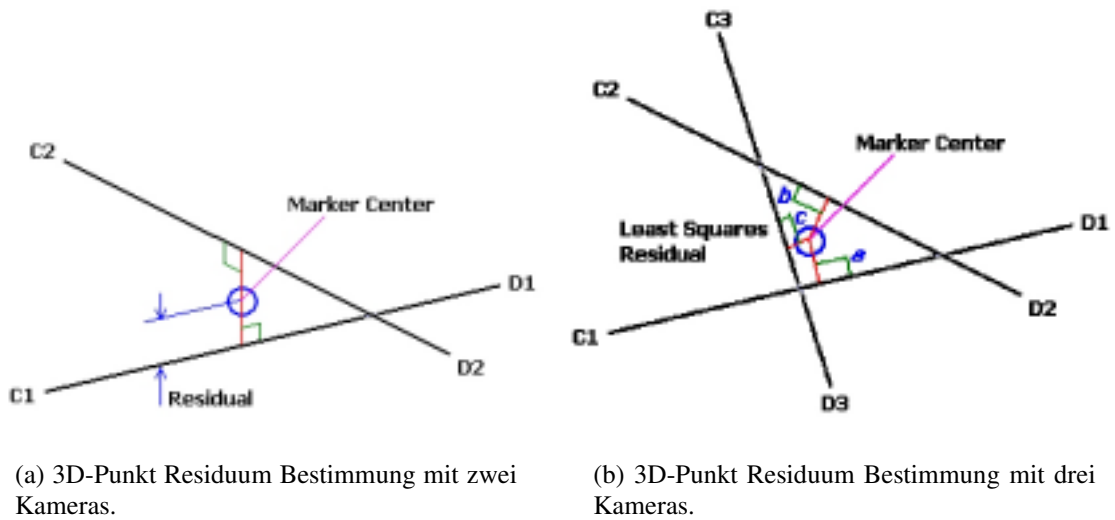


Abbildung B.1. Bestimmung von 3D-Punkt Residuen aus [Dainis 05].

zwei Kameras. Im Allgemeinen sind die Residuen, die man aus zwei Kameras erhält, geringer, als diejenigen, die man aus drei erhält, allerdings impliziert dies nicht, dass Messungen mit zwei Kameras genauer sind. In [Dainis 05] wird außerdem angeregt, Residuen als Indikatoren zu nutzen: Ein Residuum von -1.0 für ungültige Punkte und ein Residuum von 0.0 für modellierte Punkte.

Anhang C

Beispiel zur Identifikation mehrerer einzelner Marker

Das hier folgende Beispiel stellt einen Schritt des in Abschnitt 4.1.3 beschriebenen Algorithmus zur Identifikation mehrerer einzelner Marker dar. Anhand dieses Schrittes soll die Wirkung der einzelnen Parameter verdeutlicht werden. Dabei gelten folgende Voraussetzungen:

- Es existieren drei Listen L_1, L_2 und L_3 von 3D-Punkten mit
 - Identifikationsindizes $id_1 = 4, id_2 = 2$ und $id_3 = 1$,
 - Vertrauenswerten $t_1 = 1, 0, t_2 = 0, 5$ und $t_3 = 1, 0$,
 - den letzten Indizes $lst_1 = 2, lst_2 = 1, lst_3 = 3$ und
 - den für den kommenden Schritt jeweils prädierten Positionen P_1, P_2 und P_3 .
- Die für alle Listen gültigen Parameter sind
 - der Grenzwert $d_{max} = 50$ mm,
 - der Reduktionswert $t_{red} = 0, 25$ und
 - die Gewichtungparameter $w_1 = 1, 0$ und $w_2 = 0, 8$.
- Die aktuell rekonstruierten 3D-Punkte sind $\mathcal{X}_W = \{X_W^1, X_W^2, X_W^3, X_W^4\}$.

Zunächst wird eine Zuordnung der neuen Punkte zu den Listen anhand der letzten Indizes lst_i versucht. Dabei sei

$$\begin{aligned} |P_1 - X_W^2| &= 5,0 \text{ mm} < d_{max}, \\ |P_2 - X_W^1| &= 7,0 \text{ mm} < d_{max} \text{ und} \\ |P_3 - X_W^3| &> d_{max}. \end{aligned} \tag{C.1}$$

Dementsprechend wird die vorläufige Matrix W' , deren Elemente Wahrscheinlichkeitswerte sind, als 4×3 -Nullmatrix angelegt und die Elemente $W'_{12} = w_1 \cdot \frac{50-7}{50} = 0, 86$ und

$W'_{21} = w_1 \cdot \frac{50-5}{50} = 0,9$ gesetzt. Die Anzahl der Zeilen der Matrix W' entspricht dabei der Anzahl der rekonstruierten 3D-Punkte \mathcal{X}_W und die Anzahl der Spalten von W' der Anzahl der bereits existierenden Listen L_i . Die Matrix W' hat also folgende Form:

$$W' = \begin{pmatrix} 0 & 0,86 & 0 \\ 0,9 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (\text{C.2})$$

Im nächsten Schritt werden alle Punkte X_W^i auf ihre Distanz zu den prädierten Positionen P_j untersucht. Dabei sei $|P_3 - X_W^4| = 10, 0 < d_{max}$ und $|P_2 - X_W^4| = 10, 0 < d_{max}$, alle anderen Kombinationen ergeben eine Distanz die größer ist als d_{max} . Entsprechend werden die Elemente W'_{42} und W'_{43} nun mit $W'_{42} = W'_{43} = w_2 \frac{50-10}{50} = 0,64$ belegt, sodass nun

$$W' = \begin{pmatrix} 0 & 0,86 & 0 \\ 0,9 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0,64 & 0,64 \end{pmatrix}. \quad (\text{C.3})$$

ist. Die anderen Elemente von W' bleiben unverändert. Als nächstes werden die Vertrauenswerte der Listen berücksichtigt. Die Vertrauenswerte sind die Elemente der Diagonalmatrix D_t . Nun wird die Matrix W wie folgt berechnet:

$$\begin{aligned} W &= \begin{pmatrix} 0 & 0,43 & 0 \\ 0,9 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0,32 & 0,64 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 0 & 0,86 & 0 \\ 0,9 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0,64 & 0,64 \end{pmatrix}}_{W'} \underbrace{\begin{pmatrix} 1,0 & 0 & 0 \\ 0 & 0,5 & 0 \\ 0 & 0 & 1,0 \end{pmatrix}}_{D_t} \end{aligned} \quad (\text{C.4})$$

Der Matrix W kann man nun entnehmen, dass X_W^2 der Liste L_1 zugeordnet wird, da das Element W_{21} den größten Wert enthält. Der Punkt X_W^4 wird der Liste L_3 und der Punkt X_W^1 der Liste L_2 zugeordnet. Für X_W^3 wird eine neue Liste L_4 mit Identifikationsindex 3 oder 5 angelegt. Als letztes werden die Vertrauenswerte neu gesetzt: Die Vertrauenswerte der Listen, denen ein Punkt zugeordnet werden konnte, werden auf 1,0 gesetzt, also $t_1 = 1,0, t_2 = 1,0$ und $t_4 = 1,0$. Der Vertrauenswert $t_3 = 1,0 - t_{red} = 0,75$ wird reduziert, da kein Punkt der entsprechenden Liste zugeordnet werden konnte. Wäre $t_{red} = 1,0$ gewählt worden, hätte die Liste L_2 in diesem Beispiel nicht existieren können und die Liste L_3 wäre nach diesem Schritt aufgelöst worden.

Anhang D

Beispiel zur Reduktion der Liste von Punktpaaren bei Zweipunktmodellen

Das folgende Beispiel soll die in Abschnitt 4.2.3 beschriebene Reduktion einer Liste von Punktpaaren X_{ij} veranschaulichen. Die Punktconstellation in Abbildung D.1 ähnelt der Punktconstellation in Abbildung 4.7 und skizziert die Ausgangssituation für das Beispiel.

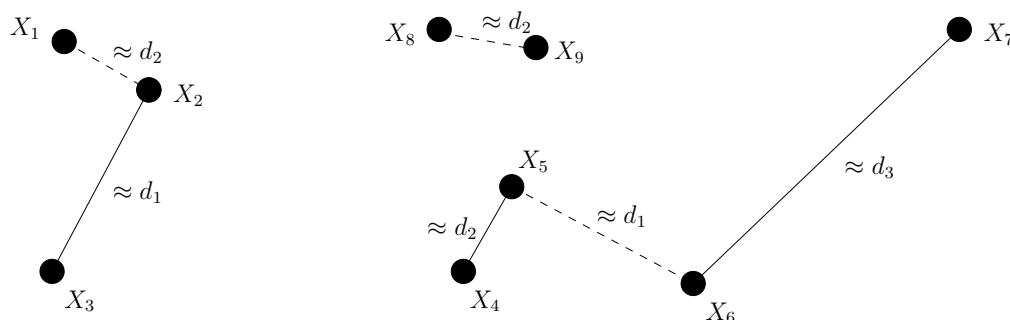


Abbildung D.1. Mehrdeutigkeiten (gestrichelte Linien) durch drei Zweipunktmodelle (durch durchgezogene Linien gekennzeichnet) und drei weitere einzelne 3D-Punkte.

Es wurden neun 3D-Punkte X_1, \dots, X_9 rekonstruiert. Dabei gilt unter anderem $\|X_2 - X_3\| - d_1 < \delta$, wobei d_1 eine der gesuchten Distanzen und δ die gegebene Distanztoleranz ist. Insgesamt gilt für die drei gesuchten Distanzen d_1, d_2 und d_3 :

$$\begin{aligned} d_1 &\approx X_{23} \approx X_{56}, \\ d_2 &\approx X_{12} \approx X_{45} \approx X_{89}, \\ d_3 &\approx X_{67}. \end{aligned} \tag{D.1}$$

Es liegen also sechs Punktpaare für drei zu findende Distanzen vor. Die Listen L_i der Punktpaare für jeden einzelnen Punkt werden wie folgt aufgebaut:

$$\begin{aligned}
L_1 &:= \{X_{12}\}, & L_2 &:= \{X_{12}, X_{23}\}, & L_3 &:= \{X_{23}\}, \\
L_4 &:= \{X_{45}\}, & L_5 &:= \{X_{45}, X_{56}\}, & L_6 &:= \{X_{56}, X_{67}\}, \\
L_7 &:= \{X_{67}\}, & L_8 &:= \{X_{89}\} & L_9 &:= \{X_{89}\}.
\end{aligned} \tag{D.2}$$

Für jedes Punktpaar wird außerdem noch der Wert v_{ij} berechnet, der die Anzahl der Punktpaare angibt, deren Abstand derselben Distanz entsprechen:

$$\begin{aligned}
v_{67} &= 1, \\
v_{23} = v_{56} &= 2 \\
v_{12} = v_{45} = v_{89} &= 3.
\end{aligned} \tag{D.3}$$

Im ersten Schritt der Reduktion soll die Liste L_2 mit $l_2 := |L_2| = 2$ reduziert werden. Sie hat gemeinsame Punkte mit L_1 und L_3 . Beide Listen haben nur ein Element, allerdings hat das Punktpaar mit v_{12} den größeren Wert. Also wird das Punktpaar X_{12} aus L_1 und L_2 gestrichen und $v_{45} = v_{89} = 2$ gesetzt.

Im zweiten Schritt der Reduktion soll die Liste L_5 mit $l_5 = 2$ reduziert werden. Sie hat gemeinsame Punkte mit L_4 und L_6 . Da $l_4 < l_6$ ist, wird das Punktpaar X_{56} aus L_5 und L_6 gestrichen und $v_{23} = 1$ gesetzt.

Im dritten Schritt der Reduktion liegt keine Liste L_i mit $l_i > 1$ mehr vor. Daher wird die Reduktion beendet. Die Mehrdeutigkeit zwischen den Punktpaaren X_{45} und X_{89} wird zu einem späteren Zeitpunkt innerhalb des in Abschnitt 4.2.1 beschriebenen Algorithmus aufgelöst.

Abbildungsverzeichnis

1.1	Gesamtkonzept dieser Arbeit. In horizontaler Ausrichtung sind aufeinanderfolgende Verarbeitungsschritte beschrieben und in vertikaler Richtung die Bandbreite der verschiedenen Ansätze.	6
2.1	Unterschiedliche Platzierung und Ausrichtung zweier Kameras in der Aufsicht. Der Interaktionsbereich ist grau unterlegt.	10
2.2	Das Lochkameramodell zeigt das Kamera-, Kamerabild-, und Pixelkoordinatensystem und einen auf die Kamerabildebene projizierten Punkt X_c	12
2.3	Aufnahme eines Kalibrierungspatterns ohne Überlagerungen des Kamerabildes (Overlays) und mit Overlays nach der Segmentierung, als Visualisierung der Segmentierungsergebnisse.	15
2.4	Relative spektrale Empfindlichkeiten beziehungsweise Durchlässigkeiten. Der ausgefüllte Bereich zwischen der CCD Kurve und der des Infrarotfilters wird von den Kameras erfasst.	17
2.5	Kamerabild mit und ohne Infrarotfilter zeigt ein mit aktiven Markern versehenes Objekt.	19
2.6	Zusammenfügung von Pixeln zu Liniensegmenten und dann zu Regionen bei einer intensitätsbasierten Segmentierung. Die Quadrate repräsentieren Pixel, deren Grauwert über dem Schwellwert liegen. Das Bild wurde aus [Figueiredo 02] entnommen.	20
2.7	Pattern, die aus punktförmigen Elementen bestehen.	21
2.8	Monoskopisches modellbasiertes Tracking aus [Comport 04].	22
3.1	Durch Infrarotfilter aufgenommenes Bildpaar eines Stereokamerasystems zeigt ein mit Markern versehenes Objekt. Nach der Segmentierung lässt sich eine Menge von 2D-Punkten aus den weiß erscheinenden Markern extrahieren.	25
3.2	Vergleich zwischen verschiedenen Markern.	26
3.3	3D-Rekonstruktion und Rekonstruktionsfehler aus zwei Kameras mit Projektionszentren P_1 und P_2	27
3.4	Farblich dargestellter Rekonstruktionsfehler an 2600 3D-Messpunkten.	30
3.5	3D-Rekonstruktion eines Punktes mit dessen Projektion auf die Kameraebenen.	31
3.6	Durchschnittlicher Fehler der Projektion auf die mit Linsenverzeichnung kalibrierten Kameras in Pixeln.	32

3.7	Farblich dargestellte Summe der 2D-Fehler, die bei der Projektion der 2600 rekonstruierten 3D-Punkte auf die Kameraebenen entstehen.	33
3.8	Farblich dargestellte Verschiebung der 2600 3D-Messpunkte durch gewichtete Mittelung in Abhängigkeit der Entfernung zu den Hauptpunkten statt arithmetischer Mittelung.	34
3.9	Toleranzbereich um die Epipolarlinie von U^1 auf der zweiten Bildebene. Zwei Punkte liegen im Toleranzbereich, der gesuchte Punkt U^2 und ein weiterer, sich zufällig innerhalb des Toleranzbereichs befindender, Punkt.	37
3.10	Algorithmus zur Umsetzung der Eindeutigkeitsvermutung.	39
4.1	Verfolgung der Kopfposition mit einem Marker an Shutterbrille.	47
4.2	Trajektorien von drei Markern. Jeder rekonstruierte 3D-Punkt wird als Kugel, der Identifikationsindex als Farbe dargestellt. Die Trajektorie im Vordergrund (rot) wird trotz kurzer Unterbrechung korrekt fortgesetzt, die mittlere Trajektorie (grün) hat keine Unterbrechung, die rechte Trajektorie (blau) erhält nach der Unterbrechung fälschlicherweise einen neuen Index.	49
4.3	3D-Punkten zugeordnete Indizes auf die Kamerabildern überlagert.	50
4.4	Aufbau der Liste der Punktlisten.	51
4.5	Algorithmus für die Verfolgung mehrerer 3D-Punkte.	52
4.6	Eine mit zwei Infrarotmarkern besetzte Kappe.	55
4.7	Mehrdeutigkeiten (gestrichelte Linien) durch drei Zweipunktmodelle (durch durchgezogene Linien gekennzeichnet) und einzelne 3D-Punkte.	55
4.8	Algorithmus zur Erkennung mehrerer 3D-Punktpaare.	57
4.9	Reduktion der Liste der Punktpaare.	60
4.10	Erkennung der Modelle aus der sortierten Liste der Punktpaare X_{ij}	61
4.11	Brillengestell als Teil eines AR-Systems zur Bestimmung der Kopfposition des Benutzers. Vier Dioden werden als aktive Marker verwendet.	62
4.12	Transformationen für die Bestimmung der Abbildung von Welt- auf Modellkoordinatensystem. Dem Modell sind hier sieben Punkte zugeordnet, wobei nur die drei schwarz markierten Punkte die tatsächlich getrackten Marker darstellen.	64
4.13	Verschiedene Anforderungen an Trackingstrategien für Multipunktmodelle, die nicht alle gleichermaßen erfüllt werden.	68
4.14	Suche nach Korrespondenzen zwischen sechs getrackten Punkten im Weltkoordinatensystem (WKS) und fünf Modellpunkten anhand von Distanzen. Die jeweils aktuell relevanten Punkte sind schwarz eingefärbt, die anderen grau.	70
4.15	Trackingalgorithmus für Multipunktmodelle basierend auf Detektion mit und ohne Vorwissen aus dem letzten Frame.	75
4.16	Ungünstige Konstellationen von Punkten für die Detektion.	81
4.17	Ungünstige Konstellationen von Punkten für die Detektion in rechteckiger Anordnung.	82
4.18	Ungünstige Konstellationen von Punkten für die Detektion.	82

5.1	AR-System für minimal-invasive Chirurgie. Infrarotkameras verfolgen das mit Markern besetzte AR-Display.	86
5.2	Links: AR-Display mit Infrarotdioden als Marker aus einer der Kameraperspektiven. Rechts: Kamerabild mit den Überlagerungen der detektierten Marker.	87
5.3	Der Lernalgorithmus für Multipunktmodelle.	88
5.4	Die graphische Oberfläche des implementierten Lernalgorithmus zeigt Distanzmessungen (links) und die daraus abgeleitete Definition eines Dreipunktmodells (rechts).	93
5.5	Schrittweises Hinzufügen neuer 3D-Punkte.	94
5.6	Drei weitere Marker sind auf der Rückseite des Displays angebracht. Im Bild sind vier der sieben alten Marker sichtbar, drei sind verdeckt.	96
5.7	Kalibrierung einer Brachitherapienadel mit integriertem elektromagnetischem 6DOF-Sensor. Die Instrumentenspitze ist an einem Punkt fixiert, während das Instrument in verschiedene Richtungen ausgelenkt und rotiert wird.	99
5.8	Aufsicht auf die Messungen mit der an neun verschiedenen Punkten fixierten Instrumentenspitze. Auf der linken Seite sind die Originalmesswerte des Sensors abgebildet, auf der rechten Seite dieselben Messungen, nachdem auf alle Messpunkte die berechnete Transformation von Sensor (altes LKS) zu Instrumentenspitze (neues LKS) angewendet wurde.	100
5.9	Ein Interaktionsgerät, an dem Marker fest angebracht sind, wird so im WKS platziert, dass das neue LKS daran ausgerichtet werden kann. Die Kamerabilder sind mit zusätzlichen Informationen überlagert.	101
5.10	Graphische Oberfläche für die interaktive, freie Platzierung eines lokalen Koordinatensystems relativ zu den Modellpunkten.	102
6.1	Zwei Handschuhe mit je drei angenähten Markern für eine Verfolgung mit Hilfe Dynamischer Modelle. Die Marker am Mittelfinger sind an den beiden Handschuhen auf unterschiedliche Weise angebracht.	104
6.2	Phasen für die Verfolgung und Erkennung von dynamischen Gruppenmodellen.	105
6.3	Gruppierung von 3D-Punkten aufgrund der räumlichen Distanzen zwischen ihnen. Die Punkte werden durch die Kugeln repräsentiert.	106
6.4	Ein auf verschiedene Arten mit drei Infrarotmarkern besetzter Handschuh für Mensch-Maschine-Interaktion. Die virtuelle Zeigerichtung und deren Erzeugung sind mit rot eingezeichnet.	109
6.5	Interaktion mit zwei Handschuhen.	110
6.6	Schematischer Ablauf der Verfolgung von zwei Handgruppenmodellen, davon ein Modell für eine Zeigegeste, das zweite Modell für Selektion.	113
6.7	Kugelförmige Markierungen, die auf der Patientenoberfläche angebracht werden und im CT gut sichtbar sind.	116
7.1	Der Kreislauf des Kalman Filters zur Filterung von Rotationen und Translationen.	127

8.1	MEDARPA-System bei einem Testaufbau in einem Operationssaal des Frankfurter Klinikums.	130
8.2	Die für das Infrarottracking relevanten Bestandteile des MEDARPA Systems.	131
8.3	Für den Abgleich der Trackingsysteme ist im Griff ein elektromagnetischer Sensor eingelassen und an dessen Ende eine Infrarotdiode befestigt.	132
8.4	Sicht des Arztes durch das AR-Display mit grüner Überlagerung des Instruments.	133
8.5	Aufbau des Infrarottrackingsystems vor einem MR-Scanner.	135
8.6	Eine Testperson mit einem Mundstück innerhalb der Kopfspule, aber außerhalb des Scanners.	136
8.7	Sieben Personen bewegen ein jeweils ihnen zugeordnetes Objekt an der Projektionsleinwand mit Hilfe der Kappen.	137
B.1	Bestimmung von 3D-Punkt Residuen aus [Dainis 05].	145
D.1	Mehrdeutigkeiten (gestrichelte Linien) durch drei Zweipunktmodelle (durch durchgezogene Linien gekennzeichnet) und drei weitere einzelne 3D-Punkte.	148

Tabellenverzeichnis

4.1	In ≈ 700 Messungen wurden zwischen 0 und 4 Modelle getrackt. Δ ist die Verarbeitungszeit für die entsprechende Anzahl von getrackten Modellen, beziehungsweise Punkten und Distanzen.	78
5.1	Distanzmessungen (je Versuch 100 Messungen) mit Mittelwert, Standardabweichung und maximaler Abweichung in mm.	91
5.2	Mittelwerte und mittlere, beziehungsweise maximale Distanzfehler des angelernten Modells in mm aus acht Versuchen.	94
5.3	Distanzfehler für jeden Versuch und Punkt im Vergleich zum jeweiligen Mittelwert der Punkte aus Tabelle 5.3.	95
5.4	Distanzfehler für jeden Versuch und die ersten sieben Punkte des Modells im Vergleich zu den Mittelwerten der Punkte aus Tabelle 5.2.	97
5.5	Mittelwerte und mittlere, beziehungsweise maximale Distanzfehler des angelernten Modells in mm aus acht Versuchen.	97
5.6	Distanzfehler der drei zusätzlich angebrachten Marker: in der oberen Hälfte wurde der Vergleich zum Mittel einschließlich Versuch 3 und 5 berechnet. In der unteren Hälfte wurde der Vergleich zum Mittel ohne die beiden Versuche berechnet.	98
6.1	Sinnvolle Kombinationsmöglichkeit verschiedener Modelle für die Verfolgung von Objekten. Dabei steht S1 für einzelne Marker, S2 für Zweipunktmodelle, S3 für Multipunktmodelle, DGP für dynamische Gruppenmodelle, DGK für dynamische Gelenkmodelle und DF für dynamische Flächenmodelle. Eine gut umzusetzende und sinnvolle Kombination wird mit + markiert, eine umsetzbare, aber eher ungeeignete Kombination mit \circ und eine ungeeignete Kombination mit $-$	119
A.1	Operationsanzahl für die Konversion zwischen Repräsentationen, Angaben gemäß [Eberly 06].	142
A.2	Operationsanzahl für die Transformation eines Vektors, Angaben gemäß [Eberly 06].	143
A.3	Operationsanzahl für die Komposition zweier Rotationen (Multiplikation), Angaben gemäß [Eberly 06].	143

Index und Abkürzungen

A

AR - Augmented Reality 1

C

CT - Computertomograph 130

D

Distanztoleranz 42

DM - Dynamisches Modell 103

DOF - Degrees of freedom 2

E

EKF - Extended Kalman Filter 126

EOS 17

Epipolar.....

-kriterium 36

-kandidat 38

-toleranz 37

Essentialmatrix 14

extrinsisch 14

F

Frame 9

Framerate 8

Fundamentalmatrix 14

I

ICP - Iterative Closest Point 117

Identifikationsindex 45

Infrarottracking 16

aktiv 16

passiv 16

Interaktionsbereich 9

intrinsisch 14

K

Kalman Filter 126

Kamerakalibrierungsmatrix 13

Kameramodell 11

Klassifikationsindex 36

Korrespondenz.....

-analyse 24

zwischen Welt- und Modellpunkt 63

L

LM - Levenberg-Marquardt 15

Lochkamera 11

LKS - lokales Koordinatensystem.... 97

M

Marker 25

Medarpa 86, 129

Modell 43

MPM - Multipunktmodell 62

MR - Mixed Reality 1

MRI - Magnetic Resonance Imaging 135

MRT - Magnetresonanz-Tomographie 135

Multikamerasystem 34

O

o.B.d.A. - ohne Beschränkung der Allgemeinheit 69

P

Pattern 14

Punkt 25

R

Rekonstruktion 24

Rekonstruktionsfehler 29
RMS - root mean square 29

S

Segmentierung 18
Stereokamerasystem 8
Störung 46
SVD - Singulärwertzerlegung 15
Synchronisation 8

T

Trackingsystem 8
Triangulierung 26
 Hartley-Sturm 33

U

Updaterate 9

V

Vertrauenswert 50, 57
VR - Virtual Reality 1

W

WKS - Welkoordinatensystem .. 27, 100
Wurzepunkt 35

2

2D-Punktmodell 39

3

3D-Punkt 25
 -modell 35
3DOF-Trackingverfahren 54

Literaturverzeichnis

- [Alexa 02] M. Alexa. *Linear Combinations of Transformations*. In SIGGRAPH 2002, ACM Transactions on Graphics, volume 21, pages 380–387, 2002.
- [ART 06a] ART. *advanced realtime tracking GmbH, Produktseiten*. <http://www.ar-tracking.de>, Juli 2006.
- [ART 06b] ART. *advanced realtime tracking GmbH, Produktseiten: tracking software DTrack*. <http://www.ar-tracking.de>, Juli 2006.
- [Auer 00] T. Auer. *Hybrid Tracking for Augmented Reality*. PhD thesis, Technische Universität Graz, 2000.
- [Azarbayejani 95] A. Azarbayejani & A. Pentland. *Camera Self-Calibration from One Point Correspondence*. Perceptual computing technical report 341, MIT media laboratory, 1995.
- [Azuma 97] R. Azuma. *Correcting for Dynamic Error*. In SIGGRAPH '97 Course Notes 30: Making Direct Manipulation Work in Virtual Reality, August 1997.
- [Beier 03] D. Beier, R. Billert & B. Brüderlin. *Model-Based Object Recognition and Localization for Augmented Reality*. In Proceedings of MIRAGE 2003, INRIA Rocquencourt (France), 2003.
- [Belta 02] C. Belta & V. Kumar. *An SVD-Based Projection Method for Interpolation on $SE(3)$* . IEEE Transactions on Robotics and Automation, vol. 18(3), June 2002.
- [Besl 92] P. Besl & N. McKay. *A Method for Registration of 3-D Shapes*. In IEEE Trans. Pattern Anal. Machine Intell., volume 14, pages 239–256, 1992.
- [Bowman 00] D. Bowman, E. Kruijff, J. LaViola & I. Poupyrev. *The art and science of 3d interaction*. Notes from the IEEE Virtual Reality 2000 Conference, IEEE Computer Society, 2000.
- [Claron 06] Claron. *Micron Tracker, Produktseiten*. <http://www.clarontech.com>, Juli 2006.

- [Comport 04] A. I. Comport, E. Marchand & F. Chaumette. *Robust model-based tracking for robot vision*. In Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'04), Sendai, Japan, 2004.
- [Cordea 01] M. D. Cordea, E.M. Petriu & N. D. Georganas et. al. *3D Head Pose Recovery for Interactive Virtual Reality Avatars*. In Proceedings of IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, 2001.
- [Curtis 93] W. Dan Curtis, Adam L. Janin & Karel Zikan. *A Note on Averaging Rotations*. In Proceedings of IEEE Virtual Reality Annual International Symposium, VRAIS'93, pages 377–386, September 1993.
- [Dainis 05] A. Dainis. *The C3D File Format User Manual*. Motion Lab Systems, Inc., 2005. www.c3d.org, Juli 2005.
- [Dold 04] C. Dold, E. Firle, M. Zaitsev, G. Sakas, O. Speck, J. Hennig & B. Schwald. *Prospective Head Motion Compensation by Updating the Gradients of the MRT*. In Proceedings of Medical Robotics, Navigation and Visualization (MRNV) 2004, Remagen, Germany, 2004.
- [Dorfmueller-Ulhaas 01] K. Dorfmueller-Ulhaas & D. Schmalstieg. *Finger tracking for interaction in augmented environments*. In Proceedings of the 2nd ACM/IEEE International Symposium on Augmented Reality (ISAR'01), New York NY, USA, 2001.
- [Dorfmueller-Ulhaas 03] K. Dorfmueller-Ulhaas. *Robust Optical User Motion Tracking Using a Kalman Filter*. Technical Report 2003-6, Klaus Dorfmueller-Ulhaas, May 2003.
- [Dorfmueller 98] K. Dorfmueller & H. Wirth. *Real-Time Hand and Head Tracking for Virtual Environments Using Infrared Beacons*. In International Workshop on Modelling and Motion Capture Techniques for Virtual Environments, CAPTECH'98, pages 113–127. Springer Verlag, 1998.
- [Dorfmueller 99a] K. Dorfmueller. *An Optical Tracking System for VR/AR-Applications*. In Proceedings of the Eurographics Workshop Virtual Environments 99. Springer ComputerScience, Vienna, Austria, 1999.
- [Dorfmueller 99b] K. Dorfmueller. *Robust tracking for augmented reality using retroreflective markers*. Computers & Graphics, no. 23, pages 795–800, 1999.
- [Eberly 06] D. Eberly. *Rotation Representations and Performance Issues*. Technical report, Geometric Tools, Inc., 2004, <http://www.geometrictools.com/Documentation/RotationIssues.pdf>, Juli 2006.

- [Faugeras 95] O. Faugeras & B. Mourrain. *On the geometry and algebra of the point and line correspondence between n images*. In 5th International Conference on Computer Vision, MIT, pages 951–956. IEEE, 1995.
- [Faugeras 01] O. Faugeras & Q. T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.
- [Figueiredo 02] P. Figueiredo. *Automatic Learning and Detection of Point-Based Models*. Master's thesis, ZGDV, Darmstadt, Germany and Minho University, Portugal, September 2002. Supervised by Bernd Schwald.
- [Fiorentino 03] M. Fiorentino, G. Monno, P. A. Renzulli & A. E. Uva. *3D Pointing in Virtual Reality: Experimental Study*. In Tools and Methods Evolution in Engineering Design, Italy, 2003.
- [Frantz 04] D. D. Frantz, S. R. Kirsch & A. D. Wiles. *Specifying 3D Tracking System Accuracy*. In Workshop Bildverarbeitung für die Medizin, Berlin, 2004.
- [Fukushige 06] S. Fukushige & H. Suzuki. *Motion Tracking with Reflections*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 428–434, Setubal, Portugal, 2006.
- [Gavrila 99] D. M. Gavrila. *The Visual Analysis of Human Movement: A Survey*. In Computer Vision and Image Understanding, volume 73, pages 82–98. Academic Press, 1999.
- [Gibson 03] S. Gibson, R. J. Hubbold, J. Cook & T. L.J. Howard. *Interactive Reconstruction of Virtual Environments from Video Sequences*. Computers & Graphics, vol. 27(2), pages 293–301, April 2003.
- [Golub 89] G. H. Golub & C. F. Van Loan. *Matrix Computations*, 2nd ed. Baltimore: Johns Hopkins University Press, 1989.
- [Govindu 04] V. M. Govindu. *Lie-Algebraic Averaging For Globally Consistent Motion Estimation*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04), pages 684–691, Washington, D.C., USA, 2004.
- [Grossman 01] T. Grossman, R. Balakrishan, G. Kurtenbach, G. Fitzmaurice, A. Khan & B. Buxton. *Interaction Techniques for 3D Modeling on Large Displays*. In Proceedings of the 2001 ACM Symposium on Interactive 3D Graphics, pages 17–23, 2001.
- [Han 05] J. Y. Han. *Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection*. In Proceedings of the UIST'05, Seattle, Washington, USA, 2005.

- [Hartley 92] R. I. Hartley. *Estimation of relative camera positions for uncalibrated cameras*. In 2nd European Conference on Computer Vision, pages 579–587. Springer Verlag, 1992.
- [Hartley 97] R. Hartley & P. Sturm. *Triangulation*. Computer Vision and Image Understanding, vol. 68, no. 2, pages 146–157, 1997.
- [Hartley 04] R. Hartley & Zissermann. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [Hornung 04] A. Hornung & L. Kobbelt. *Robust and Automatic Optical Motion Tracking*. In S. Müller, editor, *Virtuelle und Erweiterte Realität*, 1. Workshop der GI-Fachgruppe VR/AR, Chemnitz, 2004.
- [Jain 95] R. Jain, R. Kasturi & B. G. Schunk. *Machine vision*. McGraw-Hill, New York, 1995.
- [Jung 00] J. Jung, B. Adelstein & S. Ellis. *Predictive Compensator Optimization for Head Tracking in Virtual Environments*. In Proceedings of IMAGE 2000, Scottsdale, Arizona, USA, 2000.
- [Kalman 60] R. E. Kalman. *A new approach to linear filtering and prediction problems*. J. Basic Eng., vol. 82D, pages 35–45, 1960.
- [Kanatani 93] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, Oxford, 1993.
- [Kanbara 01] M. Kanbara, N. Yokoya & H. Takemura. *A stereo vision-based augmented reality registration with extendible tracking of markers and natural features*. In Proceedings IEE Conference on Computer Vision and Pattern Recognition (CVPR2001), volume Technical Sketches, pages 1045–1048, 2001.
- [Kang 06] W. Kang & S. Eiho. *3D Tracking Using 2D-3D Line Segment Corresponding and 2D Point Motion*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 278–285, Setubal, Portugal, 2006.
- [Kato 99] H. Kato & M. Billinghurst. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. In Proceedings of the 2nd International Workshop of Augmented Reality (IWAR 99), San Francisco, USA, 1999.
- [Kato 06] H. Kato, M. Billinghurst & P. Lamb. *ARToolKit 2.71 Software*. <http://www.hitl.washington.edu/artoolkit/>, Juli 2006.

- [Kim 04a] H. Kim, G. Albuquerque, S. Havemann & D. Fellner. *3D Modelling with Hand Gesture Interaction in a Semi-Immersive Environment*. Technical Report TUBSCG-2004-01, Institute of Computer Graphics, TU Braunschweig, 2004.
- [Kim 04b] H. Kim & W. Fellner. *Interaction with Hand Gesture for a Back-Projection Wall*. In Proceedings of Computer Graphics International 2004, pages 395–402, Crete, Greece, 2004.
- [Klette 96] R. Klette, A. Koschan & K. Schlüns. *Computer Vision - Räumliche Information aus digitalen Bildern*. Friedr. Vieweg & Sohn, Braunschweig, 1996.
- [Kohler 96] Markus Kohler. *Vision Based Remote Control in Intelligent Home Environments*. In B.Girod, Heinrich Niemann & Hans-Peter Seidel, editors, 3D Image Analysis and Synthesis'96, pages 147–154, University of Erlangen-Nürnberg/Germany, November 18–19th 1996. Infix-Verlag.
- [Kretschmer 04] U. Kretschmer. *Tracking einer Person im urbanen Umfeld mit Positions- und Blickrichtungsbestimmung auf Basis eines 3D-Stadtmodells*. PhD thesis, Darmstadt Technische Universität, 2004.
- [Lenz 87] Reimar Lenz. *Linsenfehlerkorrigierte Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D-Messungen in Echtzeit*. In Erwin Paulus, editor, Mustererkennung 1987, 9. DAGM-Symposium, Braunschweig, volume 149. Springer, 1987.
- [Lerner 06] T. Lerner, E. Rivlin & M. Gur. *Vision-Based Tracking System for Head Motion Correction in fMRI Images*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 213–220, Setubal, Portugal, 2006.
- [Madritsch 96] F. Madritsch & M. Gervautz. *CCD-Camera Based Optical Beacon Tracking for Virtual and Augmented Reality*. Eurographics, vol. 3, no. 15, 1996.
- [Magneau 02a] O. Magneau, P. Bourdot & R. Gherbi. *3D tracking based on infrared cameras*. In International Conference on Computer Vision and Graphics, Zakopane, Poland, September 2002.
- [Magneau 02b] O. Magneau, P. Bourdot & R. Gherbi. *Positioning and identification of markers for 3D tracking*. In Virtual Concept, Biarritz, France, October 2002.
- [Malerczyk 04] C. Malerczyk. *Interactive Museum Exhibit Using Pointing Gesture Recognition*. In Journal of WSCG, volume 12, pages 165–172, Plzen, Czech Republic, 2004.

- [Marquardt 63] D. W. Marquardt. *An algorithm for least-squares estimation of nonlinear parameters*. Journal of the Society for Industrial and Applied Mathematics, vol. 11, pages 431–441, 1963.
- [Medarpa 06] Medarpa. <http://www.medarpa.de>, Juli 2006.
- [Moakher 02] Maher Moakher. *Means and Averaging in the Group of Rotations*. SIAM J. Matrix Anal. Appl., vol. 24(1), pages 1–16, 2002.
- [Moeslund 01] Thomas B. Moeslund & Erik Granum. *A Survey of Computer Vision-Based Human Motion Capture*. Computer Vision and Image Understanding, vol. 81, pages 231–268, 2001.
- [Mulder 03] J. Mulder, J. Jansen & A. van Rhijn. *An Affordable Optical Head Tracking System for Desktop VR/AR systems*. In Proceedings of 7th International Workshop on Immersive Projection Technology, 9th Eurographics Workshop on Virtual Environments, Mai 2003.
- [NDI 06] NDI. *Northern Digital Inc., Produktseiten: Trackingsysteme Polaris und Optotrak*. <http://www.ndigital.com>, Juli 2006.
- [Neugebauer 97] P. J. Neugebauer & K. Klein. *Adaptive Triangulation of Objects Reconstructed from Multiple Range Images*. In IEEE Visualization 97, Late Breaking Hot Topics, Phoenix, Arizona, USA, 1997.
- [Nishino 98] H. Nishino, K. Utsumiya & K. Korida. *3D Object Modeling Using Spatial and Pictographic Gestures*. In Proc. of the ACM symposium on Virtual reality software and technology, pages 51–58, 1998.
- [Orin 06] Orin. *DynaSight Sensor, Produktseiten*. <http://www.orin.com>, Juli 2006.
- [Park 06] J. Park & S. Rhee. *Non-Intrusive Tracking of Multiple Users in a Spatially Immersive Display*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 464–467, Setubal, Portugal, 2006.
- [Pavlovic 97] V. Pavlovic, R. Sharma & T. Huang. *Visual interpretation of hand gestures for human-computer interaction: A review*. In IEEE Transaction on Pattern Analysis and Machine Intelligence, volume 19(7), pages 677–695, 1997.
- [Press 02] W. H. Press, S. A. Teukolsky, W. T. Vetterling & B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd ed. Cambridge: Cambridge University University Press, 2002.
- [Presscott 97] B. Presscott & G. F. McLean. *Line-based correction of radial lens distortion*. In Graphical Models and Image Processing, volume 59(1), pages 39–77, Januar 1997.

- [Rasmussen 06] N. Rasmussen, M. Störring, T. Moeslund & E. Granum. *Real-time Tracking for Virtual Environments Using SCAAT Kalman Filtering and Unsynchronised Cameras*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 333–340, Setubal, Portugal, 2006.
- [Ribo 01] M. Ribo, A. Pinz & A. L. Fuhrmann. *A new Optical Tracking System for Virtual and Augmented Reality Applications*. In IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, May 2001.
- [Ribo 03] M. Ribo, M. Brandner & A. Pinz. *A Flexible Software Architecture for Hybrid Tracking*. In Workshop on Integration of Vision and Inertial Sensors of the 11th International Conference on Advanced Robotics (ICAR) 2003, Coimbra, Portugal, June 2003.
- [Röddiger 04] S. Röddiger, D. Baltas, G. Sakas, M. Schnaider, S. Wesarg, P. Zogal, B. Schwald, H. Seibert & R. Kure. *MEDARPA - Implantation of Brachytherapy-catheters using Augmented Reality*. In Proceedings of Medical Robotics, Navigation and Visualization (MRNV) 2004, Remagen, Germany, 2004.
- [Rusdorf 04a] Stephan Rusdorf & Guido Brunnett. *A Simple VR-Control Device Using Marker-Based Tracking of the Human Hand*. In Proceedings of IEEE Virtual Reality Workshop (VR 2004), Chicgao, USA, 2004.
- [Rusdorf 04b] Stephan Rusdorf & Guido Brunnett. *Kontextabhängige Steuerung in immersiven VR-Umgebungen ohne Verwendung hand-blockierender Eingabe Hardware*. In S. Müller, editor, Virtuelle und Erweiterte Realität, 1. Workshop der GI-Fachgruppe VR/AR, Chemnitz, 2004.
- [Rusinkiewicz 02] S. Rusinkiewicz, O. Hall-Holt & M. Levoy. *Real-Time 3D Model Acquisition*. In ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH 2002, volume 21, 2002.
- [Schnaider 03] M. Schnaider, B. Schwald, H. Seibert & T. Weller. *Medarpa - A Medical Augmented Reality System for Minimal-Invasive Interventions*. In Proceedings of the 11th Annual Medicine Meets Virtual Reality (MMVR) Conference, Newport Beach, CA, USA, January 2003.
- [Schwald 02a] B. Schwald & C. Malerzczyk. *Controlling Virtual Worlds Using Interaction Spheres*. In C.A. Vidal, editor, 5th Symposium on Virtual Reality (SVR), Fortaleza, Brazil, 2002.
- [Schwald 02b] B. Schwald, H. Seibert & T. Weller. *A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window*. In International Symposium on Mixed and Augmented Reality (ISMAR'02), Darmstadt, Germany, September 2002.

- [Schwald 03a] B. Schwald & B. de Laval. *An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context*. In Vaclav Skala, editor, European Association for Computer Graphics (Eurographics): Journal of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Plzen: University of West Bohemia, volume 11(3), pages 425–432, 2003.
- [Schwald 03b] B. Schwald & B. de Laval. *Training and Assistance to Maintenance in an Augmented Reality Environment*. In Julie A. Jacko, editor, Proceedings of the 10th International Conference on Human-Computer Interaction: Cognitive, Social and Ergonomic Aspects, pages 1121–1125. Lawrence Erlbaum, 2003.
- [Schwald 04a] B. Schwald & P. Figueiredo. *Learning of Rigid Point-Based Marker Models for Tracking with Stereo Camera Systems*. In S. Müller, editor, Virtuelle und Erweiterte Realität, 1. Workshop der GI-Fachgruppe VR/AR, pages 23–34, Chemnitz, 2004.
- [Schwald 04b] B. Schwald & H. Seibert. *Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality*. In Vaclav Skala, editor, European Association for Computer Graphics (Eurographics): Journal of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Plzen, University of West Bohemia, Czech Republic, volume 12(3), pages 411–418, 2004.
- [Schwald 04c] B. Schwald, H. Seibert & M. Schnaider. *Composing 6 DOF Tracking Systems for VR/AR*. In Proceedings of Computer Graphics International (CGI) 2004, Crete, Greece, 2004.
- [Schwald 04d] B. Schwald, H. Seibert, M. Schnaider, S. Wesarg & et. al. *Supporting Minimal Invasive Interventions with Augmented Reality*. In Proceedings of the 18th Computer Assisted Radiology and Surgery (CARS 2004) Conference, 2004.
- [Schwald 04e] B. Schwald, H. Seibert, M. Schnaider, S. Wesarg, S. Röddiger & S. Dogan. *Implementation and Evaluation of an Augmented Reality System Supporting Minimal Invasive Interventions*. In International Workshop on Augmented environments for Medical Imaging and Computer-aided Surgery (AMI-ARCS) of MICCAI, 2004. published electronically at <http://publica.fraunhofer.de/eprints/N-26787.pdf>.
- [Schwald 05a] B. Schwald. *A Tracking Algorithm for Point-Based Marker Models*. In Vaclav Skala, editor, European Association for Computer Graphics (Eurographics): Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) 2005, Plzen: University of West Bohemia, pages 61–62, 2005.

- [Schwald 05b] B. Schwald. *Two-Handed Interaction with Marker-Based Optical Tracking*. In J. Zara & J. Sloup, editors, Proceedings of Central European Multimedia and Virtual Reality (CEMVR 2005) Conference, Prag, Czech Republic. European Association for Computer Graphics (Eurographics), 2005.
- [Schwald 05c] B. Schwald, T. Fröhlich & R. Kruse. *Interaction and Design for a Multi-User Edutainment Application on a Large-Scale Screen*. In Proceedings of International Conference on Human-Computer Interaction (HCI) 2005, Las Vegas, USA, 2005.
- [Shafaei 06] S. Shafaei & M. Rahmati. *Human Body Tracking for Physiotherapy Virtual Training*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 449–454, Setubal, Portugal, 2006.
- [Shen 06] S. Shen & W. Chen. *Human Body Tracking Based on Probability Evolutionary Algorithm*. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) 2006, volume 2, pages 303–309, Setubal, Portugal, 2006.
- [Shoemake 85] K. Shoemake. *Animating Rotation with Quaternion Curves*. In Proceedings of SIGGRAPH '85 (San Francisco, California, July 22–26, 1985), In Computer Graphics, volume 19(3), pages 245–254, July 1985.
- [Shoemake 91] K. Shoemake. *Quaternions and 4x4 Matrices*. Graphics Gems II, Academic Press, pages 351–354, 1991.
- [Shoemake 92] K. Shoemake & T. Duff. *Matrix Animation and Polar Decomposition*. In Proceedings of the conference on Graphics interface '92, pages 258–264, Vancouver, British Columbia, Canada, 1992. Morgan Kaufmann Publishers Inc.
- [Sonka 99] M. Sonka, V. Hlavac & R. Boyle. Image processing, analysis and machine vision - 2nd. ed. Brooks/Cole Publishing Company, 1999.
- [Stenger 04] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, St. John's College, March 2004.
- [Trucco 98] E. Trucco & A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [Trucco 99] E. Trucco, A. Fusiello & V. Roberto. *Robust motion and correspondences of noisy 3-D point sets with missing data*. Pattern Recognition Letters, vol. 20, pages 889–898, 1999.
- [Tsai 86] R. Y. Tsai. *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*. Proceedings CVPR'86, pages 364–374, June 1986.

- [Tsai 87] R. Y. Tsai. *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses*. In IEEE Journal of Robotics and Automation, volume RA-3(4), pages 323–344, August 1987.
- [van Liere 04] R. van Liere & J. D. Mulder. *Tangible Devices for Two Handed 3D Interaction*. In Workshop Beyond Glove and Wand Based Interaction, IEEE VR2004, 2004.
- [van Rhijn 04] A. van Rhijn & J. D. Mulder. *Optical Tracking using Line Pencil Fiducials*. In M. Göbel S. Coquillart, editor, Eurographics Symposium on Virtual Environments, 2004.
- [Vicon 06] Vicon. *Vicon Motion Systems, Produktseiten*. <http://www.vicon.com>, Juli 2006.
- [Welch 95] G. Welch & G. Bishop. *An Introduction to the Kalman Filter*. Technical report, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.
- [Welch 01] G. Welch & G. Bishop. *An Introduction to the Kalman Filter*. SIGGRAPH 2001, Course 8, Slides, 2001.
- [Wesarg 04] S. Wesarg, E. A. Firle, B. Schwald, H. Seibert, P. Zogal & S. Röddiger. *Accuracy of needle implantation in brachytherapy using a medical AR system - a phantom study*. In Visualization, Imag-Guided Procedures, and Display, Proceedings of SPIE, volume 5367, San Diego, California, USA, 2004.
- [Yang 01] R. Yang & Z. Zhang. *Model-based Head Pose Tracking With Stereovision, Technical Report MSR-TR-2001-102*. Technical report, Microsoft Research, Redmond, WA 98052, October 2001.
- [Zachmann 00] G. Zachmann. *Virtual Reality in Assembly Simulation - Collision Detection, Simulation Algorithms, and Interaction Techniques*. PhD thesis, Darmstadt Technische Universität, 2000.
- [Zhang 98] Z. Zhang. *A Flexible New Technique for Camera Calibration*. Technical report, Microsoft Research, Redmond, WA 98052, December 1998.
- [Zikan 94a] K. Zikan, W. D. Curtis, H. A. Sowizral & A. L. Janin. *A note on dynamics of human head motions and on predictive filtering of head-set orientations*. In SPIE Proceedings: Telemanipulator and Telepresence Technologies, volume 2351, pages 328–336, 1994.
- [Zikan 94b] K. Zikan, W. D. Curtis, H. A. Sowizral & A. L. Janin. *Fusion of absolute and incremental position and orientation sensors*. In SPIE Proceedings: Telemanipulator and Telepresence Technologies, volume 2351, pages 316–327, 1994.

Lebenslauf

Bernd Schwald

Geburtsdatum: 05.04.1972
Geburtsort: Stuttgart Bad-Cannstatt
Familienstand: ledig, keine Kinder
Staatsangehörigkeit: deutsch
Adresse: Kranichsteiner Str. 63, 64289 Darmstadt

Schulbildung und Zivildienst:

08/1979 - 07/1983 Besuch der Goetheschule (Grundschule) Asperg
08/1983 - 05/1992 Besuch des Friedrich-List-Gymnasiums in Asperg
Abschluss: Abitur
08/1992 - 10/1993 Zivildienst beim Bund für Umwelt- und Naturschutz
(BUND) im Umweltzentrum in Stuttgart

Studium und studienbegleitende Tätigkeiten:

10/1993 - 05/2000 Mathematikstudium mit Nebenfach Informatik an der Universität Stuttgart, abgeschlossen mit Diplomprüfung (Gesamturteil: sehr gut)
Diplomarbeit im Bereich geometrische Datenverarbeitung:
„Exakte Volumenberechnung von durch Doo-Sabin-Flächen begrenzten Körpern“
10/1995 - 12/1996 Hilfswissenschaftler am Mathematischen Institut der Universität Stuttgart
01/1997 - 07/1997 Auslandsstudium an der Universität Uppsala, Schweden
10/1997 - 03/1998 Tutor für Analysis I/II an der Universität Stuttgart
07/1998 - 10/1998 Praktikum im Bereich Qualitätssicherung bei der SAP AG in Walldorf
11/1998 - 04/2000 Studentische Aushilfskraft bei der SAP AG in Walldorf

Beruf:

09/2000 - 10/2006 Wissenschaftlicher Mitarbeiter am Zentrum für Graphische Datenverarbeitung e.V. (ZGDV) in Darmstadt, Abteilung Visual Computing
01/2005 - 10/2006 stellvertretender Abteilungsleiter