



Universität Karlsruhe
Fakultät für Informatik

Institut für Theoretische Informatik
Prof. Dr. Alexander Waibel

Tracking der artikularen Bewegung des Oberkörpers in Stereobildfolgen

Diplomarbeit

von
Julius Ziegler



30. SEPTEMBER 2005

Betreuer:

Prof. Dr. Alexander Waibel
Dipl.-Inform. Kai Nickel
Dr.-Ing. Rainer Stiefelhagen

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Julius Ziegler

Karlsruhe, den 30. September 2005

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation	8
1.2	Verwandte Arbeiten	8
2	Grundlagen	11
2.1	Stereobildverarbeitung	11
2.1.1	Stereogeometrie	11
2.1.2	Herstellen von Korrespondenzen	12
2.1.3	Vorverarbeitung und Rektifizierung	13
2.1.4	Nachbearbeitung der Disparitätenbilder	14
2.1.5	Räumliche Kalibrierung mehrerer Kameras	14
2.2	Kalmanfilterung	16
2.2.1	Allgemeines Rahmenwerk	16
2.2.2	Einfacher Kalmanfilter	19
2.2.3	Erweiterter Kalmanfilter (EKF)	19
2.2.4	Unscented Kalman Filter (UKF)	21
2.3	Iterative Closest Point (ICP)	25
2.4	kd-Bäume	26
2.5	Kinematische Modellierung artikularer Objekte	29
3	Trackingalgorithmus	31
3.1	Segmentierung und weitere Vorverarbeitung	31
3.1.1	Vordergrundsegmentierung auf Disparitätenbildern	31
3.1.2	Filterung der 3D-Daten	34
3.2	Tracking artikularer Objekte in Punktwolken	35
3.2.1	Systemmodell	36
3.2.2	Messmodell und Generierung des Messvektors	37
3.2.3	Ablauf der Filterung	41
3.2.4	Kollisionsvermeidung	45
4	Experimente und Ergebnisse	47
4.1	Datensammlung und Erstellung der Referenztrajektorien	47
4.2	Parametersatz und Tuning	49

4.2.1	Parametersatz	49
4.2.2	Einfluss einzelner Parameter auf die Trackinggenauigkeit .	50
4.3	Analyse kurzer Bewegungsabläufe im Detail	53
5	Zusammenfassung und Ausblick	65
A	Implementierungsdetails	67
A.1	Klassenstruktur	67
A.2	Ablauf bei der Verarbeitung eines Einzelbildsatzes	69
	Literaturverzeichnis	73

Kapitel 1

Einleitung

Im Rahmen des Sonderforschungsbereiches 588 „Humanoide Roboter“ der Deutschen Forschungsgemeinschaft (DFG) und des Projektes „Computers in the Human Interaction Loop“ (CHIL) der Interactive Systems Labs (ISL) der Universität Karlsruhe werden neuartige Methoden der Interaktion zwischen Mensch und Maschine entwickelt und erprobt. Schwerpunkt der vorliegenden Diplomarbeit ist die Entwicklung eines auf Stereobildverarbeitung basierenden Sensorsystems, das die Verfolgung (Tracking) der artikularen Bewegung des Oberkörpers ermöglicht. Unter *artikularer Bewegung* versteht man die Bewegung eines Objektes, das durch Gelenke aus mehreren starren Körpern zusammengesetzt ist (*artikulares Objekt*). Ihre Beschreibung erfordert außer der Bestimmung der Lage des Objektes im Raum auch die Erfassung aller Gelenkwinkel. Im Rahmen dieser Arbeit soll die Lage eines menschlichen Oberkörpers im Raum ermittelt werden sowie die Gelenkwinkel der oberen Extremitäten. Das vorgestellte Verfahren beruht allein auf der Auswertung von Tiefenbildern, Farbinformationen werden nicht mit einbezogen. Es ist dadurch nicht anfällig gegenüber Änderungen in der Erscheinung der aufgenommenen Personen, wie sie beispielsweise durch unterschiedliche Färbung oder Musterung der Bekleidung verursacht wird. Auch sind Tiefenbilder invariant gegenüber variablen Beleuchtungsverhältnissen. Das Verfahren greift Ansätze des bekannten Iterated Closest Point (*ICP*) Algorithmus zur Tiefenbildregistrierung auf. *ICP* ist ausschließlich zur Registrierung starrer Körper vorgesehen, durch Integration eines Unscented Kalmanfilter (*UKF*) wird aber das Verfolgen artikularer Objekte möglich.

Kapitel 2 beschreibt mathematische Grundlagen und bewährte Konzepte, die in das System eingeflossen sind, wie *ICP* und *UKF*. Die Funktionsweise des Algorithmus und die einzelnen Verarbeitungsschritte werden in Kapitel 3 im Detail beschrieben. Kapitel 4 präsentiert die Evaluation anhand von Probeaufnahmen mit vier Testpersonen, die zum Teil schwierig zu verfolgende Bewegungen wie Verschränken der Arme vor dem Körper oder Schuhebinden in der Hocke durchführten. Gemessen wurden die Winkelfehler von Körper und Gliedmaßen gegenüber einer durch Handannotierung erstellten Referenz.

1.1 Motivation

Die Anwendungsmöglichkeiten eines Systems zum artikularen Körpertracking sind vielfältig: Auf der Hand liegt die Anwendung in automatischen Überwachungssystemen zur Sicherung gefährdeter Orte, denkbar wäre aber beispielsweise auch die intuitive Steuerung von Robotern und ähnlichen technischen Geräten durch Kontroll- und Zeigegesten.

Ziel des Sonderforschungsbereiches 588 ist die Entwicklung eines menschenähnlichen Roboters, der seinen Arbeitsbereich mit dem Menschen teilt. Seine Kommunikation mit dem Menschen soll möglichst intuitiv geschehen. Ein intuitiver menschlicher Kommunikationskanal ist Gestik, die vom Roboter erkannt werden und auf die er reagieren soll. Weiterhin ist ein Trackingsystem der hier beschriebenen Art hilfreich beim Einlernen von Bewegungsabläufen, da ein menschenähnlicher Roboter über die zum direkten Nachahmen der menschlichen Bewegung erforderliche Aktorik verfügt.

Im Rahmen des CHIL-Projektes soll das System als Komponente in einem *multimodalen Raum* zum Einsatz kommen, dies ist ein Raum, der „intelligent“ auf das Verhalten seiner Benutzer reagiert. Auf der Basis des Trackingsystems können neuartige Benutzerschnittstellen zur Interaktion mit dem Raum und seinen Komponenten implementiert werden. Weiterhin soll der Raum in der Lage sein, ein vollständiges Protokoll der in ihm stattfindenden Ereignisse zu erstellen, hierzu gehören auch Bewegung, Körperhaltung und Gestikulation der Personen, die sich in ihm befinden.

1.2 Verwandte Arbeiten

Bildbasiertes Körpertracking hat sich innerhalb der letzten zehn Jahre zu einem recht aktivem Feld im Bereich digitaler Bildverarbeitung entwickelt. Dieser Abschnitt führt einige Veröffentlichungen zum Thema auf. Die Ansätze werden dabei zunächst grob in *modell-* und *merkmalsbasierte* Verfahren klassifiziert.

In diesem Abschnitt werden Verfahren als *merkmalsbasiert* bezeichnet, bei denen der Schwerpunkt auf der Extraktion von Merkmalsvektoren niedriger Dimensionalität liegt. Ein Intensitäts- oder Kantenbild ist dieser Definition nach kein Merkmal, wohl aber die Pixelposition von Körperteilen, parametrische Konturen oder ähnliches. Bei derartigen Verfahren wird versucht, die Körperpose analytisch aus solchen Merkmalen abzuleiten. Wegen der hohen Dimensionalität des Zustandsraumes und der daraus resultierenden hohen Ambiguität eines beobachteten Merkmalsatzes kann mit diesen Verfahren in der Regel kein detailliertes artikulares Verfolgen des Körpers realisiert werden. Aus den extrahierten Merkmalen lassen sich aber dennoch Informationen über die Körperpose ableiten.

Das in [Nic03] beschriebene Verfahren beispielsweise extrahiert basierend auf Farb- und Stereomerkmalen die Positionen von Kopf und Händen und kann durch

ein nachgeschaltetes Klassifizierungssystem Zeigegeesten erkennen.

Das System PFINDER (siehe [WADP97]) baut dynamisch ein zweidimensionales Modell einer Person auf. Es verfolgt und adaptiert hierzu die Statistik verschiedenfarbiger Bildregionen (*Blobs*) vor einem statischen Hintergrund und kann Kopf, Hände und Füße identifizieren.

In [RFZ05] wird ein System vorgestellt, das dynamisch Modelle (*appearance models*) für das Aussehen der einzelnen Körperteile erstellt. Die Modelle werden in Einzelbildern der Sequenz erstellt, in denen ein Klassifikator eine Person in einer von wenigen möglichen Initialisierungsposen erkannt hat. Mittels der erstellten Modelle können die Körperteile dann in nachfolgenden Einzelbildern wiedergefunden werden.

Sofern ein detailliertes, artikulares Verfolgen des Körpers angestrebt wird, kommen in der Regel modellbasierte Verfahren zum Einsatz. Das zugrundegelegte Modell kann man sich als Strichmännchen (*stick figure*) vorstellen, das die physikalische Struktur und die kinematischen Beschränkungen (*motion constraints*) des menschlichen Körpers repräsentiert. Das Trackingproblem löst man dann, indem man den Zustandsvektor, der eine Pose des Strichmännchens beschreibt, basierend auf Beobachtungen schätzt. Jede Komponente des Zustandsvektors repräsentiert einen Freiheitsgrad des Körpermodells, beispielsweise einen Gelenkwinkel.

Zahlreiche Veröffentlichungen verwenden einen *Partikelfilter* (Kondensationsfilter) zum modellbasierten Tracking. Dieses probabilistische Rahmenwerk verwendet ein Monte-Carlo-Verfahren, um die Wahrscheinlichkeitsdichtefunktion des zu verfolgenden Zustandes zu approximieren. Es wurde zuerst in [GSS93] beschrieben. Problematisch bei der Anwendung zum artikularen Körpertracking ist hierbei die hohe Dimensionalität des Zustandsraumes, da die benötigte Anzahl von Monte-Carlo-Stichproben (Partikeln) exponentiell mit dieser ansteigt.

Deutscher, Blake und Reid modifizieren den Partikelfilteralgorithmus, indem sie ihn nach Art des Simulated Annealing stufenweise ablaufen lassen. Sie erzielen hiermit eine Reduktion der benötigten Partikelanzahl und wenden diesen sogenannten *Annealed Particle Filter* in [DBR00] auf das Problem des artikularen Körpertrackings an.

Lee, Cohen und Jung reduzieren effektiv die Dimensionalität des Zustandsraumes, indem sie die Ergebnisse von Merkmalerkennern (Kopf, Hände, Körperachse) in die Partikelrepräsentation mit einbringen [LCJ02].

Veröffentlichungen, die, wie die vorliegende Arbeit, 3D-Daten zum vollartikularen Tracking heranziehen, sind rar. [JTH99] befasst sich beispielsweise mit dem Thema. Ähnlich wie im Pfinder-System (siehe oben) wird hier die Statistik von *Blobs* geschätzt, als *Blobs* werden aber dreidimensionale Punktbündel aufgefasst. Die 3D-Daten wurden durch Stereobildverarbeitung gewonnen.

Auch [UF04] beschreibt ein Verfahren zum artikularen Tracking auf mit Stereokameras gewonnenen Daten. Die Einpassung des Modells in die 3D-Daten wird hier als Optimierungsproblem formuliert und mit einem Minimierungsverfahren

durchgeführt. Schwerpunkt bei dieser Arbeit war es, das Minimierungsverfahren durch starke Bewegungsmodelle zu steuern, die durch Hauptkomponentenzerlegung (Principal Component Analysis, PCA) aus Bewegungsdatenbanken gewonnen wurden. Hierdurch wird der Posenraum auf wenige Klassen von Bewegungsabläufen beschränkt. Das System ist in der Lage, gehende oder laufende Personen robust zu tracken.

Auch das in [KBVG05] beschriebene System betrachtet artikulares Tracking als Optimierungsproblem. Der Schwerpunkt dieser kürzlich erschienenen Arbeit liegt wie in [UF04] auf der Wahl des Minimierungsverfahrens, die Autoren entscheiden sich für das Verfahren *Stochastic Meta Descent (SMD)*. Minimiert wird die Summe der Abstände einiger vom Modell abgetasteter Punkte zu korrespondierenden Punkten in der Messung. Die Messung wird nicht durch Stereobildverarbeitung gewonnen, sondern durch volumetrische Rekonstruktion aus Bildfolgen, die aus verschiedenen Blickwinkeln aufgenommen wurden. Diese Arbeit ist der vorliegenden ähnlich, allerdings wird hier nach dem globalen Minimum einer *skalaren* Objektfunktion gesucht. Im Gegensatz dazu nutzt die vorliegende Arbeit die Tatsache aus, dass von Einzelbild zu Einzelbild am Zustand nur kleine Änderungen durchzuführen sind, so dass man ihn durch ein linearisierendes Schätzverfahren (*Unscented Kalman Filter*) einem *violdimensionalen* Messvektor nachführen kann.

In [DD02] wird ein Verfahren vorgeschlagen, das einzelne Körperteile mit einem Registrierungsverfahren in eine 3D-Punktwolke einpasst. Die Körperteile werden losgelöst vom Körper betrachtet, die Beschränkungen, die sich durch die Kinematik des Körpers ergeben, werden ihnen erst nachträglich aufgeprägt. Die Methode ist deshalb eher den merkmalsbasierten Verfahren zuzuordnen.

Große Ähnlichkeit der vorliegenden Arbeit besteht zu [SMC01]. Hier wird die artikulare Bewegung einer Hand verfolgt, wie in der vorliegenden Arbeit durch lineare Schätzung mit einem Kalmanfilter. Die synthetisierte Kontur der Hand wird der aus Grauwertbildern extrahierten Kontur nachgeführt. Die vorliegende Arbeit überträgt dies von 2D-Konturen auf 3D-Punktwolken.

Kapitel 2

Grundlagen

Dieses Kapitel beschreibt bewährte Verfahren und mathematische Konzepte, die in das in dieser Arbeit entwickelte System eingeflossen sind.

2.1 Stereobildverarbeitung

Durch die digitale Verarbeitung von Bildern, die mit zwei leicht gegeneinander versetzten Kameras aufgenommen wurden, können Entfernungsmessungen durchgeführt werden. Hierzu werden Korrespondenzen zwischen Bildmerkmalen auf den verschiedenen Ansichten der Szene hergestellt und dann ihr relativer Versatz (*Disparität*) ermittelt, der in direktem Zusammenhang zur Entfernung steht. Zur Stereobildverarbeitung kam für diese Arbeit die TRICLOPS-Bibliothek der Firma Point Grey Research (siehe [Poi03]) zum Einsatz. In diesem Abschnitt werden die einzelnen Verarbeitungsschritte und die geometrischen Grundlagen kurz erläutert.

2.1.1 Stereogeometrie

Abbildung 2.1 zeigt den Aufbau und die Geometrie einer idealisierten Stereokamera in Draufsicht. Die beiden Objektive bei L und R weisen die Eigenschaften einer Lochkamera auf. Wichtig ist weiterhin die Parallelität der optischen Achsen und die Koplanarität der Bildebenen beider Kameras. Geringfügige Abweichungen von dieser Idealgeometrie können durch *Rektifizierung* ausgeglichen werden (siehe Abschnitt 2.1.3).

Den Abstand b der Objektive nennet man die *Aufnahmebasis*. Die Bildpositionen eines beobachteten Punktes P unterscheiden sich unter den beschriebenen Voraussetzungen nur in der x -Koordinate, für die linke Kamera sei diese x_L , für die rechte x_R . Ihre Differenz $d = x_L - x_R$ nennt man (*horizontale*) *Disparität* (siehe [Jä97]). Mithilfe des Strahlensatzes ergibt sich für die Entfernung z die

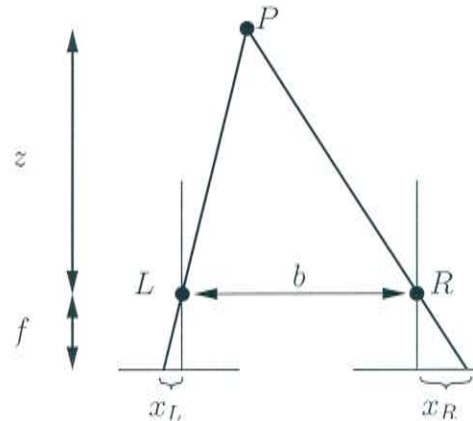


Abbildung 2.1: Stereogeometrie. Zwei Kameras mit den Brennpunkten L und R und gleicher Brennweite f beobachten Punkt P . Aus der Differenz der Bildkoordinaten x_L und x_R (Disparität) lässt sich die Entfernung z berechnen (entnommen aus [Nic03]).

Beziehung

$$z = \frac{f \cdot b}{d} \quad (2.1)$$

Legt man ein rechtshändiges Koordinatensystem so in den Brennpunkt der rechten Kamera, das die z -Achse senkrecht auf der Bildebene steht und die x -Achse nach rechts zeigt, erhält man als Koordinaten von P

$$x = \frac{x_R \cdot z}{f} \quad (2.2)$$

$$y = \frac{y_R \cdot z}{f} \quad (2.3)$$

Die Koordinaten (relativ zur Kamera) von P lassen sich also aus der Disparität d und den Koordinaten (x_R, y_R) seines Bildpunktes mittels (2.1), (2.2) und (2.3) vollständig rekonstruieren.

Für diese Arbeit kamen Stereokameras vom Typ BUMBLEBEE der Firma POINT GREY RESEARCH zum Einsatz. Sie haben eine Brennweite von zwei Millimetern und eine Aufnahmebasis von 120 Millimetern. Es wird eine Auflösung von 640 mal 480 Pixeln erreicht.

2.1.2 Herstellen von Korrespondenzen

Zur Ermittlung der Korrespondenzen kommt hier nach [Poi03] die sogenannte *sum-of-absolute-differences-correlation*-Methode zum Einsatz. Dieser Ansatz funktioniert folgendermaßen:

1. Für jedes Pixel p im Referenzbild (hier: rechte Kamera)

- (a) Wähle um p eine quadratische Region R_{ref} mit gegebener Seitenlänge m (*Korrelationsmaske*)
- (b) Vergleiche R_{ref} mit einer Anzahl Regionen R_k im anderen Bild. R_k werden entlang der Zeile von p gewählt, mit einem Versatz d_k , den man von d_{min} bis d_{max} laufen lässt.
- (c) Wähle das R_k mit der besten Passung. Der zugehörige Versatz d_k ist die ermittelte Disparität.

Die Güte der Passung zweier Regionen wird mit folgender Formel ermittelt:

$$\sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} |R_{ref}[x+i][y+j] - R_k[x+i+d][y+j]|$$

Die Wahl der minimalen Disparität d_{min} begrenzt den nutzbaren Entfernungsbereich nach hinten (von der Kamera weg), die maximale Disparität d_{max} schränkt ihn nach vorne ein. Während Schritt 1 (b) lässt man d_k im einfachsten Fall in Pixelschritten von d_{min} bis d_{max} laufen, man kann aber auch Intensitätswerte im Subpixelbereich interpolieren und so die Genauigkeit des Verfahrens erhöhen. Die mit diesem Verfahren erstellten Bilder werden *Disparitätenbilder* genannt (siehe Abbildung 3.2 auf Seite 33 für ein Beispiel) .

2.1.3 Vorverarbeitung und Rektifizierung

Die TRICLOPS-Bibliothek führt als Vorverarbeitung eine Glättung der Rohbilder mit einem Tiefpassfilter durch. Hierdurch wird niedrigfrequentes Rauschen des Bildsensors entfernt. Außerdem wird vermieden, dass die nachgeschaltete Rektifizierung (siehe unten) Aliasing-Artefakte hervorbringt. Optional kann der Stereoverarbeitung eine Kantendetektion vorgeschaltet werden. Dies ist aus zwei Gründen vorteilhaft: Erstens wird die Verstärkung der Bildsensoren beider Kameras in einem TRICLOPS-System *getrennt* geregelt und so der Gesamthelligkeit der Bilder angepasst. Hierdurch kann es Helligkeitsunterschiede zwischen den Kameras geben, was bei einer rein intensitätsbasierten Ermittlung der Korrespondenzen zu Fehlern führen kann. Die Kantenbilder sind gegenüber Helligkeitsschwankungen invariant. Zweitens kann mithilfe der Kantenbilder eine Validierung der ermittelten Disparitätenbilder durchgeführt werden, dies wird in Abschnitt 2.1.4 beschrieben.

Echte Stereokameras weichen durch Ungenauigkeiten in der Optik und der Ausrichtung der Kameras von der in 2.1.1 beschriebenen Idealgeometrie ab. So führen Linsenobjektive gegenüber einer Lochkamera im Allgemeinen zu einer leicht verzerrten und dezentrierten Abbildung. *Rektifizierung* nennt man den Prozess, der diese störenden Effekte ausgleicht. Hierzu ist eine genaue *Kalibrierung* der Kameras nötig. Die für diese Arbeit verwendeten Stereokameras sind vom

Hersteller vorkalibriert. Zu den durch die Kalibrierung ermittelten Rektifizierungsparametern gehören Brennweite, Linsendezentrierung, radialsymmetrische Verzerrung sowie das Aspektverhältnis der Bildelemente des Sensorchips, weiterhin die Aufnahmebasis und die Abweichung der optischen Achsen von der Parallelen. Der Hersteller garantiert eine Genauigkeit der Kalibrierung von 0.05 Pixel.

2.1.4 Nachbearbeitung der Disparitätenbilder

Die mit dem in Abschnitt 2.1.2 beschriebenen Verfahren erstellten Disparitätenbilder sind üblicherweise an einigen Stellen fehlerhaft in dem Sinne, dass die dort ermittelten Korrespondenzen nicht aus der tatsächlichen Parallaxe der betrachteten Punkte resultieren. Beispielsweise können nahe beieinander liegende Bildregionen mit ähnlichem Erscheinungsbild irrtümlich einander zugeordnet werden. Dies wurde häufig bei regelmäßig gemusterten Textilien beobachtet. Fehler treten auch in Bildregionen auf, die wenig Textur aufweisen, also bei einfarbigen oder sehr dunklen Flächen. Die Korrespondenzbildung liefert hier meist falsche Werte, die nicht auf Textur sondern auf dem Rauschen des Bildsensors beruhen. Eine weitere Ursache für fehlerhafte Korrespondenzen ist die Tatsache, dass, wie in Abbildung 2.2, einige Stellen der betrachteten Szene nur in einem Bild des Stereobildpaares sichtbar sind. Sinnvolle Korrespondenzen lassen sich hier offensichtlich nicht erstellen.

Die TRICLOPS-Bibliothek bietet verschiedene Verfahren an, um ungültige Disparitäten zu identifizieren. Ein *Texturfilter* identifiziert mit Hilfe des in der Vorverarbeitung (Abschnitt 2.1.3) erzeugten Kantenbildes Bildregionen mit geringer Textur und markiert diese im Disparitätenbild als ungültig. *Uniqueness Validation* überprüft, ob die ermittelte Passung für einzelne Pixel signifikant besser ist als für den Rest der Pixel innerhalb der Korrelationsmaske und invalidiert so Regionen, die in nur einer Kamera sichtbar sind. Abbildung 2.2 illustriert das Verfahren. Texturfilter und Uniqueness Validation lassen sich einzeln zuschalten und über jeweils einen Schwellwert regeln.

2.1.5 Räumliche Kalibrierung mehrerer Kameras

Die zum Tracking verwendeten Bildfolgen stammen von mehreren Stereokameras. Experimentell wurden eine bis vier Kameras erprobt (siehe Kapitel 4). Die Kameras werden zu einem gemeinsamen Bezugssystem kalibriert, wodurch ermöglicht wird, die von den einzelnen Kameras erhaltenen 3D-Daten zu einer einzigen Punktwolke zu fusionieren (siehe Abbildung 2.3). Dadurch ist implizit eine Verfolgung des Oberkörpers über mehrere Kameras realisiert.

Zur Kalibrierung der Kameras wurde auf Routinen der Softwarebibliothek OPENCV ([INT01]) zurückgegriffen. Der Ablauf des Verfahrens ist folgendermaßen: Ein Schachbrettmuster mit bekannter Dimensionierung wird so positioniert,

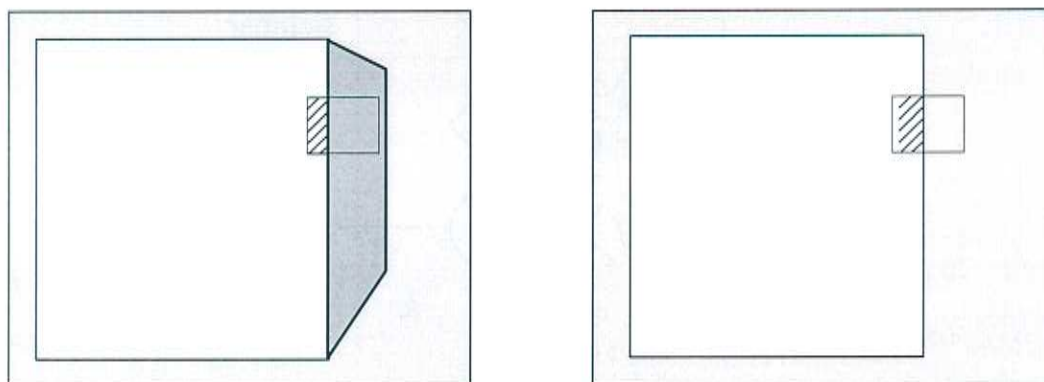


Abbildung 2.2: Uniqueness Validation. Die rechte Seite des Quaders ist im Bild der linken Kamera (rechts) nicht sichtbar. Der schraffierte Teil der Korrelationsmaske weist eine signifikant bessere Passung auf als ihr restlicher Teil. Dies ist ein Anzeichen dafür, dass ein grosser Teil der betrachteten Region in einem der Bilder verdeckt ist.

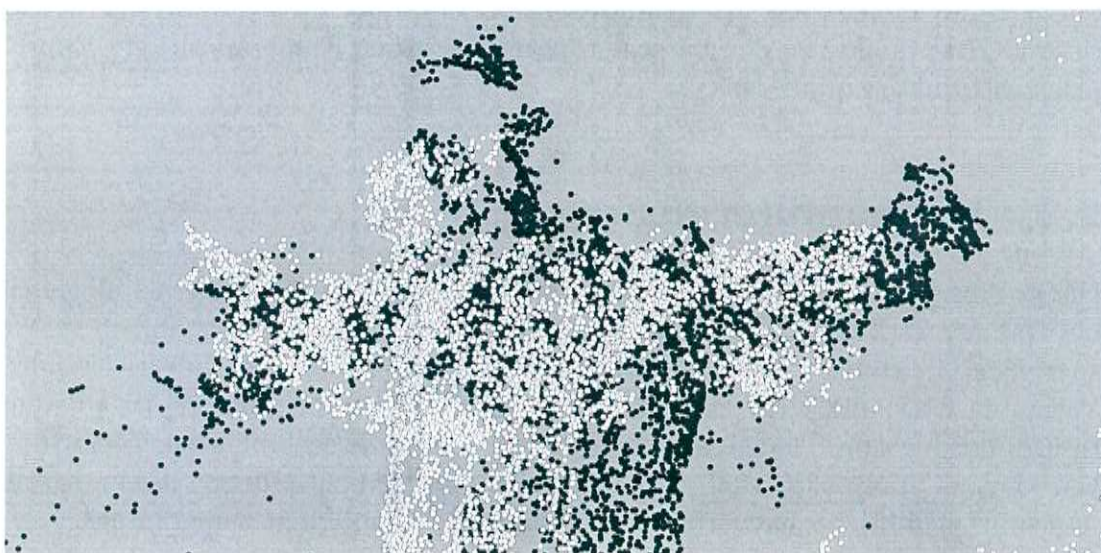


Abbildung 2.3: Fusion der 3D-Daten. Die weißen und schwarzen Punkte stammen von zwei Stereokameras, die sich ungefähr gegenüberstehen und eine Person mit ausgestreckten Armen beobachten. Durch die räumliche Kalibrierung konnten die Punkte aus den lokalen Kamerakoordinatensystemen in ein gemeinsames Weltkoordinatensystem transformiert werden.

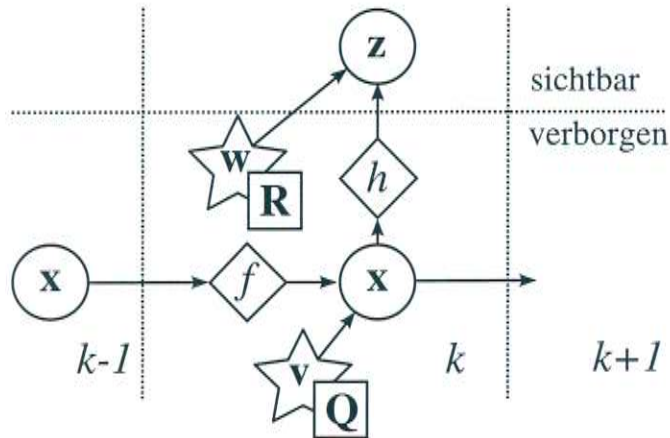


Abbildung 2.4: Das Modell, das der Kalmanfilterung zugrunde liegt. Bei Kreisen handelt es sich um Vektoren, Sterne sind Rauschprozesse (mit Kovarianzmatrix im Kasten), Rauten sind Funktionen.

dass es sich im Sichtbereich aller Kameras befindet. Die Gitterpunkte des Schachbrettes werden nun detektiert. Durch ein einfaches Optimierungsverfahren wird dann die relative Position aller Kameras zum Schachbrett bestimmt. Hierzu wird durch Gradientenabstieg der quadratische Abstand der in der Bildebene detektierten Gitterpunkte zu den unter der hypothetisierten Kameraposition rückprojizierten Punkten minimiert.

2.2 Kalmanfilterung

Dieser Abschnitt stellt zunächst das Prinzip der Kalmanfilterung als allgemeines Rahmenwerk vor. Die Unterabschnitte füllen dieses jeweils aus, indem sie die speziellen Gleichungen für den linearen Kalmanfilter und seine linearisierenden Varianten EKF und UKF einführen. Das erste Kapitel aus [May79] ist eine gute Einführung zum Thema Kalmanfilterung. Die Softwarebibliothek BAYES++ von Michael Stevens ([Ste03]) enthält effiziente Implementierungen der in diesem Abschnitt beschriebenen Verfahren und kam für diese Arbeit zum Einsatz.

2.2.1 Allgemeines Rahmenwerk

Der Kalmanfilter ist ein Algorithmus zur Zustandsschätzung eines Systems, das sich nur durch indirekte und verrauschte Messungen beobachten lässt. Er bedient sich hierzu eingebrachten Wissens um die Dynamik des Systems und Annahmen über die Größe von Messfehlern. Man modelliert das System als zeitdiskretes, nichtlineares System mit stochastischen Störungen (siehe auch Abbildung 2.4):

$$\mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{v}[k] \quad (2.4)$$

k ist der diskrete Zeitindex. Die *Systemgleichung* 2.4 beschreibt das Verhalten des Systems, indem sie den neuen Zustand $\mathbf{x}[k+1]$ funktional mit dem vorigen Zustand $\mathbf{x}[k]$ in Beziehung setzt, unter Berücksichtigung der gegenwärtigen Ansteuerung des Systems, $\mathbf{u}[k]$. Die Ansteuerung $\mathbf{u}[k]$ ist bei vielen Systemen unbekannt oder nicht vorhanden, so auch bei dem von uns betrachteten System „menschlicher Körper“. Sie wird uns im Rahmen dieser Arbeit nicht mehr begegnen. Die Funktion $f(\cdot, \cdot)$ nennt man *Systemmodell*. Die Vorhersage des Systemmodells ist mit einem additiven Fehler, dem *Systemrauschen* $\mathbf{v}[k]$, belegt. Der Zustandsvektor $\mathbf{x}[k]$ selbst ist unbekannt, man erhält aber zu diskreten Zeitpunkten Messungen \mathbf{z}_k , die mit dem Zustand über folgende Gleichung in Zusammenhang stehen:

$$\mathbf{z}[k] = h(\mathbf{x}[k]) + \mathbf{w}[k] \quad (2.5)$$

Das *Messmodell* $h(\cdot)$ beschreibt den funktionalen Zusammenhang zwischen dem Zustand $\mathbf{x}[k]$ und dem zugehörigen Messwert $\mathbf{z}[k]$. Die Messung ist mit dem Fehler $\mathbf{w}[k]$, *Messrauschen* genannt, behaftet.

Der Kalmanfilter schätzt, wie noch gezeigt wird, die Statistik zweiter Ordnung eines Systems, also Mittelwert und Standardabweichung des Zustandes.

In realen Systemen muss man von Ungenauigkeiten bei der Beschreibung bzw. Modellierung ausgehen, beispielsweise, wenn eines der Modelle vereinfacht wurde, um es mathematisch fassbar zu machen. Die Rauschprozesse $\mathbf{v}[k]$ und $\mathbf{w}[k]$ fassen alle Abweichungen vom Modellverhalten zu additivem Rauschen zusammen. Sie werden beim klassischen Kalmanfilter, wie auch bei der hier verwendeten Variante UKF als weißes, unkorreliertes Rauschen, dessen Erwartungswert verschwindet, modelliert. Sei also

$$\mathbf{Q}[k] = \text{var}(\mathbf{v}[k]) \quad (2.6)$$

die *Kovarianzmatrix des Systemrauschens*, und analog

$$\mathbf{R}[k] = \text{var}(\mathbf{w}[k]) \quad (2.7)$$

die *Kovarianzmatrix des Messrauschens*.

Im Folgenden wird mit $\hat{\mathbf{x}}[i|j]$ der Schätzwert zur Zeit i , der durch alle Messungen bis einschließlich Zeitpunkt j bedingt ist, bezeichnet:

$$\hat{\mathbf{x}}[i|j] = E \left[\mathbf{x}[j] | Z^j \right]$$

mit $Z^j = \{\mathbf{z}[1], \mathbf{z}[2], \dots, \mathbf{z}[j]\}$. Die bedingte Varianz dieser Schätzung bezeichnet man analog mit

$$\mathbf{P}[i|j] = E \left[\{\mathbf{x}[i] - \hat{\mathbf{x}}[i|j]\} \{\mathbf{x}[i] - \hat{\mathbf{x}}[i|j]\}^T | Z^j \right] \quad (2.8)$$

Der Ablauf des Algorithmus ist wie folgt:

Wiederhole:

1. **Prädiktion:** ausgehend von dem optimalen Schätzwert $\hat{\mathbf{x}}[k|k]$ und seiner Fehlerkovarianz $\mathbf{P}[k|k]$ werden der beste (*a priori*-) Schätzwert $\hat{\mathbf{x}}[k+1|k]$ und seine Fehlerkovarianzmatrix $\mathbf{P}[k+1|k]$ mit Hilfe des Systemmodells extrapoliert. $\mathbf{P}[k+1|k]$ enthält die Systemkovarianz \mathbf{Q} als additiven Anteil. Man transformiert $\mathbf{P}[k+1|k]$ und $\hat{\mathbf{x}}[k+1|k]$ durch das Messmodell und erhält als Vorhersage für den Messwert $\hat{\mathbf{z}}[k+1|k]$, mit der Varianz $\mathbf{P}_{zz}[k+1|k]$. Die Differenz zwischen vorhergesagtem und tatsächlichem Messwert nennt man Residuum (oder Innovation): $\nu[k+1] = \mathbf{z}[k+1] - \hat{\mathbf{z}}[k+1|k]$. Die Varianz dieser Größe ist $\mathbf{P}_{\nu\nu}[k+1|k] = \mathbf{P}_{zz}[k+1|k] + \mathbf{R}[k+1]$.
2. **Korrektur:** mit Hilfe der Messung $\mathbf{z}[k+1]$ wird der extrapolierte Wert $\hat{\mathbf{x}}[k+1|k]$ optimal zu $\hat{\mathbf{x}}[k+1|k+1]$ verbessert. Die Fehlerkovarianz $\mathbf{P}[k+1|k+1]$ des korrigierten Schätzwertes wird dabei minimiert. Die Korrekturgleichungen lauten:

$$\begin{aligned}\hat{\mathbf{x}}[k+1|k+1] &= \hat{\mathbf{x}}[k+1|k] + \mathbf{K}[k+1]\nu[k+1] \\ \mathbf{P}[k+1|k+1] &= \mathbf{P}[k+1|k] - \mathbf{K}[k+1]\mathbf{P}_{\nu\nu}[k+1|k]\mathbf{K}[k+1]^T\end{aligned}$$

Hierbei ist $\mathbf{K}[k+1]$ das *Kalman-Gain*. Es errechnet sich zu

$$\mathbf{K}[k+1] = \mathbf{P}_{xz}[k+1|k]\mathbf{P}_{\nu\nu}^{-1}[k+1|k]$$

$\mathbf{P}_{xz}[k+1|k]$ ist die Kreuzkovarianzmatrix

$$E\left[\{\mathbf{x}[k+1] - \hat{\mathbf{x}}[k+1|k]\}\{\mathbf{z}[k+1] - \hat{\mathbf{z}}[k+1|k]\}^T | Z^k\right] \quad (2.9)$$

von $\mathbf{x}[k+1]$ und $\mathbf{z}[k+1]$ (Vergleiche 2.8).

Der beschriebene Algorithmus beschreibt das Kalmanfilter-Rahmenwerk in sehr allgemeiner Form. Zu klären bleibt, wie $\hat{\mathbf{x}}$ und seine Kovarianz \mathbf{P} durch die funktionalen Modelle propagiert und wie die Kreuzkovarianz \mathbf{P}_{xz} ermittelt wird. Zu ermitteln sind also noch die Größen $\hat{\mathbf{x}}[k+1|k]$, $\mathbf{P}[k+1|k]$, $\hat{\mathbf{z}}[k+1|k]$, $\mathbf{P}_{zz}[k+1|k]$ sowie $\mathbf{P}_{xz}[k+1|k]$. Der Kalmanfilter und seine linearisierenden Varianten verfolgen hierzu jeweils unterschiedliche Ansätze, die in den folgenden Abschnitten erläutert werden.

Die asymptotische Komplexität eines Vorhersage-Korrekturzyklus des Kalmanfilters ist im allgemeinen Fall $\mathcal{O}(2nm^2) + \mathcal{O}(m^3) + \mathcal{O}(n^3)$ (nach [GGM97]). Hierbei ist m die Dimensionalität des Zustandsvektors, n die des Messvektors. Dies gilt auch für die im folgenden beschriebenen Varianten.

2.2.2 Einfacher Kalmanfilter

Das Kalmanfilter in seiner ursprünglichen Form war vorgesehen für Systeme mit linearer Dynamik und linearer Messgleichung. In dieser Form wurde der Kalmanfilter von seinem Erfinder Rudolf E. Kalman erdacht und in seinem berühmten Artikel [Kal60] beschrieben.

Die funktionalen Modelle aus (2.4) und (2.5) werden zu linearen Funktionen und man erhält als System- und Messgleichung

$$\mathbf{x}[k+1] = \mathbf{F}\mathbf{x}[k] + \mathbf{v}[k]$$

$$\mathbf{z}[k] = \mathbf{H}\mathbf{x}[k] + \mathbf{w}[k]$$

Die Ansteuerung $u[k]$ wurde hier wie im folgenden aus oben genanntem Grund weggelassen. \mathbf{F} und \mathbf{H} nennt man *Zustandsübergangsmatrix* und *Messmatrix*. Prädiktion und Korrektur kann man in diesem linearen Fall durch einfache Vektorgleichungen ausdrücken. Der prädizierte Zustand ergibt sich zu

$$\hat{\mathbf{x}}[k+1|k] = \mathbf{F}\hat{\mathbf{x}}[k|k]$$

Der für die Prädiktion erwartete Fehler beträgt

$$\mathbf{P}[k+1|k] = \mathbf{F}\mathbf{P}[k|k]\mathbf{F}^T + \mathbf{Q}[k]$$

Analog wird $\hat{x}[k+1|k]$ durch das Messmodell propagiert:

$$\begin{aligned}\hat{\mathbf{z}}[k+1|k] &= \mathbf{H}\hat{\mathbf{x}}[k+1|k] \\ \mathbf{P}_{zz}[k+1|k] &= \mathbf{H}\mathbf{P}[k+1|k]\mathbf{H}^T\end{aligned}$$

Die Kreuzkovarianz ergibt sich zu

$$\mathbf{P}_{xz}[k+1|k] = \mathbf{P}[k+1|k]\mathbf{H}^T \quad (2.10)$$

Der Kalmanfilter ist im beschriebenen linearen Fall ein optimaler Schätzer in dem Sinne, dass er die Wahrscheinlichkeit der Schätzung maximiert.

2.2.3 Erweiterter Kalmanfilter (EKF)

Bei *nichtlinearen* Modellen wird im einfachsten Fall durch Ableitung am aktuellen Systemzustandsvektor linearisiert (siehe auch Abbildungen 2.5 - 2.6). Für die Prädiktion des Zustandes und seines Fehlers ergeben sich dann

$$\begin{aligned}\hat{\mathbf{x}}[k+1|k] &= f(\hat{\mathbf{x}}[k|k]) \\ \mathbf{P}[k+1|k] &= \mathcal{J}_f\mathbf{P}[k|k]\mathcal{J}_f^T + \mathbf{Q}[k]\end{aligned}$$

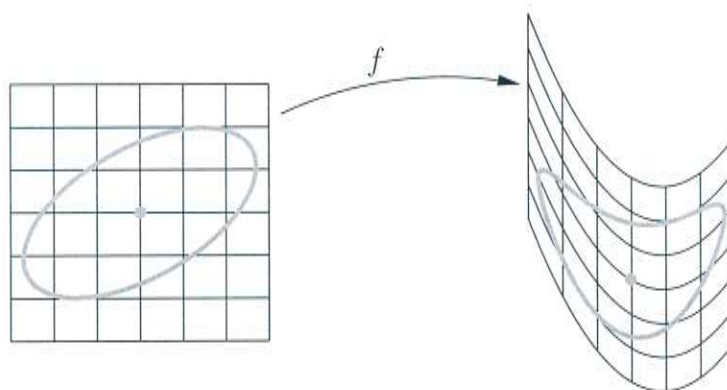


Abbildung 2.5: Unterwirft man eine normalverteilte Zufallsvariable einer nichtlinearen Transformation f , so ist sie im Allgemeinen nicht mehr normalverteilt. Linearisierende Kalmanfilter wie EKF und UKF approximieren die ersten beiden Momente der Verteilung der transformierten Variablen, also Mittelwert und Varianz.

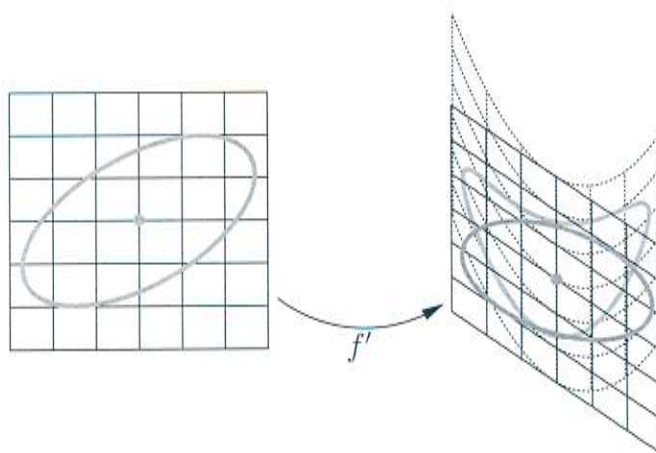


Abbildung 2.6: Beim EKF wird die nichtlineare Transformation f durch Ableitung am Mittelwert approximiert.

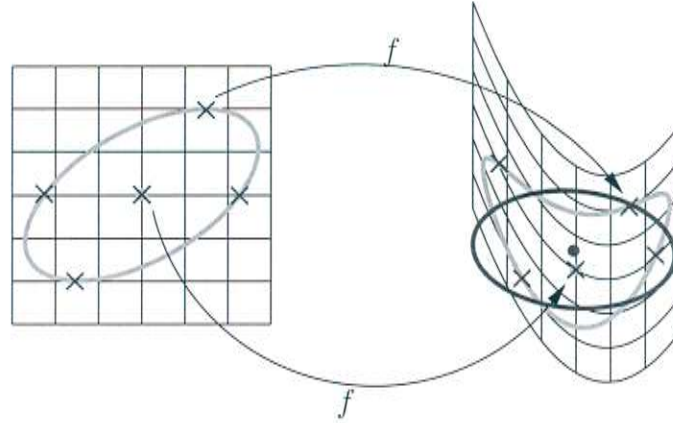


Abbildung 2.7: Beim UKF unterwirft man eine wohlgewählte Punktmenge (\times) der nichtlinearen Transformation. Anschließend bestimmt man durch gewichtete Mittelung der transformierten Punkte Mittelwert und Kovarianz.

\mathcal{J}_f ist die Jacobi-Matrix von f an der Stelle $\hat{\mathbf{x}}[k|k]$. Ganz analog wird durch das Messmodell propagiert:

$$\begin{aligned}\hat{\mathbf{z}}[k+1|k] &= h(\hat{\mathbf{x}}[k+1|k]) \\ \mathbf{P}_{zz}[k+1|k] &= \mathcal{J}_h \mathbf{P}[k+1|k] \mathcal{J}_h^T\end{aligned}$$

\mathcal{J}_h ist diesmal die Jacobi-Matrix von h , ausgewertet an der Stelle $\hat{\mathbf{x}}[k+1|k]$. Die Kreuzkovarianz ist

$$\mathbf{P}_{xz}[k+1|k] = \mathbf{P}[k+1|k] \mathcal{J}_h^T \quad (2.11)$$

Diese Form des Kalmanfilters heist *Erweiterter Kalmanfilter (EKF)*. Er wurde erstmals in [Sch70] von Stanley Schmidt vorgeschlagen und heist nach ihm auch *Kalman-Schmidt-Filter*. Die Nichtlinearitäten der Modelle werden hier bei der Propagierung der Varianzen nur bis zum Grade einer Taylorentwicklung erster Ordnung angenähert, bei großen Abtastzeiten oder stark nichtlinearen Modellen ist der EKF deshalb kein sehr verlässlicher Schätzer. Im nächsten Abschnitt wird eine Variante des Kalmanfilters vorgestellt, die Nichtlinearitäten besser approximiert als der EKF und bei der außerdem die mitunter komplizierte analytische Bildung der Jacobimatrizen nicht erforderlich ist.

2.2.4 Unscented Kalman Filter (UKF)

Der von Julier und Uhlmann in [JU97] beschriebene *Unscented Kalmanfilter (UKF)* ist eine nichtlineare Variante des Kalmanfilters, die sich bei der Propagierung der stochastischen Zustandsvariablen durch die Mess- und Systemmodelle vom EKF wesentlich unterscheidet (Abbildung 2.7). Zum Einsatz kommt hierbei die sogenannte *Unscented Transformation* (siehe [JU96]), mit der Varianz und

Mittelwert einer stochastischen Variable nach einer nichtlinearen Transformation approximiert werden können. Hierzu wird eine wohldefinierte Stichprobe des Zustandsraums der nichtlinearen Transformation unterworfen, um anschließend Mittelwert und Kovarianz durch eine gewichtete Mittelung der transformierten Punkte zu schätzen.

Sei \mathcal{N} eine n -dimensionale Normalverteilung mit Kovarianzmatrix \mathbf{P}_{xx} und Mittelwert μ_x . Man kann eine Stichprobenmenge aus $2n + 1$ Punkten generieren, die die gleiche Kovarianz aufweist. Hierzu wählt man die $2n$ Spalten der positiven und negativen Quadratwurzel der Matrix $(n + \kappa)\mathbf{P}_{xx}$. κ ist ein Skalierungsfaktor, seine Bedeutung wird weiter unten erläutert. Diese Punkte nennt man σ -Punkte $\{\sigma_1, \dots, \sigma_{2n}\}$. War \mathcal{N} nicht mittelwertfrei, so erhält man durch Addition des Mittelwertes μ zu jedem σ_i eine Stichprobenmenge, die in Kovarianz und Mittelwert mit der ursprünglichen Verteilung übereinstimmt. Man nimmt zu dieser Menge noch den Mittelwert μ_x hinzu, dies ändert Mittelwert und Kovarianz nicht. Zusammenfassend generiert man die verschobenen σ -Punkte \mathcal{X}_i folgendermaßen:

$$\sigma_i \leftarrow 2n \text{ Spalten oder Zeilen aus } \pm \sqrt{(n + \kappa)\mathbf{P}_{xx}} \quad (2.12)$$

$$\mathcal{X}_0 = \mu \quad (2.13)$$

$$\mathcal{X}_i = \sigma_i + \mu \quad (2.14)$$

Um nun \mathcal{N} einer nichtlinearen Funktion g zu unterwerfen, transformieren zunächst die Punkte \mathcal{X}_i einzeln:

$$\mathcal{Y}_i = g(\mathcal{X}_i). \quad (2.15)$$

Kovarianz \mathbf{P}_{yy} und Mittelwert μ_y der transformierten Verteilung \mathcal{N} sowie die für die Kalmanfilterung erforderliche Kreuzkorrelation \mathbf{P}_{xy} erhält man nun durch Mittelung, wobei die \mathcal{Y}_i noch unterschiedlich gewichtet werden:

$$\mu_y = \frac{1}{n + \kappa} \left\{ \kappa \mathcal{Y}_0 + \frac{1}{2} \sum_{i=1}^{2n} \mathcal{Y}_i \right\} \quad (2.16)$$

$$\mathbf{P}_{yy} = \frac{1}{n + \kappa} \left\{ \kappa [\mathcal{Y}_0 - \mu_y][\mathcal{Y}_0 - \mu_y]^\top + \frac{1}{2} \sum_{i=1}^{2n} [\mathcal{Y}_i - \mu_y][\mathcal{Y}_i - \mu_y]^\top \right\} \quad (2.17)$$

$$\mathbf{P}_{xy} = \frac{1}{n + \kappa} \left\{ \kappa [\mathcal{X}_0 - \mu_x][\mathcal{Y}_0 - \mu_y]^\top + \frac{1}{2} \sum_{i=1}^{2n} [\mathcal{X}_i - \mu_x][\mathcal{Y}_i - \mu_y]^\top \right\} \quad (2.18)$$

(2.17) und (2.18) kann man sich durch (2.8) beziehungsweise (2.9) plausibel machen.

Die von (2.12)-(2.17) durchgeführte Transformation heißt *Unscented Transformation*. Sie wird im folgenden mit dem Operator \mathcal{U} bezeichnet:

$$\begin{pmatrix} \mu_y & \mathbf{P}_{yy} \end{pmatrix} = \mathcal{U}(\mu_x, \mathbf{P}_{xx}, g).$$

Wenn die Berechnung der Kreuzkorrelation mittels (2.18) erforderlich ist, schreiben wir \mathcal{U}_{KK} :

$$\begin{pmatrix} \mu_y & \mathbf{P}_{yy} & \mathbf{P}_{xy} \end{pmatrix} = \mathcal{U}_{KK}(\mu_x, \mathbf{P}_{xx}, g).$$

Mit der Unscented Transformation kann das Kalmanfilter-Rahmenwerk aus Abschnitt 2.2.1 gefüllt werden. Die Propagierung durch das Systemmodell im Prädiktionsschritt geschieht mit

$$\begin{pmatrix} \hat{\mathbf{x}}[k+1|k] & \bar{\mathbf{P}}[k+1|k] \end{pmatrix} = \mathcal{U}(\hat{\mathbf{x}}[k|k], \mathbf{P}[k|k], f)$$

und anschließender Addition der Kovarianzmatrix $\mathbf{Q}[k]$ des Systemrauschens:

$$\mathbf{P}[k+1|k] = \bar{\mathbf{P}}[k+1|k] + \mathbf{Q}[k]$$

Es wird weiter durch das Messmodell propagiert, wobei auch die zur Berechnung des Kalman gains nötige Kreuzkorrelation \mathbf{P}_{xz} ermittelt wird:

$$\begin{pmatrix} \hat{\mathbf{z}}[k+1|k] & \bar{\mathbf{P}}_{zz}[k+1|k] & \mathbf{P}_{xz}[k+1|k] \end{pmatrix} = \mathcal{U}_{KK}(\hat{\mathbf{x}}[k+1|k], \mathbf{P}[k+1|k], h).$$

Die Kovarianzmatrix $\mathbf{R}[k]$ des additiven Messrauschens muss noch addiert werden:

$$\mathbf{P}_{zz}[k+1|k] = \bar{\mathbf{P}}_{zz}[k+1|k] + \mathbf{R}[k].$$

Hiermit stehen alle für den Korrekturschritt benötigten Größen zur Verfügung.

Der Skalierungsfaktor κ beeinflusst die räumliche Verteilung der σ -Punkte und ihre Gewichtung. Durch geeignete Wahl von κ ist es prinzipiell möglich, die σ -Punkte beliebig nah an den Mittelwert μ heranzubringen. Dies kann für bestimmte Funktionen g erstrebenswert sein, da durch eine zu weiträumige Verteilung der σ -Punkte nicht-lokale Momente der Funktion übergewichtet werden, wie Abbildung 2.8 zeigt. Möchte man die σ -Punkte sehr nah um den Mittelwert verteilen (dies war für die in Abschnitt 3.2 beschriebenen Modelle erforderlich), muss κ negativ gewählt werden. Leider kann dann nicht mehr garantiert werden, dass die transformierte Kovarianz positiv semidefinit ist, was für die Kalmanfilterung aber erforderlich ist. Julier und Uhlmann schlagen in [JU02] eine Modifikation der Unscented Transformation vor, bei der sich die Skalierung der σ -Punkte besser kontrollieren lässt. Bei dieser sogenannten *Scaled Unscented Transformation* wird der Grad der Skalierung durch einen zusätzlichen Parameter α bestimmt, der beliebig gewählt werden kann, bei garantierter positiver Semidefinitheit der transformierten Kovarianz.

Der UKF ist in den meisten praktischen Situationen ein besserer Schätzer als der EKF, da die Unscented-Transformation Nichtlinearitäten besser einfängt als die Ableitung am Systemzustand beim EKF. Der entscheidende Vorteil, der den Ausschlag für den Einsatz des UKF in dieser Arbeit gab, ist die Tatsache, dass sein Entwurf nicht die Ableitung der Modellfunktionen erfordert. Dies vereinfacht den Filterentwurf, insbesondere bei komplizierten Mess- oder Systemmodellen, enorm.

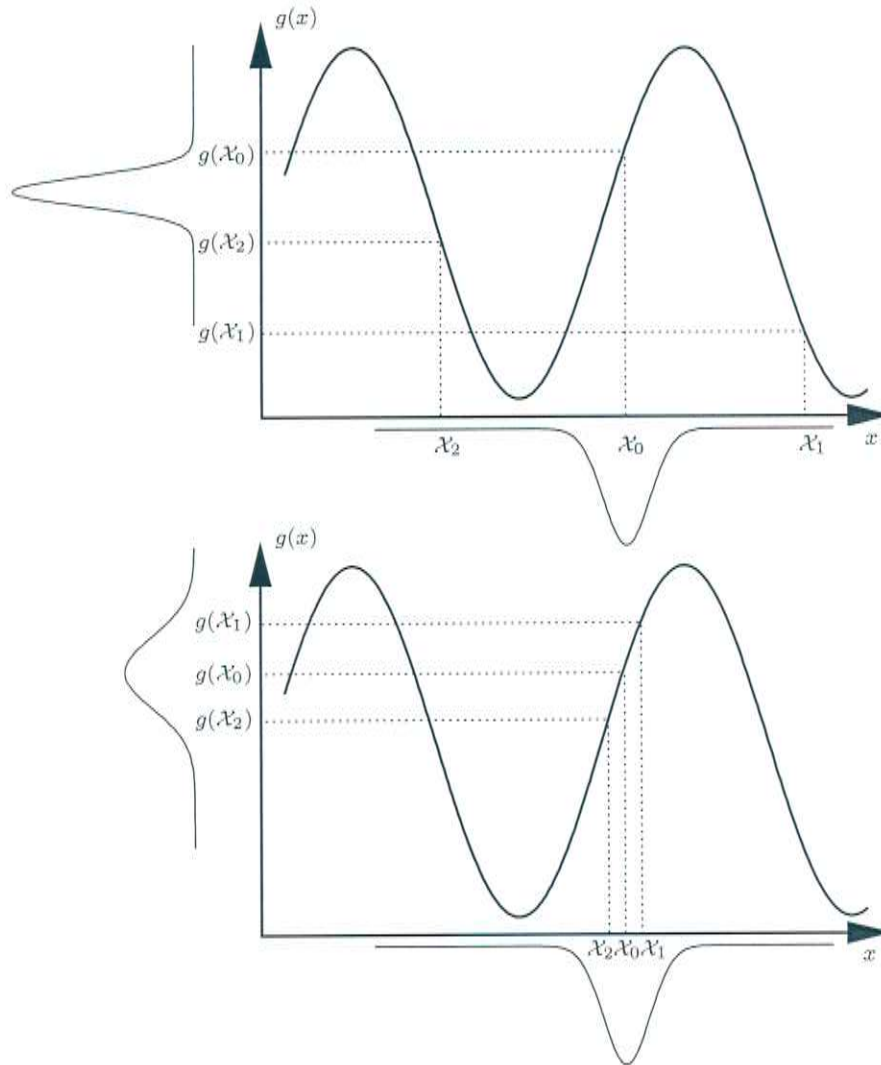


Abbildung 2.8: Auswirkung der Skalierung bei der Unscented Transformation. Eine Gaussverteilung wird durch eine Sinusfunktion transformiert. Im oberen Bild sind die σ -Punkte \mathcal{X}_i weit auseinander skaliert, wodurch nicht-lokale Momente der Funktion übergewichtet werden. Die Skalierung im unteren Bild führt zu einer besseren Approximation der transformierten Verteilung. Zu beachten ist, dass bei Skalierung auch die Gewichtung der σ -Punkte verändert werden muss. Im oberen Bild wird \mathcal{X}_0 stärker gewichtet als im unteren.

2.3 Iterative Closest Point (ICP)

Der *Iterative Closest* (treffender auch: *Corresponding*) *Point (ICP)* Algorithmus ist ein bekanntes Verfahren zur Registrierung von Datensätzen. Die Art der Daten, die der Algorithmus verwenden kann, ist prinzipiell beliebig, solange ein Abstandsmaß auf ihnen definiert werden kann (Punkte, Kurven etc.). Ein typischer Anwendungsfall ist die Einpassung der 3D-Daten (Punktwolke), die ein Sensor von einem Objekt aufgenommen hat, in ein Referenzmodell, beispielsweise das CAD-Modell des vermessenen Objektes. Die Transformation aus dem lokalen Sensorkoordinatensystem in das Koordinatensystem des Referenzmodells nennt man *Registrierung*. In diesem Abschnitt wird zunächst das Registrierungsproblem formal beschrieben und dann der ICP-Algorithmus nach [BM92] vorgestellt.

Gegeben sei eine Menge Punkte $P = \{\vec{p}_0, \dots, \vec{p}_n\}$ (*Datenpunkte*). Diese soll rigide transformiert (positioniert) werden, und zwar so, das sie sich in optimaler Ausrichtung an einer gegebenen Oberfläche X (*Modellform*) befindet. Das Abstandsmaß zwischen einem Einzelpunkt \vec{p} und der Modellform X bezeichnet man mit $d(\vec{p}, X)$. \vec{y} sei der Punkt auf X , der diesen kleinsten Abstand zu \vec{p} hat, also $\vec{y} \in X$, so dass $d(\vec{p}, \vec{y}) = d(\vec{p}, X)$. Wir führen den Operator \mathcal{C} ein, der zur Menge P die nächsten Punkte Y auf X berechnet:

$$Y = \mathcal{C}(P, X).$$

Die rigide Transformation θ sei die Registrierung der Punktmenge P und Y , also die Transformation, die P und Y unter dem Kriterium kleinster Fehlerquadrate in optimale Deckung bringt. Sie lässt sich durch Einzelwertzerlegung (Single Value Decomposition, SVD, siehe [Aru87]) oder Repräsentation im Quaternionenraum ([Hor87]) in geschlossener Form berechnen. Sei \mathcal{T} der Operator, mit dem θ und der resultierende mittlere quadratische Fehler d berechnet wird:

$$\begin{pmatrix} \theta & d \end{pmatrix} = \mathcal{T}(P, Y).$$

Der ICP-Algorithmus kann jetzt formuliert werden:

Wiederhole die folgenden Schritte bis zur Terminierung:

1. Die Iteration wird initialisiert, indem $P = P_0$, $\theta = id$ und $k = 0$ gesetzt werden.
2. Schritte a, b, c und d werden nun bis zur Konvergenz innerhalb der Toleranz τ iteriert:
 - (a) Berechne die nächsten Punkte: $Y_k = \mathcal{C}(P_k, X)$
 - (b) Berechne die Registrierung: $\begin{pmatrix} \theta_k & d_k \end{pmatrix} = \mathcal{T}(P_0, Y_k)$
 - (c) Wende die Registrierung an: $P_{k+1} = \theta_k(P_0)$

- (d) Terminiere, wenn die Änderung des kleinsten quadratischen Fehlers unter die gemäß der gewünschten Präzision gewählte Schwelle τ fällt:
 $d_k - d_{k-1} < \tau$.

Für diese Arbeit musste insbesondere der in Schritt (b) angewendete Operator \mathcal{T} angepasst werden, da die in der Literatur im Zusammenhang mit ICP beschriebenen Implementierungen dieses Operators nur für die Registrierung rigider Objekte geeignet sind. Abschnitt 3.2 beschreibt, wie das Unscented Kalmanfilter zum Einsatz kam, um auch die Registrierung artikulärer Objekte zu ermöglichen. Zur effizienten Implementierung des Operators \mathcal{C} wurde ein kd-Baum verwendet. Diese Datenstruktur wird im folgenden Abschnitt beschrieben.

2.4 kd-Bäume

Der kd-Baum (kd steht für k -dimensional) ist eine Baumstruktur zur Optimierung mehrdimensionaler Suchanfragen, die zuerst in [Ben75] beschrieben wurde. Im Rahmen dieser Arbeit wird er verwendet, um die Suche nächster Nachbarn im ICP-ähnlichen Teil des Algorithmus zu optimieren, außerdem bei der Implementierung der in Abschnitt 3.1.2 beschriebene Filterstufe. Dieser Abschnitt erklärt kurz den Aufbau von kd-Bäumen und den Ablauf der nächster-Nachbar-Anfrage.

Beim kd-Baum handelt sich um einen verallgemeinerten binären Suchbaum, der k -dimensionale Schlüssel verwendet. Ohne Beschränkung der Allgemeinheit wird angenommen, dass als Daten k -dimensionale Vektoren im Baum abgelegt werden sollen und gleichzeitig als Schlüssel verwendet werden. Das Einfügen in den Baum geschieht genauso wie beim einfachen binären Baum, man wählt aber zum Indizieren auf Ebene i des Baumes das $i \bmod k$ -te Element des Schlüssels:

\vec{p} sei der einzufügende Schlüssel. Setze initial als nächsten zu besuchenden Knoten n die Wurzel des Baumes. Iteriere nun bis zur Terminierung folgende Schritte:

- Setze den *Diskriminator* $d = i \bmod k$. i ist die Tiefe von n .
- \vec{q} sei der bei n gespeicherte Schlüssel. Vergleiche die d -ten Elemente q_d und p_d von \vec{q} und \vec{p} : Gilt $q_d < p_d$, setze als den nächsten zu besuchenden Knoten n den *linken* Nachfolger von n , ansonsten den *rechten*. Gibt es hier keinen Nachfolger, so füge \vec{p} als neuen Knoten hier ein und terminiere.

Anschaulich repräsentiert jeder Knoten n eine $(k - 1)$ -dimensionale Hyperebene, die auf der Hyperebene des Vaterknotens senkrecht steht. Neu eingefügte Punkte werden dem linken oder rechten Teilbaum von n zugeordnet, je nachdem, auf welcher Seite der Hyperebene sie sich befinden. Die Hyperebene teilt die zum Punkt gehörige *Region*, die für $k = 2$ rechteckig ist, im allgemeinen Fall ein Hyperquader. Abbildung 2.9 veranschaulicht dies für $k = 2$.

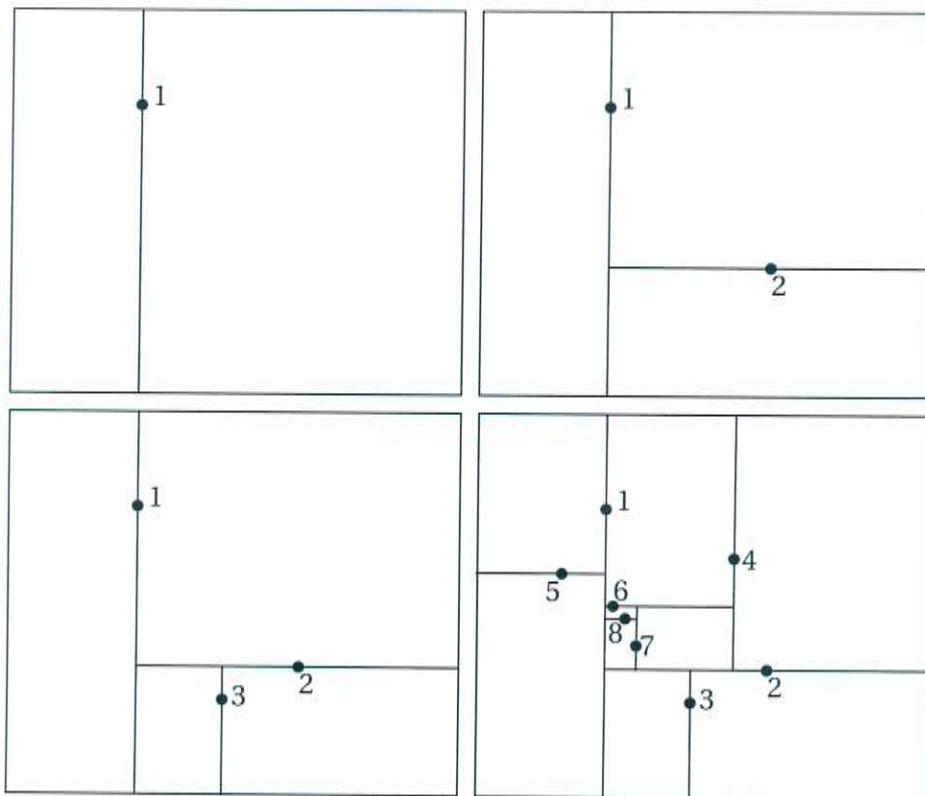


Abbildung 2.9: kd-Baum für $k = 2$. Die Punkte sind in der Reihenfolge ihres Einfügens nummeriert.

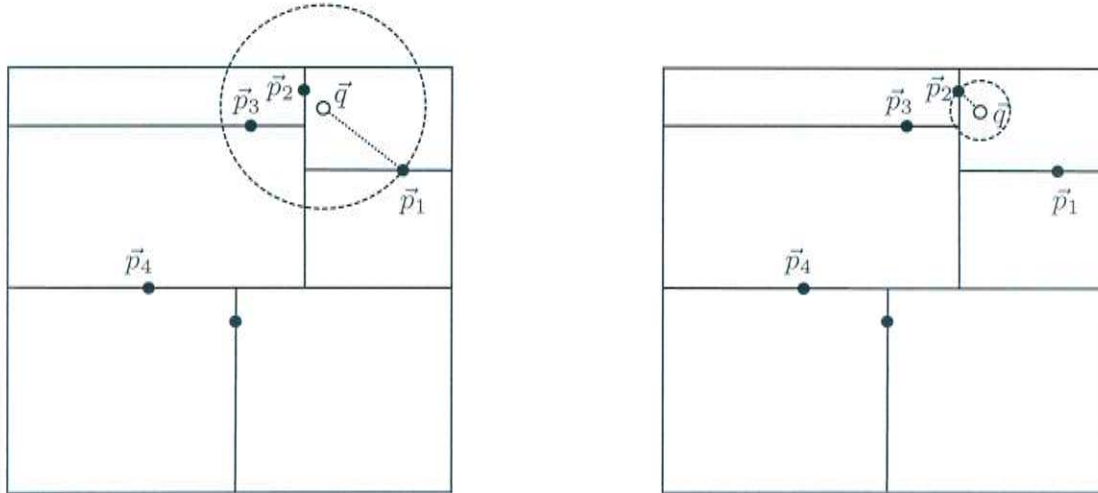


Abbildung 2.10: Nächster-Nachbar-Suche. Es soll der nächstgelegene Punkt zu \vec{q} gefunden werden. Man steigt zunächst im Baum bis zu \vec{p}_1 ab. Durch Backtracking gelangt man zunächst zum Vater \vec{p}_2 . Der Bounds-Overlap-Ball-Test (BOB) mit dem Abstand des bisher besten Kandidaten \vec{p}_1 ist für die zu \vec{p}_2 gehörige Region positiv, deshalb wird dieser Punkt überprüft. \vec{p}_2 liegt näher an \vec{q} als \vec{p}_1 , der Kreisradius für BOB wird entsprechend angepasst (rechts). \vec{p}_3 ist ein Kind von \vec{p}_2 , das noch überprüft werden muss, da sich seine Region noch mit dem Kreis überlappt. Der Algorithmus terminiert nach Backtracking zum Wurzelknoten \vec{p}_4 , da alle seine Kinder entweder bereits überprüft wurden, oder BOB fehlschlägt.

Der kd-Baum ist eine geeignete Datenstruktur, um das Problem nächster Nachbarn zu lösen (Operator \mathcal{C} aus Abschnitt 2.3), also zu einem Punkt \vec{q} den nächstgelegenen zu finden. Hierzu geht man zunächst wie beim Einfügen vor und findet so einen Blattknoten des Baumes. Der dort gespeicherte Punkt liegt in der Regel \vec{q} schon recht nah, der wahre nächste Nachbar könnte aber auch in angrenzenden Regionen liegen. Durch Backtracking werden deshalb noch alternative Äste des Baumes durchsucht. Es müssen alle Äste untersucht werden, deren Regionen sich mit einer Kugel um \vec{q} , deren Radius der Abstand zum nächsten bisher gefundenen Punkt ist, überlappen (*Bounds-Overlap-Ball-Test*). Abbildung 2.10 zeigt den Ablauf des Algorithmus.

Will man die k nächstgelegenen Punkte finden, so merkt man sich statt des besten bisher gefundenen Kandidaten eine Liste der k besten, der Ablauf ist ansonsten gleich. David Mounts Softwarebibliothek ANN (siehe [Mou98]), die für diese Arbeit zum Einsatz kam, enthält effiziente Implementierungen der beschriebenen Suchverfahren.

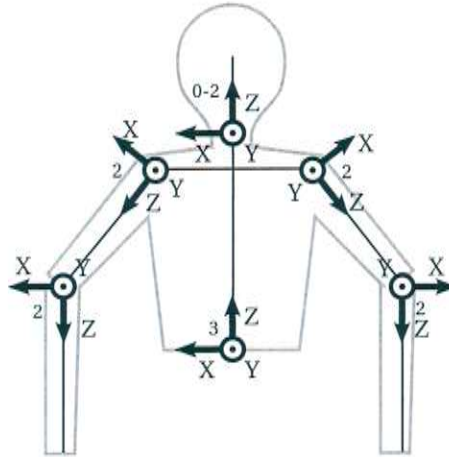


Abbildung 2.11: Kinematische Modellierung des Oberkörpers. Die Blickrichtung des Modells ist zum Betrachter hin. Jedes der gezeigten Koordinatensysteme wird durch eine Rotationsmatrix repräsentiert. Die Zahlen zeigen an, mit wievielen Freiheitsgraden diese parametrisiert werden. Um Spiegelsymmetrie herzustellen, sind die Koordinatensysteme des linken Armes linkshändig. Zu den hier gezeigten 11-13 rotatorischen Freiheitsgraden kommen drei weitere translatorische, die die Position des Modells im Weltkoordinatensystem beschreiben.

2.5 Kinematische Modellierung artikularer Objekte

Das im Rahmen dieser Arbeit für die Kalmanfilterung gewählte Messmodell enthält als wesentlichen Teil die Vorwärtskinematik des menschlichen Oberkörpers. In diesem Abschnitt werden die entsprechenden mathematischen Modelle eingeführt.

Vorwärtskinematik bedeutet hier die Transformation eines in einem beliebigen, lokalen Koordinatensystem (zum Beispiel dem Koordinatensystem eines bestimmten Körperteils) gegebenen Punktes des Oberkörpers in ein globales Weltkoordinatensystem. Die Transformation ist abhängig von den Gelenkwinkeln der Gliedmaßen. Abbildung 2.11 zeigt das in dieser Arbeit verwendete kinematische Modell des Oberkörpers und die lokalen Koordinatensysteme. Die Transformation von einem Koordinatensystem in ein in der kinematischen Kette direkt nachfolgendes (also zum Beispiel vom Torsokoordinatensystem ins Oberarmkoordinatensystem oder vom Oberarmkoordinatensystem ins Unterarmkoordinatensystem) besteht aus einer Rotation und einer Translation. *Homogene Transformationsmatrizen* erlauben es, die Kombination aus Rotation und Translation als eine

Matrixmultiplikation zu beschreiben:

$$\begin{pmatrix} x_{\text{transformiert}} \\ y_{\text{transformiert}} \\ z_{\text{transformiert}} \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

$(t_x \ t_y \ t_z)^T$ ist die translatorische Komponente, r_{ij} sind die Elemente der Rotationsmatrix. Wie man sieht, muss der ursprüngliche Koordinatenvektor um eine Komponente erweitert werden, die immer gleich eins ist.

Die Rotationsmatrizen sind mit Eulerwinkeln parametrisiert:

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix}.$$

Die Winkel α , β und γ entsprechen der Rotation um die Achsen xyz (*Pitch-Yaw-Roll*), in dieser Reihenfolge. γ ist bei allen Gelenken konstant, lediglich die Rotationsmatrix, die die Gesamtlage des Körpers beschreibt, erlaubt Drehung um alle Achsen.

Zur Transformation eines Vektors von einem Koordinatensystem in ein anderes muss man nun alle Transformationsmatrizen entlang der kinematischen Kette, die beide Koordinatensysteme verbindet, miteinander multiplizieren. Man erhält eine homogene Transformationsmatrix, die die Gesamttransformation repräsentiert.

Kapitel 3

Trackingalgorithmus

Die Gesamtverarbeitung der Stereobilder zerfällt in Vorverarbeitung und den eigentlichen Trackingalgorithmus. Ziel der Vorverarbeitung ist es, aus den Stereobildern eine Punktwolke zu gewinnen, die die zu verfolgende Person abbildet. Sie soll vom Hintergrund freigestellt sein. Die einzelnen Vorverarbeitungsschritte werden in Abschnitt 3.1 erläutert. Abschnitt 3.2 beschreibt den auf der Punktwolke operierenden Trackingalgorithmus.

3.1 Segmentierung und weitere Vorverarbeitung

Aus einem Stereobildpaar lässt sich mittels Stereobildverarbeitung (siehe Abschnitt 2.1) ein Disparitätenbild generieren, aus dem eine Punktwolke rekonstruiert werden kann, die einer 3D-Ansicht der beobachteten Szene entspricht. Dank der räumlichen Kalibrierung aller Kameras gegeneinander (Abschnitt 2.1.5) können die einzelnen Punktwolken aus deren lokalen Koordinatensystemen in ein gemeinsames Weltkoordinatensystem transformiert werden. Der Kern des Trackingalgorithmus (siehe Abschnitt 3.2) setzt eine hinreichend gute Freistellung (Segmentierung) der beobachteten Person vom Hintergrund voraus. Dieser wichtige Vorverarbeitungsschritt wird in diesem Abschnitt beschrieben. Abbildung 3.1 präsentiert einen Überblick der einzelnen Stufen der Vorverarbeitung .

3.1.1 Vordergrundsegmentierung auf Disparitätenbildern

Um eine erste, grobe Vordergrundsegmentierung zu erhalten, wurden die aus der Stereobildverarbeitung stammenden Disparitätenbilder (Abbildung 3.2 (a)) herangezogen. Dieses Verfahren unterscheidet sich von der konventionellen Vordergrundsegmentierung auf Intensitäts- oder RGB-Bildern entscheidend, ist aber prinzipiell einfacher. Die Vordergrundsegmentierung auf Disparitätenbildern ist im Vergleich zur konventionellen Segmentierung im RGB-Farbraum robuster gegenüber wechselnden Beleuchtungsverhältnissen und Schattenwürfen. Das hier

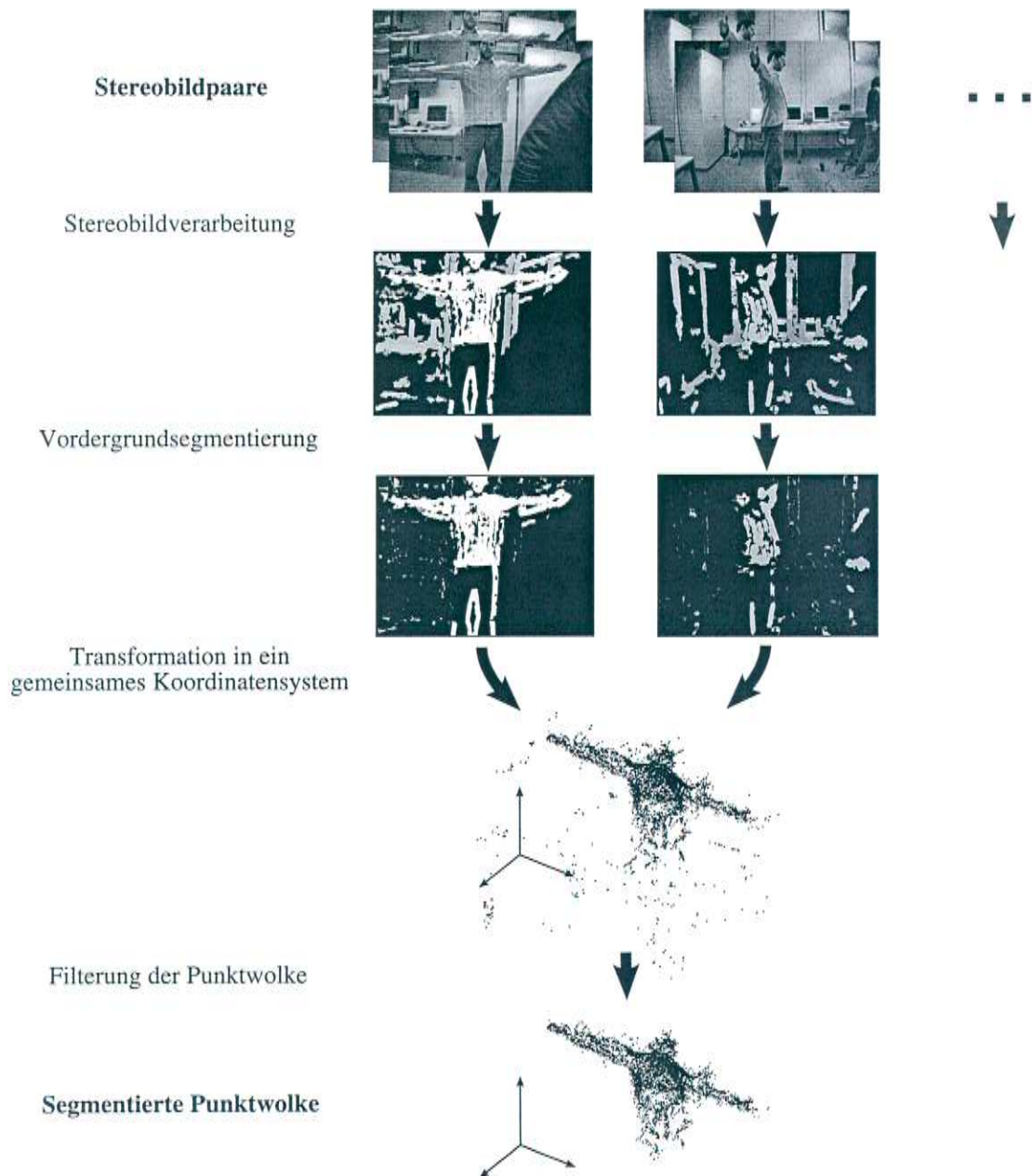


Abbildung 3.1: Die einzelnen Schritte der Vorverarbeitung.

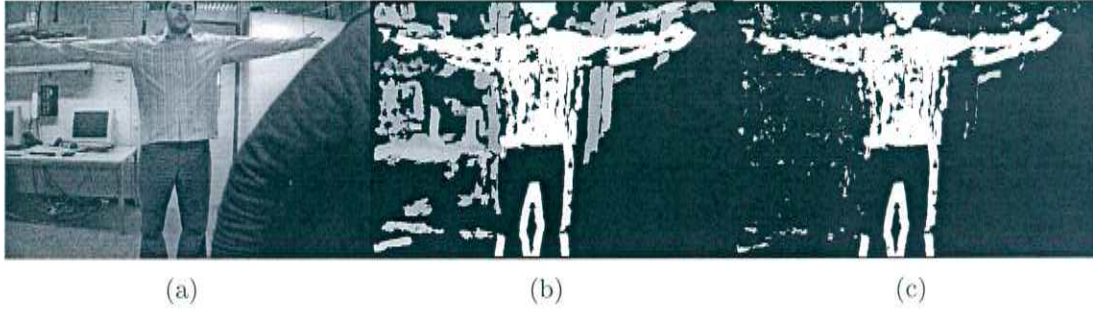


Abbildung 3.2: Vordergrundsegmentierung auf Disparitätenbildern. (a) Ein Einzelbild der Stereobildfolge. (b) Disparitätenbild. Schwarze Bereiche enthalten keine gültigen Tiefeninformationen. (c) Ergebniss der Vordergrundsegmentierung. Die Vordergrundpixel sind mit ihrer Disparität dargestellt, Hintergrund ist schwarz.

beschriebene Verfahren ist ähnlich zu [EKB98].

Die Ungenauigkeit der errechneten Disparität d ist unabhängig von der Entfernung des beobachteten Punktes. Die Entfernung z hängt mit der Disparität über einen Zusammenhang der Form $z \sim \frac{1}{d}$ (Vergleiche 2.1) zusammen, die resultierende Ungenauigkeit von z ist also nicht unabhängig vom Betrag von d beziehungsweise z . Aus diesem Grund ist es ratsam, die Disparitätenbilder statt der echten Tiefenbilder zu verarbeiten.

Existiert ein Modell D_{Modell} des statisch angenommenen Hintergrundes in Form eines Disparitätenbildes, so ist das Konzept der Vordergrundsegmentierung denkbar einfach: Um ein aufgenommenes Disparitätenbild D zu segmentieren, wird die Disparität d jedes Pixels aus D mit der des entsprechenden Pixels in D_{Modell} , d_{Modell} , verglichen. Gilt $d > d_{Modell}$, so wird das Pixel dem Vordergrund zugeordnet, ansonsten dem Hintergrund. Es wird für die Unterscheidung der Operator $\mathcal{V}(d, d_{Modell})$ eingeführt und schrittweise verfeinert. Im idealisierten Fall kann man \mathcal{V} wie beschrieben wählen:

$$\mathcal{V}(d, d_{Modell}) = \begin{cases} d > d_{Modell} & : \text{Vordergrund} \\ sonst & : \text{Hintergrund} \end{cases} \quad (3.1)$$

In der Praxis müssen einige weitere Faktoren beachtet werden: Zunächst enthalten die Disparitätenbilder durch die in Abschnitt 2.1.4 beschriebene Nachverarbeitung Pixel ohne gültige Tiefeninformation. Finden man gültige Tiefeninformation an Stellen, an denen das Modell als ungültig markiert ist, so zum Beispiel, wenn eine Person mit gemustertem Hemd vor einem sehr schwach texturierten Teil des Hintergrundes steht, so können die entsprechenden Pixel dem Vordergrund zugeordnet werden:

$$\mathcal{V}_1(d, d_{Modell}) = \begin{cases} gueltig(d) \wedge ungueltig(d_{Modell}) & : \text{Vordergrund} \\ d > d_{Modell} & : \text{Vordergrund} \\ sonst & : \text{Hintergrund} \end{cases} \quad (3.2)$$

Prinzipiell gilt auch der komplementäre Fall, steht also eine Person mit schwach texturiertem Hemd vor einer gemusterten Tapete, so könnte man auch hier Vordergrundpixel ausmachen. Dieser Fall wird im Folgenden ignoriert, da Vordergrundpixeln ohne gültige Tiefeninformation für die Weiterverarbeitung wertlos sind.

Die Disparitätenbilder sind verrauscht, hieraus würden bei einer strengen Diskriminierung wie in (3.1) und (3.2) Artefakte resultieren. Deshalb wird ein Schwellwert s in die Ungleichung eingeführt:

$$\mathcal{V}_2(d, d_{Modell}) = \begin{cases} gueltig(d) \wedge ungueltig(d_{Modell}) & : \text{Vordergrund} \\ d > d_{Modell} + s & : \text{Vordergrund} \\ sonst & : \text{Hintergrund} \end{cases} \quad (3.3)$$

Dieser Operator funktioniert bei einem guten Hintergrundmodell recht zuverlässig. Zu klären bleibt, wie das Modell des Hintergrundes erzeugt wird.

Man kann ein Hintergrundmodell erzeugen, indem man die aufgenommenen Disparitätenbilder laufend mit exponentieller Lernrate mittelt. Das neue Hintergrundmodell d_{Modell}^k (k ist der diskrete Zeitindex der Bildfolge) errechnet sich für jedes neue Disparitätenbild d^k zu

$$d_{Modell}^k = (1 - \alpha)d_{Modell}^{k-1} + \alpha d^k$$

mit der Lernrate α , die zwischen null und eins liegen muss. Ähnlich wird ein Array G_{modell} gelernt, das angibt, wo im Hintergrund mit gültigen Disparitäten zu rechnen ist:

$$g_{Modell}^k = (1 - \alpha_g)g_{Modell}^{k-1} + \alpha_g g^k.$$

g^k enthält bei einer gültigen Disparität eins, ansonsten null. Entsprechend zeigt G_{Modell} für jedes Pixel an, mit welcher Häufigkeit dort in den letzten Bildern der Folge gültige Tiefeninformation vorhanden war. Das Array G_{Modell} wird verwendet, um die Prädikate *gueltig* und *ungueltig* aus (3.3) auszuwerten. Hierzu wird der Wert von G_{Modell} an der jeweiligen Stelle mit einem Schwellwert verglichen. Abbildung 3.2 (c) zeigt die Wirkung des Verfahrens.

3.1.2 Filterung der 3D-Daten

Transformiert man die segmentierten Disparitätenbilder zu Punktwolken (siehe 2.1) und fusioniert die Messungen der Einzelkameras (2.1.5), erhält man eine Punktwolke, in der zwar die meisten zum Hintergrund gehörigen Punkte entfernt sind, einige aber noch als Artefakte verblieben sind. Sie resultieren aus fehlerhaft berechneten Korrespondenzen bei der Stereobildverarbeitung sowie mangelhafter Segmentierung. Diese Artefakte werden in einer nachgeschalteten Filterung entfernt.

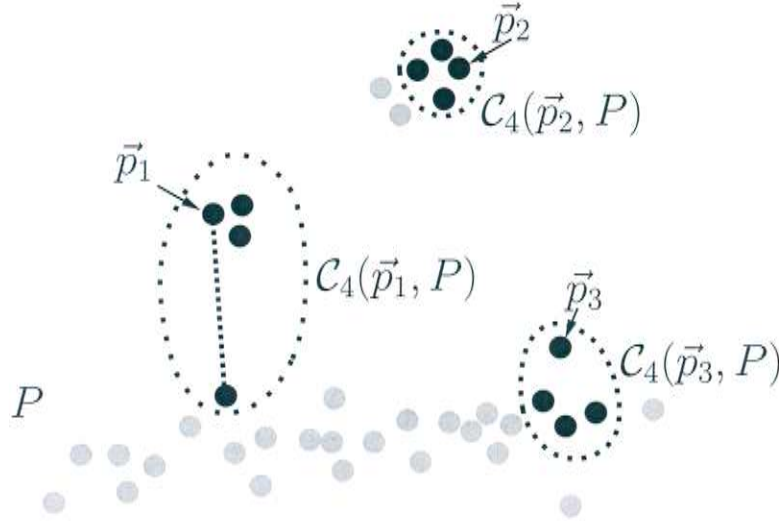


Abbildung 3.3: Filterung der vorsegmentierten Punktwolke. Exemplarisch werden \vec{p}_1 , \vec{p}_2 und \vec{p}_3 für $k = 4$ überprüft. \vec{p}_1 wird entfernt, da der eingezeichnete Abstand zu groß ist. Schrittweise würde im weiteren Verlauf das vollständige Artefakt um \vec{p}_1 entfernt werden. \vec{p}_2 und \vec{p}_3 werden nicht entfernt.

Die auftretenden Artefakte sind ihrem Charakter nach Punktbündel kleiner Ausdehnung und Mächtigkeit. Um sie zu entfernen, kommt der k -nächste-Nachbarn-Operator \mathcal{C}_k zum Einsatz. $\mathcal{C}_k(\vec{p}, P)$ ermittelt zu einem Punkt \vec{p} die k nächstgelegenen Punkte aus P . In Abschnitt 2.4 wird eine effiziente Implementierung beschrieben.

Sei $P = \{\vec{p}_0, \dots, \vec{p}_n\}$ die ungefilterte, vorsegmentierte Punktwolke. Für alle \vec{p}_i ermittelt man sukzessive die k nächsten Nachbarn in P : $\mathcal{C}_k(\vec{p}_i, P)$. Man entfernt \vec{p}_i , wenn der Abstand eines der Elemente aus $\mathcal{C}_k(\vec{p}_i, P)$ zu \vec{p}_i einen Maximalwert d_{max} überschreitet (siehe Abbildung 3.3). Abbildung 3.4 zeigt die Wirkung des Verfahrens.

Die amortisierten Kosten für die Nächster-Nachbar-Suche in n Punkten ist bei der Implementierung mit kd-Bäumen (Abschnitt 2.4) $\mathcal{O}(\log n)$, die Anwendung des Verfahrens auf alle n Punkte hat dementsprechend die Komplexität $\mathcal{O}(n \log n)$.

3.2 Tracking artikularer Objekte in Punktwolken

Die in Abschnitt 3.1 durchgeführte Vorverarbeitung liefert das vom Hintergrund freigestellte Abbild der zu verfolgenden Person als 3D-Punktwolke P_m . Das im Rahmen dieser Arbeit entwickelte System verwendet ein Registrierungsverfahren, um ein artikulares Modell des menschlichen Oberkörpers (siehe Abschnitt 2.5)

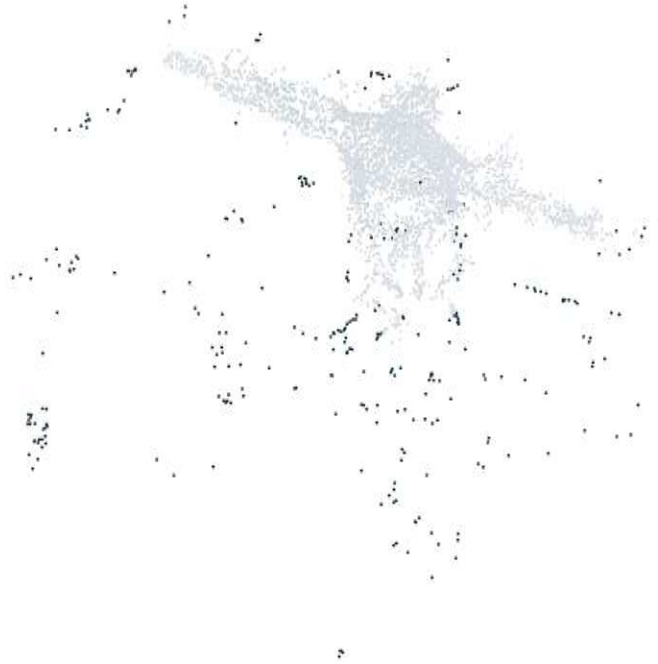


Abbildung 3.4: *Filterung der 3D-Daten. Die schwarzen Punkte wurden durch das Artefaktfilter entfernt.*

dieser Punktwolke nachzuführen. Hierzu wurde der bekannte ICP-Algorithmus modifiziert, um die Registrierung nicht-rigider Körper zu ermöglichen. Das Problem der Registrierung zweier Punktwolken wurde dazu als lineares Schätzproblem formuliert, das mit einem Kalmanfilter gelöst werden kann. Da das beteiligte funktionale Messmodell nichtlinear ist, kam eine linearisierende Variante des Kalmanfilters (*Unscented Kalmanfilter*, Abschnitt 2.2.4) zum Einsatz, um es zu linearisieren.

Abschnitt 3.2.1 beschreibt zunächst das verwendete Systemmodell, Abschnitt 3.2.2 das Messmodell sowie die Generierung eines geeigneten Messvektors aus der von der Vorverarbeitung gelieferten Punktwolke. Abschnitt 3.2.3 demonstriert dann detailliert den Ablauf der Schätzung.

3.2.1 Systemmodell

Der Kalmanfilter schätzt einen numerischen Zustandsvektor \mathbf{x} . Dieser beschreibt bei uns die Konfiguration des menschlichen Oberkörpers und enthält die Parameter der in Abschnitt 2.5 beschriebenen Gelenke beziehungsweise ihrer Rotationsmatrizen. In die Zustandsschätzung fließt wie in Abschnitt 2.2.1 beschrieben Wissen über die Dynamik des Zustandes \mathbf{x} ein, das in Form des funktionalen Systemmodells f aus Gleichung (2.4) sowie der Kovarianzmatrix \mathbf{Q} , die die Verlässlichkeit von f beschreibt, eingebracht wird. Da die Bildrate der verwendeten

Videohardware nicht besonders hoch ist (~ 10 Hz), die Bewegung einer Person aber vergleichsweise hochfrequent und schwer vorhersehbar, wurde als Dynamik des Systems Konstanz ($f(\mathbf{x}) = \mathbf{x}$) angenommen, allerdings bei entsprechend hoher Kovarianz \mathbf{Q} . Der Systemprozess ist demnach ein mehrdimensionaler, *symmetrischer Random Walk* mit Kovarianz \mathbf{Q} . Wir nehmen an, dass die Bewegungen der einzelnen Körperteile beziehungsweise Gelenkwinkel stochastisch unabhängig voneinander sind, \mathbf{Q} hat also Diagonalform.

3.2.2 Messmodell und Generierung des Messvektors

Das für die Filterung verwendete Messmodell $h(\mathbf{x})$ aus Gleichung 2.5 stützt sich auf die in Abschnitt 2.11 beschriebene Vorwärtskinematik des Oberkörpers und den Closest Point-Operator aus Abschnitt 2.4. Das Messmodell wird in diesem Abschnitt schrittweise aufgebaut.

Zunächst tastet man eine Menge Punkte $P_{\hat{z}}$ von der Oberfläche des Oberkörpermodells ab. $P_{\hat{z}}$ soll eine Hypothese für die Punktwolke sein, die man im Systemzustand \mathbf{x} beobachten würde. Das Oberkörpermodell wird aus affin transformierten Kegelstümpfen zusammengesetzt (Abbildung 3.5 (a)). Abbildung 3.5 (b) zeigt, wie ein im Koordinatenursprung liegender und an der z -Achse ausgerichteter Kegelstumpf abgetastet wird. Es gilt

$$\mathbf{p}(h, \alpha) = \left(r_1 + h \frac{r_2 - r_1}{H} \right) \begin{pmatrix} \cos \alpha \\ \sin \alpha \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix}.$$

Einige Körperteile bestehen aus mehreren derartigen Stümpfen, die gegebenenfalls gegen den Koordinatenursprung verschoben und gedreht werden müssen. Dies ist der Fall beim Torso, der aus einem großen Stumpf und zwei kleineren, die die Schultern modellieren, besteht. Der Hauptstumpf des Torsos wird in y -Richtung abgeflacht. Der Kopf wird aus zwei aneinandergesetzten Kegelstümpfen modelliert, ebenso die Unterarme, die zusammen mit den Händen als ein in sich starres Körperteil modelliert werden.

Die so erhaltenen Punkte werden nun aus den lokalen Koordinatensystemen der einzelnen Körperteile mittel Vorwärtskinematik (siehe Abschnitt 2.5) in das Weltkoordinatensystem transformiert. Die Abbildungen 3.5 (c)-(d) zeigen zwei verschiedene Möglichkeiten, die Modelloberfläche abzutasten. Abschließend werden Punkte, die im Inneren des Modelles liegen oder die durch Verdeckung auf keiner Kamera sichtbar sind, entfernt (Abbildung 3.5 (e)).

Wir führen den Operator \mathcal{G} ein, der die Punktmenge $P_{\hat{z}} = \{\mathbf{p}_{\hat{z}}^0, \mathbf{p}_{\hat{z}}^1, \dots, \mathbf{p}_{\hat{z}}^m\}$ zu einem gegebenen Zustand \mathbf{x} wie beschrieben generiert:

$$P_{\hat{z}} = \mathcal{G}(\mathbf{x}). \quad (3.4)$$

Man berechnet nun mit dem im Abschnitt 2.11 eingeführten Operator \mathcal{C} die zu $P_{\hat{z}}$ korrespondierenden Punkte $P_z = \{\mathbf{p}_z^0, \mathbf{p}_z^1, \dots, \mathbf{p}_z^m\}$ aus der von der Vorver-

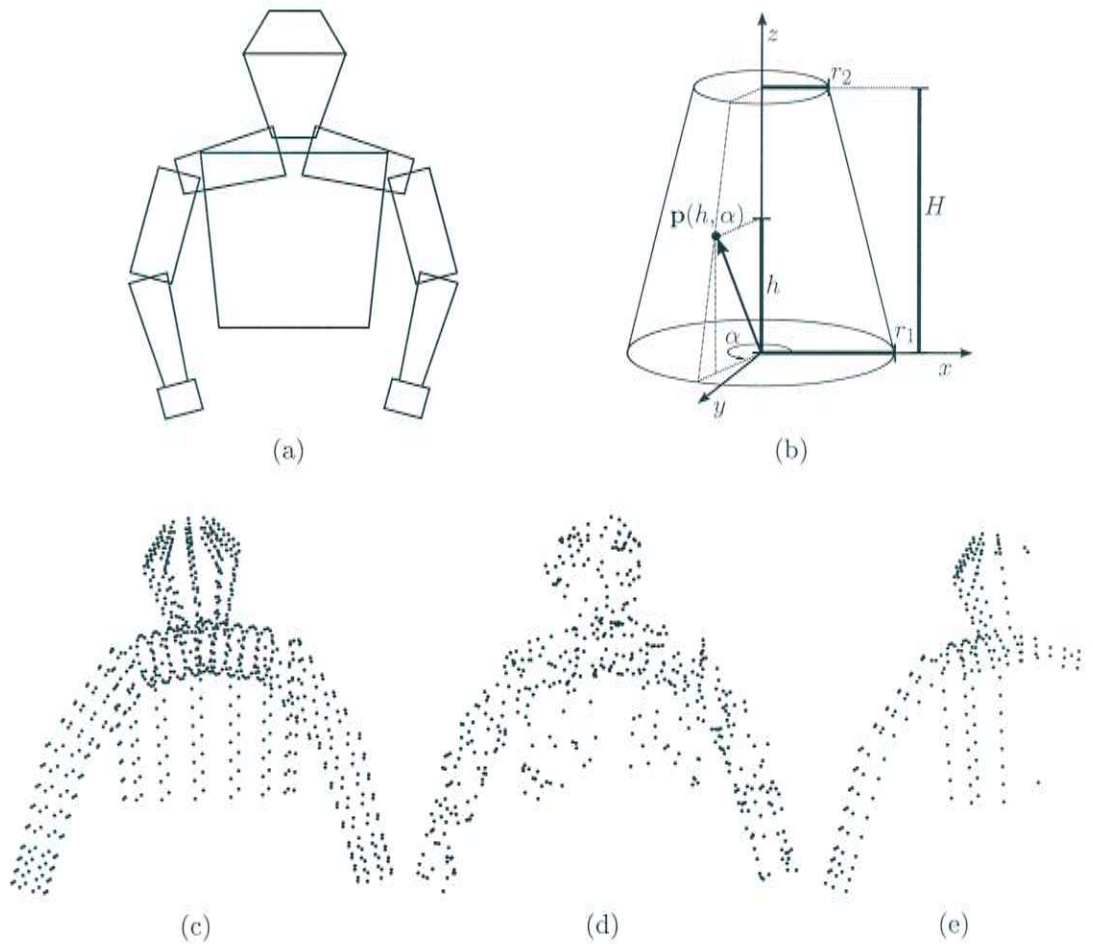


Abbildung 3.5: Generierung der Modellpunkte P_h . Bild (a): Das Oberkörpermodell ist aus 11 affin transformierten Kegelstümpfen zusammengesetzt. Bild (b) zeigt das Abtasten eines Kegelstumpfes. Man beschreibt einen Punkt \mathbf{p} des Stumpfmantels zweckmäßigerweise wie im Bild über die Höhe h und den Winkel α . Der Stumpf selber wird über die beiden Radien r_1 und r_2 der Stirnflächen sowie seine Höhe H beschrieben. Die Modellpunkte werden von den Kegelstümpfen (c) deterministisch oder (d) zufällig abgetastet. Schließlich werden Punkte, die in keiner Kamera sichtbar wären, entfernt. (e) zeigt dies für den Fall, das nur eine Kamera verwendet wird, die von links auf das Modell gerichtet ist.

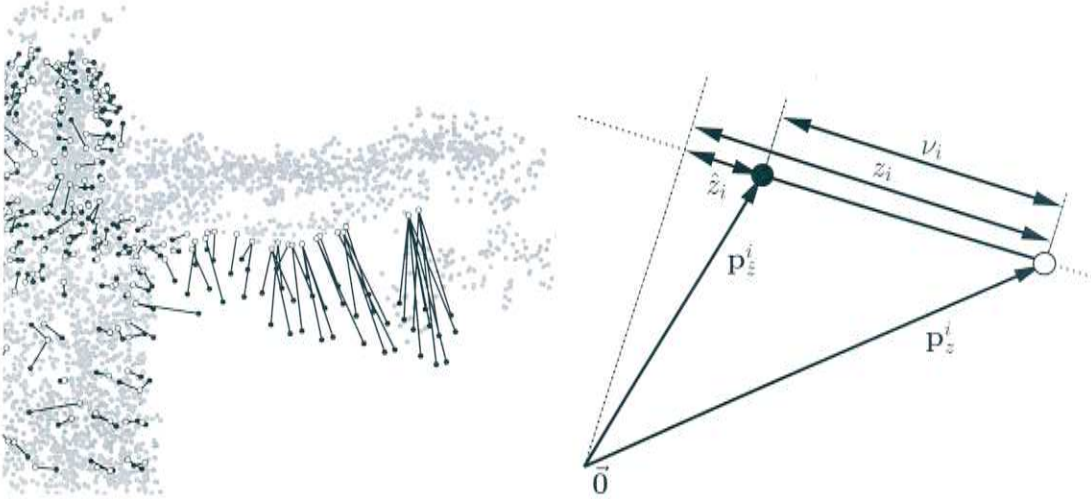


Abbildung 3.6: Messmodell. Die schwarzen Punkte \mathbf{p}_z^i entsprechen der prädizierten Messung, die weißen Punkte, \mathbf{p}_z^i , der Messung selbst. Das rechte Bild zeigt, wie aus jeweils einer Korrespondenz $(\mathbf{p}_z^i, \mathbf{p}_z^i)$ eine Komponente von $\hat{\mathbf{z}}$ und \mathbf{z} gebildet wird. Wie man sieht, sind die Komponenten des Residuums $\nu_i = \hat{z}_i - z_i$ die Abstände der korrespondierenden Punkte.

arbeitung gelieferten Punktwolke P_M . Dieser Schritt ist identisch zum Vorgehen im gewöhnlichen ICP-Algorithmus aus Abschnitt 2.3:

$$P_z = \mathcal{C}(P_{\hat{z}}, P_M).$$

Im nächsten Schritt muss $P_{\hat{z}}$ mit P_z registriert werden. Beim ICP-Algorithmus zur Registrierung starrer Körper kommen hier unterschiedliche Verfahren zum Einsatz (Singularwertzerlegung, Quaternionenrepräsentation), die aber im artikularen Fall nicht anwendbar sind. Wir betrachten deshalb das Problem der Registrierung als lineares Schätzproblem, das mit einem Kalmanfilter gelöst werden kann.

Die Menge P_z repräsentiert hier in der Notation aus Abschnitt 2.2.1 die Messung $z[k+1]$, $P_{\hat{z}}$ die vorhergesagte Messung $\hat{z}[k+1|k]$. Der diskrete Zeitindex k wird in diesem Abschnitt weggelassen und nur \mathbf{z} und $\hat{\mathbf{z}}$ geschrieben. Die Vektoren \mathbf{z} und $\hat{\mathbf{z}}$ müssen aus P_z und $P_{\hat{z}}$ generiert werden.

Der direkteste Ansatz, aus den Punktmengen Vektoren zu bilden, wäre, die Einzelpunkte direkt zu Vektoren zusammenzuführen, aus $P = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_m\}$ würde also der Vektor

$$\left(\mathbf{p}_0^x \ \mathbf{p}_0^y \ \mathbf{p}_0^z \ \mathbf{p}_1^x \ \mathbf{p}_1^y \ \mathbf{p}_1^z \ \cdots \ \mathbf{p}_{m-1}^x \ \mathbf{p}_{m-1}^y \ \mathbf{p}_{m-1}^z \right)^T,$$

mit $\mathbf{p}_i = (\mathbf{p}_i^x \ \mathbf{p}_i^y \ \mathbf{p}_i^z)^T$. Analog würden aus P_z und $P_{\hat{z}}$ die $3m$ -dimensionalen Vektoren \mathbf{z} und $\hat{\mathbf{z}}$ generiert. Durch einen Trick ist es möglich, die Dimensionalität

von \mathbf{z} und $\hat{\mathbf{z}}$ auf m zu reduzieren. Dies ist besonders lohnenswert, da die Komplexität des Korrekturschrittes des Kalmanfilters kubisch mit der Dimensionalität des Messvektors zusammenhängt (siehe Abschnitt 2.2.1). Die Idee hierbei ist, dass bei der Bewegung eines Punktes aus $\mathbf{p}_z^i \in P_z$ hauptsächlich der Anteil der Bewegung, der zu seinem korrespondierenden Punkt $\mathbf{p}_z^i \in P_z$ hin (oder davon weg) verläuft, für eine gute Passung ausschlaggebend ist. Deshalb werden \mathbf{p}_z^i und \mathbf{p}_z^i auf ihre Verbindungsgerade projiziert. Abbildung 3.6 illustriert das Verfahren. Aus \mathbf{p}_z^i und \mathbf{p}_z^i werden also die skalaren Werte

$$\hat{z}_i = \frac{\mathbf{p}_z^i - \mathbf{p}_z^i}{\|\mathbf{p}_z^i - \mathbf{p}_z^i\|} \cdot \mathbf{p}_z^i \quad (3.5)$$

$$z_i = \frac{\mathbf{p}_z^i - \mathbf{p}_z^i}{\|\mathbf{p}_z^i - \mathbf{p}_z^i\|} \cdot \mathbf{p}_z^i. \quad (3.6)$$

berechnet. Die Vektoren $\hat{\mathbf{z}}$ und \mathbf{z} werden nun durch Zusammenführen aller \hat{z}_i und z_i gebildet:

$$\hat{\mathbf{z}} = \begin{pmatrix} \hat{z}_0 & \hat{z}_1 & \cdots & \hat{z}_{m-1} \end{pmatrix}^T \quad (3.7)$$

$$\mathbf{z} = \begin{pmatrix} z_0 & z_1 & \cdots & z_{m-1} \end{pmatrix}^T \quad (3.8)$$

Die Generierung der prädizierten Messung aus den korrespondierenden Punktmengen P_z und P_z mittels (3.5) und (3.7) wird zum Operator $\hat{\mathcal{Z}}$ zusammengefasst:

$$\hat{\mathbf{z}} = \hat{\mathcal{Z}}(P_z, P_z). \quad (3.9)$$

Die Messung selbst wird mit dem Operator \mathcal{Z} generiert ((3.6) und (3.8)):

$$\mathbf{z} = \mathcal{Z}(P_z, P_z). \quad (3.10)$$

Wie man gesehen hat, muss zur Auswertung des Messmodells der Closest Point-Operator \mathcal{C} angewendet werden. Aus diesem Grund müsste \mathcal{C} im Korrekturschritt des UKF $2n+1$ mal (n : Dimensionalität des Zustandsraumes) ausgewertet werden (für jeden σ -Punkt einmal). Da man durch Skalierung der Unscented Transformation (siehe Abschnitt 2.2.4) die σ -Punkte beliebig nah zueinander bringen kann, kann man sich damit begnügen, die Punktkorrespondenzen im Korrekturschritt nur einmal herzustellen, und zwar basierend auf dem prädizierten Zustand $\hat{\mathbf{x}}$. Das entsprechende Messmodell wird hier $h_{\hat{\mathbf{x}}}(\mathbf{x})$ genannt. Es lässt sich zusammenfassend formulieren als

$$h_{\hat{\mathbf{x}}}(\mathbf{x}) = \hat{\mathcal{Z}}(P_z, P_z), \quad (3.11)$$

mit $P_z = \mathcal{G}(\mathbf{x})$ und $P_z = \mathcal{C}(\mathcal{G}(\hat{\mathbf{x}}), P_m)$. P_m ist die segmentierte und vorverarbeitete Punktwolke aus der Stereobildverarbeitung. Der zugehörige Messvektor \mathbf{z} wird mit (3.10) erzeugt.

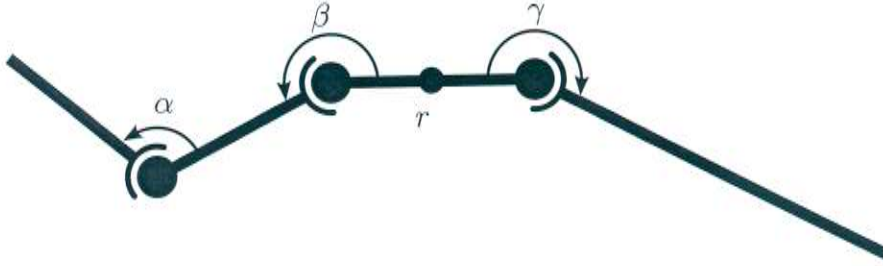


Abbildung 3.7: Einfaches artikuläres Modell mit den drei Freiheitsgraden α , β und γ . Das Modell ist bei r fest.

3.2.3 Ablauf der Filterung

In diesem Abschnitt wird ein vollständiger Vorhersage-Korrektur-Zyklus des Filters demonstriert. Es wird aber aus Gründen der Anschaulichkeit ein stark vereinfachtes Zustandsmodell mit nur 3 Winkeln verwendet, dessen Kinematik in Abbildung 3.7 dargestellt ist. Der Zustandsvektor hat die Form $\mathbf{x} = (\alpha \ \beta \ \gamma)^T$.

Man geht von der Zustandsschätzung zum letzten diskreten Zeitpunkt k aus. Die Schätzung besteht aus dem Mittelwert $\hat{\mathbf{x}}[k|k]$ und der Kovarianzmatrix $\mathbf{P}[k|k]$, die die Ungenauigkeit der Schätzung beschreibt. Zur Erinnerung: Mit $\hat{\mathbf{x}}[i|j]$ wird die Schätzung zum Zeitpunkt i , unter Einbeziehung aller Messungen bis zum Zeitpunkt j , bezeichnet (vergleiche Abschnitt 2.2.1). Die Extrapolation von $\hat{\mathbf{x}}[k+1|k]$ und $\mathbf{P}[k+1|k]$ (Prädiktion) ist im hier betrachteten Fall trivial, da als Systemmodell Konstanz angenommen wird (siehe Abschnitt 3.2.1). Es vergrößert sich also lediglich die Kovarianz der Schätzung um das Systemrauschen \mathbf{Q} :

$$\begin{aligned}\hat{\mathbf{x}}[k+1|k] &= \hat{\mathbf{x}}[k|k] \\ \mathbf{P}[k+1|k] &= \mathbf{P}[k|k] + \mathbf{Q}.\end{aligned}$$

Abbildung 3.8 veranschaulicht dies. Die folgenden Ausführungen befassen sich meistens mit den prädizierten Größen $\hat{\mathbf{x}}[k+1|k]$ und $\mathbf{P}[k+1|k]$, die Zeitindizes werden dann weggelassen.

Abbildung 3.9 zeigt die aus $\hat{\mathbf{x}}$ und \mathbf{P} mittels (2.12)-(2.14) generierten σ -Punkte $\{\mathcal{X}_0, \dots, \mathcal{X}_6\}$, die weiter unten gebraucht werden. Sie sind eine vollständige Repräsentation von Mittelwert und Kovarianz der Zustandsschätzung, $\hat{\mathbf{x}}$ und \mathbf{P} liessen sich aus ihnen durch gewichtete Mittelung wie in (2.16) und (2.17) reproduzieren.

Nun wird der Korrekturschritt vorbereitet, indem die Punktkorrespondenzen hergestellt werden, auf denen das Messmodell $h_{\hat{\mathbf{x}}}$ basiert. Zunächst wird eine wohldefinierte Menge Punkte vom Modell, das sich im prädizierten Zustand $\hat{\mathbf{x}}$ befindet, abgetastet. Es wird angenommen, dass für das hier betrachtete Modell ein Operator \mathcal{G} existiert, der analog zu (3.4) funktioniert. $\mathcal{G}(\hat{\mathbf{x}})$ liefert, wie in Abbildung 3.10 ersichtlich, die Punktmenge $P_{\hat{\mathbf{x}}} = \{\mathbf{p}_{\hat{\mathbf{x}}}^0, \dots, \mathbf{p}_{\hat{\mathbf{x}}}^5\}$. Durch Anwendung des Closest Point-Operators \mathcal{C} erhält man die Menge korrespondierender Punkte

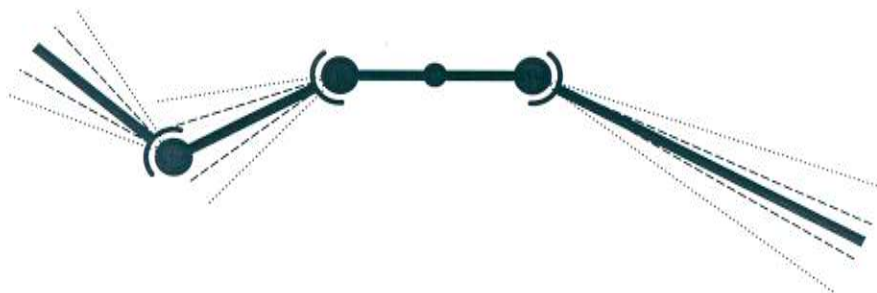


Abbildung 3.8: Prädiktion von $\hat{\mathbf{x}}[k+1|k]$ und $\mathbf{P}[k+1|k]$. Die Ungenauigkeit $\mathbf{P}[k|k]$ der letzten Zustandsschätzung $\hat{\mathbf{x}}[k|k]$ ist durch die gestrichelten Linien angedeutet. Sie vergrößert sich (gepunktete Linien) bei der Prädiktion um das Systemrauschen \mathbf{Q} .



Abbildung 3.9: Unscented-Repräsentation des prädierten Zustandes und seiner Ungenauigkeit. Jeder der 7 σ -Punkte $\{\mathcal{X}_0, \dots, \mathcal{X}_6\}$ repräsentiert eine vollständige Konfiguration des Modelles, hier schwarz dargestellt. Für die Darstellung wurde eine diagonale Kovarianzmatrix gewählt (keine Korrelation der Winkel untereinander), deshalb überlappen sich die Modelle teilweise, dies ist im Allgemeinen nicht der Fall.

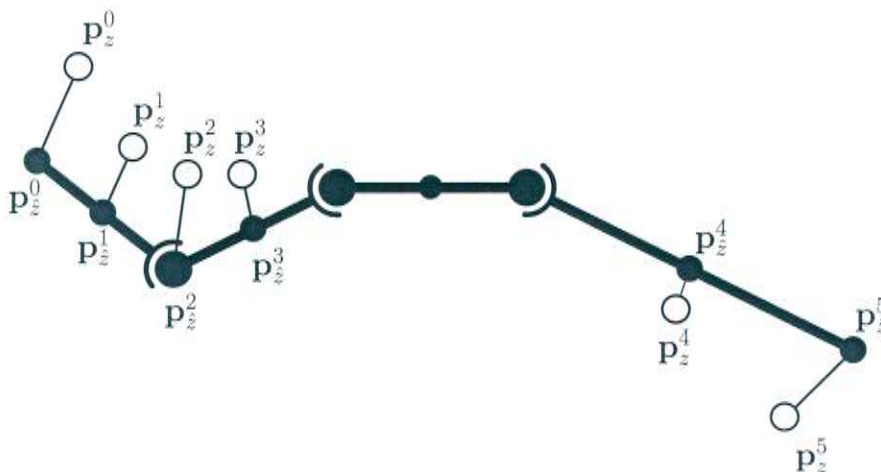


Abbildung 3.10: Korrespondenzen. Es wurde die Punktmenge $\mathcal{G}(\hat{\mathbf{x}}) = P_z = \{\mathbf{p}_z^0, \dots, \mathbf{p}_z^5\}$ vom Modell abgetastet (schwarz). Sie soll mit den korrespondierenden weisen Punkten $P_z = \{\mathbf{p}_z^0, \dots, \mathbf{p}_z^5\}$ registriert werden.



Abbildung 3.11: Transformation von $\{\mathcal{X}_0, \dots, \mathcal{X}_6\}$ durch das Messmodell $h_{\hat{x}}$, erster Schritt: Die zu registrierenden Punkte werden für alle \mathcal{X}_i abgetastet.

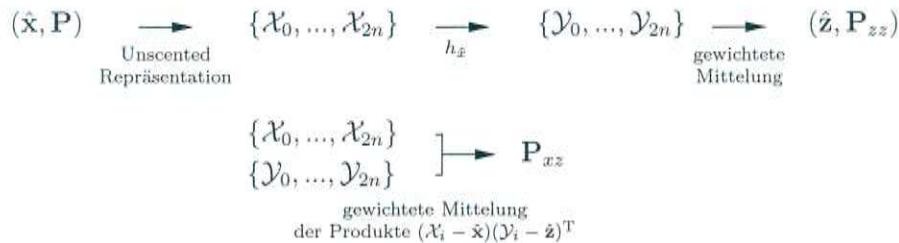


Abbildung 3.12: Die Unscented-Transformation im Überblick. n ist die Dimensionalität des Zustandsvektors, im Beispiel $n = 3$. Das Messmodell $h_{\hat{x}}$ hängt hier vom prädizierten Zustand \hat{x} ab, da auf ihm die ermittelten Punktkorrespondenzen basieren.

$P_z = \{\mathbf{p}_z^0, \dots, \mathbf{p}_z^5\}$. $\mathbf{P}_{\hat{z}}$ soll nun durch geeignete Korrektur von α , β und γ mit \mathbf{P}_z registriert werden.

Das Kalmanfilter ermittelt die optimale Korrektur des Zustandsvektors, indem er einen linearen Zusammenhang zwischen dem Zustandsvektor und dem Messvektor herstellt. Das UKF bedient sich hierzu der Unscented Transformation (Abbildung 3.12), die hier folgendermaßen abläuft: Die σ -Punkte $\{\mathcal{X}_0, \dots, \mathcal{X}_6\}$ werden zunächst durch das Messmodell transformiert. Hierzu werden in einem ersten Schritt für alle \mathcal{X}_i die Punkte $\mathcal{G}(\mathcal{X}_i)$ abgetastet (Abbildung 3.11). Dann erfolgt die Projektion auf die Verbindungsgerade des jeweiligen korrespondierenden Punktpaares. Durch zusammenführen der Projektionen aller sechs Punkte des Modells erhält für jeden σ -Punkt \mathcal{X}_i einen sechsdimensionalen Vektor, \mathcal{Y}_i .

Durch Mittelung der \mathcal{Y}_i erhält man den prädizierten Messvektor $\hat{\mathbf{z}}$ und seine Varianz \mathbf{P}_{zz} . Die Projektion und anschließende Mittelung ist für eine Komponente von $\hat{\mathbf{z}}$ in Abbildung 3.13 illustriert. Die dargestellte Varianz von \hat{z}_0 ist das 0-te Diagonalelement der Matrix \mathbf{P}_{zz} . Die Unscented-Transformation ermittelt alle Elemente dieser Matrix, gegebenenfalls auch Korrelation der Komponenten

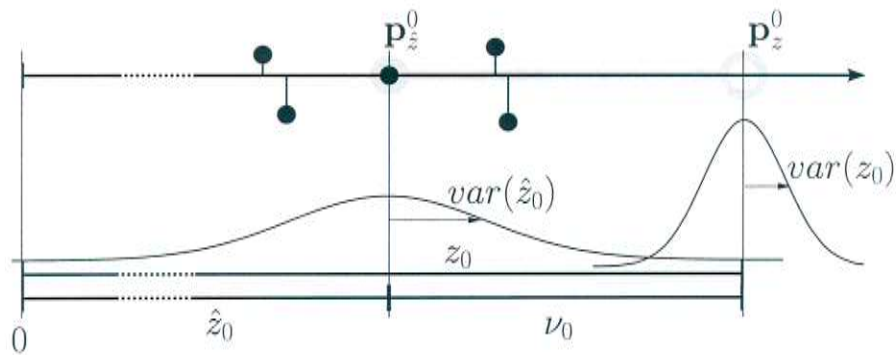


Abbildung 3.13: Transformation von $\{\mathcal{X}_0, \dots, \mathcal{X}_6\}$ durch das Messmodell $h_{\hat{z}}$, zweiter Schritt. Man betrachte die Umgebung von $\mathbf{p}_{\hat{z}}^0$ (Kasten aus Abbildung 3.11), die hier so rotiert wurde, dass $\mathbf{p}_z^0 - \mathbf{p}_{\hat{z}}^0$ entlang der Abszisse liegt. Die Projektion der zu $\mathbf{p}_{\hat{z}}^0$ gehörenden Punkte ist angedeutet (es sind tatsächlich sieben Punkte, sie liegen aber teilweise übereinander). Die Komponente \hat{z}_0 und ihre Varianz erhält man nun durch gewichtete Mittelung dieser Projektionen. Die Varianz von z_0 wird durch das Messrauschen \mathbf{R} vorgegeben und entspricht dem erwarteten Fehler der Messung. Wie man sieht, enthält jede skalare Komponente $\nu_i = \hat{z}_i - z_i$ des Residuums die Abstände der korrespondierenden Punkte.

	ν_0	ν_1	ν_2	ν_3	ν_4	ν_5
α						
β						
γ						

Abbildung 3.14: Kreuzkovarianz. Qualitative Darstellung der Kreuzkovarianzmatrix \mathbf{P}_{xz} für das in diesem Abschnitt beschriebene Beispiel. Schwarz bedeutet starke positive Korrelation, weiss starke negative Korrelation. Man kann beispielsweise ablesen, dass eine Vergrößerung des Winkels γ den Abstand $\overline{\mathbf{p}_{\hat{z}}^5 \mathbf{p}_z^5}$ verkleinern würde. Die Abstände ν_0 bis ν_3 dagegen sind mit dem Winkel γ unkorreliert.

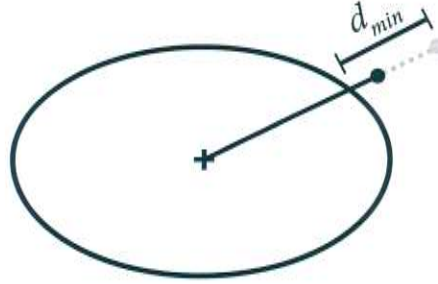


Abbildung 3.15: Bestimmung der neuen Hand- oder Ellbogenposition. Die Ellipse stellt einen Querschnitt des Torsos dar. Die alte Position (schwarz) wird soweit radial vom Zentrum der Ellipse weg verschoben, dass der Abstand d_{min} eingehalten wird. Wie man sieht, wird der Einfachheit halber statt des tatsächlichen Abstandes von der Ellipse ein „Pseudoabstand“ betrachtet.

von $\hat{\mathbf{z}}$ untereinander. Beispielsweise ist eine positive Korrelation von \hat{z}_0 und \hat{z}_1 zu erwarten, da die entsprechenden Punkte \mathbf{p}_z^0 und \mathbf{p}_z^1 nahe beieinander auf dem gleichen Schenkel des Modells liegen. Das Residuum $\nu = (\nu_0 \ \nu_1 \ \nu_2 \ \nu_3 \ \nu_4 \ \nu_6)^T$ ist die Differenz $\hat{\mathbf{z}} - \mathbf{z}$ von prädiziertem und aquiriertem Messvektor. Wie man sieht, enthalten es hier die Abstände der korrespondierenden Punkte. Seine Varianz ergibt sich aus \mathbf{P}_{zz} durch Addition des Messrauschens: $\mathbf{P}_{\nu\nu} = \mathbf{P}_{zz} + \mathbf{R}$.

Aus dem Residuum wird eine Korrektur des Zustandes $\hat{\mathbf{x}}$ berechnet. Es wird hierzu mit dem Kalmangain \mathbf{K} gewichtet: $\hat{\mathbf{x}}[k+1|k+1] = \hat{\mathbf{x}} + \mathbf{K}\nu$. In das Kalmangain gehen die Varianz $\mathbf{P}_{\nu\nu}$ des Residuums und die mittels (2.18) berechnete Kreuzkovarianzmatrix \mathbf{P}_{xz} , die ein Maß für den linearen Zusammenhang zwischen einer Änderung des Zustandsvektors und der resultierenden Änderung am Residuum darstellt, ein. Die Kreuzkovarianz ist für das hier betrachtete Beispiel in Abbildung 3.14 qualitativ dargestellt. Das Kalmangain berechnet sich zu $\mathbf{K} = \mathbf{P}_{xz}\mathbf{P}_{\nu\nu}^{-1}$. Die Korrektur der Zustandsschätzung kann nun durchgeführt werden:

$$\begin{aligned}\hat{\mathbf{x}}[k+1|k+1] &= \hat{\mathbf{x}} + \mathbf{K}\nu \\ \mathbf{P}[k+1|k+1] &= \mathbf{P} - \mathbf{K}\mathbf{P}_{\nu\nu}\mathbf{K}^T.\end{aligned}$$

Man iteriert den in diesem Abschnitt beschriebenen Korrekturschritt mehrmals, da analog zum gewöhnlichen ICP-Algorithmus zu erwarten ist, dass sich die Korrespondenzen mit jedem Mal verbessern und die Registrierung nach einigen Iterationen konvergiert.

3.2.4 Kollisionsvermeidung

Situationen, in denen Körperteile sich einander sehr nahe kommen, sind für das System schwer handhabbar, da es hier häufig zur Bildung falscher Korrespon-

denzen kommt. Liegt beispielsweise ein Arm am Körper an, kann es vorkommen, dass Modellpunkte des Armes mit Punkten des Körpers assoziiert werden. Daraus folgt oft eine fehlerhafte Verfolgung der Bewegung, im schlimmsten Fall bleiben Körperteile dauerhaft an anderen Körperteilen „hängen“. Diese Neigung wird dadurch verstärkt, dass die Bewegung des Körpermodells bisher keinen geometrischen Beschränkungen unterworfen ist, das heißt Körperteile können sich prinzipiell überschneiden. Häufig auftretende Fehler dieser Art sind das Hängenbleiben des Arms am oder im Torso sowie das „Einklappen“ eines Armes (der Unterarm liegt vollständig innerhalb des Oberarmes).

Um dies zu vermeiden, werden Posen, bei denen sich Körperteile sehr nahe kommen oder überschneiden, von vornherein vermieden. Wird nach der Registrierung des Modells mit der Punktwolke eines Satzes Einzelbilder eine solche Pose festgestellt, wird das Modell in einen Zustand gesetzt, bei dem die Körperteile bestimmte Mindestabstände einhalten. Mindestwerten werden hierbei erzwungen für

- den Abstand des Ellbogens vom Torso
- den Abstand der Hand vom Torso
- den Winkel, den Ober- und Unterarm bilden.

Wird bei Hand oder Ellbogen die Unterschreitung des Mindestabstandes festgestellt, wird die neue Position wie in Abbildung 3.15 bestimmt. Die Winkel (jeweils zwei Freiheitsgrade) von Schulter beziehungsweise Ellbogen werden nun so korrigiert, dass Hand beziehungsweise Ellbogen diesen Positionen möglichst nahe kommen. Da eine Änderung der Schulterwinkel ausser der Ellbogenposition auch die Handposition ändert, merkt man sich diese, bevor Korrekturen irgendwelcher Art durchgeführt werden. Dann korrigiert man (wenn nötig) als erstes die Schulterwinkel und den Winkel, den Ober- und Unterarm bilden. Abschließend korrigiert man die Ellbogenwinkel so, dass die (gegebenenfalls korrigierte) alte Handposition wieder möglichst gut erreicht wird.

Kapitel 4

Experimente und Ergebnisse

Das Trackingsystem wurde auf realen Videodaten getestet. Ziel war außer der Evaluierung auch die Ermittlung optimaler Werte für zahlreiche Parameter des Systems (*Tuning*, siehe Abschnitt 4.2.1 für eine Übersicht der Parameter). Abschnitt 4.1 beschreibt zunächst das verwendete Szenario und erläutert, wie eine Referenztrajektorie erstellt wurde, mit der die Ausgaben des Systems für verschiedene Parametersätze verglichen wurden. Abschnitt 4.2.2 präsentiert die Ergebnisse dieser Vergleiche, und Abschnitt 4.3 beschreibt, wie sich das System bei ausgewählten, unterschiedlich schwierig zu trackenden Bewegungsabläufen verhält.

4.1 Datensammlung und Erstellung der Referenztrajektorien

Das Trackingsystem wurde anhand von vier Stereobildfolgen evaluiert, die mit jeweils vier Stereokameras aus verschiedenen Blickwinkeln aufgenommen wurden (Abbildung 4.1). In jeder Bildfolge wurde eine andere Person aufgenommen (Abbildung 4.2). Die Aufnahmen sind zwischen zwei und drei Minuten lang und enthalten 1200 bis 2000 Einzelbilder. Der Hintergrund wurde nicht besonders modifiziert und enthält Möbel, Gegenstände und gelegentlich eine weitere Person. Um eine fehlerfreie Initialisierung des Trackers zu gewährleisten, wurden die Personen instruiert, zu Beginn eine definierte Pose einzunehmen (Blickrichtung Nord, Arme ausgestreckt). Anschließend konnten die Personen sich frei bewegen und führten teilweise komplizierte Bewegungen aus, wie Binden der Schuhe, Anlegen der Arme an den Körper oder das Verschränken der Arme vor dem Körper. Die letztgenannten Aktivitäten sind aus den in Abschnitt 3.2.4 genannten Gründen mit dem erstellten System schwierig zu verfolgen, Abschnitt 4.3 geht deshalb gesondert darauf ein.

Die aufgenommenen Bildfolgen wurden wie in Abbildung 4.3 manuell annotiert. Aus den gewonnenen Daten wurde eine Grundwahrheit ermittelt, indem das

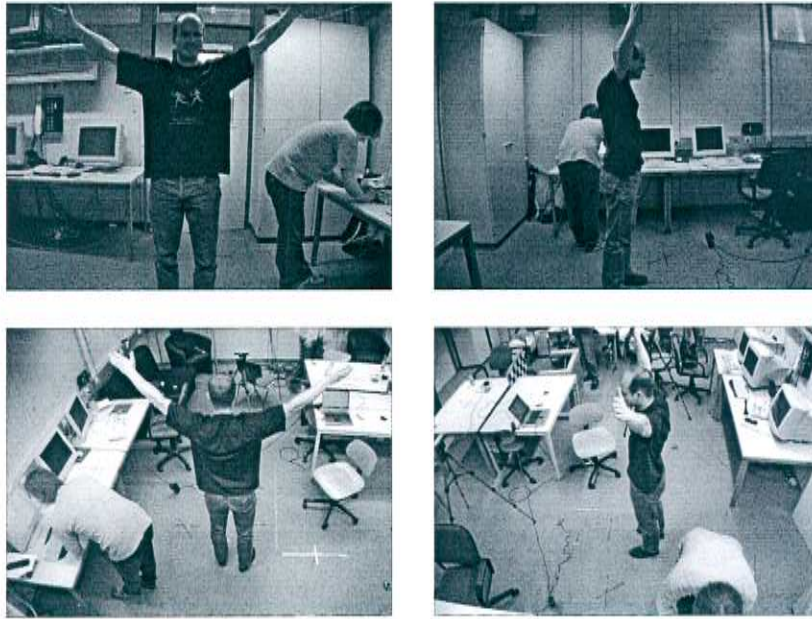


Abbildung 4.1: Datensammlung mit vier Stereokameras. Die Kamerapositionen werden nach den Himmelsrichtungen von links oben nach rechts unten mit N, O, S und W bezeichnet. Die Person konnte sich in einem Bereich von ungefähr drei mal drei Metern frei bewegen.

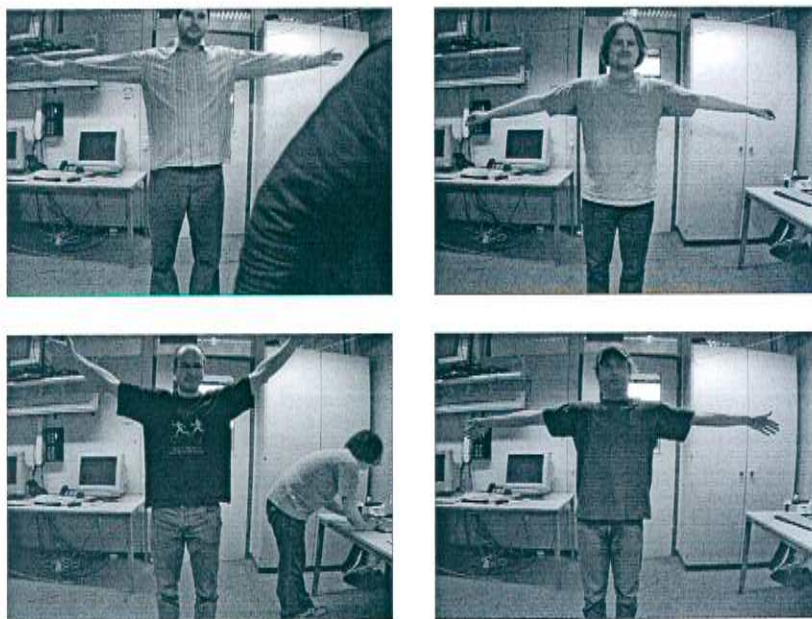


Abbildung 4.2: Datensammlung. Die Experimente wurden mit vier unterschiedlichen Personen durchgeführt, im Folgenden mit P1 bis P4 durchnummeriert (von links oben nach rechts unten). Hier sieht man sie in der Initialisierungspose.

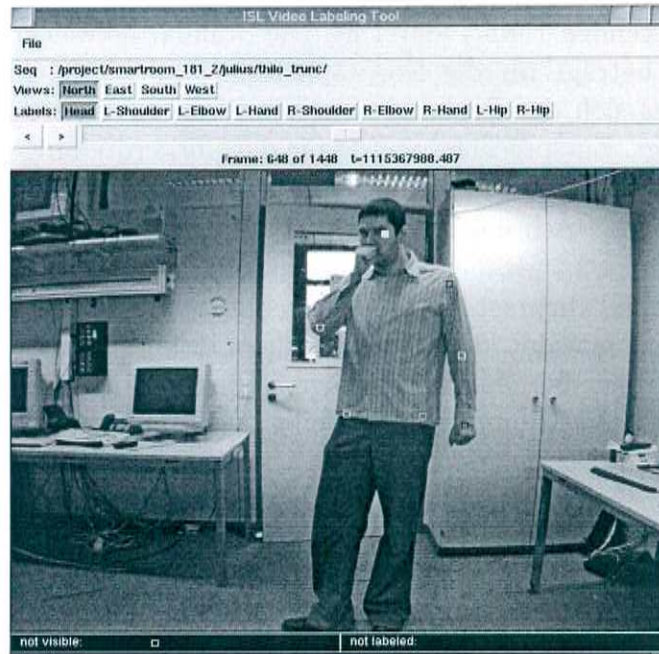


Abbildung 4.3: Handannotierung der Bildfolgen. Neun Körperstellen wurden per Mausklick in allen Kameraansichten markiert (Hände, Ellenbogen, Schultern, Kopf und Hüftknochen).

Oberkörpermodell aus Abschnitt 2.5 den Marken angepasst wurde. Hierzu wurde das Modell mit einem Kalmanfilter den handannotierten Marken nachgeführt. Gegenüber der erhaltenen Modellkonfiguration ergab sich dann ein mittlerer Fehler der Annotierung von 10 Pixel. Hieraus wurde eine Genauigkeit der Grundwahrheit von 10° geschätzt.

4.2 Parametersatz und Tuning

4.2.1 Parametersatz

Zahlreiche Parameter des Trackingsystems können variiert werden. Experimentell wurde ein Referenzparametersatz ermittelt, der für alle Testsequenzen akzeptable Ergebnisse liefert. Dieser Abschnitt führt die Parameter und ihre Referenzwerte auf.

Messrauschen und Systemrauschen (\mathbf{R} und \mathbf{Q} aus Abschnitt 2.2.1) können in recht weiten Grenzen variiert werden ohne signifikante Änderung der Trackinggenauigkeit. Für den Standardparametersatz wurde die Standardabweichung des Messrauschens auf zwei Zentimeter festgesetzt. Dies bedeutet, dass angenommen wird, dass die Position eines zu einem abgetasteten Modellpunkt durch *nearest-neighbour* zugeordneten Messpunktes im Mittel um zwei Zentimeter vom tatsäch-

lich korrespondierenden Punkt abweicht. Die Standardabweichung des gewählte Systemrauschen beträgt für die Körperposition $\sigma_{pos} = 100$ mm, es wird also angenommen, das sich der Proband von Einzelbild zu Einzelbild im Mittel 10 Zentimeter bewegt. Entsprechend wurde für die Körperdrehung (drei Achsen) eine Genauigkeit von $\sigma_{rot} = 20^\circ$ angenommen, für Ellbogen- und Schultergelenke (jeweils zwei mal zwei Achsen) $\sigma_{gelenk} = 40^\circ$.

Wie in Abbildung 3.5 gesehen gibt es grundsätzlich zwei Möglichkeiten, Punkte vom Körpermodell abzutasten. Im Allgemeinen wurde für die Experimente eine zufällige Abtastung gewählt, da sich zeigte, dass, insbesondere wenn wenig Punkte abgetastet wurden, bei deterministischer Abtastung die Neigung besteht, das der Tracker in lokalen Minima hängenbleibt. Der Standardwert für die Anzahl abgetasteter Punkte ist $m = 64$.

Die Proportionen des Kinematikskelettes wurden durch Vermessen des Körpers dem jeweiligen Probanden grob angepasst. Alternativ wurden personenunabhängige Proportionen gewählt, die aus den individuellen Körpermaßen gemittelt wurden.

Wie gesehen läuft das Tracking nach Art einer iterativen Registrierung ab. Für jedes Einzelbild werden standardmäßig $i = 4$ Iterationsschritte verwendet.

In Abschnitt 3.2.4 wurde erläutert, wie ein Mindestabstand der Hand beziehungsweise des Ellbogens vom Torso erzwungen wird. Als Mindestabstand wurde $d_{min} = 150$ mm gewählt.

Im Allgemeinen wurden alle vier Kameras zum Tracking herangezogen (N, O, S und W).

4.2.2 Einfluss einzelner Parameter auf die Trackinggenauigkeit

Die Auswirkung, die die Variation einzelner Parameter auf die Trackinggenauigkeit hat, werden in diesem Abschnitt einander gegenübergestellt. Gemessen durch Vergleich mit der Referenztrajektorie wurden jeweils die Abweichung der Torso-position, der Torsolängsachse, der Torsoquerachse (von Schulter zu Schulter) und der Längsachsen der Ober- und Unterarme. Die Abweichungen wurden über alle Einzelbilder gemittelt.

Tabelle 4.1 führt zunächst die Ergebnisse für den Referenzparametersatz auf. Die Ergebnisse für P3 und P4 sind etwas schlechter, dies ist einerseits auf deren besonders kontrastarme Kleidung zurückzuführen (siehe Abbildung 4.2), andererseits darauf, das in den Sequenzen besonders schwierig zu verfolgende Aktivitäten durchgeführt wurden (siehe auch Abschnitt 4.3).

Tabelle 4.2 zeigt, wie sich die Ergebnisse verändern, wenn die Anzahl der Modellpunkte variiert wird. Der Gewinn an Genauigkeit fällt geringer als erwartet aus, von $m = 24$ abgesehen. Zu erwähnen ist hier, dass die Rechenzeit des Algorithmus vornehmlich von m (Komplexität: $\mathcal{O}(m^3)$) abhängt. Abbildung 4.4

	P1	P2	P3	P4	Zusammen
Torsoposition [mm]	65	53	85	67	67
Torsolängsachse [°]	9,1	8,2	11,2	9,2	9,4
Torsoquerachse [°]	9,5	13,6	16,4	16,1	13,9
Oberarme [°]	22,0	33,5	40,3	32,8	32,1
Unterarme [°]	27,6	25,6	27,8	42,3	30,8

σ_{pos}	σ_{rot}	σ_{gelenk}	Abtastung	m	Proportionen	i	d_{min}	Kameras
100 mm	20°	40°	zufällig	64	individuell	4	150 mm	NOSW

Tabelle 4.1: Ergebnisse für den Referenzparametersatz.

	$m = 24$	$m = 40$	$m = 64$	$m = 120$
Torsoposition [mm]	89	71	67	83
Torsolängsachse [°]	10,8	9,1	9,4	9,3
Torsoquerachse [°]	18,5	13,7	13,9	13,1
Oberarme [°]	41,4	31,8	32,1	31,9
Unterarme [°]	41,9	31,1	30,8	32,0

σ_{pos}	σ_{rot}	σ_{gelenk}	Abtastung	m	Proportionen	i	d_{min}	Kameras
100 mm	20°	40°	zufällig	s. Tabelle	individuell	4	150 mm	NOSW

Tabelle 4.2: Auswirkung der Anzahl abgetasteter Punkte m auf den mittleren Trackingfehler. Die Ergebnisse des Referenzparametersatzes sind hervorgehoben.

	Kameras NS	Kameras NOSW
Torsoposition [mm]	64	62
Torsolängsachse [°]	7,6	8,8
Torsoquerachse [°]	12,1	13,1
Oberarme [°]	34,1	29,4
Unterarme [°]	39,8	31,8

σ_{pos}	σ_{rot}	σ_{gelenk}	Abtastung	m	Proportionen	i	d_{min}	Kameras
100 mm	20°	40°	zufällig	64	individuell	4	150 mm	s. Tabelle

Tabelle 4.3: Tracking mit zwei Kameras anstatt vier. Die Ergebnisse des Referenzparametersatzes sind hervorgehoben. Achtung: P3 wurde aus diesem Vergleich entfernt. Sie war zu einem frühen Zeitpunkt der Sequenz nur in Kamera S sichtbar. Der Track der Arme ging hier verloren und konnte auch später nicht wieder hergestellt werden.

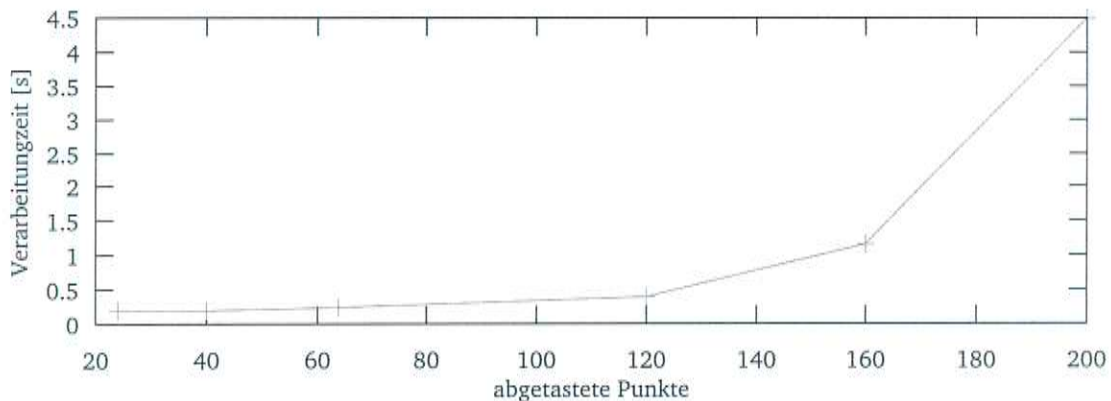


Abbildung 4.4: Abhängigkeit der Verarbeitungszeit von der Anzahl der Modellpunkte. Abgetragen ist die Verarbeitungszeit für ein Einzelbild.

zeigt die experimentell ermittelte Abhängigkeit zwischen m und der für ein Einzelbild benötigten Verarbeitungszeit. Hierzu kommt noch die Rechenzeit für die Stereobildverarbeitung (0.05 Sekunden pro Kamera) und Segmentierung (0.13 Sekunden für die fusionierte Punktwolke).

Probeweise wurde nur auf den Daten von zwei Stereokameras getrackt, die Ergebnisse zeigt Tabelle 4.3. Die Ergebnisse sind, von der P3-Sequenz abgesehen, unerwartet gut. Verschlechterung ergibt sich nur bei den Armen, dies kann darauf zurückzuführen sein, dass diese im Vergleich zum Torso leichter in Verdeckung geraten. Verdeckung ist offensichtlich kritischer, wenn die Szene nur aus wenigen Blickwinkeln beobachtet wird.

Im Allgemeinen wurde das kinematische Modell den ungefähren Proportionen der Probanden individuell angepasst. Tabelle 4.4 zeigt die Trackingergebnisse mit

	gemittelt	individuell
Torsoposition [mm]	75	67
Torsolängsachse [°]	9,4	9,4
Torsoquerachse [°]	14,2	13,9
Oberarme [°]	43,9	32,1
Unterarme [°]	35,9	30,8

σ_{pos}	σ_{rot}	σ_{gelenk}	Abtastung	m	Proportionen	i	d_{min}	Kameras
100 mm	20°	40°	zufällig	64	s. Tabelle	4	150 mm	NOSW

Tabelle 4.4: Tracking unter Annahme gleicher Proportionen aller Probanden.

einem einheitlichen Modell, das durch Mittlung der individuellen Proportionen entstanden ist. Die Verschlechterung beim Tracking der Arme ist signifikant.

4.3 Analyse kurzer Bewegungsabläufe im Detail

Einige Arten von Bewegungsabläufen sind mit dem beschriebenen Verfahren schwieriger zu verfolgen als andere. In diesem Abschnitt werden verschieden anspruchsvolle Bewegungsabläufe einander gegenüber gestellt, die den Sequenzen aus 4.1 entnommen wurden. Auch die präsentierten Trackingergebnisse entstammen den vollständigen Sequenzen (es fand keine separate Initialisierung der hier betrachteten Einzelbewegungen statt). Getrackt wurde jeweils mit den Referenzparametern aus Tabelle 4.1.

Schwierigkeiten ergeben sich meistens dann, wenn, wie in Abschnitt 3.2.4 beschrieben, Gliedmaßen sich stark einander annähern oder berühren. Die Zuordnungen zwischen Modell- und Messpunkten basiert auf räumlicher Nähe, deshalb kommt es in diesen Fällen häufig zu falschen Korrespondenzen. Die in 3.2.4 beschriebene Kollisionsvermeidung kann in vielen Fällen Divergenz verhindern, allerdings dauert es mitunter einige Zeit, bis die richtige Zuordnung wiedergefunden wird. Gelegentlich bleibt eine Gliedmaße auch für lange Zeit falsch eingepasst, beispielsweise bleibt dann eine Hand am Körper oder am Kopf „hängen“.

Abbildung 4.5 auf Seite 56 zeigt fünf Einzelbilder der P4-Sequenz, daneben ist jeweils eine Visualisierung des Trackingergebnisses aus der selben Perspektive abgebildet. Stark artikulierte Bewegungen wie diese, bei denen die Arme klar vom Körper entfernt sind, werden zuverlässig getrackt. Dies bestätigt auch die Fehleranalyse in Abbildung 4.10 auf Seite 61. Die Fehler betragen im Mittel ungefähr 20°.

Ähnlich problemlos wird die Bewegung von P2, die in Abbildung 4.7 (Seite 58) dargestellt ist, verfolgt. Die Fehler sind in Abbildung 4.8 (Seite 59) aufgetragen.

In Abbildung 4.9 (Seite 60) ist ein Ausschnitt der P3-Sequenz wiedergegeben.

P3 beugt sich vor und geht in die Hocke, um sich die Schuhe zu binden. Abbildung 4.10 (Seite 61) zeigt, wie sich die Fehler in diesem Abschnitt entwickeln. Es gibt hier teilweise beträchtliche Abweichungen von bis zu 170° gegenüber der Referenz, trotzdem wird die Szene augenscheinlich recht gut getrackt, insbesondere werden die Fehler nach dem Aufrichten der Person (ungefähr beim Einzelbild 320) rasch wieder geringer.

Abbildung 4.11 (Seite 62) zeigt fünf Einzelbilder der P4-Sequenz. P4 geht mit dem rechten Arm aus einer stark artikulierten Pose in eine Haltung mit vor dem Körper verschränkten Armen über. Der Tracker verhält sich fehlerhaft und variiert sehr stark um die wahre Pose herum. Abbildung 4.12 (Seite 62) zeigt, wie sich die Fehler in diesem Abschnitt entwickeln. Erst ab Bild 520 wird die Pose wieder korrekt wiedergefunden. Posen wie diese, bei denen die Arme dicht am Körper anliegen, sind besonders schwierig zu tracken.

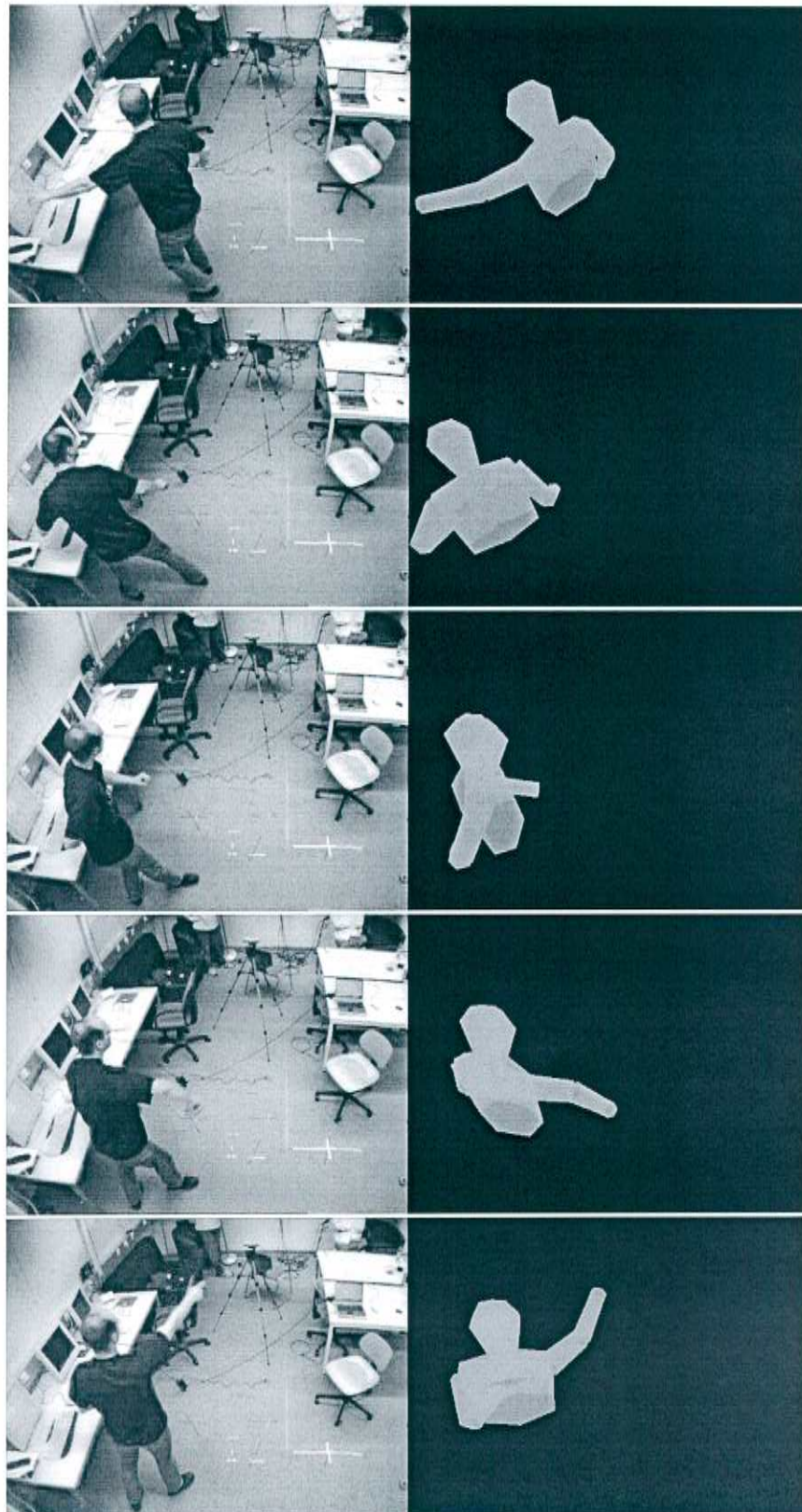


Abbildung 4.5: Bewegungsablauf: „Peitschenhieb“. Einzelbildindices (von oben): 1479, 1507, 1577, 1651, 1753

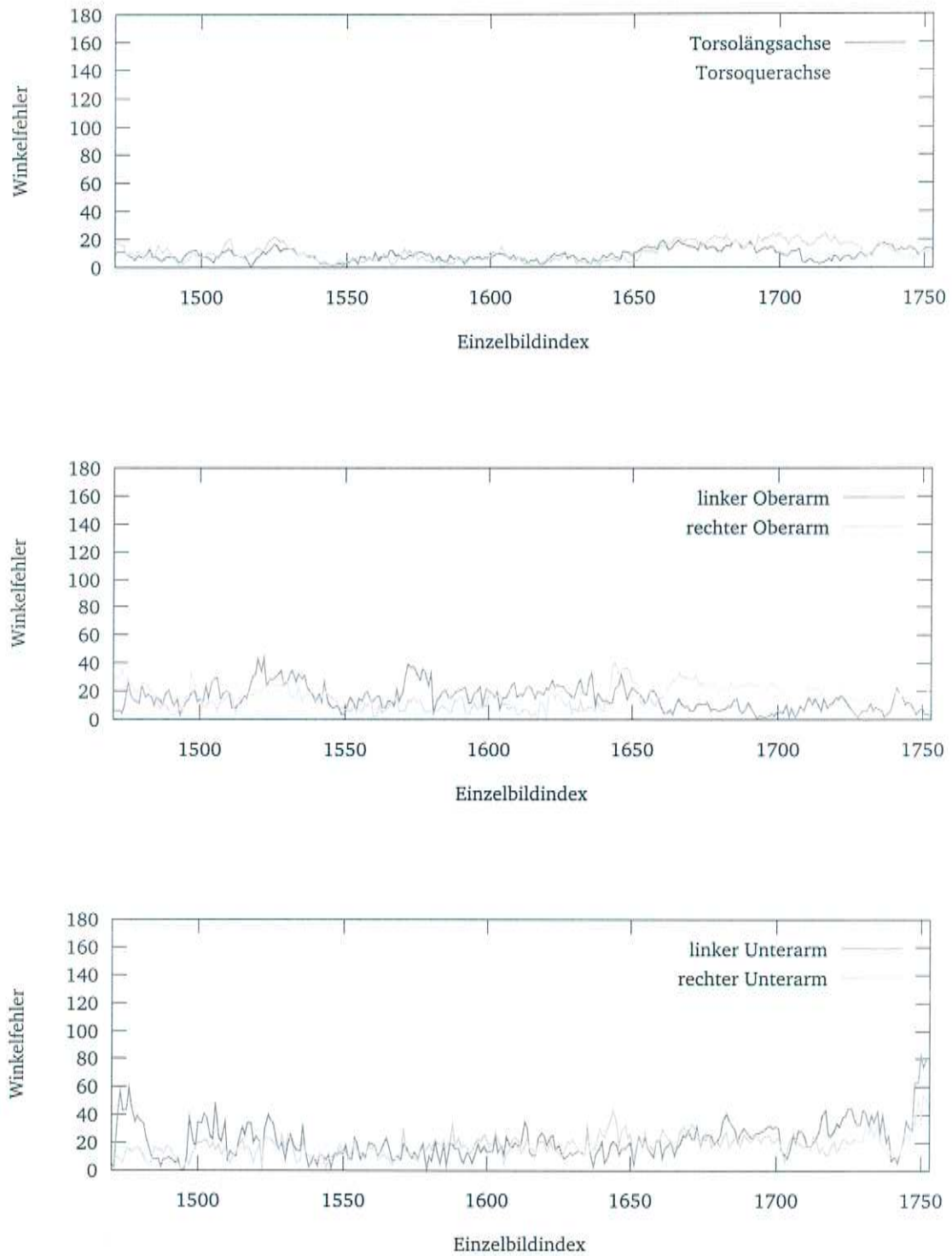


Abbildung 4.6: Fehlerentwicklung während des Bewegungsablaufes „Peitschenhieb“

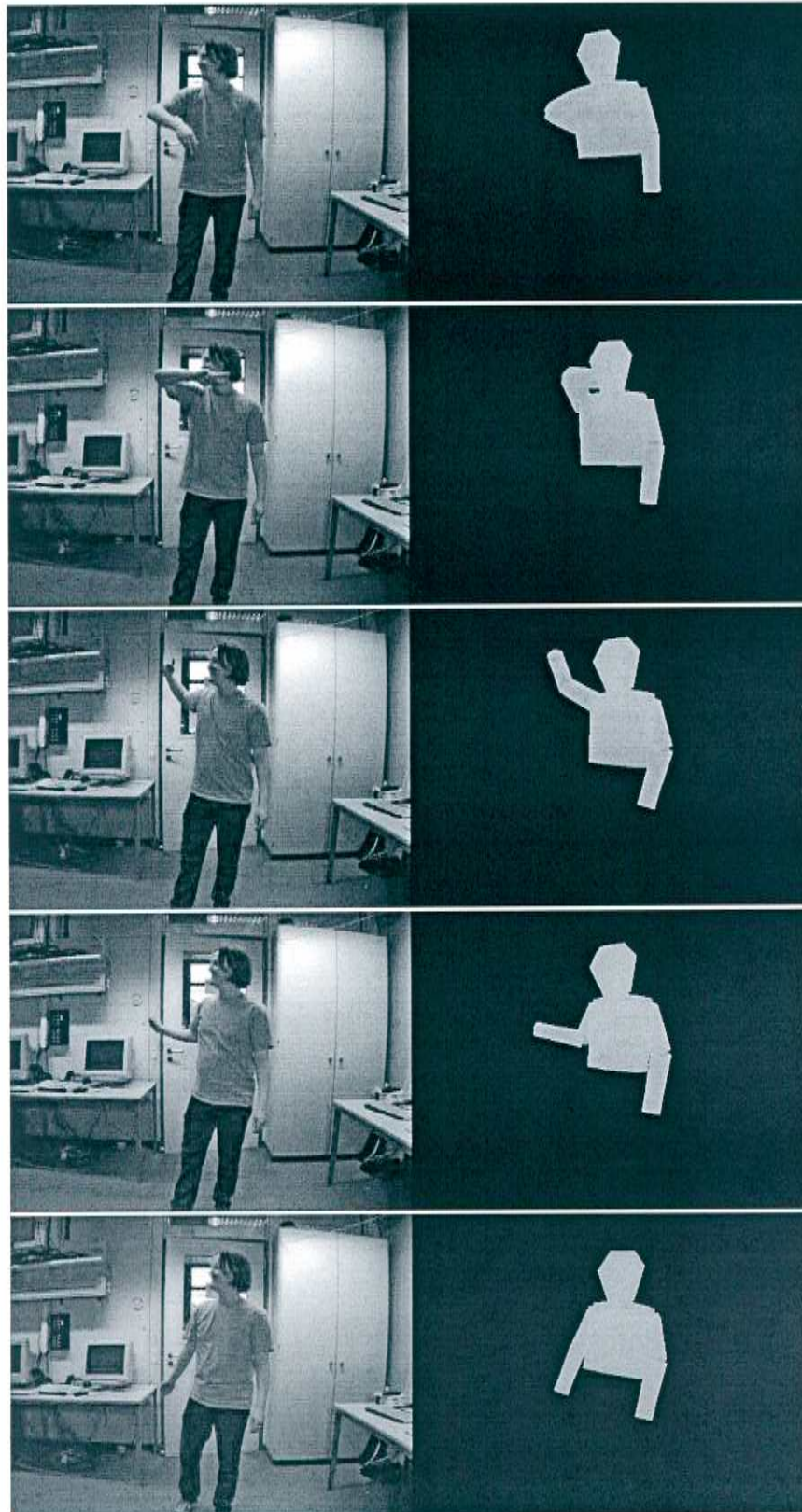


Abbildung 4.7: Bewegungsablauf „Travolta“. Einzelbildindices (von oben): 412, 421, 436, 443, 450

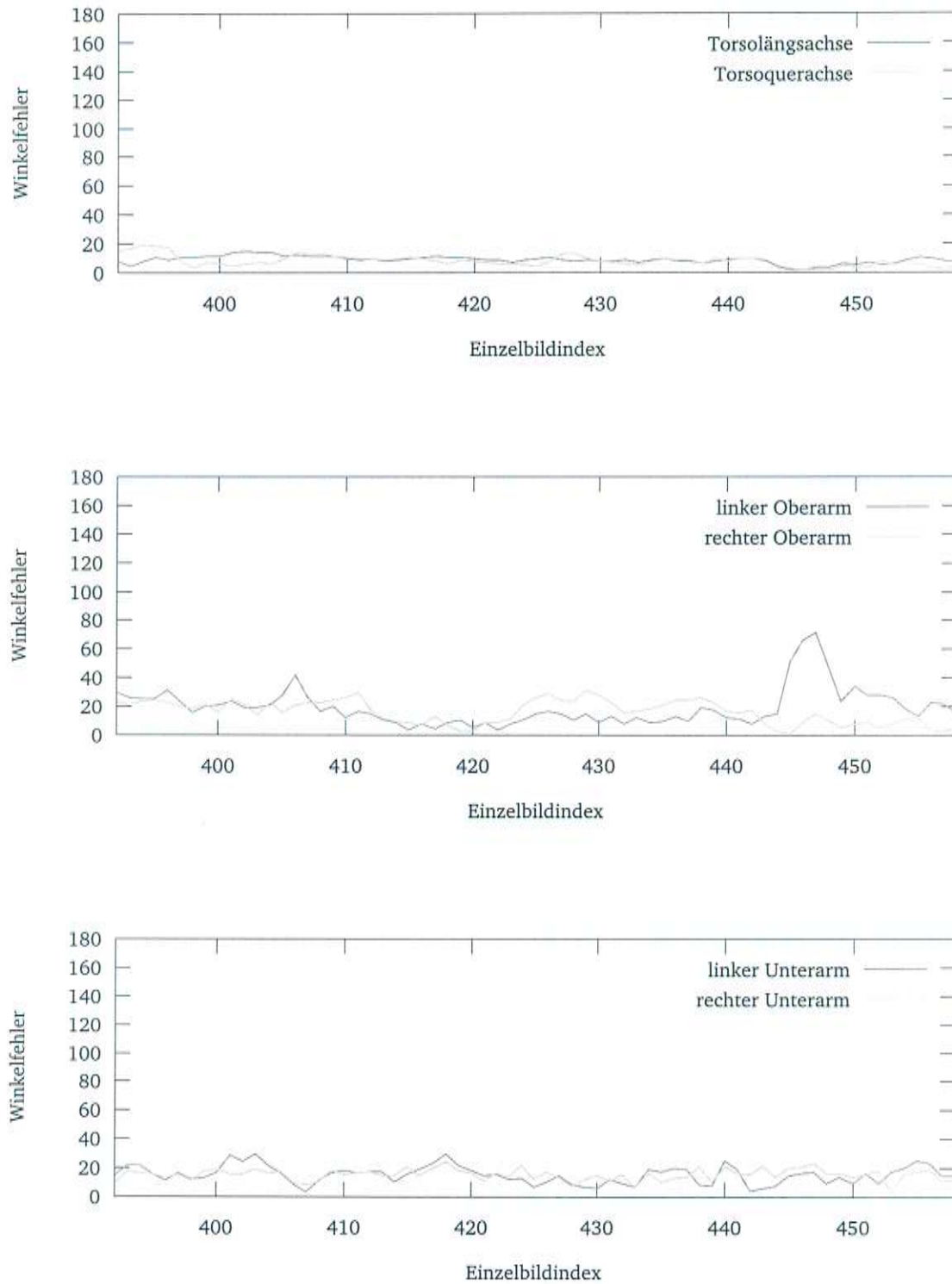


Abbildung 4.8: Fehlerentwicklung während des Bewegungsablaufes „Travolta“.

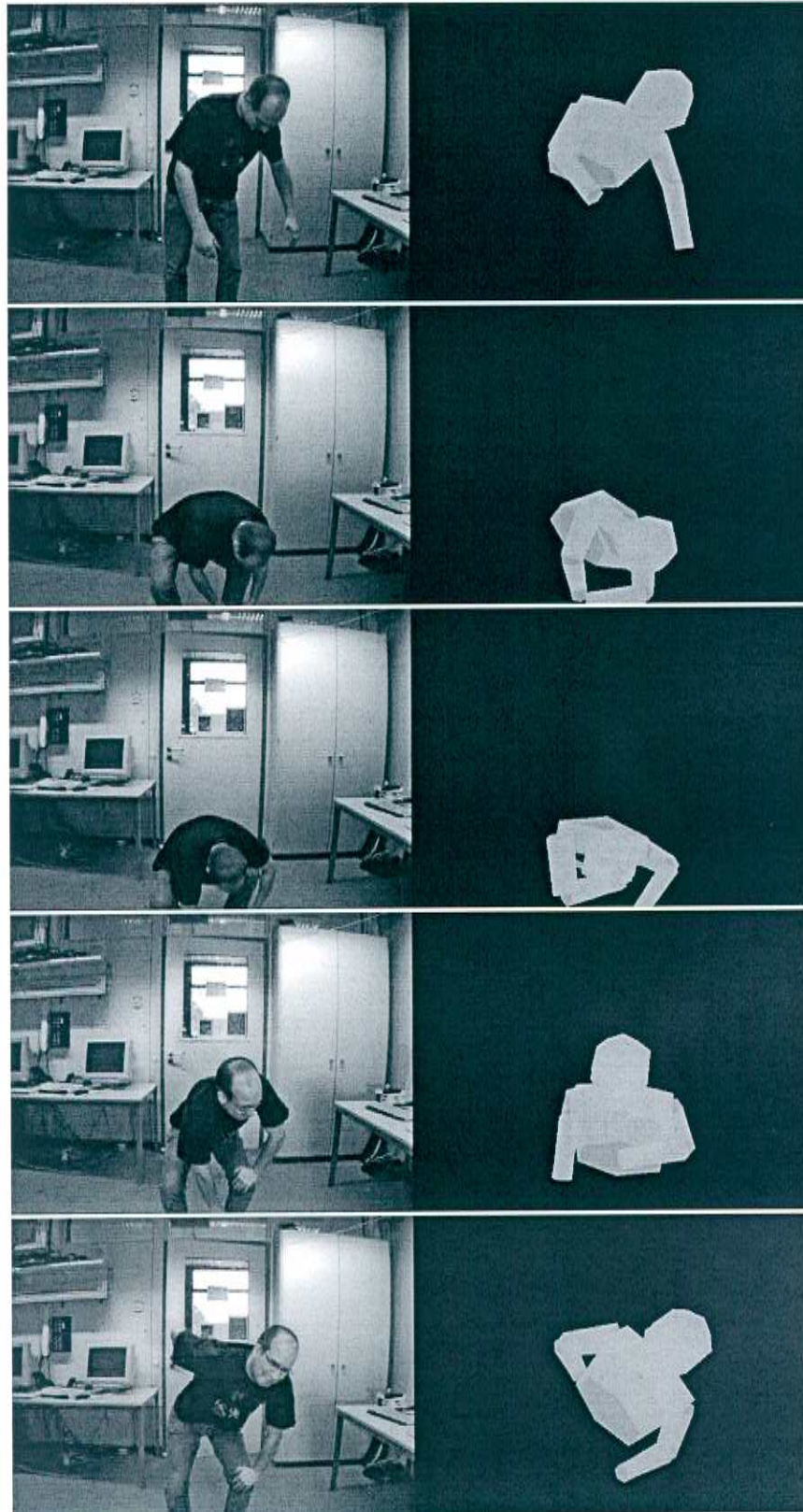


Abbildung 4.9: Bewegungsablauf: „Schuhe binden“. Einzelbildindices (von oben): 113, 148, 230, 312, 328

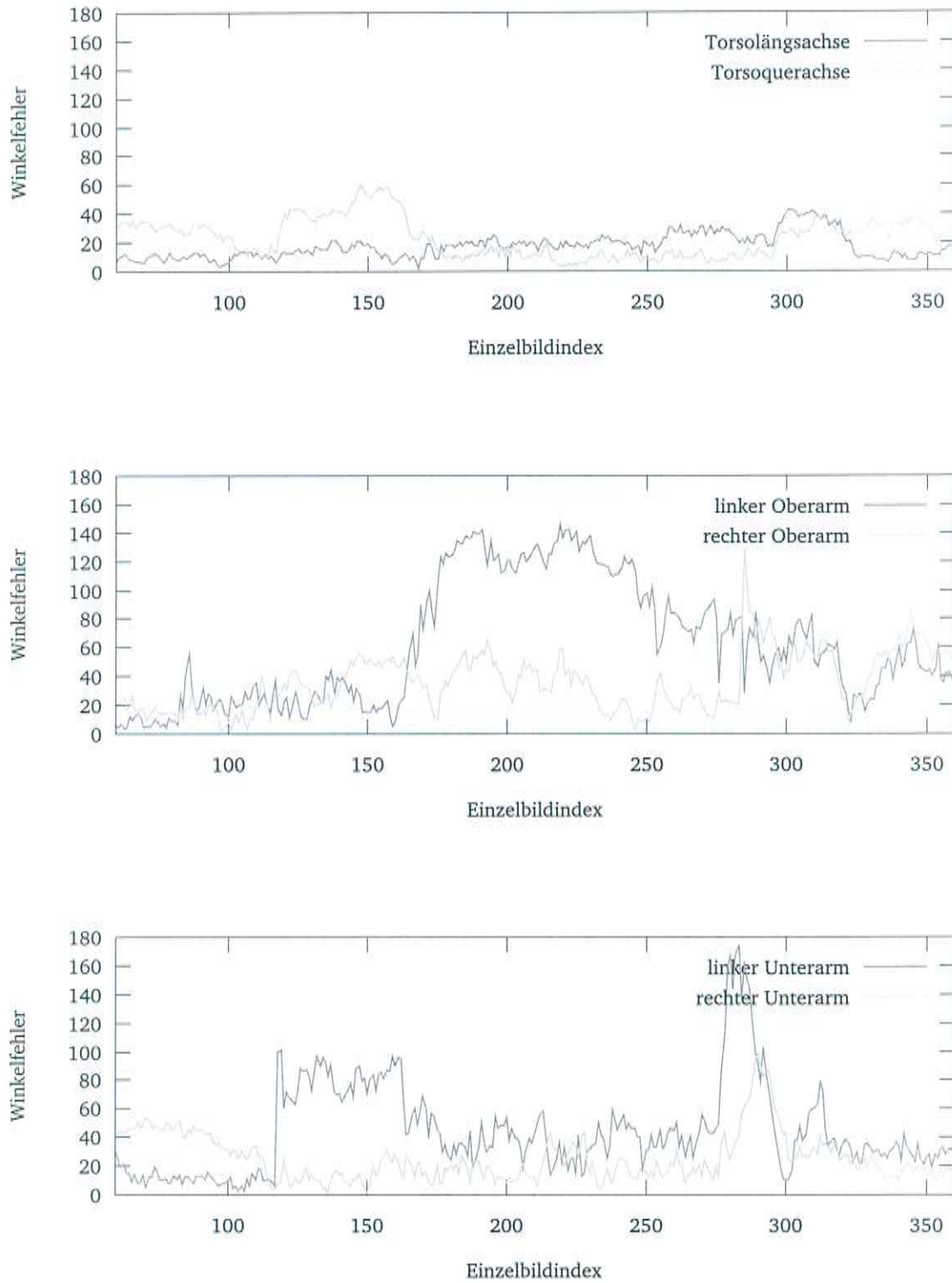


Abbildung 4.10: Fehlerentwicklung während des Bewegungsablaufes „Schuhe binden“.

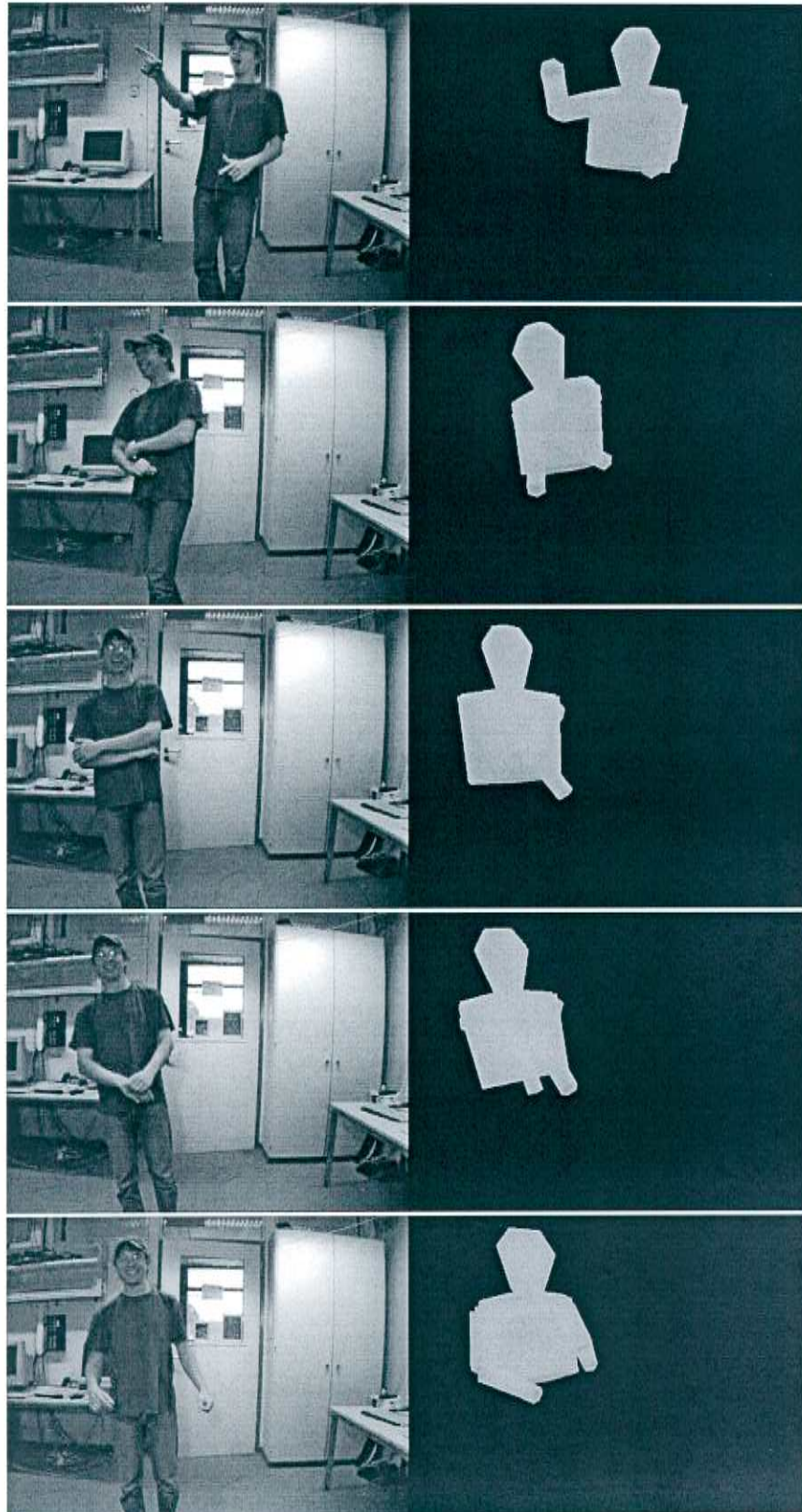


Abbildung 4.11: Bewegungsablauf: „Arme kreuzen“. Einzelbildindices (von oben): 390, 455, 488, 503, 520

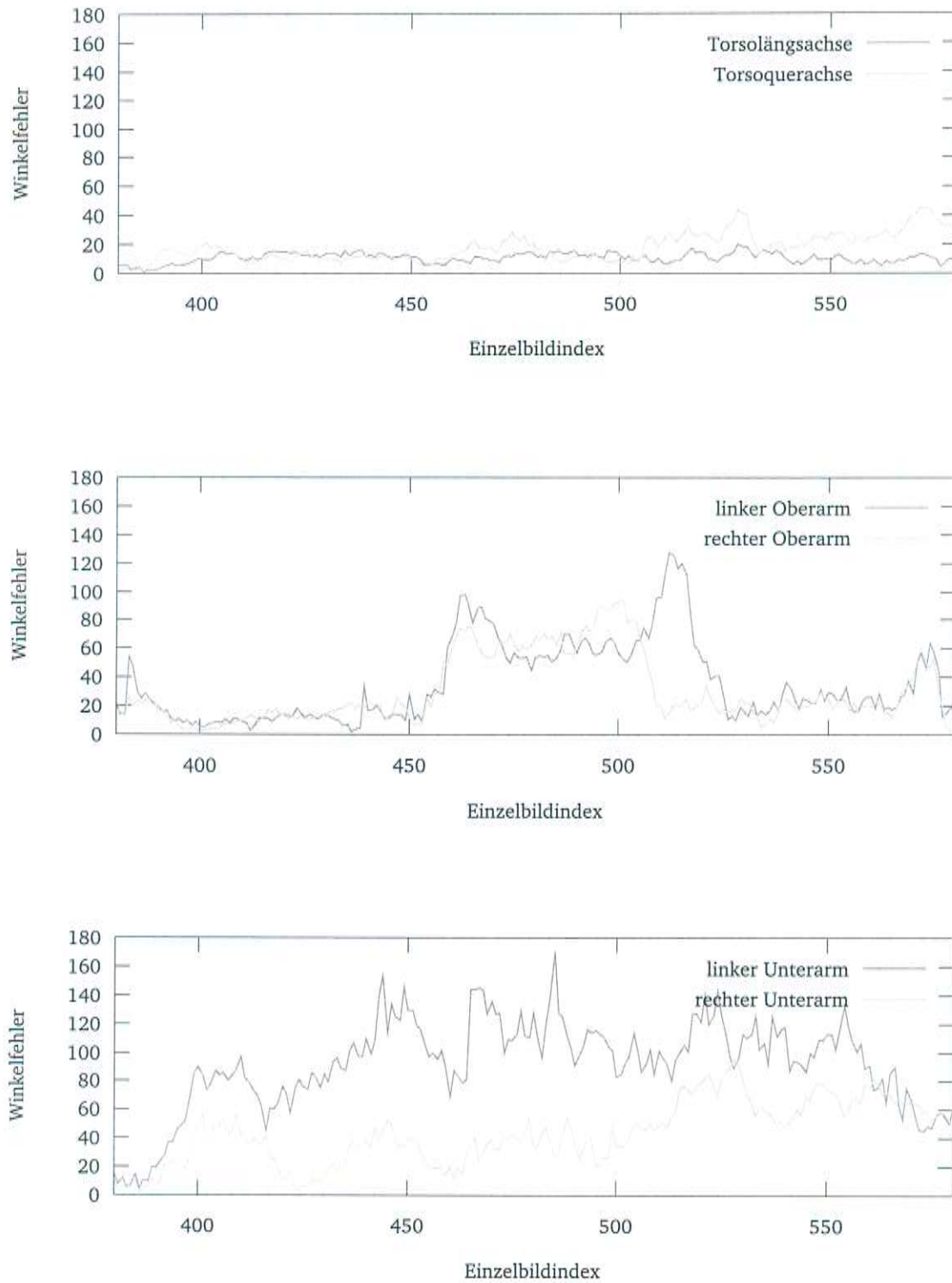


Abbildung 4.12: Fehlerentwicklung während des Bewegungsablaufes „Arme Kreuzen“. Die Arme waren von Bild 455 bis 505 gekreuzt.

Kapitel 5

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein System zur Verfolgung der artikularen Bewegung des menschlichen Oberkörpers entwickelt. Der Algorithmus führt ein dreidimensionales kinematisches Modell des Oberkörpers einer 3D-Punktwolke nach, die durch Stereobildverarbeitung und verschiedene Segmentierungsschritte aus den Bildfolgen mehrerer Stereokameras gewonnen wurde. Das Trackingproblem wurde hierzu als die Registrierung zweier Punktwolken aufgefasst. Die Registrierung wurde als ein lineares Schätzproblem formuliert, das mit einem Unscented Kalmanfilter gelöst wurde.

Die Evaluation des Systems auf vier Bildsequenzen mit vier verschiedenen Personen ergab einen Fehler der Winkelbestimmung von Ober- und Unterarmen von etwa 30° , für die Lage des Oberkörpers etwa 13° . Diese Werte sind gemittelt über die Gesamtheit der ca. 7000 (pro Kamera) aufgenommenen Einzelbilder. Besonders gut verfolgbar sind stark artikulierte Posen, bei denen die Gliedmaßen deutlich vom Körper entfernt sind. In diesen Fällen liegt der gemessene mittlere Trackingfehler ungefähr bei 20° (Arme) beziehungsweise 10° (Torso). In den aufgenommenen Bildfolgen wurden auch schwierig zu verfolgende Posen eingenommen (Verschränken der Arme, Binden der Schuhe). Hier traten bei der Verfolgung der Arme Fehler auf, der Track ging aber nicht vollständig und auch nicht dauerhaft verloren. Alle Bildfolgen (1200 bis 2000 Einzelbilder) wurden vom System durchgehend und ohne zu divergieren getrackt. Der Algorithmus verarbeitet vier Einzelbildsätze pro Sekunde, hinzu kommen Stereobildverarbeitung (0,05 Sekunden pro Kamera) und Segmentierung (0,13 Sekunden für die fusionierte Punktwolke).

Das System wurde ursprünglich mit den Daten von vier Kameras betrieben. Der Betrieb mit weniger Kameras ist auch möglich, die Beschränkung auf zwei Kameras brachte geringe Einbußen bei der Trackinggenauigkeit.

Das Tracking basiert ausschließlich auf der Verwendung von durch Stereobildverarbeitung gewonnenen Tiefeninformationen. Diese sind im wesentlichen invariant gegenüber den Beleuchtungsverhältnissen, Schattenwürfen sowie unterschiedlicher Farbe und Musterung der Bekleidung. Das System zeichnet sich des-

halb durch eine hohe Robustheit gegenüber einer Veränderung dieser Faktoren aus.

Eine sinnvolle Erweiterung des Systems wäre die Einbeziehung von Farbinformationen bei der Bildung der Korrespondenzen. Die natürlichste und einfachste Möglichkeit hierzu ist es, die betrachteten 3D-Punkte um Farbraumkomponenten zu erweitern, so dass Punkte unterschiedlicher Farbe einen hohen Abstand erhalten, wodurch Korrespondenzbildung zwischen ihnen erschwert würde. Das Verfahren wurde im Rahmen dieser Arbeit erprobt, Probleme bereitete hierbei aber die fehlende Farbkalibrierung der Kameras und die damit verbundene Schwierigkeit, einheitliche, für alle verwendeten Kameras gültige Farbmodelle zu erstellen.

Der Kalmanfilter ist ein geeignetes Mittel zur Fusion mehrerer, auch verschiedenartiger, Messungen zu einer optimalen Zustandsschätzung. An dieser Stelle sei auf den *single-constraint-at-a-time*-Ansatz (SCAAT) aus [WB97] verwiesen. Er nutzt die Flexibilität des Kalmanfilter-Rahmenwerks im Umgang mit verschiedenartigen, asynchron eintreffenden Messungen. Entsprechend könnten merkmalsbasierte Verfahren (Gesichtserkennung, Handerkennung) verwendet werden, um die Leistung des in der vorliegenden Arbeit entwickelten Systems zu verbessern. Die Softwarearchitektur erlaubt es, solche Merkmale einfach in die Zustandsschätzung mit einzubringen, im Wesentlichen ist hierzu die Formulierung eines funktionalen Messmodells, das heißt, der Ein-Ausgaberelation zwischen dem die Pose beschreibenden Zustandsvektor und dem Merkmalsvektor, erforderlich. Die Bildung der Jacobimatrix entfällt dank des Einsatzes des Unscented Kalmanfilters. Probeweise wurde das System nach diesem Ansatz so erweitert, dass die mit dem in [Nic03] beschriebenen Verfahren ermittelten Hand- und Kopfpositionen (bisher wurden nur 2D-Informationen verwendet) in die Schätzung der Körperpose mit eingebracht wurden. Die Ergebnisse sind sehr vielversprechend, auch beim Betrieb mit nur einer Kamera wurde eine Trackinggenauigkeit erzielt, die mit den vom ursprünglichen System beim Betrieb mit vier Kameras erzielten Werten vergleichbar sind und sie teilweise sogar übertreffen. Auch in Hinblick auf eine automatische Initialisierung der Verfolgung ist dieser Ansatz interessant.

Anhang A

Implementierungsdetails

Dieser Abschnitt gibt einen Überblick über die Softwarearchitektur des Systems und gibt Hinweise, wie Erweiterungen durchgeführt werden können. Die Stereobildverarbeitung wird dabei ausgespart, sie beruht weitgehend auf [Poi03]. Abschnitt A.1 gibt einen Überblick über die beteiligten Klassen. Abschnitt A.2 demonstriert das Zusammenspiel der Klassen während der Verarbeitung eines Einzelbildsatzes.

A.1 Klassenstruktur

Abbildung A.1 zeigt die am Tracking beteiligten Klassen. Um die Übersichtlichkeit zu wahren, wurden die Objektfunktionen nicht vollständig aufgeführt. Funktionen, die nicht direkt am Tracking beteiligt sind, wurden ausgelassen, im wesentlichen sind dies jene, die der Initialisierung des Systems oder der Visualisierung der Trackingergebnisse dienen. In Einzelfällen betrifft dies auch die Parameterlisten der Funktionen.

Die Klasse `ArticulatedTracker` steuert den Ablauf der Kalmanfilterung und delegiert hierzu an `ArticulatedTrackerObserveModel`, `ArticulatedTrackerPredictModel`, `Scaled_unscented_filter` und `DispFilter`.

Die Klassen `ArticulatedTrackerObserveModel`, `ArticulatedTrackerPredictModel` und `Scaled_unscented_filter` leiten von Klassen der BAYES++-Bibliothek ab (siehe [Ste03]).

`Scaled_unscented_filter` erweitert die Funktionalität der von BAYES++ für den Unscented Kalmanfilter bereitgestellten Filterklasse um das Skalierungsverfahren aus [JU02]. Die Klasse enthält als öffentliche Attribute die momentane Zustandsschätzung, bestehend aus dem Vektor x und seiner Kovarianzmatrix P .

`ArticulatedTrackerPredictModel` enthält das für die Kalmanfilterung verwendete dynamische Modell, derzeit nur die einfache Implementierung aus Abschnitt 3.2.1. In Form der Kovarianzmatrix Q könnte hier zusätzliches Wissen über die Dynamik des menschlichen Körpers eingebracht werden.

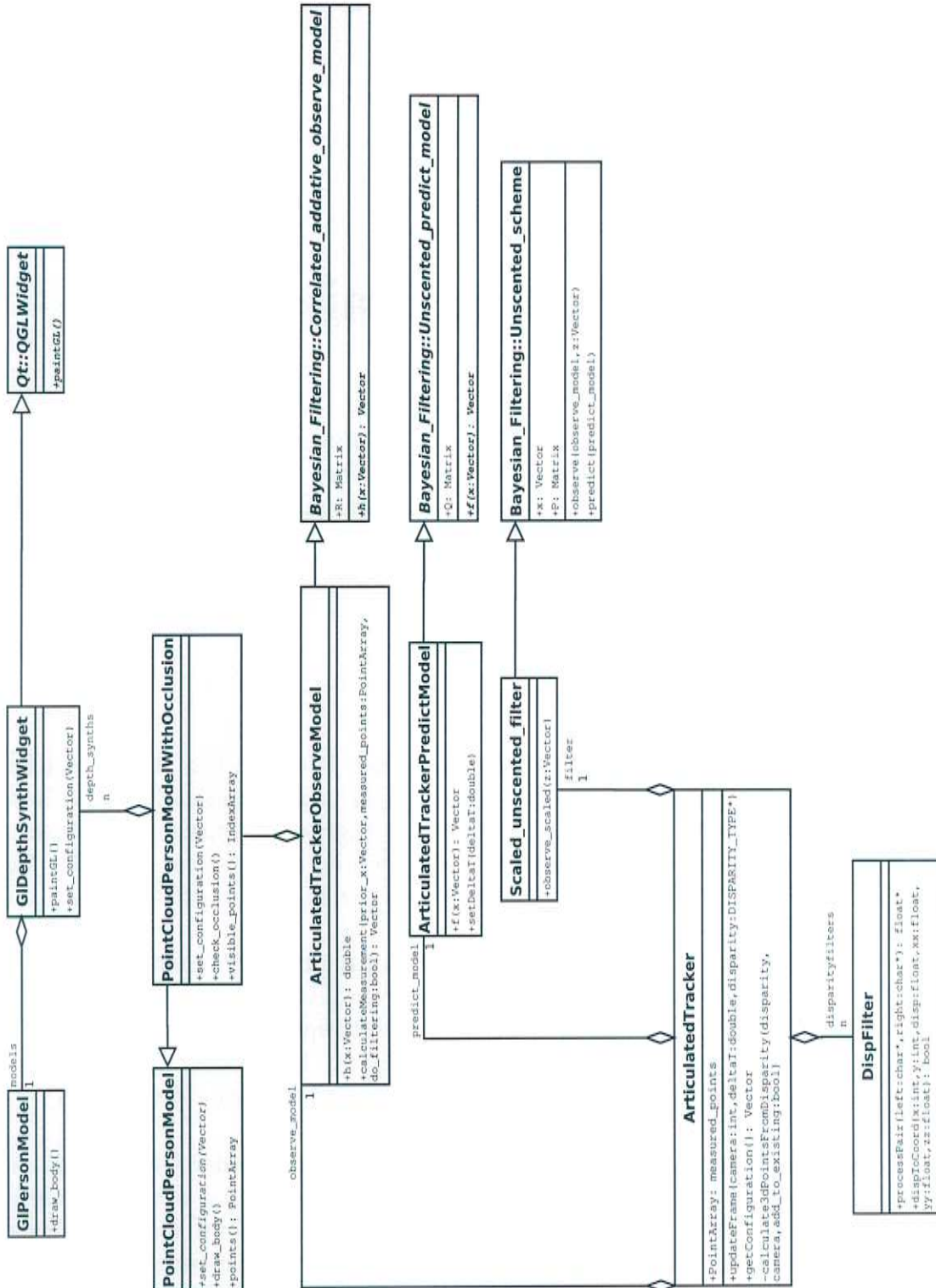


Abbildung A.1: Klassendiagramm 1. n ist die Anzahl verwendeter Kameras. Die zum Messmodell gehörende Gruppe von Klassen ist durch Schattierung hervorgehoben.

Objektbezeichner	Typ
tracker	ArticulatedTracker
predict_model	ArticulatedTrackerPredictModel
observe_model	ArticulatedTrackerObserveModel
filter	Scaled_unscented_filter
disparityfilters[n]	DispFilter[n]
pointsynth	PointCloudPersonModelWithOcclusion
depthsynths[n]	GlDepthSynthWidget[n]
models[n]	GlPersonModel[n]

Tabelle A.1: Objekte des Trackingsystems. `disparityfilters[n]`, `depthsynths[n]`, und `models[n]` sind jeweils ein Arrays aus n Objekten, n ist die Anzahl verwendeter Stereokameras.

`ArticulatedTrackerObserveModel` enthält die Implementierung des Messmodells aus Abschnitt 3.2.2 und delegiert hierzu an die Klasse `PointCloudPersonModelWithOcclusion`. Die Klasse `GlDepthSynthWidget` dient zur Generierung eines synthetischen Tiefenbildes, das bei der Entfernung verdeckter Punkte verwendet wird und verwendet hierzu Routinen der OPENGL-Bibliothek [WNDS99]. Die Anbindung an OPENGL geschah über das QT-GUI-Toolkit [BS04].

Die Klasse `DisparityFilter` hat im Wesentlichen zwei Funktionen: Mit `processPair(...)` wird aus einem Stereobildpaar ein Disparitätenbild erzeugt. `dispToCoord(...)` führt die Konvertierung eines Pixels des Disparitätenbildes in Weltkoordinaten durch.

A.2 Ablauf bei der Verarbeitung eines Einzelbildsatzes

Tabelle A.1 führt alle am Tracking beteiligten Objekte und ihre Typen auf. Abbildung A.2 und A.3 zeigen die Kollaboration der Objekte des Trackingsystems bei der Verarbeitung eines Satzes Disparitätenbilder. Die Pfeile bedeuten Funktionsaufrufe, sie sind entsprechend der Aufrufhierarchie durchnummeriert und mit Funktionsname und Parametererliste versehen.

`tracker.updateFrame(...)` (0) wird nacheinander für die mit der Klasse `FGSegmenter` vorsegmentierten Disparitätenbilder aller Kameras aufgerufen. Alle Aufrufe bis auf den Letzten verhalten sich wie in Abbildung A.2 und führen ausschließlich dazu, das im Array `measured_points` die aus den Disparitätenbildern mittels des entsprechenden `disparityfilters` errechneten 3D-Punkte akkumuliert werden (1 - 1.1).

Erst der letzte Aufruf stößt zusätzlich die Verarbeitung (2 - 4.2) an. Zu-

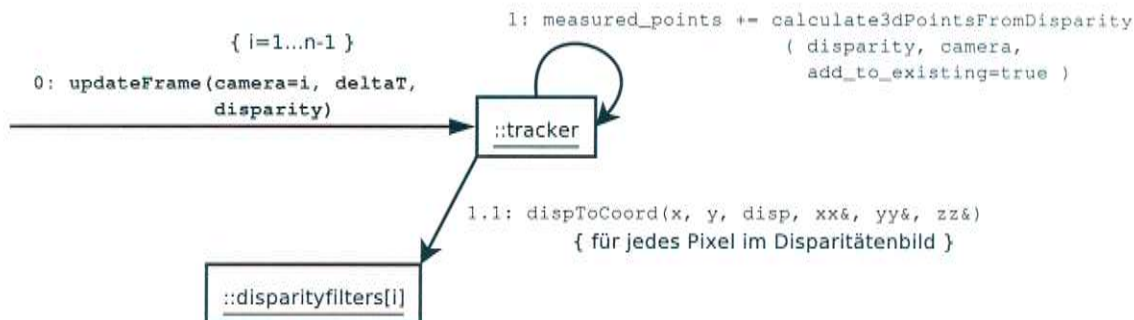


Abbildung A.2: Kollaborationsdiagramm des Trackingsystems, Akkumulierung der 3D-Daten. n ist die Anzahl verwendeter Stereokameras.

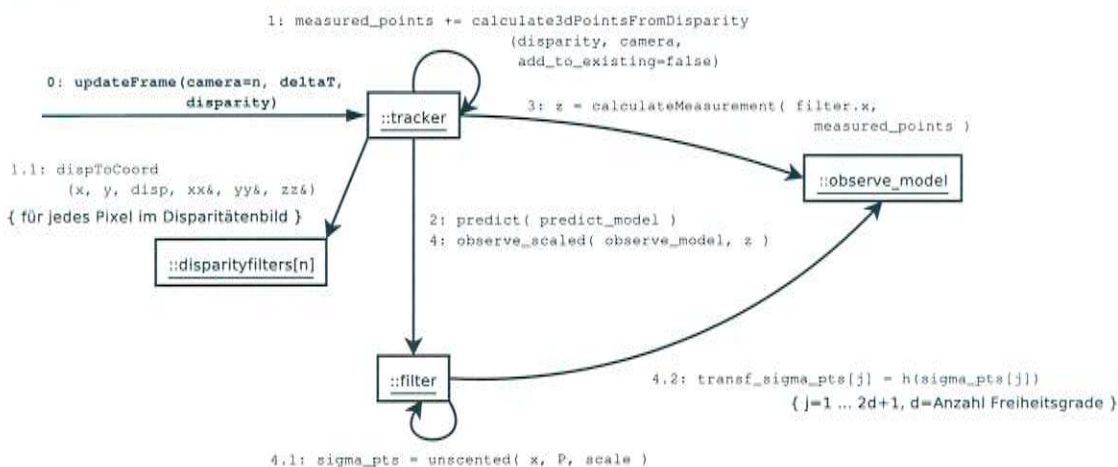


Abbildung A.3: Kollaborationsdiagramm des Trackingsystems, Verarbeitung. n ist die Anzahl verwendeter Stereokameras.

nächst wird der Prädiktionsschritt des Kalmanfilters durchgeführt (2). Anschließend wird der Messvektor z berechnet, der aus der gemessenen Punktwolke in `measured_points` und der momentanen Zustandsschätzung `filter.x` abgeleitet wird (3). Die Berechnung des Messvektors ist in Abbildung A.4 im Detail wiedergegeben. Der Messvektor wird schließlich in die Zustandsschätzung eingebracht (Kalmanfilter-Korrekturschritt), hierzu wird er über die Funktion `observe_scaled` zusammen mit dem zu verwendenden Messmodell an das Filterobjekt übergeben (4). Im Rahmen des Korrekturschrittes werden die σ -Punkte berechnet (4.1) und durch das Messmodell transformiert (4.2). Die Auswertung des Messmodells ist in Abbildung A.5 dargestellt.

A.2. ABLAUF BEI DER VERARBEITUNG EINES EINZELBILDSATZES 71

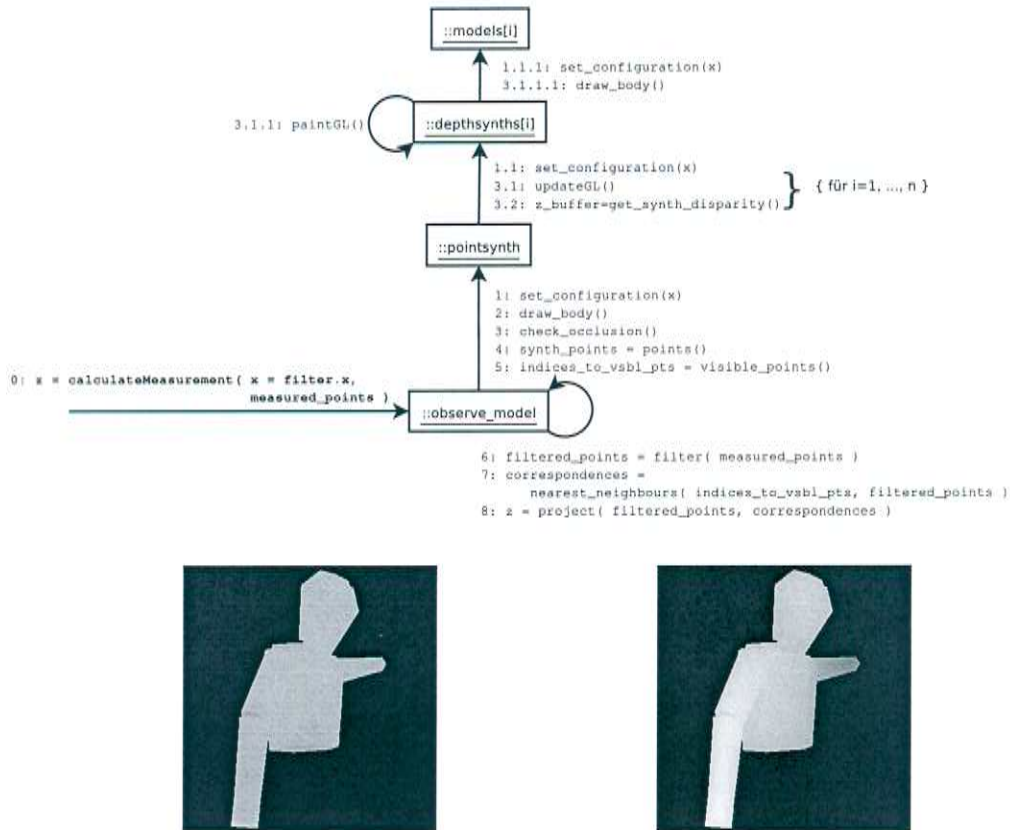


Abbildung A.4: Kollaborationsdiagramm zur Berechnung des Messvektors z (oben), n ist die Anzahl verwendeter Stereokameras. Die Entfernung verdeckter Punkte geschieht mit Hilfe eines synthetisierten Tiefenbildes. Dieses wird erzeugt, indem eine polygonale Repräsentation (links unten) des Oberkörpermodells mittels `GLPersonModel.draw_body()` (3.1.1.1) in einen OpenGL-Kontext gezeichnet wird. Die OpenGL-Library kann gleichzeitig ein Tiefenbild der gezeichneten Szenerie erstellen, den sogenannten z-Buffer (rechts unten). Dieser wird von `GLDepthSynthWidget` ausgelesen (3.2).

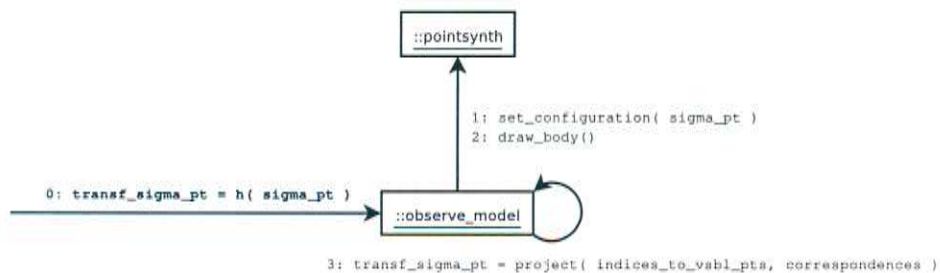


Abbildung A.5: Kollaborationsdiagramm zur Auswertung des Messmodelles.

Literaturverzeichnis

- [Aru87] ARUN, Huang T.S. Blostein S.: Least-squares fitting of two 3D point sets. In: *Transactions on Pattern Analysis and Machine Intelligence* 9 (1987), Nr. 5, S. 698–700
- [Ben75] BENTLEY, Jon L.: Multidimensional Binary Search Trees Used for Associative Searching. In: *Communications of the ACM* 18 (1975), Nr. 9, S. 509–517
- [BM92] BESL, P. ; MCKAY, N.: A Method for Registration of 3-D Shapes. In: *Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2
- [BS04] BLANCHETTE, Jasmin ; SUMMERFIELD, Mark: *C++ GUI Programming with Qt 3*. Prentice Hall PTR, 2004
- [DBR00] DEUTSCHER, J. ; BLAKE, A. ; REID, I.: Articulated Body Motion Capture by Annealed Particle Filtering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* Bd. 2, 2000, S. 2126–2133
- [DD02] DEMIRDJIAN, D. ; DARRELL, T.: 3-D Articulated Pose Tracking for Untethered Diectic Reference. In: *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, 2002, S. 267
- [EKB98] EVELAND, C. ; KONOLIGE, K. ; BOLLES, R. C.: Background Modeling for Segmentation of Video-Rate Stereo Sequences. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998, S. 266–271
- [GGM97] GORIS, Malcolm J. ; GRAY, Douglas A. ; MAREELS, Iven M. Y.: Reducing the Computational Load of a Kalman filter. In: *IEE Electronics Letters* 33 (1997), Nr. 18, S. 1539–1540
- [GSS93] GORDON, N.J. ; SALMOND, D.J. ; SMITH, A.F.M.: Novel Approach to Non-linear/Non-Gaussian Bayesian State Estimation. In: *IEE Proceedings-F* Bd. 140, 1993, S. 107–113

- [Hor87] HORN, Berthold K. P.: Closed-Form Solution of Absolute Orientation Using Unit Quaternions. In: *Journal of the Optical Society of America A* 4 (1987), Nr. 4, S. 629
- [Int01] Intel Corporation: *Open Source Computer Vision Library Reference Manual*. 2001
- [Jä97] JÄHNE, Bernd: *Digitale Bildverarbeitung*. Springer-Verlag, 1997
- [JTH99] JOJIC, Nebojsa ; TURK, Matthew ; HUANG, Thomas S.: Tracking Articulated Objects in Dense Disparity Maps. In: *International Conference on Computer Vision*, 1999, S. 123–130
- [JU96] JULIER, S. J. ; UHLMANN, J. K.: A General Method for Approximating Nonlinear Transformations of Probability Distributions / Department of Engineering Science, University of Oxford. 1996. – Forschungsbericht
- [JU97] JULIER, S. ; UHLMANN, J.: A New Extension of the Kalman Filter to Nonlinear Systems. In: *SPIE AeroSense Symposium*, 1997
- [JU02] JULIER, S. J. ; UHLMANN, J. K.: The Scaled Unscented Transformation. In: *Proceedings of the IEEE American Control Conference*, IEEE, 2002, S. 4555–4559
- [Kal60] KALMAN, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME–Journal of Basic Engineering* 82 (1960), S. 35–45
- [KBVG05] KEHL, R. ; BRAY, M. ; VAN GOOL, L.J.: Full Body Tracking from Multiple Views Using Stochastic Sampling. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, S. 129–136
- [LCJ02] LEE, Mun W. ; COHEN, I. ; JUNG, Soon K.: Particle filter with analytical inference for human body tracking. In: *Workshop on Motion and Video Computing*, 2002, S. 159
- [May79] MAYBECK, Peter S.: *Mathematics in Science and Engineering*. Bd. 141: *Stochastic models, estimation, and control*. 1979
- [Mou98] MOUNT, David M. *Approximate Nearest Neighbor Library (ANN), Version 0.2*. Kann bei <http://www.cs.umd.edu/~mount/ANN> heruntergeladen werden. 1998
- [Nic03] NICKEL, Kai: *Erkennung von Zeigegesten basierend auf 3D-Tracking von Kopf und Händen*, Universität Karlsruhe, Diplomarbeit, 2003

- [Poi03] Point Grey Research Inc.: *Triclops Stereo Vision System Manual Version 3.1*. 2003
- [RFZ05] RAMANAN, D. ; FORSYTH, D. A. ; ZISSERMAN, A.: Strike a Pose: Tracking People by Finding Stylized Poses. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* Bd. 1, 2005, S. 271–278
- [Sch70] SCHMIDT, Stanley F.: Computational Techniques in Kalman Filtering. In: *AGARDograph 139 - Theory and Applications of Kalman Filtering*. NATO Advisory Group for Aerospace Research and Development, 1970
- [SMC01] STENGER, B. ; MENDONÇA, P. R. S. ; CIPOLLA, R.: Model-Based Hand Tracking Using an Unscented Kalman Filter. In: *Proceedings of the British Machine Vision Conference* Bd. I, 2001, S. 63–72
- [Ste03] STEVENS, Michael. *Bayes++ the Bayesian Filtering Library*. Kann bei <http://bayesclasses.sourceforge.net/Bayes++.html> heruntergeladen werden. 2003
- [UF04] URTASUN, Raquel ; FUA, Pascal: 3D Human Body Tracking using Deterministic Temporal Motion Models. In: *European Conference on Computer Vision*, 2004
- [WADP97] WREN, Christopher R. ; AZARBAYEJANI, Ali ; DARRELL, Trevor ; PENTLAND, Alex: Pfunder: Real-Time Tracking of the Human Body. In: *Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), Nr. 7, S. 780–785
- [WB97] WELCH, Greg ; BISHOP, Gary: SCAAT: Incremental Tracking with Incomplete Information. In: *Computer Graphics* 31 (1997), S. 333–344
- [WNDS99] WOO, Mason ; NEIDER, Jackie ; DAVIS, Tom ; SHREINER, Dave: *OpenGL 1.2 Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley, 1999

